Licenční ujednání

ČTĚTE POZORNĚ:

Používáním tohoto produktu potvrzujete svůj souhlas s Licenčními podmínkami, na jejichž základě je Vám poskytnuta jedna ze dvou následujících licencí k užívání programu a omezená záruka.

Upozornění:

Program je chráněn Autorským zákonem a mezinárodními úmluvami. Porušování licenčních podmínek bude stíháno dle zákona.

Licenční podmínky:

1. Prohlášení výrobce

Program není ani Public domain, ani Freeware.

2. Volná licence - pro neregistrovanou kopii (demo verze programu)

Neregistrovanou kopii produktu (tzv. demo verzi) můžete používat zdarma a dále volně šířit za podmínky, že program šíříte zdarma a každá další, vámi šířená kopie, bude úplná, včetně těchto licenčních podmínek a 100% shodná s originálem.

3. Placená licence - pro registrovanou kopii (plná verze programu)

Registrovanou kopii produktu máte právo používat v jednu chvíli pouze na jednom počítači, a za podmínky, že jste legální vlastník této registrované kopie.

Legální vlastník je ten, na jehož jméno je program zaregistrován u prodejce nebo ten, kdo legálně získal originální distribuční médium (disk, CD-ROM).

Používáte-li registrovanou kopii v počítačové síti, musíte mít zakoupenu licenci pro všechny počítače zapojené do sítě, na kterých bude program spouštěn.

Registrace je nepřenosná, tj. nemůže být žádným způsobem ani převedena ani zapůjčena, a to ani dočasně, na žádný další subjekt.

Originální distribuční médium (disk, CD-ROM) je možno přenechat další osobě za těchto podmínek:

a) Produkt dosud nebyl zaregistrován na žádné jméno ani u prodejce ani u výrobce.

b) Předávající osoba si neponechá žádnou kopii produktu ani žádné jeho součásti.

Vlastní aplikace (zdrojové texty, obrázky, zvuky) vytvořené v tomto produktu můžete šířit bez omezení a bez dalších licenčních poplatků.

Omezená záruka

Výrobce produktu prohlašuje, že produkt je poskytován v podobě, v jaké je, a výrobce neposkytuje žádné záruky ohledně funkce produktu nebo pro jeho užití k dalším specifickým účelům.

Výrobce nedává žádné záruky a nenese žádnou odpovědnost za žádné případné škody, ať přímé, či nepřímé, způsobené produktem, ať přímo, či nepřímo.

Maximální výše odpovědnosti prodejce je omezena pouze na výměnu produktu nebo vrácení zaplacené ceny.

Proto doporučujeme, abyste si před koupí licence na plnou verzi programu, nejprve důkladně vyzkoušeli demo verzi.

Poznámka: V každé zemi mohou platit jiné zákony vymezující zákonné záruky prodejce.

Výrobce

SGP Systems, s.r.o.
L. Janáčka 180
686 01 Uherské Hradiště
Česká republika
Tel./fax: +420-632-551089
E-mail: sgp@sgp.cz
http://www.sgp.cz

Podrobný obsah podle témat

Stručný obsah podle témat

Základy

- Co je to Baltík
- Předměty a banky předmětů
- Scéna

Režim Skládat scénu (od 4 let)

- Skládání scény
- Uchovávání scén v souborech
- Kreslení vlastních předmětů
- Nástroje grafického editoru Paint
- Jak si nakreslit svůj vlastní předmět

Režim Čarovat scénu (od 5 let)

Režim Programovat

Úroveň Začátečník

- Používané prvky
- Počítané opakování a bloky příkazů
- Průhlednost - využití v programech

Úroveň Pokročilý

Literál a typy dat

Konstanty a proměnné

- Konstanty
  - Systémové konstanty
- Proměnné
  - Použití proměnných
  - Sledování hodnot proměnných za běhu programu
- Modifikace **vlastností**
- Operátor **přiřazení**

Strukturované datové typy

- Pole
- Hromady
- Souřadnice
- Nějaký předmět
- Tabulka oblastí
- Datum a čas,
- Řetězce
  - Převod na číslo
- Soubory,
  - Tabulka souborů
  - Složky

Programové konstrukce

- Počítané opakování a bloky příkazů
- Podmínky
- Větvení "if" a "if - else"
- Větvení "switch - case"
- Cyklus s řídící proměnnou - cyklus "for"
- Cykly s podmínku - cykly "while" a "do - while"
- Předčasné ukončení cyklu - break

Pomocníci

- Baltíkův pomocník (procedura)
- Tabulka pomocníků

Animace

- Jednoduché animace
- Animace
- Ruční animace
- Nezobrazovat

Aplikace a multimédia

- Tabulka souborů
- Aplikace
- Multimedia
  - Přehrávání zvukových CD
  - Video
  - Obrázky
  - Zvuky

Práce s číselnými daty

- Číselné výrazy
- Matematické funkce
- Náhodné číslo,
- Stopky

Vstupy a výstupy

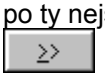
- Práce s obrazovkou
- Zobrazení
- Grafika
- Klávesnice
- Znak
- Myš
- Základní informace o průhlednosti
- Pokročilé použití průhlednosti
- Události
- Chyby při běhu programu
- Nezobrazovat

Používání nápovědy

Doporučený postup výkladu

Doporučení k používání nápovědy a programu:

Tato nápověda je vytvořena tak, že zároveň slouží jako **podrobný návod** k používání programu. Zajímá-li vás pouze jedno jediné téma, najděte si je ve stručném nebo podrobném obsahu. Chcete-li se však seznámit s Baltíkem podrobněji, doporučujeme vám postupovat po jednotlivých sekcích nápovědy. Ty jsou seřazeny od nejjednodušších pro začátečníky (vysvětlení základních pojmů - Předměty, Scéna) až po ty nejsložitější pro pokročilé (Programovat).



Budete-li chtít **číst nápovědu plynule od počátku do konce**, stačí od této chvíle přecházet k dalšímu tématu pouhým stiskem tlačítka se symbolem **>>** na panelu pod hlavní nabídkou.

Vždy, když budete tuto elektronickou příručku opouštět, zadejte v základní nabídce nápovědy povel ***Záložka → Definovat*** a v následně otevřeném dialogovém okně ***Definice záložky*** zadejte nějaký název (např. **Tady**) a své zadání potvrďte stiskem ***OK***.

Pozn.: Učí-li se vás podle této nápovědy několik, můžete jako název záložky zadat např. svoje jméno.

Když se příště k nápovědě vrátíte, otevřete nabídku ***Záložka*** a zadejte název příslušné záložky. Nápověda se vám otevře v sekci, kterou jste si minule založili.

Druhou možností, jak navázat na své minulé studium, je otevřít sekci **Doporučený postup výkladu,** v níž jsou jednotlivá témata seřazena přesně v tom pořadí, v jakém jsou vám postupně předkládána při přecházení k další sekci stiskem tlačítka **>>**.

Pozn.: **K tomu, abyste spustili nápovědu, nepotřebujete bezpodmínečně spouštět Baltíka.** Soubor BALTIE.HLP, v němž je nápověda uložena, můžete spustit i samostatně (stačí na něj poklepat), a to dokonce i na počítači, kde Baltík nejen neběží, ale kde by dokonce ani běžet nedokázal - např. na počítači s operačním systémem Windows 3.11. Ke studiu tak můžete využít i starší, méně výkonné počítače.

Doporučený postup výkladu

Stručný obsah podle témat Podrobný obsah podle témat

V této sekci najdete seznam témat seřazených v tom pořadí, v jakém jsou vám postupně předkládána při přecházení k další sekci stiskem tlačítka **>>**. Má vám pomoci vrátit se k tématu, v němž jste nápovědu opustili nebo k tématu, které jste již prošli, ale potřebujete se na něj ještě jednou podívat. Chronologické řazení by vám mělo pomoci najít hledané téma co nejrychleji.

Základní informace

- Co je to Baltík
- Předměty a banky předmětů
- Scéna

Režimy práce

Režim Skládat scénu (od 4 let)

- Skládání scény
- Uchovávání scén v souborech
- Kreslení vlastních předmětů
- Nástroje grafického editoru Paint
- Jak si nakreslit svůj vlastní předmět

Režim Čarovat scénu (od 5 let)

Režim Programovat

- Základní informace
  - Vytváření a úpravy programů
  - Tlačítka nad pracovní plochou
- Práce v úrovni Začátečník (od 6 let)
  - Používané prvky
  - Počítané opakování a bloky příkazů
  - Průhlednost - využití v programech
  - Jednoduché animace
- Práce v úrovni Pokročilý (od 9 let)
  - Seznam témat
  - Komentáře

  - Místní nabídky
  - Další tlačítka nad pracovní plochou

První kroky pokročilého programátora

- Animace
- Nastavení vlastností Baltíka
- Aplikace a multimédia

Data a datové typy

- Literál a jednoduché datové typy
- Konstanty a jednoduché proměnné
  - Systémové konstanty
- Strukturované datové typy

Program v akci

- Programové konstrukce
- Pomocníci
- Práce s číselnými daty

Vstupy a výstupy

Co je to Baltík

Baltík je výukový programovací nástroj pro děti od 4 let. **Baltík** je také hlavní hrdina tohoto programu - malý kouzelník, který dokáže vykonávat různé příkazy a čarovat obrázky (**předměty**) ve své **scéně**. Pomocí Baltíka děti rychle pochopí, co to je počítač, jak se ovládá a jak se programuje. To vše formou hry.

Baltík také slouží k procvičování logického myšlení. Neklade žádné nároky na vědomosti, požaduje pouze hravost a fantazii.

Program Baltík má tři režimy s rostoucí složitostí a možnostmi: **Skládat scénu** , **Čarovat scénu**, **Programovat**. Mezi jednotlivými režimy se přepínáte klepnutím na záložku s názvem režimu nebo výběrem z nabídky ***Zobrazit***.

1. Skládat scénu: První režim slouží k ovládnutí počítače. Děti se naučí ovládat myš, nakreslit obrázek, uložit soubor, načíst soubor, pojmenovat soubor. To vše pouze pomocí obrázků a ikon.

2. Čarovat scénu: Ve druhém režimu se děti už neučí ovládat počítač, ale kamaráda Baltíka. Naučí se v něm zadávat správně příkazy, které pak Baltík bez odmlouvání plní. Příkazy se zadávají pouhým klepnutím na příslušnou ikonu. Baltík každý příkaz ihned po jeho zadání provede. V tomto režimu se děti učí rozložit každou složitější akci na řadu nejjednodušších příkazů, kterým Baltík rozumí.

3. Programovat: Nejvyšší a nejzajímavější režim. V této úrovni děti při vytváření svého vlastního programu nejlépe pochopí, co to je program a co to je počítač. Program děti vytvářejí snadno, protože pouze skládají za sebe prvky, které znamenají příkazy pro Baltíka. Po stisknutí tlačítka START (PLAY) se takto vytvořený program spustí.

Tlačítko ***Start***

Baltík je cesta od prvního kontaktu s počítačem přes pochopení jeho činnosti až po osvojení si nejdůležitějších postupů při algoritmizaci úloh.

Předměty a banky předmětů

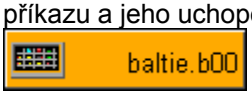
Rozměry předmětu

Baltíkův **předmět** je obrázek o rozměrech 39x29 bodů. Baltík umí předměty vyčarovat, odstranit a animovat. Může při tom použít již připravené předměty nebo předměty, které mu **nakreslíte** sami. Jednotlivé předměty jsou uloženy v **bankách předmětů**.

Ukázku banky předmětů zvětšíte tak, že na ni klepnete myší

Každá **banka předmětů** obsahuje 150 předmětů(10 řad po 15 předmětech), které můžete vložit do scény nebo které může Baltík vyčarovat.

Pozn.: V režimu **Programovat** můžete načíst banku ve svém programu. Pro vytvoření příslušného příkazu a jeho uchopení slouží tlačítko ***Uchopit***.



Tlačítko ***Uchopit***

Scéna

Ukázku scény zvětšíte tak, že na ni klepnete myší

Scéna je plocha, na kterou můžete v režimu **Skládat scénu** vkládat **předměty.** Ve vyšších režimech se po ní navíc Baltík i pohybuje.

Na scénu se vejde **150** předmětů uspořádaných do **10** řad po **15** předmětech.

Scénu, kterou program zobrazuje, na kterou vkládáte předměty a po které se Baltík pohybuje, označujeme jako **pracovní scéna**.

Pracovní scéna je na počátku černá.

Scény můžete mít **uloženy na disku**. Program může mít uloženo **100** vlastních scén. Tyto scény jsou uloženy v souborech se stejným názvem, jaký má soubor s programem, a s příponami ***.s00*** až ***.s99***.

Baltík umí scénu načíst a uložit i **v programu** (viz prvek **Obrazovka**).

Rozměry scény:

Rozměry scény můžeme udávat buďto v políčcích o rozměrech Baltíkových předmětů nebo v obrazovkových bodech. Scéna má rozměry:

- Ü **10×15 políček** (10 řádků po 15 sloupcích, přičemž řádky číslujeme od 0 do 9 a sloupce od 0 do 14) nebo
- Ü **290×585 bodů** (290 linek po 585 bodech přičemž linky číslujeme od 0 do 289 a body od 0 do 584).

Režim Skládat scénu

Uchovávání scén v souborech

Skládání scény je nejjednodušším režimem programu. V tomto režimu je na obrazovce **scéna**, na kterou můžete vkládat **předměty** a z jednotlivých předmětů skládat složitější obrázky.

Ukázku obrazovky zvětšíte tak, že na ni klepnete myší

Při práci se scénou můžete:

Vložit do scény nový předmět
Odstranit předmět ze scény
Přesouvat či kopírovat předměty
Vložit scénu do programu
Nakreslit vlastní předmět
Požádat o nápovědu
Ukončit práci s Baltíkem
Scény je navíc možno **uchovávat na disku**.

Vložení předmětu do scény

Chcete-li do scény vložit nějaký Předmět:,

1. Klepněte na tlačítko ***Předměty*** nebo kamkoliv na šedou plochu kolem scény, kde se ukazatel myši změní na mřížku.
2. Není-li v následně otevřeném okně ***Výběr předmětu*** zobrazena požadovaná banka, vyberte ji klepnutím na záložku. Můžete také použít tlačítka ***Předchozí banka*** a ***Další banka***
3. Klepněte na požadovaný předmět. Tím jej uchopíte a vrátíte se zpět do scény.
4. Umístěte předmět do jeho budoucí pozice na scéně a klepněte.

Tlačítko ***Předměty***

Tlačítko ***Předchozí banka***

Tlačítko ***Další banka***

Odstranění předmětu ze scény

Chcete-li ze scény odstranit nějaký předmět:

1. Klepněte levým tlačítkem na odstraňovaný předmět.
2. Přesuňte předmět na okolní šedou plochu tak, aby se na něm objevilo tmavošedé přeškrtnutí a klepněte levým tlačítkem.

Přesouvání a kopírování předmětů

Chcete-li předmět přesunout nebo zkopírovat do nové pozice:

1. Uchopte předmět tím, že na něj klepnete. Chcete-li jej **přesunout**, použijte **levé** tlačítko, chcete-li jej **zkopírovat**, použijte **pravé** tlačítko.
2. Přesuňte předmět, resp. jeho kopii do nové pozice.
3. Klepněte stejným tlačítkem, jako při uchopení předmětu.

Vedle popsaného postupu můžete použít i klasický způsob přetažení předmětu do nové pozice:

1. Najeďte na předmět a uchopte jej stisknutím příslušného tlačítka myši (levého pro přesun, pravého pro kopírování).
2. Při stisknutém tlačítku přeneste předmět do nové pozice
3. Puštěním tlačítka myši předmět do nové pozice umístěte.

Vložení scény do programu

V režimu **Programovat** můžete načíst vytvořenou scénu ve svém programu. Pro vytvoření příslušného příkazu a jeho uchopení slouží tlačítko ***Uchopit***.

Tlačítko ***Uchopit***

Nakreslení vlastních předmětů

Chcete-li upravit stávající předměty nebo nakreslit předměty vlastní, stiskněte tlačítko ***Kreslení***. Tím spustíte grafický editor **Paint,** který vám nabídne řadu užitečných **nástrojů**.

Tlačítko ***Kreslení***

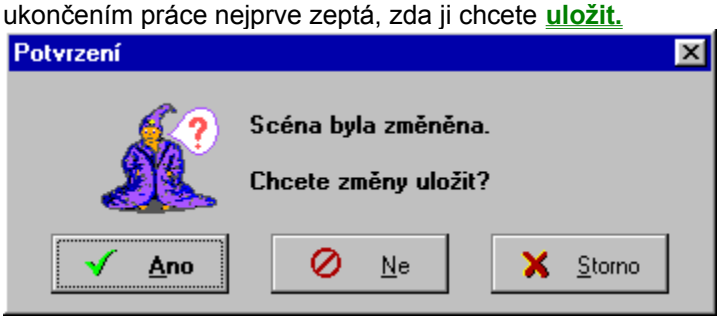
Vyvolání nápovědy

Stiskem tlačítka ***Nápověda*** vyvoláte kdykoliv tuto nápovědu.

Ukončení práce s Baltíkem

Stiskem tlačítka ***Konec*** ukončíte práci s celým programem. Pokud jste změnili scénu, Baltík se vás před

ukončením práce nejprve zeptá, zda ji chcete **uložit.**



- Stisknete-li **Ano**, scéna se uloží a Baltík se ukončí,
- stisknete-li **Ne**, Baltík se ukončí aniž by se scéna uložila,
- stisknete-li **Storno**, dialogové okno se zavře a vrátíte se zpět k původní práci, aniž by se cokoliv změnilo.

Uchovávání scén v souborech

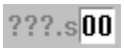
Vytváření scény

Při práci s diskem můžete:

Uložit scénu na disk
Načíst scénu z disku
Otevřít novou, prázdnou scénu
Přesunout se na další nebo předchozí scénu

Tip: Dříve uloženou **scénu opravíte** tak, že ji nejprve **načtete** a po opravě ji znovu **uložíte.**

Název souboru se scénou



Přibližně uprostřed šedé plochy nad vytvářenou scénou je zobrazen název souboru, v němž je scéna uložena.

Pokud takový soubor ještě neexistuje, zobrazí se pouze **tři otazníky**. Při ukládání nenazvané scény do souboru budete na tento název dotázáni.

Názvy souborů se scénami sestávají z nejvýše 8znakového jména a tříznakové přípony. Jméno smí obsahovat pouze písmena, číslice a znaky

! # $ % & $ ( ) - @ _ { }

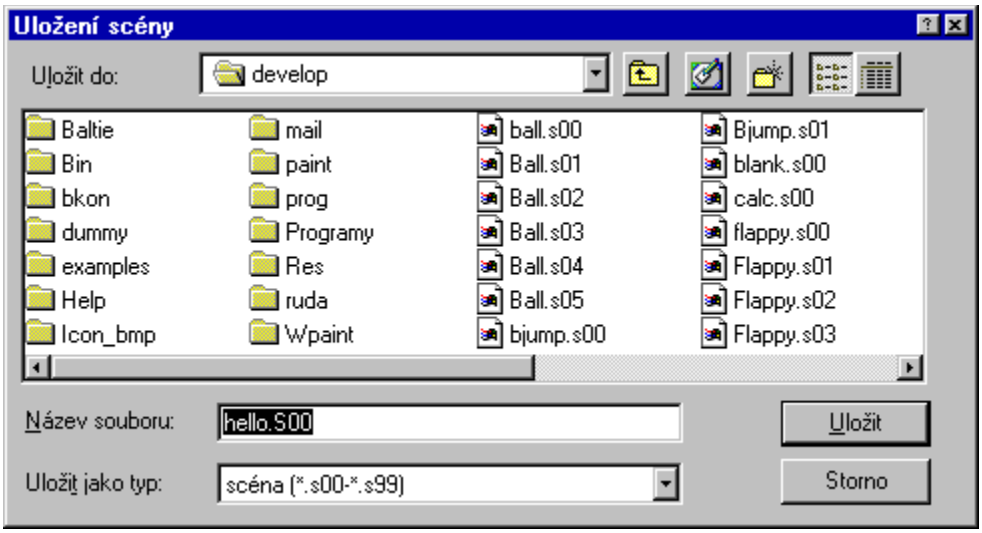
Názvy souborů se scénami mají přípony **.s00**, **.s01** atd. až **.s99**. Číslo, které je součástí přípony, nastavíte v poli ***Číslo scény***.

Uložení scény na disk

Hotovou scénu uložíte na disk tlačítkem ***Uložit scénu***. Scéna se uloží do souboru s názvem, který je zobrazen v poli nad scénou (viz název souboru se scénou výše.

Tlačítko ***Uložit scénu***

Chcete-li přiřadit souboru (a tím i scéně) vlastní jméno, zadejte příkaz ***Soubor → Uložit jako***, čímž otevřete dialogové okno ***Uložit scénu***. Stejné dialogové okno se otevře i v případě, kdy jsou při obyčejném ukládání zobrazeny nad scénou místo názvu souboru **tři otazníky.**



Nezadáte-li při ukládání souboru příponu sami, použije se standardní přípona s číslem, které bylo uvedeno v poli nad scénou.

Zadáte-li **název existujícího souboru**, program se nejprve zeptá, zda jej chcete přepsat. **Nehrozí tedy, že byste nějaký soubor přepsali nechtěně, aniž byste o tom věděli.**

Budete-li chtít používat scénu později v nějakém programu (režim **Programovat**), uložte ji se stejným názvem, jako je název souboru programem, v němž budete chtít scénu použít. Jmenuje-li se soubor s vaším programem např. *Hra1.bpr*, pak příslušné scény k tomuto programu ukládejte s názvy *Hra1.s00* až *Hra1.s99*.

Načtení scény z disku

Chcete-li načíst uloženou scénu, stiskněte tlačítko ***Otevřít scénu*** a v následně otevřeném dialogovém okně zadejte název příslušného souboru.

Tlačítko ***Otevřít scénu***

Pokud jste předchozí scénu od jejího načtení změnili, program se nejprve zeptá, zda ji chcete uložit.

Načtení nové, prázdné scény

Tlačítkem Nová scéna smažete aktuální scénu a můžete skládat novou.

Tlačítko ***Nová scéna***

Pokud jste předchozí scénu od jejího načtení změnili, program se nejprve zeptá, zda ji chcete uložit.

Přepínání mezi scénami s týmž jménem

Mezi jednotlivými scénami k jednomu programu, tj. mezi scénami se stejným názvem ale různými příponami, se můžete přepínat tlačítky ***Předchozí scéna*** a ***Další scéna***.

Tlačítko ***Další scéna***

Tlačítko ***Předchozí scéna***

Kreslení předmětů - grafický editor Paint

Nástroje editoru Paint Jak nakreslit svůj vlastní předmět

Chcete-li spustit Editor Paint:

- Ü stiskněte tlačítko ***Kreslení*** nebo
- Ü klepněte pravým tlačítkem myši na předmět v programu a vyberte ***Kreslení*** nebo
- Ü jednoduše přetáhněte předmět levým tlačítkem myši na tlačítko ***Kreslení.***

Tlačítko ***Kreslení***

Ukázku obrazovky grafického editoru Paint zvětšíte tak, že na ni klepnete myší

Popis grafického editoru Paint

Paint je grafický editor, který slouží především ke kreslení předmětů, používaných v programovacím nástroji **Baltík**. Mezi hlavní rysy tohoto grafického editoru patří:

- Ü úplná sada **běžných nástrojů pro kreslení** (body, čáry, obdélníky, elipsy, gumy, spreje)
- Ü komfortní **práce s výřezy** (přesun, kopie, převracení, otáčení, změna velikosti)
- Ü zvláštní podpora práce s Baltíkovými **předměty** (přesun, kopie, průhledná barva)
- Ü práce se zvláštním formátem obrázku - s Baltíkovou **bankou předmětů** (soubory **.b00** - **.c99**)
- Ü možnost importu a exportu souborů **.bmp** (***Soubor*** → ***Import, Soubor*** → ***Export***).

V levé části okna je připraven **sloupec nástrojů** uspořádaných do skupin. Po stisku levého tlačítka myši na ikoně některé skupiny se rozbalí nabídka nástrojů v této skupině. Paint nabízí následující skupiny nástrojů (**klepnutím na ikonu nebo název nástroje se přesunete na příslušnou nápovědu**):

Lupa

Kreslení

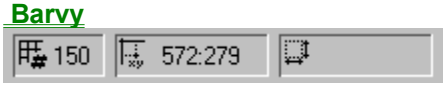
Nástroje

Výřezy

Práce s předměty

Zeď

Barvy



Stavové informace

Přepínač bank

Nástroje grafického editoru Paint

Jak nakreslit svůj vlastní předmět

Paint nabízí následující skupiny nástrojů (klepnutím na ikonu nebo název nástroje se přesunete na příslušné téma):

Lupa

Kreslení

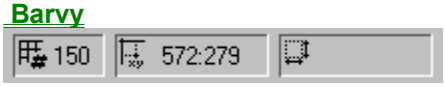
Nástroje

Výřezy

Práce s předměty

Zeď

Barvy



Stavové informace

Přepínač bank

1) Lupa

Lupou si můžete libovolnou část banky **2x**, **4x**, **6x** nebo **8x** zvětšit, případně volbou **1x** zobrazit opět celou banku nezvětšenou.

2) Kreslení

V této skupině najdete základní nástroje pro kreslení samostatných **bodů**, kreslení **od ruky**, kreslení **paprsků** vycházejících z jednoho bodu, rovných **čar**, prázdných a vyplněných **obdélníků** a **elips**.

Při vybrání kteréhokoliv z uvedených kreslících nástrojů s výjimkou nástroje pro kreslení bodů se vlevo pod sloupcem nástrojů zobrazí **měřítko tloušťky čáry**. Na začátku je tloušťka čáry nastavena na nejtenčí (jeden obrazovkový bod).

Pozn.: Pro kreslení používejte **levé tlačítko** myši.
Pravé tlačítko myši slouží k **nabírání barvy** - funkce **kapátko**: chcete-li nastavit pro kreslení některou z barev v obrázku již použitých, klepněte pravým tlačítkem myši na bod, který má požadovanou barvu.

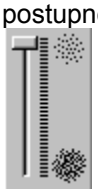
3) Nástroje

Další skupina obsahuje nástroje pro malování ploch: oválný a obdélníkový **sprej**, **vyplňování** souvislé plochy stejné barvy, barevnou, kruhovou a obdélníkovou **gumu** a **text**.

Sprej

Sprej funguje tak, že nejdříve někde v kreslící oblasti stisknete levé tlačítko myši a táhnutím určíte velikost a tvar oblasti, pomocí níž budete sprej nanášet. Pak tlačítko pustíte a dalšími tahy při stisknutém tlačítku již sprej používáte.

Při rozhodování mezi obdélníkovým a oválným sprejem mějte na paměti, že **obdélníkový sprej** pokrývá zadanou oblast stejnoměrně a hodí se proto lépe k vytvoření rovnoměrně vybarvených ploch, kdežto **oválný sprej** pokrývá zadanou oblast nejhustěji uprostřed a směrem k okrajům oblasti se hustota nanášené barvy snižuje. Hodí se proto lépe pro kreslení různých nápisů, mraků a dalších objektů s postupně slábnoucími okraji.



Při používání spreje se vlevo pod sloupcem nástrojů zobrazí měřítko hustoty spreje (počet bodů, vykreslených za jednotku času). Na začátku je hustota nastavena na nejmenší.

Vyplňování

Nástroj ***Vyplňování*** změní na barvu popředí barvu všech bodů v souvislé oblasti stejně barevných bodů obsahující bod, na nějž jste klepli,.

Teď ještě jednou **méně učeněji**. Nástroj ***Vyplňování*** slouží ke změně barev bodů **na barvu popředí.** Doporučený postup je následující:

1. Zadejte barvu popředí.

2. Zvolte nástroj ***Vyplňování***.
3. Klepnete na některý z bodů, jejichž barvu chcete změnit.

Bod svoji barvu změní a předá požadavek na změnu barvy všem svým **stejně barevným sousedům**, ale jenom jim. Ty po něm tuto akci zopakují, tj. změní svoji barvu a předají tento požadavek svým stejně barevným sousedům. Barva se tak rozlévá po obrázku až se zarazí o body, které mají jinou barvu, než měl bod, na který jste klepli.

Pozn.: Vyplňování je vhodné na přebarvení jednobarevné plochy. Chcete-li změnit barvu posprejované plochy, použijte raději nějakou **gumu** (viz dále).

Pozn.: Chcete-li přebarvit nějakou ohraničenou plochu, prověřte, že **v ohraničení není žádná díra**. Jinak by se vám mohl barva touto dírkou vylít i mimo požadované hranice.

Guma

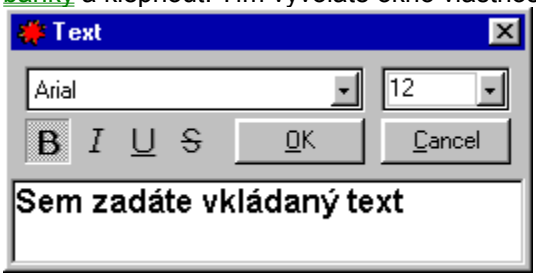
Gumy fungují podobně jako sprej: prvním táhnutím určíte velikost a tvar gumy a dalšími tahy gumujete.

Kruhová a **obdélníková guma** mění barvu všech bodů, přes něž přejedete zadanou oblastí, na barvu pozadí. Naproti tomu **barevná guma** mění pouze barvu bodů, které mají barvu popředí.

Tip: Vhodnou volbou barvy popředí a pozadí můžete za pomoci barevné gumy snadno **změnit jednu barvu na jinou**.

Text

Text je posledním nástrojem z této skupiny. Po jeho vybrání musíte přesunout kurzor myši na plochu banky a klepnout. Tím vyvoláte okno vlastností textu.



V tomto okně zadáte text, který chcete vložit do obrázku, písmo, jímž chcete text napsat, jeho velikost a styl (**B=tučné**, *I=kurziva*, U=podtržené, ~~S=přeškrtnuté~~). Výsledek všech nastavení vidíte ihned na textu v dialogovém okně.

Po potvrzení správnosti zadání stiskem ***OK*** se text objeví na obrazovce jako výřez. Tento výřez můžete dále **posouvat** (musíte na něj ale nejprve najet myší tak, aby její kurzor měl tvar do čtyř stran ukazující šipky), měnit jeho **rozměr** (kurzor myši musí mít tvar dvoustranné šipky, jejíž směr naznačuje i směr možné změny rozměru výřezu) a **barvu textu** (tu změníte klepnutím na příslušnou barvu v paletě barev).

Stiskem pravého tlačítka na výřezu a výběrem volby ***Vlastnosti*** otevřete opět okno vlastností, kde můžete změnit jak vlastní text, tak vlastnosti jeho písma.

4) Výřezy

Pomocí nástrojů v této skupině můžete pracovat s obdélníkovým výřezem obrázku, který určíte prvním nástrojem: **výřez**.

Najedete-li na zadaný výřez myší tak, aby její kurzor měl tvar do čtyř stran ukazující šipky, můžete tento výřez levým tlačítkem **přesouvat** a pravým tlačítkem **kopírovat**.

Najedete-li myší na okraj výřezu tak, aby kurzor měl tvar dvojité šipky, můžete tahem za hranu při stisknutém levém tlačítku měnit jeho rozměr a odpovídajícím způsobem i rozměr jeho obsahu.

Dokud je výběr označen, můžete jej pomocí dalších nástrojů převracet a otáčet.

Při používání výřezů se vlevo pod sloupcem nástrojů zobrazí přepínač průhledného (dolní ikona) a neprůhledného (horní ikona) zobrazování posouvaných či kopírovaných výřezů. Při neprůhledném zobrazování obsah posouvaného (kopírovaného) výřezu v nové pozici zcela zakryje původní obrázek. Při průhledném zobrazování bude v nové pozici na místě bodů v **barvě pozadí** prosvítat původní obrázek.

5) Práce s předměty

Na rozdíl od běžných grafických editorů Paint podporuje práci s celými **předměty.**

První nástroj této skupiny zapíná práci s předměty. Zapnutou práci s předměty poznáte podle toho, že banka je rozdělena mřížkou a ukazatel myši má tvar lupy.

Máte-li zapnutou práci s předměty, můžete jednotlivé předměty levým tlačítkem myši přesouvat, pravým tlačítkem myši kopírovat.

POZOR! Na rozdíl od skládání scény, zde nestačí pouze na předmět klepnout. Zde musíte předmět "držet" (tj. mít stisknuté příslušné tlačítko) po celou dobu jeho přesunu.

Stisknete-li na některém předmětu pouze levé tlačítko, předmět se **zvětší** na největší možnou velikost a vy jej můžete upravovat.

Druhý nástroj ovládá nastavení **průhlednosti.** Po výběru tohoto nástroje se u každého předmětu zobrazí v levém horním rohu čtvereček, označující **průhlednou barvu.**

Stiskem levého tlačítka myši na některém předmětu změníte jeho průhlednou barvu na **barvu popředí.** (barva popředí se stane jeho **průhlednou barvou**) Dalším stiskem levého tlačítka tuto barvu vypnete a předmět se stane neprůhledným.

Posledním nástrojem z této skupiny vypnete práci s předměty a zapnete **práci s celou bankou**. Dalším výběrem tohoto nástroje práci s celou bankou opět vypnete a nastavíte původní režim, tj. buď práci s předměty nebo nastavování průhlednosti podle toho, který nástroj jste měli zapnutý předtím.

6) Zeď

Tento přepínač slouží k omezení kreslení na jeden předmět. Máte-li Zeď zapnutu, bude se při kreslení měnit pouze jeden předmět - ten, ve kterém kreslení započalo.

7) Barvy

Při spodním okraji aplikačního okna editoru Paint se nachází paleta barev. Dva překrývající se obdélníky vlevo zobrazují nastavené barvy popředí a pozadí (tj. barvy pera a papíru).

Barvou popředí jsou kresleny všechny body, čáry, okraje obdélníků a elips, spreje a jsou jí vyplňovány souvislé oblasti. Barva popředí je také důležitá pro barevnou **gumu**, jež pouze mění barvu bodů, které mají barvu popředí.

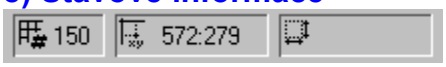
Barvou pozadí jsou vyplňovány obdélníky a elipsy a používají ji také obě gumy, které tuto barvu přiřazují mazaným bodům.

Pokud při používání výřezů přepnete zobrazování na **průhledné**, body mající aktuální barvu pozadí se při přesouvání a kopii výřezu nezobrazují a na jejich místě prosvítá původní obrázek.

V grafickém editoru Paint můžete pro popředí a pozadí rychle zadat jednu ze 16 barev, které používá Baltík. Tyto barvy jsou zobrazeny v paletě vpravo vedle ukazatele nastavené barvy popředí a pozadí. Stiskem levého tlačítka myši na některé barvě nastavíte tuto barvu pro **popředí**, stiskem pravého tlačítka myši nastavíte tuto barvu pro **pozadí**.

Tip: Barvu popředí můžete měnit také přímo při kreslení stiskem pravého tlačítka myši na bodu zvolené barvy - funkce **kapátko**.

8) Stavové informace



Vpravo vedle přepínače bank jsou stavové informace. Ty postupně zobrazují číslo předmětu, na kterém se právě nacházíte, souřadnice ukazatele myši a velikost zadaného výřezu.

9) Přepínač bank

Vpravo vedle palety barev je přepínač **bank předmětů.** Skládá se z tlačítka ***Předchozí banka***, pole ***Číslo***

banky a tlačítka ***Další banka***. Tlačítky ***Předchozí banka*** a ***Další banka*** se snadno přepínáte mezi sousedními bankami. Přitom za sousední jsou považovány i banky č. 0 a 199.

Chcete-li zobrazit banku určitého čísla, klepněte levým tlačítkem myši na pole ***Číslo banky*** a vložte číslo požadované banky.

Názvy bank

Baltík očekává, že jméno banky bude shodné se jménem programu nebo že se banka bude jmenovat ***Baltie***. Přípona je pak odvozena z čísla banky podle následujícího klíče:

- banky **0** až **99** mají příponu **.b##** kde **##** zastupuje číslo banky (**00**, **01**, **02** až **99**),
- banky **100** až **199** mají příponu **.c##**, kde **##** zastupuje poslední dvojčíslí čísla banky.

Soubor s bankou **5** má tedy příponu **.b05**, soubor s bankou **105** má příponou **.c05**.

Tip: Při číslování bank byste měli používat čísla od 10 do 99 a od 110 do 199, abyste nezakrývali banky dodávané se systémem včetně případných budoucích rozšíření.

Jak si nakreslit svůj vlastní předmět

Nástroje editoru Paint

1) Zapnutí práce s předměty

Po spuštění programu Paint je práce s **předměty** vždy **zapnuta.** Pokud jste ji z nejrůznějších důvodů vypnuli, musíte nejprve **zapnout práci s předměty.**

2) Výběr předmětu

Vyberte si předmět, s nímž chcete pracovat. Chcete-li nakreslit nový předmět, zvolte některý z prázdných předmětů, chcete-li změnit již existující předmět, zvolte tento předmět.

Klepněte na zvolený předmět myší. Tím jej zvětšíte na největší velikost a můžete jej začít upravovat.

3) Výběr nástroje

Z levého sloupce **nástrojů** vyberte nástroj, kterým chcete kreslit. Nevyberete-li žádný nástroj, budete kreslit rovné čáry - úsečky.

4) Výběr barvy

Nevyhovují-li vám nastavené barvy, vyberte z dolního řádku barev levým tlačítkem myši barvu popředí a pravým tlačítkem myši barvu pozadí.

5) Kreslení

Nyní již můžete kreslit nebo upravovat vámi zvolený předmět.

Pozn.: Body 2 až 4 můžete libovolně opakovat a kombinovat. Teprve až jste s výsledným předmětem spokojeni, přistupte k bodu 5.

6) Uložení banky

Upravenou banku můžete uložit příkazem ***Soubor*** → ***Uložit***, případně ***Soubor*** → ***Uložit jako***. Po uložení banky můžete buď pokračovat v editaci předmětů této banky nebo **zvolit jinou banku.**

7) Opuštění programu

Program opustíte např. tlačítkem ***Opusť Paint***:

Tlačítko ***Opusť Paint***

Pokud jste před opuštěním programu změněnou banku neuložili, budete dotázáni při opouštění, zda chcete provedené změny uložit.

Po opuštění grafického editoru Paint již můžete nově vytvořené předměty používat ve svých scénách a programech.

Režim Čarovat scénu

Ukázku obrazovky zvětšíte tak, že na ni klepnete myší

Čarování scény s pomocí Baltíka je druhým režimem programu. Zde se již objevuje samotný Baltík, který na počátku práce stojí v levém dolním rohu **scény** a čeká na vaše příkazy. V tomto režimu se naučíte ovládat postavičku Baltíka a pochopíte jeho základní příkazy. Naučíte se s ním pohybovat a naučíte jej **čarovat předměty.**

Pozn.: V režimu **Programovat** můžete navíc načíst vytvořenou scénu ve svém programu. Pro vytvoření příslušného příkazu a jeho uchopení slouží tlačítko ***Uchopit***.

Tlačítko ***Uchopit***

V režimu Čarovat scénu můžete zadávat následující příkazy:

Tlačítko ***Vlevo vbok***
Baltík provede obrat vlevo o 90°. Pozor! Baltík dělá obraty vždy ze svého pohledu!

Tlačítko ***Vpravo vbok***
Baltík provede obrat vpravo o 90°. Platí totéž, co u obratu vlevo.

Tlačítko ***Popojdi***
Není-li před Baltíkem neprůchodný předmět (např. okraj scény), popojde o jedno políčko tím směrem, kterým se dívá.

Tlačítko ***Vyber a čaruj***
Baltík vyčaruje předmět, který si vyberete z **banky předmětů**.

Pozn.: Banku předmětů můžete otevřít také stiskem levého tlačítka myši kdekoliv na šedé ploše kolem scény, kde se ukazatel myši změní na mřížku.

Tlačítko ***Čaruj***
Po vyčarování prvního předmětu se vpravo vedle tlačítka Vyber a čaruj objeví tlačítko s naposledy vyčarovaným předmětem. Po jeho stisku Baltík před sebou tento předmět vyčaruje (samozřejmě pouze není-li před ním okraj scény).

Tlačítko ***Kreslení***
Chcete-li nakreslit svůj vlastní předmět, případně změnit některý už existující předmět, spusťte tlačítkem ***Kreslení*** grafický editor **Paint.**



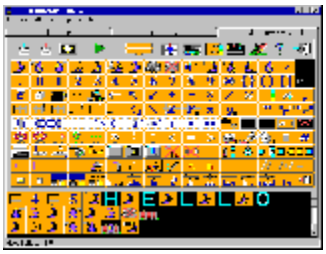
Otevřít scénu ***Uložit scénu*** ***Nová scéna***

Pro **ukládání scény**, její **načtení,** a otevření **nové scény** platí vše, co jsme řekli v pasáži **Uchovávání scén v souborech**

Připomínáme, že scény mají příponu **.s00**, **.s01** atd. až **.s99**. Budete-li chtít svou scénu používat později v nějakém svém programu (režim **Programovat**), uložte ji se stejným názvem, jako je název souboru s tímto programem.

Režim Programovat

Příkazy základní nabídky



Ukázku obrazovky s programem zvětšíte tak, že na ni klepnete myší

Programování je nejvyšší režim Baltíka. Umožňuje definovat předpis - program, který vysvětlí Baltíkovi jak splnit zadaný úkol. Pomocí Baltíka můžete naprogramovat příběhy, výukové programy nebo také hry, ovládané z klávesnice nebo myší.

Program je předpis, který Baltíkovi říká, jak má splnit zadanou úlohu. Skládá se z **posloupnosti příkazů**, které Baltík postupně plní.

Každý **příkaz je tvořen jedním nebo několika prvky** (ikonami) poskládanými v přesně určeném pořadí.

Příkazy se provádějí jeden za druhým ve stejném pořadí, v jakém čtete slova v knize, tj. **zleva doprava a shora dolů**.

Programování Baltíka má dvě úrovně, mezi kterými se přepínáte zadáním příslušného povelu v nabídce ***Program***:

1. Začátečník: Po nastavení této úrovně budete mít dostupné příkazy pro Baltíkův pohyb, čarování, **počítané opakování**, pípnutí a načtení scény nebo banky.

Tyto možnosti odpovídají možnostem **Baltíka verze 2.0**. Doporučujeme vám zvládnout nejprve beze zbytku tuto úroveň, a teprve pak přejít do úrovně **Pokročilý**!

2. Pokročilý: Tato úroveň již umožňuje plné programování včetně příkazů pro řízení běhu programu, práce s konstantami a proměnnými, animacemi a práce s multimediálními soubory.

Obě úrovně mají společné základy takže **vytváření a úpravy programů,** stejně jako **využití tlačítek nad pracovní plochou** je pro obě úrovně téměř totožné.

Vytváření a úpravy programů

Počítané opakování a bloky příkazů

Mezi společné základy obou úrovní patří:

- **Základní operace**, které jste se naučili při **skládání scény**
  - **Vložení prvku**
  - **Přesouvání, kopírování a odstraňování prvků**
- **Používání prvků Mezera a Konec řádku**
- **Práce s celými bloky prvků**
  - **Označování bloků**
  - **Přesun, kopírování a odstraňování bloků prvků**
  - **Vodorovný posun prvků bloku**
- **Načítání programů** jejich **ukládání** a otevírání **nových**
- **Spouštění programů**
- **Práce s předměty**

Základní operace

Při vytváření a úpravách programů využijete dovednosti, které jste se naučili při **skládání scény.** Proto bude tento výklad stručnější. Kdo potřebuje podrobnější výklad, najde jej v příslušné pasáži.

Vložení prvku

Nový prvek vložíte do programu tak, že na ně v **panelu příkazů** **klepnete levým** tlačítkem myši. Tím jej uchopíte a můžete jej přenést dolů na libovolné místo programu. Tam znovu klepnete a tím prvek umístíte.

Můžete také použít techniku drag and drop (Popadni-Přenes-Pusť) a tlačítko do nové pozice **přetáhnout** - v tom případě můžete používat i pravé tlačítko myši.

Po přechodu do úrovně **Pokročilý** se objeví ještě **další možnosti.**

Tip: Chcete-li umístit prvek na konec řádku, nemusíte jej dopravovat až na místo. Stačí jej "upustit" kdekoliv za koncem řádku a Baltík jej sám na konec řádku přesune.

Přesouvání, kopírování a odstraňování prvků

V programu můžete prvky libovolně přeskládávat. Prvky **přesouváte levým** tlačítkem a **kopírujete pravým**. Přitom je jedno, zda prvek přetáhnete do nové pozice klasickým způsobem uchopit-přetáhnout-pustit, nebo zda použijete techniku klepnout-přenést-klepnout.

Prvek odstraníte z programu tak, že jej přesunete (levým tlačítkem myši) mimo pracovní plochu programu.

Používání prvků Mezera a Konec řádku

Většina z prvků na ***Panelu příkazů*** slouží k vytváření příkazů. Některé však slouží především ke zpřehlednění programu. Mezi nimi mají zvláštní postavení prvky ***Mezera*** a ***Konec řádku***.

Jedinou funkcí obou těchto prvků je **zpřehlednění programu** a **oddělení jednotlivých příkazů**. Narazí-li

na ně Baltík při vykonání programu, **neudělá nic** a přejde k dalšímu příkazu.

Prvek ***Mezera*** se používá především ve dvou případech:

- Ü **Oddělení příkazů.** Sousedí-li spolu v programu dva prvky, z nichž levý má ukončovat jeden příkaz a pravý má uvozovat příkaz následující, může se stát, že program náš příkaz špatně pochopí a přiřadí prvky do příkazů jinak. Takovéto situace se nejlépe vyřeší tím, že mezi tyto dva příkazy, tj. mezi ony sousedící prvky, vložíme prvek ***Mezera*** nebo ***Konec řádku***.
- Ü **Odsazení počátku řádku.** U profesionálních programů bývá zvykem odsazovat bloky příkazů od levého okraje tak, aby bylo dobře vidět, kde dotyčný blok příkazů začíná a kde končí. Tohoto odsazení dosáhneme nejlépe pomocí prvku ***Mezera***.

Prvek ***Konec řádku (Enter)*** se používá především ve dvou případech:

- Ü **Oddělení příkazů.** Prvek ***Konec řádku*** zde řeší stejný problém, o němž jsme před chvílí hovořili u prvku ***Mezera***.
- Ü **Ukončení řádku.** Jedním z charakteristických rysů dobrých programů je to, že jsou přehledné. Jedním z pravidel je používání řádků, které se celé vejdou na obrazovku.

Tip: Prázdný řádek vložíte do programu tak, že zkopírujete prvek ***Konec řádku***.

Tip: Chcete-li rozdělit řádek na dva, zkopírujte prvek ***Konec řádku*** mezi prvky, kterými bude první z řádků končit a druhý začínat.

Práce s celými bloky prvků

Při úpravách programu bývá často výhodné pracovat místo s jednotlivými prvky přímo s celými bloky prvků

Pozn.: Nebude-li vám něco z výkladu o práci s celými bloky jasné, zkuste se k tomuto tématu vrátit po prostudování sekce ***Počítané opakování a bloky příkazů.***

Označování bloků

Blok prvků můžeme označovat několika způsoby:

Vybírání po prvcích:

1. Najeďte myší na prvek, kterým bude blok začínat.
2. Ukažte kurzorem na ten jeho kvadrant (čtvrtinu), který je protilehlý směru, v němž bude blok pokračovat (Bude-li blok pokračovat např. vpravo dolů, ukažte na levou horní čtvrtinu prvku).
3. Stiskněte na klávesnici přeřaďovač ***Shift***.
4. Stisknete levé tlačítko myši a přesuňte kurzor tak, aby byly zašeděny všechny prvky budoucího bloku.
5. Pusťte levé tlačítko myši

Pozn.: Budete-li chtít pracovat s celými řádky, nezapomeňte vybrat do bloku příslušné prvky ***Konec řádku***.

Pozn.: Přejedete-li při výběru bloku myší dolů, přidají se do výběru všechny prvky od původní pozice kurzoru do konce řádku včetně prvku ***Konec řádku*** a zároveň prvky od počátku následujícího řádku po prvek, na nějž ukazuje myší kurzor.

Vybírání po blocích:

Při vybírání po blocích využíváme toho, že každým poklepáním (dvojitým klepnutím) na prvek se definovaným způsobem zvětší skupina vybraných prvků - blok:

Ü poklepáním na prvek vybereme **příkaz,** jehož je daný prvek součástí,

Ü poklepáním na vybraný příkaz vybereme celý řádek, v němž se daný prvek nachází,

Pozn.: Řádek můžete vybrat také jednoduchým klepnutím při stisknutém tlačítku ***Shift*** nebo ***Alt***.

Ü poklepáním na vybraný řádek vybereme celý blok stejně odsazených řádků,

Ü poklepáním na vybraný blok Baltík tento výběr rozšíří o další řádky podle následujícího algoritmu:

1. Je-li vybrán řádek, za nímž bezprostředně následují řádky s větším odsazením, vyberou se všechny následující více odsazené řádky.
2. Neplatí-li předpoklad bodu **1.** a nejsou-li ještě vybrány všechny sousední řádky se stejným či větším odsazením, přidají se do výběru.
3. Neplatí-li ani předpoklady bodu **2.**, přidá se do výběru řádek před vybraným blokem.

Příklad:
Máli váš program např. následující podobu (předpokládáme, že jako všichni správní programátoři poctivě odsazujete vnořené bloky), bude při postupném poklepávání na prvek v šestém řádku rozšiřovat výběr tak, jak je naznačeno v jednotlivých sloupcích (znak **@** označuje, že řádek je zahrnut do výběru):

	Program	**Počet poklepání**							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	**1.úroveň**	-	-	-	-	-	-	-	@
2	**1.úroveň**	-	-	-	-	-	-	@	@
3	**2.úroveň**	-	-	-	-	-	@	@	@
4	**2.úroveň**	-	-	-	-	@	@	@	@
5	**3.úroveň**	-	-	-	@	@	@	@	@
6	**3.úroveň**	-	@	@	@	@	@	@	@
7	**4. úroveň**	-	-	@	@	@	@	@	@
8	**3.úroveň**	-	-	-	@	@	@	@	@
9	**4. úroveň**	-	-	-	@	@	@	@	@
10	**2.úroveň**	-	-	-	-	-	@	@	@
11	**3.úroveň**	-	-	-	-	-	@	@	@
12	**1.úroveň**	-	-	-	-	-	-	-	@

Přesun, kopírování a odstraňování bloků prvků

S bloky prvků pracujeme stejně jako s jednotlivými prvky: přesouváme je pomocí levého tlačítka myši, kopírujeme je pomocí pravého tlačítka myši a mažeme je přesunutím mimo pracovní plochu.

Vodorovný posun prvků bloku

Chcete-li odsadit či přisadit několik sousedních řádků, tj. chcete-li před všechny tyto řádky přidat mezeru nebo naopak mezeru z počátku všech těchto řádků odebrat, stačí označit blok a pomocí **levého** tlačítka myši jej **posunout ve vodorovném směru**. Při pohybu myši se objeví tenký bílý rámeček, který ukazuje budoucí pozici přesouvaného bloku.

Pozn.: Podrobněji si o odsazování bloků povíme v sekci ***Počítané opakování a bloky příkazů.***

Tlačítka nad pracovní plochou

Tlačítka dostupná pouze v režimu Pokročilý

Klepnutím na tlačítko se přesunete na příslušné téma nápovědy:


Otevřít program


Uložit program


Nový program


Start


Scéna


Předměty


Kreslení


Nápověda


Konec

Protože byste význam tlačítek měli znát z úrovně **Skládat scénu**, zmíníme se hlavně o zvláštnostech úrovně **Programovat**

Uložení programu na disk

Program uložíte na disk stiskem tlačítka ***Uložit program***

Chcete-li přiřadit souboru (a tím i scéně) vlastní jméno, zadejte příkaz ***Soubor → Uložit jako***, čímž otevřete dialogové okno ***Uložit program***. Stejné dialogové okno se otevře i v případě, kdy jste požádali o obyčejné ukládání a program přitom ještě neměl přiřazené jméno.

Názvy souborů s programy mají přípony **.bpr** (**B**alties **pr**ogram).

Zadáte-li **název existujícího souboru**, program se nejprve zeptá, zda tento soubor chcete přepsat. **Nehrozí tedy, že byste nějaký soubor přepsali nechtěně, aniž byste o tom věděli.**

Otevření uloženého programu

Chcete-li načíst uložený program, stiskněte tlačítko ***Otevřít program***.

Pokud jste současný program od jeho načtení změnili, Baltík se nejprve zeptá, zda nechcete upravený nejprve program uložit.

Otevření nového programu

Chcete-li začít vytvářet nový program, stiskněte tlačítko ***Nový program***. Tím otevřete nový, prázdný program. Tento program nebude mít na počátku ještě jméno. To mu přidělíte až ve chvíli, kdy jej budete poprvé ukládat.

Pokud jste současný program od jeho načtení změnili, Baltík se nejprve zeptá, nechcete-li upravený program nejprve uložit.

Spuštění programů

Vytvořený program spustíte tlačítkem ***Start***

Klepnete-li na ně **pravým** tlačítkem myši, spustí se **pouze označený blok**. Tato možnost je vhodná zejména k **ověřování správnosti** napsaných částí programu.

Scéna

Stiskem tlačítka ***Scéna*** se přepnete do režimu **Skládat scénu**, ve kterém můžete vytvořit scénu, uložit ji a vrátit se zpět k programování

Práce s předměty

Stiskem tlačítka ***Předměty*** otevřete (stejně jako v režimu **Skládat scénu**) okno ***Výběr předmětu***. V tomto okně zadáte, který **předmět** ze které banky chcete do programu vložit.

V okamžiku, kdy Baltík při plnění programu na tento předmět v programu narazí, tak jej na pracovní scéně vyčaruje (samozřejmě před sebou nesmí mít v daném okamžiku okraj scény).

Výběr čarovaného předmětu při běhu programu

Klepnete-li na ikonu ***Předměty*** pravým tlačítkem myši, rozevře se místní nabídka. Zadáte-li volbu ***Uchopit***, program vám předá prvek ***Čaruj vybraný předmět***, který můžete vložit do programu **všude tam, kam můžete vložit obyčejný předmět**.

Když Baltík při plnění programu narazí na prvek ***Vybraný předmět***, otevře okno ***Výběr předmětu,*** v němž **uživatel vybere předmět**, **který Baltík vzápětí vyčaruje**. Pokud uživatel žádný předmět nevybere, vyčaruje Baltík prázdné políčko, tedy předmět č. **0**.

Vložíte-li za prvek ***Vybraný předmět*** **číslo**, otevře se okno ***Výběru předmětů*** na bance s tímto číslem.

Vložíte-li za prvek ***Vybraný předmět*** nějaký předmět, bude otevřena banka s tímto předmětem.

Vytvoření vlastního předmětu

Stiskem tlačítka ***Kreslení*** spustíte grafický editor **Paint,** pomocí nějž můžete vytvořit vlastní **předměty** i celé **banky** předmětů.

Úprava předmětu za běhu programu

Klepnete-li na ikonu ***Kreslení*** pravým tlačítkem myši, rozevře se místní nabídka. Zadáte-li volbu ***Uchopit***, program vám předá prvek ***Spusť editor Paint***, který můžete vložit do programu **všude tam, kam můžete vložit obyčejný příkaz** - např. příkaz ***Popojdi***.

Když Baltík při plnění programu narazí na prvek ***Spusť editor Paint***, spustí grafický editor **Paint,** a po jeho ukončení pokračuje v plnění programu.

Vložíte-li za prvek ***Spusť editor Paint*** **číslo,** editor **Paint,** se otevře na bance s tímto číslem.

Vložíte-li za prvek ***Spusť editor Paint*** nějaký předmět, **Paint,** se otevře na bance s tímto předmětem a tento předmět bude zvětšen.

Vyvolání nápovědy

Stiskem tlačítka ***Nápověda*** vyvoláte kdykoliv tuto nápovědu.

Nápovědu ke konkrétnímu prvku můžete získat také tak, že prvek přesunete nad tlačítko ***Nápověda***.

Ukončení práce s Baltíkem

Stiskem tlačítka ***Konec*** ukončíte práci s celým programem. Pokud jste program od jeho načtení změnili, Baltík se vás před ukončením práce nejprve zeptá, zda jej nechcete uložit.

- Ü Stisknete-li **Ano**, program se uloží a Baltík se ukončí,
- Ü stisknete-li **Ne**, Baltík se ukončí aniž by se program uložil,
- Ü stisknete-li **Storno**, dialogové okno se zavře a vrátíte se zpět k původní práci, aniž by se cokoliv změnilo.

Programovací možnosti úrovně Začátečník

Úroveň **Začátečník** zahrnuje základní sadu příkazů: příkazy pro Baltíkův pohyb, načtení scény nebo banky obrázků, rychlost, viditelnost čarování, pípnutí. Navíc obsahuje prvky **Mezera** a **Konec řádku.** Z programových konstrukcí tato úroveň umožňuje **Opakování bloku příkazů.**

Zajímá-li vás pouze význam a použití některého prvku, klepněte na příslušný prvek na panelu dole.

Popojdi
Není-li Baltík před okrajem scény, popojde o jedno políčko tím směrem, kterým se dívá.

Vlevo vbok
Baltík provede obrat vlevo vbok. **Pozor!** Baltík dělá obraty vždy ze svého pohledu!

Vpravo vbok
Baltík provede obrat vpravo vbok. O směru platí totéž, co u obratu vlevo.

Neviditelný
Baltík bude od této chvíle neviditelný.

Viditelný
Baltík bude od této chvíle viditelný.

Čekej
Tento příkaz slouží k zadání čekání na stisk klávesy nebo tlačítko myši, případně zadanou dobu. Pokud je tento prvek použit samotný, Baltík čeká (stojí a klepe nožkou), dokud nestisknete klávesu nebo tlačítko myši. Zadáte-li za tímto prvkem **číslo,** bude toto číslo znamenat dobu v milisekundách, po kterou se bude čekat na stisk klávesy.

Čekej na stisk klávesy, nejdéle však **2 s.**

Tento příkaz čeká dvě sekundy. Stisknete-li v této době klávesu nebo tlačítko myši, program pokračuje ihned dále. Není-li stisknuta klávesa ani tlačítko myši, program pokračuje po uplynutí zadaných dvou sekund.

Pozn.: Následující klávesy Baltík ignoruje,a proto čeká dál i po jejich stisku: ***Shift***, ***Ctrl***, ***Alt***, ***CapsLock***, ***ScrollLock***, ***NumLock***.

Další možnosti práce s klávesnicí se otevřou při přechodu do režimu **Pokročilý**

Čaruj s obláčkem
Baltík bude od této chvíle vždy čarovat předměty s doprovodným obláčkem.

Čaruj bez obláčku
Baltík bude od této chvíle čarovat předměty bez doprovodného obláčku.

Obrazovka
Tento prvek můžete použít pro smazání obrazovky nebo k načtení scény či banky.

Samotný prvek ***obrazovka*** slouží jako příkaz ke **smazání obrazovky**.

Prvek ***Obrazovka*** následovaný **číslem** od **0** do **99** načte scénu s příslušným číslem. Bude-li se tedy váš program uložen v souboru **Pokus.bpr**, načte příkaz na obrázku scénu uloženou v souboru **Pokus.s01**.

Prvek ***Obrazovka*** následovaný číslem od **1000** do **1199** načte do pracovní scény banku určenou posledními třemi číslicemi podle pravidel pro **tvorbu názvů bank.** Bude-li se tedy váš program uložen v souboru **Pokus.bpr**, načte příkaz na obrázku banku uloženou v souboru **Pokus.b01**.

Další možnosti práce s obrazovkou se otevřou při přechodu do režimu **Pokročilý**

Průhlednost
Tento příkaz slouží k přepínání průhlednosti zobrazování předmětů, scén a bank. Podrobnosti viz pasáž **Průhlednost - využití v programech**

Pípni

Použijete-li tento příkaz, Baltík pípne.

Otoč se na východ, jih, západ, sever
Baltík se otočí na danou stranu (doprava, dolů, doleva, nahoru) bez ohledu na to, na kterou stranu byl otočený předtím.

Mezera
Baltík neudělá nic. Tento prvek pouze ukončuje příkaz a umožňuje rozdělovat program do více řádků. Podrobnosti viz **vytváření a úpravy programů.**

Nastav rychlost
Za tímto příkazem může být **číslo**, které určuje, jakou rychlostí se bude provádět další část programu až do příští změny rychlosti.

Není-li číslo zadáno, rychlost se nastaví rychlost **0**.

Význam jednotlivých rychlostí je následující:

0 Rychlost **0** znamená, že Baltík bude **po** každém příkazu čekat na stisk klávesy nebo tlačítka myši.

1 Rychlost **1** je nejmenší plynulá rychlost, tj. rychlost, při níž již nemusíme Baltíka ke každé akci pobízet.

2- 6 Se zvyšujícím se číslem rychlosti se Baltík postupně zrychluje.

7 Rychlost **7** je největší "zdržovaná" rychlost, tj. rychlost, u níž je mezi jednotlivé animované fáze vkládána čekací smyčka. Je to největší rychlost, při níž Baltík vykonává své akce na všech počítačích zhruba stejně rychle.

8 Rychlost **8** je největší zobrazovaná rychlost, tj. rychlost, při níž můžeme Baltíka při práci vidět (alespoň na pomalejších počítačích). Rychlost provádění jeho akcí však již závisí na rychlosti daného počítače.

9 Při rychlosti **9** je Baltík tak rychlý, že není vidět ani na nejpomalejším počítači, protože se již nezdržuje vykreslováním sebe ani čarovacích obláčků a plně se věnuje svěřenému úkolu. Uvidíte jej až v okamžiku, kdy se zastaví a čeká na nějakou vaši reakci.

∞ Rychlost **nekonečno** je zkratka pro příkazy ***Rychlost 9, Neviditelný, Bez obláčku***. Neuvidíte jej proto ani v okamžiku, kdy se zastaví a čeká na vaši reakci. Po nastavení některé nižší rychlosti se vrátí opět původní nastavení obláčku a viditelnosti.

Číslice
Z prvků s číslicemi lze sestavovat čísla, pomocí nichž můžete zadávat Baltíkovu **rychlost**, počet **opakování** označené akce, číslo **nahrávané scény či banky** apod.

Nekonečno
Prvek ***Nekonečno*** se používá pro nekonečné **opakování** příkazu nebo bloku příkazů:

Opakuj donekonečna { popojdi, vlevo vbok }

Konec řádku (Enter)
Baltík neudělá nic. Tento prvek pouze ukončuje příkaz a umožňuje rozdělovat program do více řádků. Podrobnosti viz **vytváření a úpravy programů.**

Označení začátku a konce bloku příkazů
Tento prvek slouží k označení **bloku příkazů, k němuž se bude Baltík chovat jako k příkazu jedinému**. Často se proto blok příkazů nazývá **složený příkaz**.

Při označování bloku příkazů:

1. Klepneme v panelu příkazů na prvek ***Blok příkazů***.
2. Přesuneme vybraný prvek **před první** prvek budoucího bloku.
3. Klepnutím sem umístíme prvek s otevírací složenou závorkou.
4. Přesuneme držený prvek **za poslední** prvek budoucího bloku.
5. Klepnutím sem umístíme prvek s uzavírací složenou závorkou.

Podrobnosti o **používání** bloků příkazů viz **Počítané opakování a bloky příkazů.**

Řádkový komentář
Baltík při provádění programu přeskočí všechny další prvky od tohoto prvku až do konce řádku. Pro zdůraznění ignorovaných prvků jsou tyto v programu přešktnuty

Textový komentář
Textový komentář se používá k slovnímu popisu programu. Na provádění programu nemá žádný vliv. Baltík totiž **při provádění programu textové komentáře ignoruje**.

Jediné místo, kam textový komentář **vložit nesmíte**, je mezi jednotlivé číslice čísla sestaveného z prvků s číslicemi.

Používáním vhodně zvolených textových komentářů se program zpřehlední a jeho pozdější čtení je snadnější. Správné používání textového komentáře vám umožní si i po čase vzpomenout na fígle, které jste kdysi v programu použili a využít jich při dalších úpravách a vylepšeních programu.

Po **vložení** prvku ***Textový komentář*** do programu se v něm objeví textový kurzor a můžete vepsat libovolný text. Zadávání textu ukončíte stiskem klávesy ***Enter*** nebo klepnutím myší mimo text komentáře. Velikost prvku ***Textový komentář*** se průběžně automaticky upravuje podle délky zadávaného textu.

Upravovaný znakový komentář s (blikajícím) kurzorem na konci vkládaného textu

Chcete-li **změnit hodnotu** prvku ***Textový komentář***, ukažte na něj myší a stiskněte klávesu **F2** nebo na

něj klepněte **pravým** tlačítkem myši a v místní nabídce zadejte povel ***Upravit***.

Text v komentáři se po této akci zpřístupní pro úpravy. Na počátku je všechen text vybrán do bloku a reaguje následovně:

- Ü stisknete-li některou z kurzorových kláves, vybrání se zruší a kurzor se přesune na místo určené právě stisknutou klávesou,
- Ü stisknete-li některou ze znakových kláves, vybraný text se smaže a je nahrazen vkládaným textem.

Rozmyslíte-li si svůj záměr s vkládáním či úpravou komentáře, stiskněte klávesu ***Esc*** a vše se vrátí do původního stavu.

Při vkládání i úpravách textu můžete využít všechny nabízené možnosti editace textu

Při programování v úrovni **Pokročilý** budete mít k dispozici **další komentářové prvky.**

Počítané opakování a bloky příkazů

Práce s celými bloky prvků

Opakování příkazu

Chcete-li, aby Baltík provedl nějaký příkaz několikrát za sebou, vložte do programu **před** tento příkaz **číslo** udávající požadovaný počet opakování.

Baltík 5 krát popojde o 1 políčko

Opakování skupiny příkazů

Chcete-li, aby Baltík provedl několikrát celou skupinu příkazů, musíte tuto skupinu **označit jako blok příkazů.** K bloku příkazů se pak Baltík chová stejně jako k příkazu jedinému.

Baltík čtyřikrát vyčaruje stromek, popojde a otočí se vlevo

Vnořování bloků

Protože blok příkazů vystupuje v programu jako jediný příkaz. Může se proto stát součástí jiného bloku příkazů - bloky můžeme do sebe **vnořovat**.

Vnořený blok pak označujeme jako **vnitřní** a blok, do kterého jsme tento blok vnořili, jako **vnější**.

Baltík ohradí stromečky prázdný čtverec velký 4×4 pole (pátý stromek přijde do rohu).
Udělá to tak, že vždy 5 krát za sebou vyčaruje stromek a popojde. Po těchto pěti čarováních s popojitím udělá vlevo vbok a to vše čtyřikrát zopakuje.

Úprava vzhledu bloků příkazů

Abychom se v programech dobře vyznali, odsazujeme všechny příkazy uvnitř bloku od levého kraje. S každou úrovní vnoření odsadíme příkazy vnořeného bloku o další prvek.

Pozn.: Všimněte si, že každá úroveň vnoření je označena závorkami jiné barvy. Tento detail výrazně zvyšuje přehlednost programu a snadnost odhalování chyb způsobených špatným uzavřením bloků do závorek.

Baltík vysadí obrazec ze stromků a postaví se do jeho středu.
V programu můžete kromě odsazování vnořených bloků používat i komentáře, které naznačují, co bude dělat následující část programu.

Další použití bloků příkazů

V úrovni **Pokročilý** se bloky příkazů používají zejména v **cyklu s řídící proměnnou** (cyklus **for**), v **cyklu s podmínkou** (cyklus **(do -) while**), ve **větvení** (příkaz **if**) a v **příkazech pro animaci**.

Průhlednost - využití v programech

Průhlednost pro pokročilé

Prvek ***Průhlednost*** slouží k přepínání průhlednosti zobrazování předmětů, scén, bank a další grafiky.

Pozn.: **Průhlednou barvu** pro daný předmět vidíte v programu v levém horním rohu předmětu, máte-li zapnutu volbu ***Možnosti → Zobrazit*** průhlednost nebo ji zjistíte v grafickém editoru **Paint**.

Pozn.: **Na začátku** programu je čarování nastaveno na **neprůhledné.**

1) Přepínání průhledného a neprůhledného zobrazování

Samostatně

Stojí-li prvek ***Průhlednost*** v programu samotně, přepne zobrazování pro další běh programu na průhledné. Od této chvíle budou **předměty**, **scény**, **banky** i **obrázky** zobrazovány s uplatněním průhledné barvy, tzn. že tam, kde tyto obrázky mají průhlednou barvu, bude na obrazovce vidět podklad.

S číslem

Použijete-li prvek ***Průhlednost*** s **číslem**, nastaví se průhlednost podle tohoto čísla:

- Ü **0** přepíná čarování a načítání na **neprůhledné**,
- Ü **ostatní čísla** je přepínají na **průhledné**.

Nastaví neprůhledné zobrazování

2) Nastavení průhlednosti zobrazení pro jeden příkaz

Použijete-li prvek ***Průhlednost***, resp. prvek následovaný číslem, **za příkazem**, bude se dané nastavení vztahovat **pouze na příslušný příkaz**. Nastavené zobrazování průhlednosti se však pro zbytek programu tímto příkazem nezmění.

Začátečníci mohou takto změnit nastavení průhlednosti **čarovaného předmětu,** nebo **načítané scény či banky,**.

Průhledně čaruj omráčenou opici

Pokročilí mohou tuto možnost využít u všech příkazů načítajících či kreslících něco na obrazovku včetně příkazů pro načítání obrázků a příkazů tisku.

Jednoduché animace

Animace pro pokročilé

Baltík umožňuje **animovat** předměty, tj. vyvolávat dojem jejich pohybu po obrazovce. Tím nám umožňuje naprogramovat takové efekty, jakými jsou např. například chůze panáčka, let ptáka, růst houby nebo otevírání dveří.

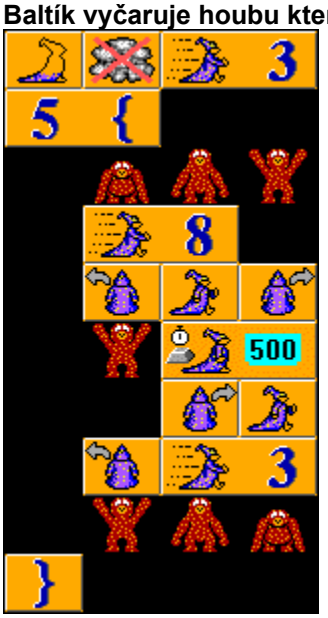
Vlastní animaci sestává z postupného zobrazování jednotlivých **fází**, tj. jednotlivých podob animovaného předmětu.

Zobrazení jedné fáze animace je zobrazením definovaného předmětu v definované pozici. Při animaci:

- Ü **změnou pozice** zobrazovaného předmětu mezi jednotlivými fázemi vyvoláme **dojem plynulého pohybu**,
- Ü **změnou** samotného **zobrazovaného předmětu** vyvoláme **dojem "života" animovaného předmětu** (růst houby, otevírání dveří apod.).

Pozn.: Při animaci pohybujících se předmětů, které vypadají pořád stejně, vystačíme s jedinou fází.

Baltík vyčaruje houbu která vyroste



Opice 5 krát vyskočí

Příkazy základní nabídky

Místní nabídky

Většina funkcí, které Baltík nabízí, je (mimo jiné) vyvolatelná prostřednictvím nabídek. Základní nabídka (to je ta, která je neustále vidět pod titulkovou lištou) je tvořena podnabídkami:

Program

Upravit

Zobrazit

Možnosti

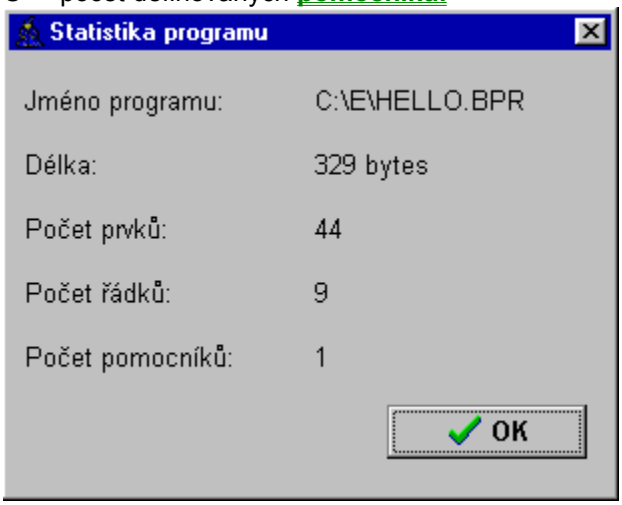
V následujících odstavcích probereme pouze ta témata, která nejsou podrobněji diskutována v jiných sekcích nápovědy.

Program

Statistika

Zadáním tohoto povelu otevřete dialogové okno ***Statistika programu***, které zobrazuje základní parametry programu:

- Ü jméno programu včetně kompletní cesty (nevejde-li se celá cesta na obrazovku, objeví se na spodním okraji okna posuvník),
- Ü jeho délku v bajtech (počítejte s tím, že místo, které příslušný soubor zabírá na disku, je díky některým vlastnostem operačního systému větší)
- Ü počet prvků (ikon), které jej tvoří, včetně komentářů, konců řádků a mezer,
- Ü počet řádků programu včetně řádků ukončených prvkem **rozdělení řádku,**
- Ü počet definovaných **pomocníků.**



Konec

Ukončí veškerou práci s Baltíkem. Pokud jste do svého programu zanesli od posledního uložení nějaké změny, zeptá se vás nejprve, zda jej chcete uložit. Pak vše ukončí a vrátí řízení zpět operačnímu

systému.

Naposledy uložené soubory
Na konci nabídky najdete seznam několika naposledy otevřených souborů. Jejich počet si můžete nastavit v dialogovém okně ***Možnosti prostředí*** v poli ***Počet programů zapamatovaných v historii.***

Upravit

Zpět
Uděláte-li nějakou chybnou akci (např. upustíte-li přenášený prvek na špatném místě, nějaký prvek omylem smažete apod.), můžete tuto akci odvolat.

Baltík umožňuje odvolat **nejvýše jednu akci** (některé akce jsou neodvolatelné). Abyste proto mohli nepovedenou akci úspěšně odvolat, musíte o její odvolání požádat **ihned** poté, co ji provedete. V opačném případě vám bude Baltík nabízet k odvolání akci, kterou jste provedli jako poslední.

Označit blok
Zadáním povelu ***Program → Označit blok*** přepnete Baltíka do režimu označování bloku. V tomto režimu můžete **označit blok prvků** bez použití klávesy ***Shift***.

Označit řádky
Tento povel umožňuje vybírat prvky do bloku po celých řádcích. Při výběru bloků tvořených celými řádky má tento výběr několik výhod:

- Ü výběr je rychlejší,
- Ü při výběru bloků tvořených celými řádky máte jistotu, že se do výběru dostane i prvek ***Konec řádku*** na konci posledního řádku bloku (to oceníte zejména tehdy, je-li poslední řádek delší než pracovní plocha).

Označit vše
Tímto povelem vyberete do bloku celý program. Toho můžete využít např. tehdy, když chcete celý program rychle smazat nebo když chcete jeho kód přenést prostřednictvím ***schránky*** do jiného programu.

Vyjmout
Označený blok prvků odstraní z programu a přesune jej do ***schránky.***

Kopírovat
Označený blok prvků zkopíruje do ***schránky.*** V programu blok nadále zůstane v podobě, v jaké byl před kopírováním.

Vložit
"Předá myšímu kurzoru" obsah ***schránky,*** který pak můžete klepnutím umístit na požadované místo programu. Obsah ***schránky.*** se touto operací nezmění (jinými slovy: můžete jej dále používat).

Odstranit

Odstraní **označený blok prvků** z programu. Obsah ***schránky.*** se touto operací nijak nezmění.

Seřadit konstanty a proměnné

Baltík umožňuje zadat jeden ze dvou způsobů uspořádání konstant a proměnných v jejich bankách:

- Ü při řazení **podle Názvu** řadí Baltík objekty v jednotlivých bankách podle jejich názvů a zcela přitom ignoruje značky, které jsou napsány na listech, šuplících a košících,
- Ü při řazení **podle značek** naopak Baltík při řazení objektů ignoruje jejich názvy a řadí pouze podle značek na listech, šuplících a košících.

Nápověda

Schránka ve Windows umožňuje předávat objekty jakéhokoliv druhu v rámci aplikace nebo mezi jednotlivými aplikacemi. Cokoliv uložíte do ***Schránky***, zůstane tam uloženo tak dlouho, dokud do ní neuložíte něco jiného.

Části programů zůstávají ve schránce i poté, co je program ukončen otevřen program jiný. Toho lze s výhodou použít pro přenášení částí kódu mezi jednotlivými programy.

- **Program**
  - **Nový**
  - **Otevřít**
  - **Uložit**
  - **Uložit jako**
  - **Začátečník**
  - **Pokročilý**
  - **Start**
  - **Spusť označený blok**
  - **Statistika**
  - **Konec**
  - **Naposledy uložené soubory**

- Upravit
  - Zpět
  - Označit blok
  - Označit řádky
  - Označit vše
  - Vyjmout
  - Kopírovat
  - Vložit
  - Odstranit
  - Hledat
  - Hledat další
  - Hledat předchozí
  - Kreslení
  - Tabulka souborů

Zobrazit

- **Skládat scénu**
- **Čarovat scénu**
- **Programovat**

Možnosti

- Tabulka kódování znaků
- Seřadit konstanty a proměnné
- Prostředí

Nastavení parametrů prostředí

Baltík umožňuje uživateli, aby si jeho prostředí přizpůsobil svým zvyklostem a dovednostem. Nastavitelné parametry jsou rozděleny do šesti bloků:

1) Možnosti zobrazení prvků

Zobrazovat průhlednou barvu předmětů

Při nastavení této volby budou v programu u všech průhledně čarovaných předmětů v jejich levém horním rohu zobrazena přeškrtnutá kolečka s průhlednou barvou daného předmětu.

Nebude-li tato volba nastavena, nebude průhledná barva u čarovaných předmětů nijak zdůrazněna.

Zobrazovat příkazy v celku

Nebude-li tato volba nastavena, bude každý prvek v programu zobrazen samostatně nezávisle na tom, vystupuje-li v programu opravdu samostatně nebo je-li naopak součástí nějakého většího celku.

Při nastavení této volby smaže Baltík hraniční čáry mezi prvky, které tvoří dohromady jeden příkaz.

Doporučujeme vám tuto volbu **nastavit**, protože pak je program výrazně čitelnější a přehlednější.

Zobrazovat chyby jako vykřičníky

Je-li tato volba nastavena, bude Baltík u všech prvků, jejichž významem si není jist, zobrazovat červené vykřičníky, kterými upozorňuje uživatele na spornost dané konstrukce.

Není-li volba nastavena, program na žádné možné nesrovnalosti předem neupozorňuje.

Doporučujeme tuto volbu **nastavit**, protože se tak okamžitě dozvídáte o možných potížích při následném běhu. Potlačení volby je sice výhodné při psaní programu, kdy programátora nerozptylují neustále se objevující a opět mizející vykřičníky, ale méně zkušeným programátorům bychom doporučovali překousnout tyto dočasné nepříjemnosti a volbu nastavit.

2) Možnosti editace

Automaticky odsazovat

Při nastavení této volby Baltík předřadí před každý vkládaný blok tolik odsazujících mezer, kolik jich má předchozí příkaz.

Není-li tato volba nastavena, vkládá se blok tak, jak je.

Výhodnost nastavení této volby závisí na zvyklostech a preferencích programátora. Doporučíme vám proto pouze to, abyste vnořené bloky důsledně odsazovali, ale rozhodnutí, zda dáte přednost "ručnímu" **vodorovnému posunu bloků** nebo využijete nabízené možnosti automatického odsazování, necháme na vašem uvážení.

Automaticky rozvinovat příkazy

Po nastavení této volby se po každém najetí myší na tlačítko ***Příkazy*** rozvine ***panel příkazů.***

Není-li tato volba nastavena, panel na nájezdy myši nereaguje.

Zkušenější programátoři většinou dávají přednost potlačení této volby, protože si pak nemusejí dávat pozor, kudy jedou myší, aby omylem nerozvinuli panel příkazů a nemuseli se zdržovat s jeho opětovným sbalováním.

Zobrazovat číslo řádku a prvku

Po nastavení této volby bude Baltík zobrazovat na pravém okraji stavového řádku řádek a sloupec, v němž se nachází prvek, na nějž právě ukazujete myší.

Doporučujeme vám tuto volbu **nastavit**, protože jejím nastavením nic neztratíte a jejím potlačením nic nezískáte.

3) Vztah ke spouštěným programům

Minimalizovat před spuštěním

Nastavením této volby požádáte Baltíka, aby před každým spuštěním programu minimalizoval aplikační okno, v němž jste program vyvíjeli. To je výhodné v okamžiku, kdy program někomu předvádíte a nestojíte o to, aby pod oknem s Baltíkovým prostorem vykukovalo okno s vaším programem.

Potlačením této volby naopak dosáhnete toho, že po spuštění programu zůstane okno s programem otevřeno, takže vhodným nastavením zobrazované části programu a posunováním okna s Baltíkovým prostorem můžete na pozadí děje číst program, který je právě prováděn a hledat v něm např. případné chyby.

Ukládat program před spuštěním

Nastavením této volby požádáte Baltíka, aby před každým spuštěním modifikovaného programu tento program nejprve uložil. Zabráníte tak tomu, abyste v případě zhroucení počítače (pamatujte: v každém programu je alespoň jedna chyba) přišli o svoji celodenní práci.

Potlačení této volby je výhodné v případech, kdy si jen něco zkoušíte a netrváte na uchování testovaného programu.

Kompromisem mezi oběma možnostmi je nastavení volby ***Ukládat program před spuštěním*** spolu s nastavením podvolby ***Před uložením se dotázat***. Pak se vás bude Baltík před každým uložením modifikovaného programu ptát, zda jej chcete opravdu uložit.

Tomu, koho věčné odpovídání na podobné dotazy obtěžuje, doporučujeme volbu raději nastavit.

Počet programů zapamatovaných v historii

V tomto poli můžete nastavit počet naposledy otevřených a uložených programů, které si má Baltík pamatovat. Seznam těchto programů seřazený podle času, kdy jste s nimi naposled pracovali (nejmladší jako první) naleznete na konci nabídky ***Program.***

Zvuky

Po nastavení této volby bude Baltík doprovázet své počínání doprovodnými **zvuky.**

4) Barvy komentářů

V bloku ***Barvy komentářů*** můžete zadat, jakou barvou a na jakém podkladě se budou zobrazovat vaše komentáře. Vhodným nastavením dosáhnete přiměřeného odlišení komentářů od textu, které program zpřehlednění.

5) Podbarvení kontrolních tisků proměnných

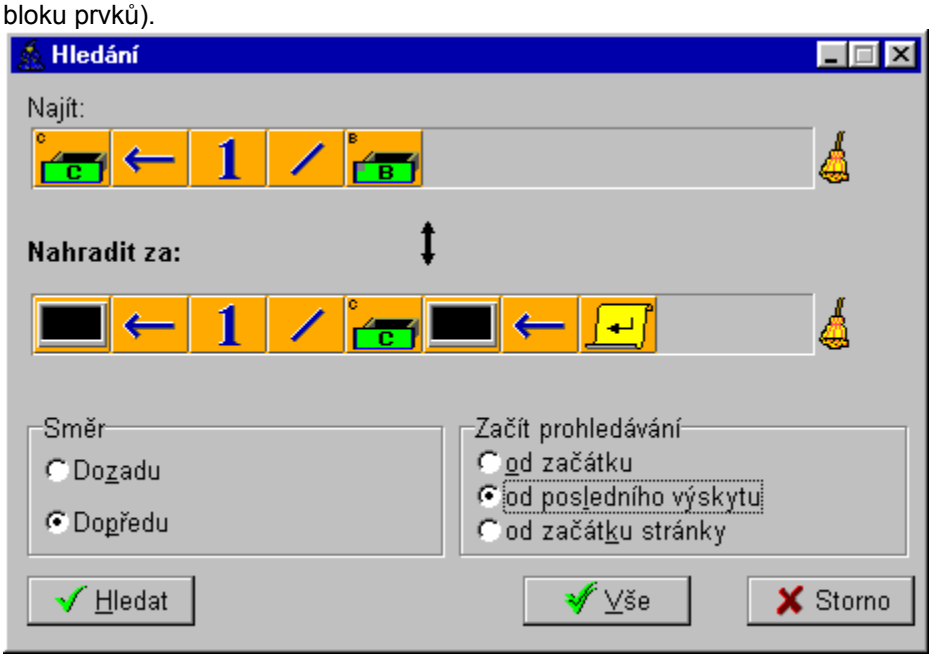
Na různých monitorech jsou různé kombinace barev různě zřetelné. V bloku ***Podbarvení kontrolních tisků proměnných*** si proto můžete nastavit barvu podkladu pod názvy jednotlivých proměnných **v kontrolním výpisu obsahu proměnných,** který můžete při pozastavení programu zobrazit pod Baltíkovým prostorem.

6) Konverze tabulačních znaků na výstupu

Po zaškrtnutí volby ***Zobrazovat tabulátory jako mezery*** bude při **výpisu souboru** na obrazovku každý znak tabulátoru nahrazen takovým počtem mezer, jaký nastavíte v příslušném editačním poli. Pokud není tato volba zaškrtnuta, tabulátory jsou zobrazeny jako ostatní nezobrazitelné znaky.

Hledání a nahrazování bloků prvků

Při úpravách rozsáhlejších programů často využijete možnost najít zadanou posloupnost prvků a případně ji nahradit posloupností jinou. K tomuto účelu slouží dialogové okno ***Hledání***, které vyvoláte povelem ***Hledat*** z nabídky ***Program*** (v úrovni ***Pokročilý*** je tento povel k dispozici i v místní nabídce bloku prvků).



Pro zadání hledané posloupnosti prvků máte několik možností:

- Ü Máte-li vybraný blok, tak se po otevření okna automaticky objeví v poli ***Najít***.
- Ü Nemáte-li vybraný blok, který chcete hledat, ale víte-li o nějakém, který je hledanému bloku podobný, můžete z okna odskočit, hledaný blok prvků vybrat v programu (okno ***Hledat*** nechte v průběhu vybírání bloku otevřené), uchopit jej a zkopírovat do pole ***Najít*** (obě tlačítka myši se v tomto případě chovají jako kopírovací).
- Ü Nevíte-li ani o podobném bloku, musíte do pole ***Hledat*** zadat postupně jednotlivé prvky hledané posloupnosti obdobně, jako je vkládáte do porgramu.

Chcete-li hledanou posloupnost prvků nahradit jinou posloupností, zadáte je obdobným způsobem do pole ***Nahradit za***.

Obsahy polí ***Najít*** a ***Nahradit za*** můžete editovat (upravovat). Prvky můžete v rámci polí i mezi poli přesouvat a kopírovat. Nemůžete však uvnitř pole vybrat blok, takže potřebujte-li provést nějakou akci s několika prvky najednou, musíte ji provést s každým prvkem zvlášť.

Po vložení nějakého prvku do pole ***Najít*** nebo ***Nahradit za*** se vpravo vedle pole objeví malé oranžové **koště**. Najedete-li na ně myší, zjistíte, že se jedná o ikonu na tlačítku. Stiskem tohoto tlačítka **smažete celý obsah** příslušného vstupního pole.

Způsoby hledání

Před započetím vlastního hledání zkontrolujte, zda jsou přepínače ***Směr*** a ***Začít prohledávání*** ve spodní části okna nastaveny podle vašich požadavků. Význam jejich nastavení je následující:

Směr

- Ü **Dozadu** - prohledává od pozice zadané v přepínači ***Začít prohledávání*** směrem k počátku programu,
- Ü **Dopředu** - prohledává od pozice zadané v přepínači ***Začít prohledávání*** směrem ke konci programu,

Začít prohledávání

- Ü **Od začátku** - začne prohledávat od počátku programu,
- Ü **Od konce** - začne prohledávat od konce programu,
- Ü **Od posledního výskytu** - začne prohledávat od místa, kde naposledy našel hledaný řetězec,
- Ü **Od začátku stránky** - nezávisle na směru prohledávání začne prohledávat od prvního zobrazeného prvku, tj. od prvku v levém horním rohu obrazovky.

Hledat další

Po tomto povelu Baltík hledá další výskyt zadané posloupnosti prvků.

Hledat předchozí

Po tomto povelu Baltík otočí směr hledání a hledá předchozí výskyt zadané posloupnosti prvků.

Úroveň Pokročilý

Do úrovně ***Pokročilý*** se přepnete zadáním příkazu ***Program → Pokročilý***. Po přechodu do této úrovně se před vámi otevře nový svět plný netušených možností, z nichž mnohé vám budou závidět i zkušení programátoři.

Stiskem tlačítka **>>** na panelu nástrojů nápovědy se přesunete do následujících sekcí, kde se vše postupně naučíte.

Pro ty, kteří nemohou číst dále, aniž by se před tím dozvěděli, co vše se v následujících sekcích naučí, jsme připravili **stručný přehled základních** tematických oblastí:

Práce se systémem:

- Ü Panel příkazů bude výrazně větší a bude mít nové, zajímavé vlastnosti
- Ü Budete moci používat **místní nabídky,**
- Ü **Komentářové závorky** vám umožní zamaskovat části programu, které zatím nechcete spouštět,
- Ü Při přesouvání a kopírování bloků příkazů budete moci využít **Schránku,**
- Ü Nad pracovní plochou se objeví **nová tlačítka,** která vám zpřístupní práci s **konstantními** i **proměnnými** daty, s multimedii a umožní vytvářet Baltíkovy **pomocníky.**

Obrazové a zvukové efekty

- Ü Vedle jednoduchých animací, při nichž se mohly animované předměty pohybovat pouze po celých políčcích, se před vámi otevře svět plnohodnotné **animace,** jež vám umožní dosahovat netušené efekty,
- Ü vaše programy budou umět pracovat s **multimediálními** soubory, pouštět zvuky či přehrávat videoukázky,
- Ü budete schopni naučit své programy **přehrávat zvuková CD** podle vámi zadaného schématu,
- Ü vaše programy budou schopny samostatně **kreslit** zajímavé obrázky a **pracovat s obrázky** uloženými v souborech.

Zpracování dat

- Ü Ve svých programech budete umět pracovat s **čísly,** s **texty,** se **souřadnicemi** i s **časem.**
- Ü budete umět ukládat hodnoty nejen do proměnných, ale budete mít k dispozici i složitější "schránky", jakými jsou **pole** a **hromady,**
- Ü dokážete z programu **spouštět cizí aplikace,**
- Ü budete umět ukládat data do **souborů** a příště je z nich opět číst.

Vstup a výstup údajů

- Ü Budete schopni číst v programu data **zadávaná z klávesnice,**
- Ü budete umět naučit své programy reagovat na **myš,**
- Ü vámi naprogramované hry budou schopny reagovat na nejrůznější **události,** jakými jsou stisk, držení či puštění zadané klávesy, pohyb myší nebo vypršení zadaného času,
- Ü údaje, které budete tisknout na obrazovku budete moci velmi efektně **formátovat** včetně možnosti definovat **písmo** nebo **barvu,** kterou se budou vypisovat.

Programové konstrukce

- Vaše programy nebudou pouze plnit slepě příkazy jeden za druhým, ale budou se moci na základě otestovaných **podmínek** rozhodnout o dalším postupu,
- budou si umět **vybrat z několika možných pokračování,**
- budou schopny **opakovat** zadanou akci tak dlouho, dokud bude platit předem zadaná **podmínka.**

Novinky editace a místní nabídky

V této sekci si povíme o dvou novinkách úrovně ***Pokročilý***: o nových vlastnostech ***Panelu příkazů*** a o **místních nabídkách.**

Panel příkazů

V režimu ***Pokročilý*** je příkazový panel mnohem větší. Abyste viděli vše, co potřebujete, a aby přitom panel nezabíral zbytečně místo na pracovní ploše, máte možnost jej pomocí myši zvětšit nebo zmenšit. Dosáhnete toho následovně:

1. Najedete myší na horní hranu pracovní plochy tak, aby se kurzor myši změnil na dvě vodorovné čárky se svislými šipkami.

2. Uchopíte hranu a stáhnete ji tak, abyste viděli požadovaný počet řádků panelu.

Počet průběžně viditelných řádků příkazového panelu můžete nastavit od nuly (celý panel je skrytý) do devíti (všechny řádky jsou vidět).

Operativní zobrazování příkazového panelu

V režimu ***Pokročilý*** můžete nechat panel skrytý. Potřebujete-li vybrat nějaký prvek, stačí najet myší na prázdnou černou plochu vpravo vedle programu tak, aby se kurzor myši změnil na malou bílou tabulku (viz obrázek). Následným klepnutím levým tlačítkem pak panel rozvinete.

Po vybrání požadovaného prvku se panel opět schová. Pokud si svůj původní úmysl rozmyslíte a nebudete chtít vybrat žádný prvek, sbalíte panel tím, že klepnete kamkoliv mimo něj nebo stisknete klávesu Esc.

Místní nabídky

Jednou z vymožeností, kterou práce v režimu ***Pokročilý*** přináší, je možnost využití místních nabídek, které vyvoláte klepnutím pravým tlačítkem myši na objekt vybavený místní nabídkou.

Většinu povelů z místních nabídek naleznete také v **základní nabídce.** V této sekci proto budeme hovořit o těch několika povelech, které v základní nabídce nejsou.

Zkopírovat sem označený blok

Tento povel se objeví v místní nabídce prvku v programu tehdy, když je současně vybrán nějaký blok prvků.

Příkaz zkopíruje onen vybraný blok do místa, kam při vyvolání místní nabídky ukazoval myší kurzor. Zkopírovaný blok (tj. kopie) bude po provedení příkazu vybrán (tj. zašedivěn).

Tento příkaz nemá žádný vliv na obsah ***Schránky,*** ani není jejím obsahem ovlivněn. Po provedení příkazu proto můžete dále vkládat do programu původní obsah schránky.

Přesunout sem označený blok

Tento povel se objeví v místní nabídce prvku v programu tehdy, když je současně vybrán nějaký blok prvků.

Příkaz přesune onen vybraný blok do místa, kam při vyvolání místní nabídky ukazoval myší kurzor. Z původního místa bude přesouvaný blok odstraněn. Přesunutý blok bude po provedení příkazu vybrán (tj. zašedivěn).

Ani tento příkaz nemá žádný vliv na obsah ***Schránky,*** ani není jejím obsahem ovlivněn. Po provedení příkazu proto můžete dále vkládat do programu původní obsah schránky.

Kreslení

Tento příkaz se objevuje v místní nabídce obrázků (přesněji řečeno prvků zastupujících obrázky). Po jeho zadání Baltík zobrazí obrázek zastupovaný v programu prvkem, na nějž při vyvolání místní nabídky ukazoval myší kurzor.

Komentáře pro pokročilé

Komentáře jsou při provádění programu ignorovány a nemají na něj proto žádný vliv. Slouží pouze k zpřehlednění programu (**dokumentační** komentáře) a/nebo k vymaskování jeho dočasně nepoužívaných částí (**maskovací** komentáře).

Dokumentační komentáře

Textový komentář

Textový komentář se používá k slovnímu popisu programu. **Popis jeho použití** najdete v sekci **Programovat - začátečník**.

Rozdělení řádku

Tento prvek **opticky** rozděluje řádek. Rozdělení řádku můžete s výhodou použít například tehdy, když je příkaz, který musí být zapsán vcelku, tak dlouhý, že se nevejde na šířku obrazovky (obsahuje např. hodně parametrů se souřadnicemi a dlouhými čísly), takže by jeho zápis byl nepřehledný.

Rozdělení řádku nesmíte vložit mezi jednotlivé číslice dlouhého čísla.

V Baltíkových programech rozlišujeme fyzické a logické řádky:

- Ü **Fyzický řádek** je řádek, který vidíme na obrazovce. Může být ukončen jak prvkem ***Konec řádku***, tak prvkem ***Rozdělení řádku***. Je to tedy řádek, který vidíme my.
- Ü Naproti tomu **Logický řádek** je řádek tak, jak jej vidí Baltík - tj. řádek pro vypuštění prvků ***Rozdělení řádku***, protože tyto prvky Baltík ignoruje. Logický řádek je tedy **posloupnost prvků mezi dvěma znaky *Konec řádku***.

Pozn. Je-li řádek ukončen prvkem ***Rozdělení řádku***, jsou **mezery** na začátku následujícího řádku ignorovány. Tyto komentářové mezery mají uprostřed výraznější tečku komentářovou barvou (šedou).

Maskovací komentáře

Řádkový komentář

Baltík při provádění programu přeskočí všechny další prvky od tohoto prvku až do konce logického řádku.

Pro zdůraznění ignorovaných prvků jsou tyto v programu přešktnuty

Tip: Řádkový komentář můžete s úspěchem použít při laděník maskování zarážek v programu a kontrolních tisků.

Po otočení vlevo a vyčarování stromku se již Baltík nepřepne na krokovací rychlost 0.

Otevírací a zavírací komentářová závorka

Baltík při provádění programu přeskočí všechny příkazy od otevírací komentářové závorky po zavírací komentářovou závorku včetně.

Chcete-li přeskočit provádění určité části programu, uzavřete ji do komentářových závorek. Prvky mezi začátkem a koncem komentáře nebudou brány v úvahu.

Pro zdůraznění ignorovaných prvků uvnitř komentáře jsou prvky v komentáři **přešktnuty**

Komentáře mohou být vnořené. Otevřete-li uvnitř komentáře další komentářovou závorku, potřebujete k zavření celého komentáře dvě zavírací komentářové závorky - viz obrázek

Další tlačítka nad pracovní plochou

Po přepnutí do úrovně **Pokročilý** ožijí nad pracovní plochou do té doby potlačená tlačítka. (**Klepnutím na obrázek tlačítka se přesunete na příslušné téma nápovědy.**)


Panel příkazů


Tabulka pomocníků


Banky dat


Tabulka souborů a oblastí

Příkazy

Klepnutím na toto tlačítko rozvinete skryté řádky ***Příkazového panelu***. Po výběru příkazu se řádky zas skryjí.

Tip: Obtěžuje-li vás neustálé zmenšování a zvětšování pracovní plochy způsobované odkrýváním a opětným skrýváním řádků ***Příkazového panelu***, stačí **klepnout** levým tlačítkem v pracovní ploše kamkoliv **na černou plochu** vpravo od programu. Objeví se okno s duplikátem ***Příkazového panelu***, které po vybrání příkazu opět zmizí.

Pomocníci

Tímto tlačítkem otevřete tabulku **pomocníků**, což jsou Baltíkovy ekvivalenty **procedur** z klasických programovacích jazyků. **Tabulka pomocníků** umožňuje velmi snadno:

- Ü **vložit** na konec programu **definici** nového pomocníka,
- Ü **vložit** do programu **volání** určeného pomocníka,
- Ü **zobrazit kód** určeného pomocníka,
- Ü **vrátit se** do definice pomocníka na místo, kde jsme v ní byli naposledy.

Zvláštní banky

Konstanty, **proměnné**

Po stisku tohoto tlačítka se zobrazí **zvláštní banky**, které obsahují **klávesy**, konstanty a proměnné.

Tabulka souborů a oblastí

Stiskem tlačítka se otevře **tabulka oblastí a souborů,** do které můžete vložit **odkazy na oblasti** či soubory, a to jak **datové** tak **multimediální.**

Literál a typy dat

Termín **literál** se v programování používá k označení přímého zadání hodnoty do programu. Číselné hodnoty můžeme zadávat pomocí **prvků s číslicemi** obdobně, jako jsme to dělali v úrovni **Začátečník.** V úrovni pokročilý navíc přistupuje možnost zadávat hodnoty do programu prostřednictvím prvku ***Literál***

Baltík rozlišuje **tři typy hodnot**:

- Ü **celá čísla,** (označuje je azurovou barvou)
- Ü **reálná čísla** (označuje je zelenou barvou) a
- Ü **řetězce** (lidově bychom řekli texty - označuje je žlutou barvou).

Pozn.: Vedle výše uvedených tří typů hodnot se v programu občas vyskytují také **logické hodnoty** (např. jako odpověď na otázku, zda je 3<5). Jak si však za chvíli vysvětlíme, Baltík je interpretuje jako čísla.

Typ hodnoty ovlivňuje to,

kde se smí daná hodnota použít a

jak se s ní bude nakládat.

Čísla

Tip: Přestože číslo můžete v Baltíkovi zapsat pomocí jednotlivých prvků s číslicemi přímo, **doporučujeme používat** i pro čísla prvek ***Literál***. Váš program bude přehlednější.

1. Prvek *Literál*

Prvek ***Literál*** se používá pro **přímé zadání** čísla nebo řetězce.

Vložení prvku do programu

Po vložení prvku ***Literál*** do programu se ve vloženém prvku objeví textový kurzor a můžete do prvku vepsat libovolný text. Zadávání textu ukončíte klávesou ***Enter*** nebo klepnutím myší mimo text literálu. Velikost prvku se průběžně automaticky upravuje podle délky zadaného textu.

V průběhu zadávání se může měnit **podkladová barva** pole, do nějž vepisujete text. Tato barva **určuje typ zadané hodnoty** a tím i to, kde se smí daný literál použít a jak se bude s jeho hodnotou nakládat.

- Ü dokud je Baltík schopen převést zadávaný text na **celé číslo**, bude pokladová barva **azurová**,
- Ü dokud je Baltík schopen převést zadávaný text na **reálné číslo** (ale ne na celé), bude podkladová barva **zelená**,
- Ü není-li Baltík schopen převést text na číslo, bude zadávaný text považovat za **řetězec** a pokladová barva bude proto **žlutá**.

Chcete-li **změnit typ** zadávaného literálu z celého čísla na reálné nebo z čísla na řetězec, **klepněte** levým tlačítkem myši **na jednu z barevných plošek** v pravé části prvku nebo použijte šipky nahoru a dolů.

Zadanou hodnotu potvrdíte klávesou ***Enter*** nebo stiskem tlačítka myši kdekoliv mimo prvek.

Chcete-li přerušit zadávání hodnoty a odstranit prvek z programu, stiskněte klávesu ***Esc***.

Změna hodnoty prvku

Chcete-li změnit hodnotu prvku ***Literál***, klepněte na něj **pravým** tlačítkem myši a v místní nabídce zadejte povel ***Upravit*** nebo nastavte myší kurzor nad ***Literál*** a pak stiskněte klávesu ***F2***.

Při úpravách řetězce platí totéž, co platí pro zadávání a úpravy textu **komentářů.**

Chcete-li přerušit zadávání hodnoty a zachovat původní hodnotu, stiskněte klávesu ***Esc***.

Číselný literál můžete použít všude tam, kde se používá **číslo:**

Vyčaruj předmět č. 1025

Řetězcový literál můžete použít všude tam, kde se používá **řetězec:**

Zobraz na obrazovce text "Ahoj lidi"

2. Celá čísla

Rozsah

Od **-2 147 483 648** do **2 147 483 647**.

Zápis

Aby bylo celé číslo chápáno jako celé, musí být zapsáno pouze jako posloupnost číslic od **0** do **9** a musí být v rozpětí
od **-2 147 483 648** do **2 147 483 647** (zhruba od mínus dvou miliard do plus dvou miliard).

Čísla, jejichž absolutní hodnota je větší, považuje Baltík za reálná stejně jako čísla, při jejichž zápisu bylo použito některé z možností platných jen pro **reálná čísla.** (Vyzkoušejte si to zadáním hodnoty čísla do literálu.)

Stejně jako v matematice se před zápornými čísly uvádí znaménko mínus (-), kdežto znaménko plus (+) se před kladnými čísly uvádět nemusí (ale může).

6340 (šest tisíc tři sta čtyřicet)

-32 (mínus třicet dva)

Tip: Přestože číslo můžete v Baltíkovi zapsat i pomocí jednotlivých prvků s číslicemi přímo, **doporučujeme používat** i pro čísla prvek ***Literál***. Váš program bude přehlednější.

Zobraz 6340 (šest tisíc tři sta čtyřicet)

Zobraz -32 (mínus třicet dva)

Pozn.: Celé číslo můžete v literálu **zadat** také **v šestnáctkové soustavě**. V tom případě musí číslo začínat znakem **$** nebo znaky **0x** (nula-x), případně **0X**. Za těmito znaky následuje posloupnost složená z číslic **0** až **9** a z písmen **a** až **f** a **A** až **F**, např. číslo **0xFF** má v desítkové soustavě hodnotu **255**.

3. Reálná čísla

Reálné číslo je číslo, které

- Ü má kromě celé části (viz celé číslo) také desetinnou část nebo
- Ü nevejde se do rozsahu platného pro celá čísla nebo
- Ü v jeho zápisu je použita exponenciální část.

Rozsah

Absolutní hodnota reálného čísla se musí pohybovat v rozsahu
od **2.9×10^-39** do **1.7 * 10^+38**.

Čísla s **menší** absolutní hodnotou jsou považována za **nulová**,
čísla s **větší** absolutní hodnotou **způsobí chybu**.

Baltík si reálná čísla pamatuje s přesností na přibližně **10 platných cifer**. Zadáte-li mu tedy např. číslo **1.234567890123456789**, bude si pamatovat, že jste zadali přibližně **1.234578901**.

Zápis

Na rozdíl od zvyklostí běžných v matematice se v programování odděluje desetinná část v zápisu čísla **desetinnou tečkou**:

Prvek ***Desetinná tečka***

123.45 (sto dvacet tři celých, čtyřicet pět setin)

Zápis reálného čísla může obsahovat i **exponenciální část**. Ta je uvozena prvkem ***E*** za nímž následuje (kladné nebo záporné) celé číslo, které udává mocninu deseti, jíž je třeba předchozí číslo vynásobit.

Prvek *Exponent*

12.34*10^5 (dvanáct celých, třicet čtyři setin krát deset na pátou)

Toto číslo má hodnotu **12.34×10^5**. Deset na pátou je **10×10×10×10×10**,
tj. **100 000**. Výše uvedené číslo je tedy rovno **12.34*100 000**,
tj. **1 234 000**.

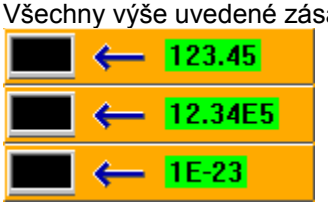
Přestože je toto číslo celé a vejde se do potřebného rozsahu, považuje je Baltík za **reálné**, protože při jeho zápisu byl použit exponent.

Exponent může být i záporný.

1E-23 (jedna krát deset na mínus dvacátou třetí)

10^-23 (deset na mínus dvacátou třetí) je číslo, které má za desetinnou tečkou **22 nul**, tedy číslo **0.000 000 000 000 000 000 000 01**.

Všechny výše uvedené zásady platí i pro zápis čísel v prvku ***Literál***:

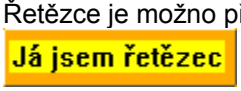


4. Řetězce

Vedle čísel umí Baltík pracovat i s řetězci, což jsou posloupnosti znaků (lidově bychom řekli texty). Délka řetězců, tj. počet jejich znaků, je omezena pouze velikostí paměti počítače.

Pozn.: V nápovědě jsou řetězce odlišeny od ostatního textu tím, že jsou uzavřeny do uvozovek, např.: "Stiskni klávesu", "Petr", "123456" apod.

Řetězce je možno přímo zadat pouze prostřednictvím prvku ***Literál***.



Řetězcový literál s textem "Já jsem řetězec"

Mezi řetězci má zvláštní postavení tzv. **prázdný řetězec**, který neobsahuje ani jeden znak.

Literál s prázdným řetězcem

Součástí řetězce mohou být i **řídící znaky** (např. přechod na další řádek), avšak tyto znaky není možno zadat prostřednictvím prvku ***Literál*** a musí se zadat jiným způsobem.

Další podrobnosti o řetězcích najdete v pasáži **Řetězce.**

5. Logické hodnoty

Vedle výše uvedených tří typů hodnot se v programu občas vyskytují také logické hodnoty, tj. hodnoty se dvěma možnými stavy: **ANO** a **NE**. (Takové hodnoty získáme např. jako odpověď na otázku, zda je **3<5**).

V místech, kde je očekávána logická hodnota, můžeme použít hodnotu jakéhokoliv typu. Baltík ji zpracuje tak, že

Ü nulové hodnoty a prázdné řetězce interpretuje jako **NE** a

Ü nenulové hodnoty a neprázdné řetězce interpretuje jako **ANO**.

Má-li Baltík sám spočítat nějakou logickou hodnotu (např. odpovědět na otázku, zda je **3<5**), pak místo **NE** vrátí celočíselnou hodnotu **0** a místo **ANO** vrátí celočíselnou hodnotu **1**.

Souřadnice

Pokročilá práce se souřadnicemi

Souřadnice jsou čísla, která udávají **polohu objektu** na pracovní scéně. Tímto objektem přitom může být předmět, Baltík, **myš**, **oblast** nebo **obrázek**.

U větších objektů je **referenčním bodem**, vůči němuž se všechny souřadnice počítají, **levý horní roh objektu**.

Souřadnice jsou dvě:

- Ü **vodorovná souřadnice**, kterou značíme **X**, udává **vzdálenost od levého** okraje scény a
- Ü **svislá souřadnice**, kterou značíme **Y**, udává **vzdálenost od horního** okraje scény.

Objekt **přilepený k levému hornímu rohu** scény má od obou hran nulovou vzdálenost a má proto **obě souřadnice nulové**.

Baltík rozeznává dva druhy souřadnic:

- Ü souřadnice **políčka** a
- Ü souřadnice **bodu**.

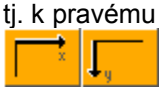
Souřadnice políčka

Políčka jsou malé oblasti, do nichž se v režimu **Skládat scénu** umísťují předměty. Políčko je stejně veliké jako **předmět.** a scéna má proto rozměry **15×10** políček.

Souřadnice políček značíme velkými písmeny **X** a **Y**.

Políčko v levém horním rohu scény má souřadnice **X0 Y0**,
políčko v pravém dolním rohu scény má souřadnice **X14 Y9**.

Vyčaruj strom na pozici X14 Y5,
tj. k pravému okraji scény, těsně pod jeho střed



Souřadnice bodu

Baltík umožňuje zadávat souřadnice s přesností na jeden obrazovkový bod.

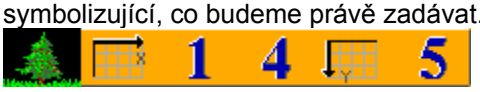
Souřadnice obrazovkových bodů zadáváme malými písmeny **x** a **y**.

Z toho, že scéna má rozměr **15×10** políček a že políčko má rozměr **39×29** bodů, snadno spočítáte, že

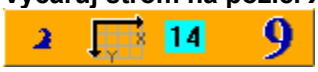
bod v levém horním rohu scény má souřadnice **X0 Y0**,
bod v pravém dolním rohu scény má souřadnice **X584 Y289**.

Zadávání souřadnic

Souřadnice zadáváme tak, že **za** prvek symbolizující požadovanou akci umístíme prvek/prvky symbolizující, co budeme právě zadávat.

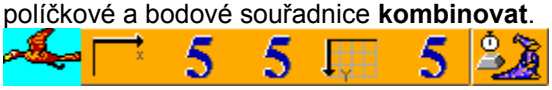


Vyčaruj strom na pozici X14 Y5, tzn. k pravému okraji scény, těsně pod jeho střed.



Přesuň Baltíka na pozici X14 Y9, tzn. do pravého dolního rohu scény.

Řadě prvků je jedno, jestli za nimi uvedete políčkové nebo bodové souřadnice. Většinou dokonce můžete políčkové a bodové souřadnice **kombinovat**.



Vyčaruj letícího ptáka na pozici x55Y5.

Předmět, **vyčarovaný na bodové souřadnice** se čaruje **bez obláčku** a není umístěn v žádném políčku scény, tzn. nelze zjišťovat jeho přítomnost na určité pozici scény pomocí **podmínek**.

Baltík přesunutý na bodové souřadnice se přesune do toho políčka, v němž leží bod o zadaných bodových souřadnicích.

Pozn.: Při zadávání souřadnic můžete zadávat potřebné hodnoty prostřednictvím **číselných výrazů.** Přitom můžete používat **funkce**, které vrací současné (aktuální) souřadnice Baltíka, animovaného předmětu, myši nebo neviditelného kurzoru

Pozn.: Pro práci s rozměry scény, obrazovky a předmětu můžete použít **systémových konstant**.

Náhodné číslo

V programech často potřebujeme zařídit, aby se některé věci děly zdánlivě náhodně. K tomu s výhodou použijeme prvek ***Náhodné číslo***, který dokáže generovat pseudonáhodná čísla, jejichž hodnotu není uživatel schopen dopředu určit.

Použijeme-li v programu prvek ***Náhodné číslo*** **samotný**, vrátí **reálné číslo z intervalu <0;1)**.

Pozn.: Všimněte si, že interval je z pravé strany otevřený, takže hodnoty **0** číslo nabýt může, ale hodnoty **1** ne.

Chcete-li generovat **náhodné reálné číslo v nějakém rozsahu** hodnot, musíte si pomoci násobením a sčítáním:

Vytiskni náhodné číslo od -1 do 1 (tj. 0 - 0.999… krát 2 -1, takže výsledkem je -1.. až 0.99..)

Potřebujete-li generovat **náhodné celé číslo** (např. když simulujete házení kostkou), zadáte požadovaný rozsah jako parametr. Přitom platí:

Ü je-li parametrem **číslo**, generuje se celé číslo z intervalu **<0; Parametr)**.

Ü je-li parametrem **interval**, generuje se číslo ze zadaného intervalu.

Vygeneruj některé z čísel 0, 1, 2, 3, 4, 5.

Vygeneruj některé z čísel 1, 2, 3, 4, 5, 6.

Příklady použití náhodného čísla:

Vyčaruj někde na scéně náhodně žábu

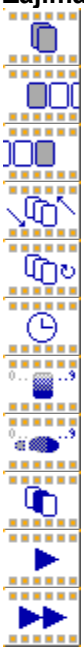
Otoč se na východ a náhodně krát (avšak ne dále, než ke kraji) opakuj {vyčaruj strom, popojdi}.

Pozn.: **14-XBaltika** je vzdálenost Baltika od pravého okraje scény.

Animace

Jednoduché animace

Zajímá-li vás pouze funkce konkrétního prvku, klepněte na něj a přesunete se na příslušný výklad.



V úrovni Pokročilý nabízí Baltík řadu nových možností, s jejichž pomocí lze vytvářet mnohem efektnější animace, než tomu bylo u jednoduchých animací v režimu Začátečník.

Animovaný předmět se může **přesouvat po bodech** a navíc je možno nastavovat řadu parametrů, které mohou ještě umocnit celkový dojem. Největším přínosem je však to, že po nastavení příslušných parametrů je Baltík schopen přehrát jednodušší animace **automaticky**.

Pro animaci předmětů nabízí Baltík **dvě možnosti**:

- Ü Vystačíte-li s jednoduchou animací stojícího nebo rovnoměrně přímočaře se pohybujícího předmětu, oceníte **jednoduchost použití automatické animace,**.
- Ü Chcete-li vytvářet složitější animace jako animace po křivce nebo animace s různými doprovodnými efekty odvozenými od kolizí s jinými animacemi apod., využijete **univerzálnost ruční animace,** která vám umožní přesně ovládat každé zobrazení animovaného předmětu, takže se animovaný předmět může chovat přesně tak, jak potřebujete.

Automatická animace

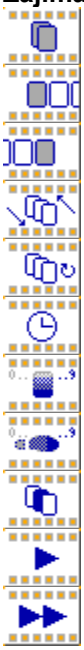
Viz též:Ruční animace

Proces programování animace sestává z několika fází:

- **1) Definice jednotlivých fází animace**
- **2) Zadání čísla (identifikátoru) animace**
- **3) Nastavení parametrů celé animace**
  - Zadání předmětů pro zobrazení fází
  - Styl změny fází - animační cyklus
  - Počet opakování
  - Trvání
- **4) Nastavení parametrů pro počátek a konec**
  - Souřadnice
  - Viditelnost
  - Otisk
  - Fáze
- **5) Spuštění** nebo
- **6) Přetočení** animace

U řady nastavení budeme hovořit o **animačním cyklu,** jehož délka leccos ovlivňuje.

Zajímá-li vás pouze funkce konkrétního prvku, klepněte na něj a přesunete se na příslušný výklad.



1) Definice jednotlivých fází animace

1. Rozmyslete si, z kolika fází (postupně vykreslovaných předmětů) se bude animace skládat a jak budou jednotlivé fáze vypadat.
2. **Nakreslete jednotlivé fáze animovaného předmětu** tak, aby byly v **bance předmětů** umístěny za sebou.

Tip: Pro inspiraci se můžete podívat do dodávaných bank, kde najdete příklady rozkreslených pohybů: letící pták, poskakující opice, rostoucí hříbek, otevírající se dveře, chodící chlapeček a

holčička, kráčející Baltík, čarovací obláček a další.

Jednotlivé fáze otevírajících se dveří

2) Určení animace

Urči animovaný předmět

Baltík umožňuje definovat až **100** nezávislých animovaných předmětů. Tyto předměty označujeme čísly od **1** do **100**.

Každý blok příkazů pro animaci musí začínat prvkem *Urči animovaný předmět* následovaným číslem (identifikátorem) příslušného předmětu. Nezadání čísla má stejný efekt, jako zadání čísla **1**.

Číslo animace nemá jen význam identifikační, ale zároveň **zařazuje animaci do příslušné vrstvy**. Čím je vyšší číslo animace a tedy i její vrstvy, tím víc je animace na vrchu. Při svém pohybu po obrazovce pak **zakrývá animace s čísly nižšími a je zakrývána animacemi s čísly vyššími**.

Blok příkazů pro animaci, následující za určením animovaného předmětu, musí být **na jediném logickém řádku nebo uvnitř příkazových (složených) závorek**.

Nastavíte-li jednou nějaké parametry animace a rozhodnete se později některé z nich změnit, stačí zadat pouze měněné parametry. **Měníte-li některou z vlastností animace, žádná jiná vlastnost se nemění.** Změníte-li např. předměty tvořící jednotlivé **fáze** animace, pak **souřadnice**, **viditelnost**, **čas**, **styl** atd. se nemění.

3) Nastavení parametrů celé animace

3.1 Zadání předmětů pro zobrazení fází

Abyste vůbec mohli nějakou animaci spustit, musíte především zadat její jednotlivé fáze, tj. určit, z jakých předmětů se bude animace skládat.

Fáze animace zadáte tak, že za určení animovaného předmětu umístíte první předmět a za něj poslední předmět ze skupiny po sobě jdoucích **předmětů zobrazujících jednotlivé fáze** budoucí animace.

Animace 3 bude zobrazovat otevírající se dveře.

Baltík netrvá na tom, abyste zadávali předměty ve vzestupném pořadí. Můžete je zadat i obráceně - např. u otevírajících se dveří zadáte nejprve otevřené dveře a pak dveře zavřené. Baltík bude za **počáteční předmět animace** považovat vždy **ten, který je uveden jako prvý** a naopak za poslední předmět animace ten, který je uveden jako druhý.

Nezadáte-li předměty pro zobrazení fází, použije se **prázdný předmět** (předmět číslo **0**).

Tip: Pokud nechcete animaci ihned přehrávat, je vhodné při zadávání fází nastavit současnou **viditelnost** animovaného předmětu na **0** (neviditelný), jinak by se hned zobrazila první fáze.

3.2 Styl animace - animační cyklus

Prvek ***Styl animace***
Číslem za prvkem ***Styl animace*** zadáváte styl přehrávání animace:

- Ü **0** nastavuje opakování fází **cyklicky - dokola**, tj. za poslední fází se zobrazí opět první fáze atd.
- Ü číslo **různé od 0** nebo zadání stylu bez čísla nastavuje styl **ping-pong**, tj. za poslední fází se zobrazí předposlední fáze atd.

Nenastavíte-li nic je nastaveno **cyklické** opakování.

Příklad:
Má-li animace 4 fáze, bude se přehrávat:
Cyklicky: **1 - 2 - 3 - 4 - 1 - 2 - 3 - 4 - 1** atd.
Ping-pong: **1 - 2 - 3 - 4 - 3 - 2 - 1 - 2 - 3** atd.

Na nastaveném stylu animace závisí počet fází, které zahrnuje jeden **animační cyklus**:

- Ü je-li nastaven **cyklický** styl animace, přehrají se v jednom animačním cyklu všechny fáze od začátku do konce,
- Ü je-li nastaven styl animace **ping-pong**, přehrají se v jednom animačním cyklu nejen všechny fáze od první do poslední, ale hned se pokračuje zpětným během, při němž se zpátky přehrají fáze od předposlední do první.

3.3 Počet opakování

Prvek ***Počet opakování***
Číslem za příkazem ***Počet opakování*** zadáváte, kolik **animačních cyklů** se má **přehrát** během jedné animace.

Nezadáte-li počet opakování nebo zadáte-li počet opakování menší než **1**, zobrazí se posloupnost fází jednou.

Pamatujte na odlišnosti v délce animačního cyklu u cyklické animace a u animace typu ping-pong:

- Ü u cyklické animace se při **N** opakováních animace o **K** fázích zobrazí celkem **K*N** jednotlivých fází,
- Ü u animace typu ping-pogk se při **N** opakováních animace o **K** fázích celkem zobrazí **2*N*(K-1)+1** fází. Chcete-li tedy 3 krát opakovat animaci, která má 5 fází, přehraje se celkem 2*3*4+1=25 fází.

3.4 Trvání animace

Prvek ***Trvání automatické animace***
umožňuje nastavit (číslem v milisekundách), jak dlouho má celá animace trvat.

Nastavením trvání animace proto spolu s počtem opakování **určujete počet promítnutých snímků** (zobrazených fází) animace.

Počítač standardně "promítá" rychlostí **18,2 snímků za sekundu**. Před spuštěním animace vždy spočte počet promítaných snímků a rozdělí je mezi jednotlivé fáze animace. Přitom zjistí, zda bude některé fáze **zobrazovat v několika snímcích po sobě** anebo zda naopak některé fáze zobrazit nestihne a bude je proto muset **přeskočit**.

Baltík zaručuje, že **vždy promítne závěrečnou fázi animace**. S přibývajícím časem, který mu pro animaci poskytnete, se budou postupně objevovat další fáze přičemž počáteční fáze se objeví jako

poslední.

Nezadáte-li trvání animace, bude promítnut zadaný počet **animačních cyklů,** a to maximální rychlostí, tj. 18,2 snímků/sekundu. 10násobné opakování cyklické animace o 9 fázích (dohromady 90 fází) tak bude promítnuto za zhruba 5 sekund, v případě animace ping-pong vzroste počet animací na 161 a jejich promítnutí bude trvat necelých 9 sekund.

4) Nastavení pro začátek a konec animace

Některé parametry animace je třeba zadávat zvlášť pro začátek animace a zvlášť pro její konec. To, zda zadáváte parametry pro začátek či konec určíte pomocí prvků ***Začátek animace*** a ***Konec animace***.

Prvek ***Začátek automatické animace***
uvozuje nastavení parametrů pro první snímek automatické animace.

Prvek ***Konec automatické animace***
uvozuje nastavení parametrů pro poslední snímek automatické animace.

Má-li být některé nastavení shodné pro začátek i konec animace, je možné je uvodit oběma prvky a nastavit hodnoty pro začátek i konec společně.

Tip: Chcete-li nastavit některé parametry pro začátek a konec animace stejně a některé jinak, vložte za dvojici prvků ***Začátek animace*** a ***Konec animace*** shodné parametry a pak použijte znovu prvek ***Začátek animace*** nebo ***Konec animace*** a zadejte další (rozdílné) parametry.

Nastavíte-li některý z dále uvedených parametrů **bez určení, zda jej nastavujete pro začátek či konec**, nastaví se příslušný parametr pouze pro aktuálně zobrazený snímek. Toho využíváme při **ruční animaci,**

4.1 Souřadnice

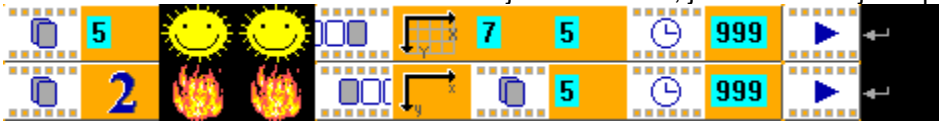
Jednou z důležitých charakteristik je určení **souřadnic** místa, kde má animace začínat, resp. končit.

Můžete používat jak souřadnice políček (souřadnice s velkými písmeny), tak přesnější souřadnice bodů (souřadnice s malými písmeny).

Pro zadání souřadnic můžete **použít i souřadnic objektů**, jejichž polohu lze snadno zjistit. Můžete tak použít např. **souřadnice myši** nebo souřadnice jiné animace(např. vyskakuje-li animovaný parašutista z animovaného letadla, bude animace jeho seskoku začínat na pozici jeho mateřského letadla).

Nezadáte-li souřadnice začátku (konce), bude animace začínat (končit) na políčku před Baltíkem.

Souřadnice animace je možné nejen nastavovat, ale i zjišťovat. V takovém případě musíte za prvek ***Souřadnice*** zadat určení animovaného objektu obdobně, jako v následujícím příkladu.



Z pozice před Baltíkem vycestuje sluníčko přibližně do poloviny obrazovky. Z koncové pozice sluníčka vyběhne ohýnek a vrátí se před Baltíka.

4.2 Viditelnost

Prvek ***Viditelnost animovaného předmětu***
Číslo za příkazem určuje, jak viditelný bude animovaný předmět na začátku (konci) animace. Viditelnost **0** znamená neviditelný předmět, **1** znamená velmi málo viditelný, atd. až viditelnost **9** znamená předmět viditelný úplně.

Nezadáte-li viditelnost, bude předmět **zcela viditelný**, viditelnost je **9**.

4.3 Stopa

Prvek ***Viditelnost stopy***
Číslo za příkazem určuje jak výrazné stopy za sebou animace zanechává. **0** označuje **žádnou stopu**, s rostoucími čísly se otisk zviditelňuje až **9** označuje **plně viditelnou stopu**.

Nezadáte-li intenzitu stopy, je nastavena na nulu, animace nezanechává **žádnou stopu**.

4.4 Fáze

Prvek ***Nastav fázi***
Číslo za prvkem specifikuje fázi, kterou má animace začínat, resp. končit. První fáze má číslo **1**.

Nezadáte-li počáteční fázi, nastaví se jako **počáteční první fáze**. Nezadáte-li koncovou fázi, nastaví se
- Ü pro styl animace dokola poslední fáze,
- Ü pro styl animace tam a zpět znovu první fáze.

5) Přehrání animace

Prvek ***Přehraj automatickou animaci***
Přehraje animaci od současného snímku.

Chcete-li přehrát pouze předem definovaný počet snímků, zadejte jejich počet za prvkem ***Přehraj automatickou animaci***. Příští přehrávka bude začínat následujícím snímkem.

- Ü **nezadáte-li nic**, animace se přehraje až do konce **animačního cyklu** a pak se přetočí zpět na začátek,
- Ü zadáte-li **záporné číslo**, animace se bude přehrávat zpět.
- Ü zadáte-li pouze prvek ***Mínus***, animace se přehraje na začátek.

Prvek ***Minus***

6) Přetočení animace

Prvek ***Přetoč automatickou animaci***
přetočí (skrytě přehraje) animaci o zadaný počet snímků. Jeho parametry jsou totožné s parametry prvku ***Přehraj automatickou animaci,*** tj.:

- Ü Nezadáte-li počet snímků, přetočí se animace na konec.
- Ü Zadáte-li záporné číslo, přetočí se animace o zadaný počet snímků zpět.
- Ü Zadáte-li pouze prvek ***Mínus***, animace se přetočí na začátek.

Ruční animace

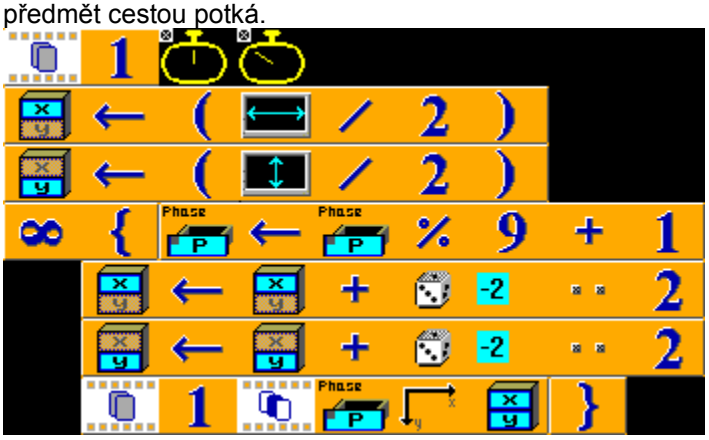
Automatická animace

Možnosti ruční animace využijeme tehdy, pokud potřebujeme dosáhnout efektů, které nám automatická animace neumožňuje nebo které se s ní dosahují obtížně.

Při ruční animaci nespouštíme animaci příkazy ***Přehraj automatickou animaci,*** ale využíváme toho, že příkaz ***Určení animovaného předmětu*** zobrazí fázi specifikovanou následnými parametry. (Nechcete-li ji zobrazit, musíte ji nastavit neviditelnou.) To nám umožňuje definovat pro každou zobrazovanou fázi její vlastní parametry.

Při ruční animaci nedefinujeme parametry pro **začátek a konec animace,** ale definujeme je přímo pro určený animovaný předmět.

Pomocí ruční animace tak můžeme snadno pohybovat animovaným předmětem po libovolně křivolaké dráze a doprovázet ji nejrůznějšími efekty odvozenými např. z toho, které jiné animované předměty tento předmět cestou potká.



V programu vlevo Baltík náhodně pohybuje animovanými hodinami (animace o 8 fázích).
Musí průběžně vypočítávat číslo následně zobrazované fáze. Využívá k tomu proměnné ***Phase***, v níž má uloženu naposledy použitou fázi a do níž ukládá fázi příští (zapamatujte si použitý vzoreček - může se vám hodit).
Souřadnice si pamatuje v proměnné ***xy***. Před zobrazením další fáze vždy posune obě souřadnice o náhodné číslo v rozsahu <-2;2>.

Konstanty

Proměnné

V této pasáži jsou zpracována následující témata:

1) Co je to konstanta

Chcete-li mít program opravdu přehledný, musíte používat co nejméně **literálů.** Pro pevné hodnoty, které jsou definovány při běhu programu a po celou dobu programu se nemění, je výhodnější používat **konstanty**.

Konstanta je pojmenovaný prvek, který má po celou dobu běhu programu stejnou hodnotu.

Hlavní rozdíl mezi konstantou a literálem je ten, že **konstanta je pojmenovaná**, kdežto literál je pouze přímým zobrazením hodnoty. To má velký význam pro přehlednost programu. Podrobněji viz **Použití konstant a jeho výhody.**

Prvky představující konstanty jsou uloženy ve zvláštních bankách uložených odděleně od bank předmětů.

Ukázku banky konstant zvětšíte tak, že na ni klepnete myší

2) Rozdělení konstant

Rozdělení podle typu

Aby bylo na první pohled patrné, jakého typu je hodnota konstanty, jsou jednotlivé typy konstant odlišeny barvou:

Celočíselná konstanta (azurová)

Reálná konstanta (zelená)

Řetězcová konstanta (žlutá)

Konstanty každého typu mají pro sebe vyhrazenu jednu zvláštní banku.

Rozdělení podle účelu

V rámci jedné banky jsou konstanty dále rozděleny:

- Ü na posledním řádku se nacházejí **násobné konstanty**, které využijeme především při práci se **souřadnicemi.**
- Ü na předposledním řádku (u celočíselných konstant na dvou předposledních řádcích) se nacházejí předdefinované **systémové konstanty,**
- Ü zbytek banky tvoří **jmenné konstanty**, které můžete vytvořit a pojmenovat.

3) Vložení konstanty do programu

Konstantu lze vložit do programu několika způsoby:

Z panelu příkazů vyberte prvek ***Nová proměnná*** a umístěte jej na požadované místo do programu. Po umístění prvku se otevře okno **Vlastnosti prvku**, ve kterém určíte typ konstanty, její název a hodnotu.

nebo

Stiskem tlačítka ***Konstanty, proměnné*** v panelu tlačítek nad nabídkou příkazů zobrazte **zvláštní banky** a v nich požadovanou konstantu vyberte. Vyberete-li jednu ze dosud nepojmenovaných jmenných konstant, otevře se okno **Vlastnosti prvku**, kde ji budete moci pojmenovat a přidělit jí hodnotu.

4) Změna vlastností konstanty

Budete-li chtít ve svém programu změnit vlastnosti některé konstanty (název, typ, hodnotu apod.), stačí na tuto konstantu klepnout pravým tlačítkem myši a z místní nabídky vybrat možnost ***Vlastnosti***. Tím otevřete okno **Vlastnosti prvku**, kde budete moci požadované vlastnosti upravit.

5) Použití konstant a jeho výhody

Hlavní rozdíl mezi konstantou a literálem je ten, že **konstanta je pojmenovaná**, kdežto literál je pouze přímým zobrazením hodnoty. To má velký význam pro přehlednost programu. Konstanta, na rozdíl od literálu, nevystupuje v programu pod svou hodnotu, ale pod svým jménem.

Rozhodneme-li se v programu změnit nějakou klíčovou pevnou hodnotu, musíme při použití literálu procházet celý program a zaměňovat všechny jeho výskyty. Přitom si jej nesmíme splést s jiným literálem, který má náhodou stejnou hodnotu.

Rozhodneme-li se např. změnit termín maximální věk účastníků soutěže z **12** na **15** roků, nesmíme si splést literály (dvanáctky), které je třeba změnit, s literály (dvanáctkami), kde číslo **12** představuje např. počet měsíců v roce. Naproti tomu při použití konstant stačí **změnit hodnotu** v definici konstanty a tím je vše hotovo.

Konstanty používejte všude tam, kde chcete v programu použít vícekrát stejné číslo nebo řetězec

(tzv. literály).

Konstantu **pojmenujte podle toho, co skutečně obsahuje**. Pokud např. vytváříte hru, ve které přibude hráči další život za určitý počet bodů, řekněme 1000, vytvoříte si v okně **Vlastnosti prvku** konstantu s názvem Bonus a naplníte ji hodnotou 1000:

Celočíselná konstanta Bonus

Použití konstant:

- Ü usnadňuje údržbu programu (všechny konstanty jsou uvedeny přehledně na jednom místě),
- Ü usnadňuje zápis programu (nemusíte pokaždé vypisovat celé číslo),
- Ü zpřehledňuje program (z názvu konstanty poznám, jaký druh informace obsahuje),
- Ü usnadňuje pozdější modifikace (úpravy) programu.

6) Systémové konstanty

Za jmennými konstantami jsou **systémové konstanty**, které mají předem danou hodnotu i jméno a jsou obecně použitelné ve všech programech, např.

Šířka obrazovky v bodech = 585 (příklad systémové konstanty)

6.1. Celočíselné konstanty

Šířka scény v políčcích = 15

Výška scény v políčcích = 10

Šířka obrazovky v bodech = 585

Výška obrazovky v bodech = 290

Počet předmětů v bance = 150

Šířka předmětu = 39

Výška předmětu = 29

Maxint - nejvyšší celé číslo = 2147483647

Pozn.: Nejmenší celé číslo je -2147483648, tedy **-Maxint-1**

Prázdné políčko = 0

Okraj scény = 151

Nepravda, Ne, Vypnout = 0

Pravda, Ano, Zapnout = 1

Tiskárna = 0

6.2. Reálné konstanty

Pí - Ludolfovo číslo = 3.14159265359

e - základ přirozeného logaritmu = 2.718281828459

Minreal - nejmenší kladné reálné číslo = 2.9e-39

Maxreal - největší kladné reálné číslo = 1.7e38

Pozn.: Záporná čísla mohou být v rozsahu -Maxreal až -Minreal.

Převod z radiánů na stupně = 180/pí

Převod ze stupňů na radiány = pí/180

6.3. Řetězcové konstanty

Styl písma:tučné = "B"

Styl písma:kurziva = "I"

Styl písma:podtržené = "U"

Styl písma:přeškrtnuté = "S"

Konec řádku = #13#10

Pozn.: **Zobrazením** této konstanty bez udání souřadnic se přesunou aktuální souřadnice zobrazení na nový řádek o výšce nastaveného písma.

Prázdný řetězec = ""

Mezera = " "

Pozn.: Následující čtyři řetězce používáme při reakcích na **Události**

Stisknuta klávesa = "D"

Puštěna klávesa = "U"

Přečten znak = "C"

Vypršel čas časovače = "T"

Pozn.: Následující dva řetězce používáme při práci se složkami a soubory

Složka/název spuštěného programu = "*\"

Dočasná složka/soubor = "?\"

Proměnné

Konstanty

Obsah

V této pasáži jsou zpracována následující témata:

- **Co je to proměnná**
- **Rozdělení proměnných**
- **Banky proměnných**
- **Vložení proměnné do programu**
- **Použití proměnné**
- **Sledování proměnných v programu**

1) Co je to proměnná

Proměnnou si můžete představit jako **pojmenovanou schránku**, do níž ukládáte **hodnotu**, kterou chcete později v programu použít.

Pamatujte, že v programech je třeba jasně **rozlišovat, kdy se jedná o vlastní proměnnou** (tj. o onu pojmenovanou schránku) **a kdy se jedná o její hodnotu** (tj. o to, co je ve schránce uloženo).

Narazíte-li tedy v dalším textu na pojem proměnná, vždy si dobře uvědomte (podle použití), o čem je řeč, zda o schránce nebo o obsahu anebo o obojím.

Prvky představující proměnné jsou uloženy ve zvláštních bankách uložených odděleně od bank předmětů.

Ukázku banky proměnných zvětšíte tak, že na ni klepnete myší

2) Rozdělení proměnných

Rozdělení podle typu

Celočíselné proměnné (azurové)

Reálné proměnné (zelené)

 **Řetězcové proměnné (žluté)**

Rozdělení podle viditelnosti

Baltík rozeznává dva druhy viditelnosti proměnných:

Ü **globální** proměnné (**šuplíky**) jsou viditelné z každého místa programu, kdežto

Ü **lokální** proměnné (**košíky**) používá pouze nějaká část programu a nikdo jiný o nich neví.

 **Globální proměnné (šuplíky)**

Globální proměnné, tj. ty, co jsou odevšad přístupné, jsou v Baltíkovi zobrazeny jako šuplíky. Jejich dostupnost si můžete zdůvodnit tak, že kdesi na veřejně přístupném místě je šuplíková skříň, kam si kdokoliv může dojít a podívat, co je v šuplíku (přečíst a použít hodnotu této proměnné) nebo do šuplíku něco uložit (přiřadit proměnné hodnotu).

Globální proměnná může mít (na rozdíl od lokální) hned po spuštění programu přirazenu **počáteční hodnotu**, která se nastavuje v **okně vlastností**.

Aby bylo na první pohled zřejmé, zda má proměnná **nastavenu počáteční hodnotu**, zobrazuje Baltík u proměnných s nastavenou počáteční hodnotou v levém horním rohu čelní strany jejich šuplíku malý šedý čtvereček.

 **Lokální proměnné (košíky)**

Lokální proměnná je taková proměnná, kterou vidí (a může používat) pouze ten **pomocník**, v jehož **definici** se daná proměnná vyskytuje. Ostatní pomocníci ani samotný program k ní nemají přístup.

Lokální proměnné jsou v Baltíkovi zobrazeny jako košíky. Jejich nedostupnost si můžete zdůvodnit tak, že si každý pomocník nosí své košíky pořád s sebou a nikomu nedovolí, aby se mu do nich díval nebo se v nich dokonce přehraboval.

Podrobnější výklad lokálních proměnných najdete v pasáži o **pomocnících** v části nazvané **Používání proměnných.**

3) Banky proměnných

Proměnné se nacházejí v **bankách proměnných**.

Obrázkové proměnné

Prvních 30 proměnných je **obrázkových**. Jejich názvy jsou předem definovány a nemůžete je změnit. Chcete-li vědět, jak se proměnná jmenuje, ukažte na ni v okně ***Výběr předmětu*** myší a podívejte se na stavový řádek okna.

Obrázkové proměnné je výhodné používat pro uchovávání hodnot souvisejících s obrázkem - např. do proměnné ***Klávesa*** ukládejte číslo přečtené **klávesy**, do proměnné ***Náhodné číslo*** můžete uložit **náhodné číslo**, jehož hodnotu budete později potřebovat apod.. Tím se váš program stane zase o trochu přehlednějším a čitelnějším.

Jmenné proměnné

Dalších 89 proměnných je **jmenných**. Každou z těchto proměnných, musíte před tím, než ji v programu použijete, **pojmenovat**, aby bylo v každém okamžiku jasné, o kterou proměnnou se jedná. Zadaný název se pak bude objevovat v horní části obrázku daného prvku.

Název, kterým proměnnou pojmenujete,

- Ü **nesmí** obsahovat písmena s diakritickými znaménky (čárky, háčky, kroužky atd.) a
- Ü **musí** být v dané bance jedinečný, tj. nesmí jej mít žádná proměnná tohoto typu.

Indexované proměnné

Za poslední jmennou proměnnou je prvek, používaný pro **indexované** (číslované) **proměnné**. Indexovanou proměnnou vytvoříte tak, že za prvek ***Indexovaná proměnná*** zadáte do hranatých závorek její **index** - libovolné **celé číslo**. Indexovanou proměnnou pak můžete používat jako každou jinou proměnnou.

Pole

Na dalším řádku jsou pole. O jejich používání se dozvíte více v pasáži **pole**.

Souřadnicové proměnné

Poslední řádek obsahuje **souřadnicové proměnné**. Proměnné **X**, **Y**, **x**, **y**, **x1**, **y1**, **x2** a **y2** můžete používat jako ostatní proměnné. Navíc však můžete odpovídající dvojice spojit do dvojproměnné a tu pak používat jako **souřadnice** bodu, případně spojit proměnné **x1**, **y1**, **x2** a **y2** do čtyřproměnné, kterou je možno použít jako souřadnice **oblasti**.

Na konci řádku jsou prvky pro **indexované dvojproměnné a čtyřproměnné**, které vytvoříte tak, že prvek pro příslušnou proměnnou zadáte do hranatých závorek její **index** - libovolné **celé číslo**. Indexovanou souřadnicovou proměnnou pak můžete používat jako každou jinou proměnnou.

Použití proměnných v programu

V této sekci se dozvíte

- Ü jak **vložit** proměnnou do programu,
- Ü jak proměnnou v programu **používat** a
- Ü jak **sledovat** chování proměnných při běhu programu.

1) Vložení proměnné do programu

Proměnnou můžete do programu vložit několika způsoby.

Chcete-li mít možnost vybírat **ze všech** proměnných, pak:

1. Stiskněte tlačítko  (***Konstanty, proměnné***) v panelu tlačítek nad nabídkou příkazů
nebo
klepněte pravým tlačítkem na místo, kam chcete proměnnou vložit, a z místní nabídky zadejte ***Vložit prvek → konstantu, proměnnou***.

2. V následně zobrazeném okně **Výběr předmětu**, vyberte zvolenou proměnnou.

3. Vyberete-li dosud nepojmenovanou jmennou proměnnou, otevře se okno **Vlastnosti prvku**, kde ji budete moci pojmenovat a případně i přiřadit počáteční hodnotu.

Stačí-li vám vybírat **pouze z pojmenovaných proměnných**, můžete v nabídce příkazů vybrat prvek ***Nová proměnná***. Baltík pak rovnou otevře okno **Vlastnosti prvku.**

Pozn.: Budete-li chtít později změnit některou vlastnost (název, typ, počáteční hodnotu) proměnné, kterou máte ve svém programu, stačí na tuto proměnnou klepnout pravým tlačítkem myši a z místní nabídky vybrat možnost **Upravit.**.

2) Použití proměnné

Proměnnou v programu používáme dvěma způsoby:

- Ü použijeme uloženou hodnotu nebo
- Ü uložíme do ní novou hodnotu

Použití hodnoty

Hodnotu uloženou v proměnné použijeme tak, že do příkazu vložíme místo požadované hodnoty danou proměnnou.

Příklady:

Délka krát popojdi - Baltík popojde o tolik políček, kolik je uloženo v proměnné ***Délka***.

Na souřadnicích XY vyčaruj okno - Baltík vyčaruje okno na políčku, jehož souřadnice jsou v proměnných X a Y.

Na pozici před Baltíkem zobraz obsah proměnné *Pozdrav* - před Baltíkem bude zobrazen řetězec (text), který je uložen v proměnné ***Pozdrav***.

Pozn.: Obecně můžeme na takové místo vložit libovolný **číselný** nebo **řetězcový** výraz

Uložení nové hodnoty

K uložení nové hodnoty do proměnné slouží **příkaz přiřazení,** který je podrobně probrán v samostatné sekci.

3) Sledování hodnot proměnných při běhu programu

Je smutným pravidlem, že v každém programu se vyskytne nějaká chyba. Proces odhalování chyb v programu se nazývá **ladění**.

Chcete-li zjistit, zda je v daném okamžiku v proměnné požadovaná hodnota, můžete postupovat následovně:

1. Na místo programu, kde vás zajímá hodnota některé proměnné umístěte příkaz pro čekání na zadání znaku z klávesnice.
2. Spusťte program a počkejte až se dostane k příslušnému místu.
3. Nebudou-li pod pracovní scénou zobrazeny hodnoty použitých proměnných, stiskněte tlačítko ***Ukázat proměnné***.

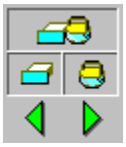
Tlačítko ***Ukázat proměnné*** se nachází pod levým okrajem pracovní scény

Po jeho stisku se tlačítko změní na trojici tlačítek: ***Schovat proměnné***, ***Schovat šuplíky*** a ***Schovat košíky***. Vpravo od této trojice budou zobrazeny názvy a obsahy proměnných.

Tlačítky ***Ukaž/schovej globální (lokální) proměnné*** přepínáte, zda chcete vidět hodnoty použitých globálních (lokálních) proměnných, nebo ne.

Proměnné budou seřazeny podle svého pořadí v bankách, tj.
- Ü nejprve podle typů v pořadí celočíselné - reálné - řetězce.
- Ü v rámci typů budou uvedeny nejprve globální a pak lokální,
- Ü uvnitř skupiny podle umístění v příslušné bance.

Názvy a hodnoty budou mít barvu podle typu proměnné. Globální proměnné budou zobrazeny tučně, lokální tučnou kurzívou.



Nevejde-li se výpis hodnot všech proměnných do vyhrazeného prostoru, doplní se ***Ovládací panel sledování hodnot proměnných*** dole tlačítky ***Další proměnné*** a ***Předchozí proměnné***, pomocí nichž můžete přepínat zobrazované stránky

Okno vlastností

Proměnné Použití proměnných v programu

Vložení konstanty nebo proměnné

Při vkládání nové proměnné, při výběru nepojmenované jmenné **konstanty** nebo **proměnné** a při výběru možnosti ***Vlastnosti*** v místní nabídce prvků programu, případně po stisku pravého tlačítka myši ve zvláštních bankách se otevře okno ***Vlastnosti***.

Obrázek okna vlastností zvětšíte tak, že na něj klepnete myší

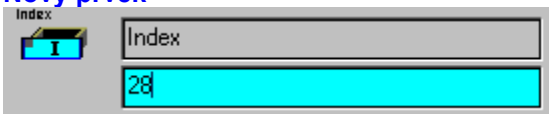
Původní prvek

V levé části okna ***Vlastnosti*** se nahoře nachází **původní** obrázek, název a hodnota prvku, jehož vlastnosti jste se rozhodli změnit.

- Ü Klepnutím na tento obrázek **obnovíte původní stav**.
- Ü Klepnutím **pravým** tlačítkem **otevřete banku předmětů** na stránce, na které se původní prvek nachází. Z banky můžete **vybrat nový prvek**.

Pozn.: Při zadávání nového prvku se v okně vlastností původní prvek nezobrazuje.

Nový prvek



Pod původním prvkem je zobrazen nový prvek, tedy ten prvek, kterým bude původní prvek nahrazen v případě, že se rozhodnete záměnu dokončit.

Můžete **měnit název i hodnotu** prvku, výsledek ihned vidíte na obrázku nového prvku.

Barva pozadí pole hodnota ukazuje **typ** zadávané hodnoty (azurová - celé číslo, zelená - reálné číslo, žlutá - řetězec, bílá - nic).

Pokud **typ neodpovídá** typu vybraného prvku, je vedle pole hodnota zobrazen **otazník**.

Je-li u globální proměnné zadána **počáteční hodnota**, je na obrázku nového prvku zobrazena **šedá značka**.

Pozn.: U **systémových konstant** nemůžete měnit název ani hodnotu.

Pozn.: U **předmětů** a **kláves** nezadáváte název, ale do pole hodnota zadáváte číslo předmětu, případně klávesy.

Klepnutím **pravým** tlačítkem na obrázek nového prvku **otevřete banku předmětů** na stránce, na které se Nová proměnná nachází. Z této banky pak můžete **vybrat nový prvek**.

Existující prvek se stejným názvem

Zadáte-li název (stačí začátek názvu) proměnné nebo konstanty, která **již existuje**, tento prvek bude zobrazen pod novým prvkem.

Zobrazení tohoto prvku je jak nabídkou, tak i varováním, že takovýto prvek již existuje. Za hodnotou existujícího prvku jsou údaje o výskytu tohoto prvku v programu ve formátu

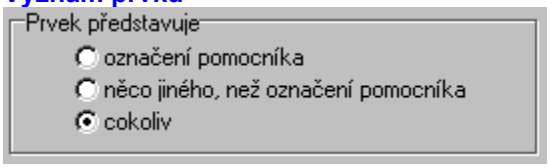
pvtp / pvcp (pppp)

kde **pvtp** je počet výskytů v aktuálním pomocníkovi,
pvcp je počet výskytů v celém programu a
pppp je počet pomocníků, používajících tento prvek.

Klepnutím na obrázek existujícího prvku **přenesete** tento prvek a jeho vlastnosti **do** pole **nového prvku**.

Klepnutím **pravým** tlačítkem na obrázek existujícího prvku **otevřete banku** předmětů na stránce, na které se tento prvek nachází. Z této banky pak můžete vybrat nový prvek.

Význam prvku

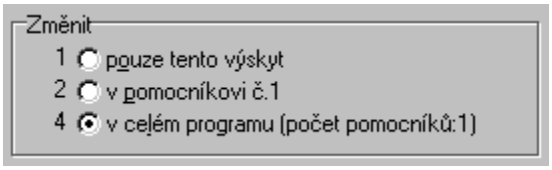


Přepínač významu prvku využijete zejména tehdy, máte-li některé pomocníky pojmenované předmětem:

- Ü chcete-li změnit tento předmět pouze tam, kde znamená označení pomocníka (tedy ve volání a v definici pomocníka), zvolte ***označení pomocníka***,
- Ü chcete-li naopak změnit pouze předmět v příkazech čaruj předmět, urči animovaný předmět a podobně, zvolte ***něco jiného, než označení pomocníka***,
- Ü chcete-li nahradit všechny stanovené výskyty bez ohledu na význam prvku, zvolte ***cokoliv*** (tato volba je přednastavena).

Podle vaší volby se ihned aktualizují počty výskytů původního prvku v okně ***přepínač rozsahu nahrazování***.

Rozsah nahrazování



Při změně vlastností prvků v programu si můžete vybrat, zda chcete změnit

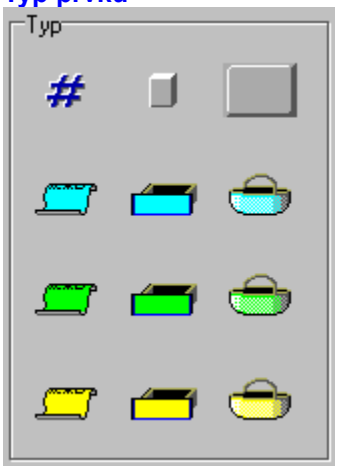
Ü pouze tento výskyt daného prvku (tj. ten, na kterém jste vyvolali z lokální nabídky okno vlastností),

Ü všechny výskyty v tomto pomocníkovi nebo

Ü všechny výskyty daného prvku v programu.

Před každým stavem přepínače je zobrazeno, **kolik se provede nahrazení**.

Typ prvku



V pravé části okna jsou přepínače typu a druhu prvku (číslo, klávesa, předmět, jednotlivé typy konstant, globálních a lokálních proměnných).

Klepnutím na obrázek typu **přepnete typ prvku** v poli ***Nová proměnná***.

Klepnutím pravým tlačítkem **otevřete banku předmětů** na stránce, na které se tento typ nachází. Z této banky pak můžete vybrat nový prvek.

Tip: Typ prvku můžete **přepínat** také stiskem klávesy ***Ctrl*** a kurzorových ***šipek***.

Přiřazení hodnoty proměnné

Příklady přiřazení

Přiřazovací příkaz musí mít vždy tvar:

<seznam proměnných> **<operátor přiřazení>** **<hodnota>**

<seznam proměnných>

Seznam proměnných je většinou tvořen jedinou proměnnou, ale může jím být i výčet několika proměnných nebo **interval proměnných**

Proměnné x, y, X, Y

Proměnné N, U, V, A1 až V1

Souřadnicové proměnné x1, y1, xy[5] až xy[15]

<hodnota>

Co můžeme přiřadit, závisí na typu proměnné, jíž danou hodnotu přiřazujeme:

- **číselným proměnným** (celočíselným i reálným) můžeme vedle **čísla** přiřadit i **předmět** nebo **klávesu,**
- **řetězcovým proměnným** musíme přiřadit **řetězec,**
- **číselným dvojproměnným** můžeme přiřadit **souřadnice** bodu nebo dvě čísla (kdyby mohlo dojít k záměně, musí se oddělit čárkou),
- **číselným čtyřproměnným** **souřadnice dvou bodů** nebo čtyři čísla (opět v případě nutnosti oddělená čárkou) nebo kombinaci dvojproměnné a dvou čísel,
- **řetězcovým dvojproměnným, resp. čtyřproměnným** dvojice, resp. čtveřice textů (např. jméno a příjmení do dvojproměnné nebo adresu do čtyřproměnné).

Pozn.: Připomínáme, že všude tam, kde smí být číslo, může být i jakýkoliv **číselný výraz.**

Pozn.: Pokud přiřazujete **reálné číslo do celočíselné proměnné**, je nejdřív převedeno na celé (je useknuta desetinná část).

<operátor přiřazení>

Operátorů přiřazení je několik. Rozdělujeme je na

- **jednoduché**, které sestávají z jediného prvku a
- **složené** vzniklé sestavením dvou prvků.

Jednoduché operátory

Operátor ***Přiřaď*** uloží do proměnných v seznamu hodnotu <hodnota>

Pozn.: Operátor ***Přiřaď*** můžete použít i k přiřazení do **souboru**, do **složky**, na **obrazovku**, do **oblasti**, při **předávání parametrů**, při **zobrazení**, k **nastavení stopek** a dalším akcím. Obdobně širší je i používání operátorů ***Zvětši*** a ***Zmenši***, o nichž budeme hovořit vzápětí.
V této pasáži zabýváme pouze přiřazením hodnoty proměnné.

Operátor ***Zvětši*** k hodnotě **celočíselné proměnné přičte** číslo <hodnota>, k hodnotě **řetězcové proměnné přidá** na její konec text <hodnota>.

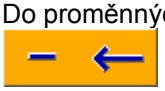
Operátor ***Zmenši*** od hodnoty **celočíselné proměnné odečte** číslo <hodnota>, od hodnoty **řetězcové proměnné odebere** text <hodnota>, tj. vyjme z řetězce v proměnné první výskyt podřetězce <hodnota>.

Pozn.: Více se o **práci s řetězci** dozvíte v pasáži **Řetězce**.

Složené operátory

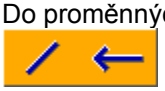
Složené přiřazovací operátory vzniknou použitím **aritmetického operátoru** následovaného prvkem ***Přiřaď***:

Do proměnných v seznamu uloží jejich obsah zvětšený o hodnotu <hodnota>.

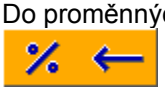


Do proměnných v seznamu uloží jejich obsah zmenšený o hodnotu <hodnota>.

Do proměnných v seznamu uloží jejich obsah vynásobený hodnotou <hodnota>.



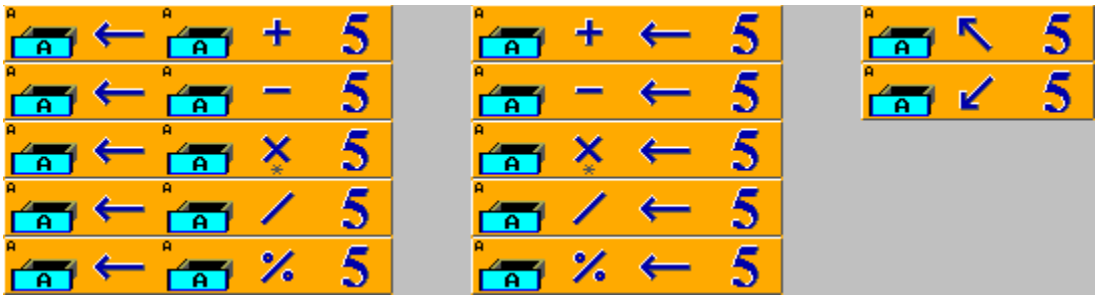
Do proměnných v seznamu uloží jejich obsah vydělený hodnotou <hodnota>.



Do proměnných v seznamu uloží zbytek po dělení jejich obsahu hodnotou <hodnota>.

Tabulka použití různých druhů přiřazení:

Baltík/C/BASIC/Pascal	Baltík/C	Zkrácený zápis



Příklady přiřazení:

Proměnné č. 5 přiřaď 84 - 2 x 5
Do indexované proměnné číslo **5** uloží hodnotu výrazu, tj. **74**.

***A* a *V* zvětši o číslo z klávesnice**
Proměnné ***A*** a ***V*** zvětší o číslo, které zadáte z klávesnice.

Proměnné Předmět přiřaď číslo vybraného předmětu

Proměnné *Panáček* přiřaď začátek + konec
Do řetězcové proměnné ***Panáček*** uloží řetězec, vzniklý spojením řetězců z proměnných ***Začátek*** a ***Konec***.

Proměnným *xy*, *XY*, *xy(0)* přiřaď souřadnice bodu [5;15]
Do obou bodových souřadnicových proměnných ***xy*** a ***xy(0)*** se přiřadí souřadnice bodu **[5;15]**, a do souřadnicových proměnných políčka ***XY*** se přiřadí souřadnice políčka, ve kterém tento bod **[5;15]** leží, tedy souřadnice **[0,0]**.

***x1y1x2y2* přiřaď 100, 100, 200, 200**
Do reálné čtyřproměnné ***x1y1x2y2*** se přiřadí postupně hodnoty **100**, **100**, **200**, **200**, takže v proměnných ***x1*** a ***y1*** bude hodnota **100** a v proměnných ***x2*** a ***y2*** bude hodnota **200**.

Vlastnosti Baltíka

Prvek ***Baltík***
slouží k ovládání řady vlastností Baltíka.

V této sekci vás upozorníme na **nové možnosti** vlastností, které jste s nimiž jste se seznámili v režimu ***Začátečník,*** nastavit či zjistit **polohu Baltíka** a **směr,** kam je natočen. zapnout nebo vypnout čarování s obláčkem, zvuky a také nastavit, jak bude Baltík, případně obláček vypadat.

1) Dříve nastavitelné vlastnosti

Přidáte-li za prvek ***Baltík*** některý z následujících prvků, bude výsledek stejný, jako kdybyste tento prvek umístili v programu samostatně **(klinutím na tlačítko se přesunete na vysvětlení jeho funkce)**:

Obdobná situace je u prvku ***Rychlost***, jediným rozdílem je to, že za prvek musíte ještě přidat číslo:

Uvedené vlastnosti můžete nyní nejenom nastavovat, ale můžete i **testovat** jejich současné nastavení a hodnotu **rychlosti** můžete dokonce **zjistit** a např. uložit do proměnné.

2) Zjištění a nastavení polohy Baltíka

Baltíkovy souřadnice můžete zjišťovat i nastavovat. Poloha Baltíka se zjišťuje a nastavuje pomocí **souřadnic:**

Zapamatuj si aktuální Baltíkovy souřadnice v proměnné XY

Přesuň Baltíka na políčko *X0 Y5*

Přesuň Baltíka na (0,0)

3) Zjištění a nastavení směru Baltíka

Stejně jako souřadnice můžete nastavovat i Baltíkův směr.

Prvek ***Směr***

Chcete-li zjistit směr, kterým je právě Baltík natočen, použijte samotný prvek ***Směr***. Tato funkce vrátí hodnotu 1 až 4 (1=východ, 2=jih, 3=západ, 4=sever).

Vytiskni na obrazovku číslo směru natočení Baltíka

Použijete-li prvek ***Směr*** s parametrem, Baltík se natočí do zadaného směru. Parametr smí přitom nabývat libovolných kladných hodnot (číslo se vždy vydělí 4 a směr otočení se nastaví podle velikosti zbytku po dělení):

Otoč Baltíka čelem vzad

Zjistí aktuální směr natočení a natočí jej do směru o 2 většího.

4) Políčko před Baltíkem

Vložením prvku ***Směr*** před Baltíkovu souřadnicí oznamujete, že se chystáte pracovat se souřadnicí políčka před Baltíkem.

Souřadnice X políčka před Baltíkem

Příkaz:

vyčaruje tulipán před Baltíka, ať už je Baltík otočen kamkoliv.

5) Animace obláčku

Pokud za prvek ***Čaruj s obláčkem*** vložíte předmět, bude tento předmět použit jako první předmět 8-mi předmětové posloupnosti pro animaci obláčku.

Čaruj s obláčkem, pro animaci obláčku použij předměty 3107 až 3114, tj. animaci výbuchu.

6) Animace Baltíka

Předměty, použité pro animaci Baltíka se mění zadáním prvního předmětu animační posloupnosti. Těchto předmětů je pro každý směr 5:

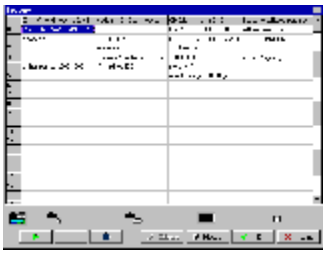
- Ü stojící Baltík,
- Ü dva předměty animující Baltíkovu chůzi,
- Ü čekající Baltík (obrázek střídaný při čekání s obrázkem stojícího Baltíka) a
- Ü čarující Baltík.

Těchto 5 předmětů se opakuje pro každý směr (východ, jih, západ, sever), takže celkem je k animaci použito 20 předmětů.

Použij pro Baltíkovu animaci opici
Za touto opicí musí být v bance dalších 19 předmětů s jednotlivými fázemi.

Tabulka souborů a oblastí

Oblasti Použití souborů z Tabulky



Tabulka souborů a oblastí usnadňuje práci se soubory a oblastmi. Obsahuje souřadnice **oblastí**, názvy **zvukových** souborů, hudebních souborů, **grafických** souborů, **audiovizuálních** souborů a **ostatních** souborů, které můžete používat ve svém programu.
Tabulku souborů zvětšíte tak, že na ni klepnete myší.

V této sekci se dozvíte:

1) Pravidla pro zadáváni názvů souborů

Název souboru může obsahovat:

- Ü **úplnou cestu**, tj. cestu od kořenové složky včetně případného označení mechaniky (např.: C:\Data\Obrázky\Baltík\Hra.B01),
- Ü **relativní cestu**, tj. cestu bez označení mechaniky (např. Zvuky\Auto.wav)
- Ü nemusí obsahovat **žádnou cestu** - stačí jen název souboru (např. Portrét.bmp).

Při spuštění programu je jako aktuální složka automaticky nastavena domovská složka spouštěného programu (tj. složka, ve které je uložen spouštěný Baltíkův program).

Neobsahují-li názvy souborů úplnou cestu, tj. cestu od kořenové složky včetně označení mechaniky, vztahují se všechny názvy k aktuální složce, tj. k domovské složce načteného programu (.bpr).

Příklad: Máte načtený program **C:\Programy\Baltík\Hra.bpr**. Pak úplná cesta k souboru zadanému jako **Pravidla.txt** je **C:\Programy\Baltík\Pravidla.txt** a úplná cesta k souboru zadanému jako **Data\Seznam.txt** je **C:\Programy\Baltík\Data\Seznam.txt**.

Pokud v programu aktuální složku změníte a chcete jednoznačně určit cestu k domovské složce spuštěného programu (.bpr), zadejte na začátek neúplné cesty speciální kombinaci dvou znaků *****. Názvy souborů z předchozího příkladu by pak měly tvar ***\Pravidla.txt** a ***\Data\Seznam.txt**.

Pozn.: Pokud používáte soubory s dlouhými názvy, obsahujícími mezery, uzavřete jméno souboru do uvozovek, např. **"Můj soubor.txt"**.

2) Otevření tabulky souborů a oblastí

Tabulku otevřete klepnutím na tlačítko ***Tabulka*** nad pracovní plochou položky nebo zadáním povelu ***Upravit → Tabulka souborů***.

Tlačítko ***Tabulka***

Tabulka souborů a oblastí má 6 sloupců o 200 řádcích, očíslovaných 1 až 200. Každý sloupec je vyhrazen jednomu druhu objektů, číslo řádku pak odpovídá číslu objektu, pomocí nějž se na daný objekt odkazujete v programu.

3) Úprava obsahu buněk tabulky

Do buňky, která je ve sloupci odpovídajícím druhu objektu a v řádku s číslem rovným číslu objektu, zapisujete základní identifikaci objektu: u oblastí souřadnice, u souborů jejich název spolu s případnými dalšími parametry.

Po otevření tabulky bude zvýrazněna **aktuální buňka**, což je buňka, jejíž hodnotu můžeme upravovat. Zvýraznění aktuální buňky můžeme **přesouvat** kurzorovými klávesami nebo klepnutím myší.

Text v aktuální buňce můžete **upravovat** z klávesnic po stisku **F2** nebo po klepnutí myší (první klepnutí = nastavení aktuální buňky, druhé klepnutí přepnutí do režimu editace).

Druhou možností je využít prostředků, které zadávání usnadní. Klepnete-li na buňku pravým tlačítkem, můžete z místní nabídky zadat

- Ü pro oblasti povel ***Editovat***, čímž otevřete **editor oblastí** a
- Ü pro soubory povel ***Hledat***, čímž otevřete dialogové okno ***Hledat ???***, kde na místě otazníků bude označení druhu hledaného souboru.

Dialogové okno ***Hledat ???*** otevřete i stiskem tlačítka ***Hledat*** na spodním okraji okna.

Souřadnice oblasti můžete také editovat stiskem pravého tlačítka myši a zadáním volby ***Editovat*** v místní nabídce. Na soubory zadané v této tabulce se potom odkazujete v programu jako na čísla řádků tabulky v **příkazech pro multimédia a obrázky** nebo v práci se **soubory**.

4) Vložení prvku do programu

V programu se na soubory a oblasti z tabulky odkazujeme prvkem určujícím druh objektu (a tím i sloupec tabulky), za nímž následuje číslo určující řádek v tabulce.

Nezadáte-li v programu číslo řádku, bude to mít stejný efekt, jako když zadáte číslo **1**.

Prvek označující druh objektu je v tabulce vždy pod příslušným sloupcem. Tento prvek můžete uchopit a vložit do programu. Výhodnější postup je však zadání povelu ***Uchopit*** v místní nabídce buňky, protože tak vložíte do programu nejen prvek označující druh souboru, ale i číslo příslušného řádku tabulky.

O dalších možnostech práce s jednotlivými druhy objektů se dozvíte v sekci **Oblasti** nebo **Použití souborů z tabulky.**

5) Které typy souborů můžete v tabulce použít

Zajímají-li vás pouze možnosti některého konkrétního sloupce, klepněte na příslušný obrázek.






Oblasti **Soubory Aplikace** **Zvuky WAVE** **Zvuky MIDI** **Obrázky** **Animace AVI**

Soubor / Aplikace

Z Baltíkových programů můžete spouštět všechny druhy aplikačních souborů, tj.:

- Ü soubory s příponou **.exe**, v nichž jsou uloženy běžné spustitelné programy, které běží v operačním systému DOS nebo Windows,
- Ü soubory s příponou **.com**, což jsou krátké příkazové soubory operačního systému DOS,
- Ü soubory s příponou **.bat**, což jsou dávkové soubory, které při svém vykonávání zpravidla spouští jiné spustitelné soubory a
- Ü soubory s příponou **.pif**, což jsou pomocné soubory s parametry definujícími jak pod systémem Windows spustit program, který byl původně napsán pro systém DOS.

Pozn.: Budete-li chtít spouštět dávkové soubory pro Windows, které mají příponu **.js** nebo **.vbs**, musíte spustit program **WScript**, kterému pak předáte spouštěný dávkový soubor jako parametr.

Za jménem spustitelného souboru může následovat seznam parametrů, které soubor používá, např.

scandisk.exe a: c: /p

Datové soubory

Do sloupečku ***Soubor/Aplikace*** můžete zadávat i názvy souborů, které nechcete spouštět, ale chcete s nimi pracovat jako s běžnými datovými soubory. **Na přípony těchto souborů nejsou kladena žádná omezení**. Podrobnosti o práci s těmito soubory se dozvíte v sekci **Soubory.**

Zvuky WAVE

Do tohoto sloupce zadáváme názvy souborů s příponou **.wav**. Tyto soubory jsou podobné nahrávce na magnetofonovém pásku a vaše zvuková karta je pouze reprodukuje.

Za názvem souboru mohou být čárkami oddělená dvě čísla: čas začátku přehrávání v milisekundách a čas konce přehrávání v milisekundách. Tak např. záznam:

Hello.waw,120000,180000

zabezpečí přehrání třetí minuty (tj. od 120. do 180. sekundy) skladby v souboru **Hello.waw**.

Zvuky v souborech **.wav** můžete nejenom přehrávat, ale můžete si přes mikrofon nebo zvukový vstup nahrát i své vlastní. Bližší podrobnosti se dozvíte v kapitole **Nahrávání zvuků.**

Zvuky MIDI

Do tohoto sloupce zadáváme názvy souborů s příponou **.mid** nebo **.rmi**. Tyto soubory jsou podobny notovému záznamu a vaše karta používá své vlastní hudební nástroje k jejich přehrání. Kvalita výsledné hudby tedy závisí na kvalitě hudebních nástrojů, které vaše karta emuluje (tj. hraje místo nich).

Parametry specifikující počátek a konec přehrávání jsou stejné jako u zvuků **.wav.**

Na rozdíl od souborů **.wav** si soubory MIDI nemůžete nahrávat pomocí Baltíka. Musíte použít buď hotové soubory **.mid** nebo **.rmi** anebo pro tvorbu těchto souborů použít nějaký jiný, specializovaný program.

Obrázky

Baltík podporuje několik formátů obrázkových souborů:

- Bitové mapy s příponou **.bmp**. Tyto soubory jsou podporovány téměř všemi grafickými editory a můžete je tedy snadno nakreslit v kreslících programech, které jsou součástí operačního systému - např. v programu **Malování**, jenž je součástí lokalizovaných Windows. Pokud Vám však stačí 16 Baltíkových barev, doporučujeme použít grafický editor **Paint,** který je Baltíkovou součástí.
- Ikony s příponou **.ico**. Tyto soubory používají Windows pro zobrazení ikon. Ikony si jsou schopny zapamatovat průhlednou barvu a při zobrazení na displej ji použít.
- Vektorové grafické soubory s příponou **.wmf** (Windows Meta File). Tyto soubory jsou příkazové soubory pro tvorbu grafiky popř. textu.
- Baltíkovy banky předmětů s příponami **.b00 - .b99** nebo **.c00 - .c99**. Tyto soubory si můžete **nakreslit** v Baltíkovi sami, případně můžete použít standardní banky.

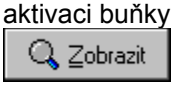
Za názvem souboru můžete zadat čtveřici čísel, která znamenají **souřadnice výřezu**, jenž se má v programu použít. Stejně jako u oblastí se nejdříve zadávají souřadnice levého horního rohu a pak pravého dolního.

Chcete-li zobrazit například levou polovinu obrázku **domek.bmp**, který má rozměry **320×200** bodů, můžete napsat:

domek.bmp, 0, 0, 159, 199

Kontrolní obrázek

Abyste viděli, že v příslušné buňce je uveden název souboru se správným obrázkem, objeví se při aktivaci buňky se zadaným názvem obrázkového souboru malé okno s náhledem obrázku.



Budete-li chtít vidět obrázek v plné velikosti, stačí na náhled klepnout nebo stisknout tlačítko ***Zobrazit***.

Audiovizuální soubory AVI

Do tohoto sloupce zadáváme soubory s příponou **.avi**, které obsahují digitalizovanou audiovizuální nahrávku (video) nebo animaci vytvořenou pomocí některých animačních programů.

6) Kontrolní přehrání multimediálního souboru

Kterýkoliv z multimediálních souborů, tj. soubory WAVE, MIDI a AVI, můžete pro kontrolu přehrát, abyste si ověřili, že opravdu obsahují to, co obsahovat mají.

Multimediální soubory přehrajete stiskem tlačítka ***Přehrát soubor***.

Po spuštění přehrávání ožije tlačítko ***Zastavit přehrávání nebo nahrávání***, jehož stiskem můžete přehrávání dlouhého souboru předčasně zastavit.

7) Nahrávání vlastních zvukových souborů

Soubory **.wav** můžete nejen přehrávat, ale můžete si nahrát i své vlastní. Kdykoliv aktivujete některou buňku z tohoto sloupce, ožijí na ovládacím panelu zvuků vedle tlačítka ***Přehrát soubor*** také tlačítko ***Nahrát soubor WAVE***.

Ovládací panel zvuků

Nahrávání spustíte stiskem tlačítka ***Nahrát soubor WAVE***. Po jeho stisku začne počítač nahrávat.

Nahrávání ukončíte stiskem tlačítka ***Zastavit přehrávání nebo nahrávání***. Po jeho stisku se otevře dialogové okno ***Uložit nový zvuk WAVE***, v němž zadáte název souboru, který se pak zapíše do aktivní buňky.

Pozn.: Dejte pozor na to, abyste před zapnutím nahrávání měli aktivní opravdu tu buňku, do které chcete název souboru uložit. Baltík vám již nedovolí před uložením názvu souboru aktivní buňku změnit.

Oblasti

Oblast je obdélníková část obrazovky, kterou budete moci v programu používat

- Ü v **grafických příkazech** pro zadání polohy a rozměrů obrazce,
- Ü v **podmínkách** k zjištění, zda se v dané oblasti nachází Baltík, předmět, animovaný předmět, ukazatel myši nebo zda tam bylo stisknuto tlačítko myši,
- Ü pro určení výřezu při **práci s obrazovkou,** při **vkládání obrázků** nebo **pouštění videa**.

Oblasti můžeme definovat jak **před spuštěním** programu, tak **za běhu** programu.

1) Definice oblastí

Baltík umožňuje definovat až **200** oblastí které označuje čísly od **1** do **200**. Definice jednotlivých oblastí se uchovávají v **tabulce oblastí a souborů** ve sloupci ***Oblasti*** a řádku s číslem příslušné oblasti.

Oblast je definována čtveřicí čísel, které postupně určují **bodové souřadnice jejího levého horního a pravého dolního rohu**. Tato čísla se v definici oddělují čárkami.

Např. čtveřice **10,20,30,40** definuje oblast, jejíž levý horní roh má souřadnice **x10y20** a pravý dolní roh má souřadnice **x30y40**.

Baltíkova scéna je **585** bodů široká a **290** bodů vysoká, takže oblast pokrývající **celou scénu** bude definována čtveřicí **0,0,584,289**.

Nejmenší možná oblast zabírá jediný bod. Definice takovéto oblasti v pravém horním rohu scény by měla tvar **289,0,289,0**.

2) Definice oblastí před spuštěním programu

Definici oblasti musíme zapsat do tabulky oblastí a souborů. Tu otevřeme klepnutím na tlačítko ***Tabulka*** nad pracovní plochou.

Tlačítko ***Tabulka***

Definice oblastí můžeme do tabulky zadat buď

- Ü **ručně** - pak klepneme do příslušné kolonky a zadáme příslušné údaje z klávesnice,
- Ü nebo můžeme k zadání oblasti využít **editor oblastí.** který otevřeme klepnutím pravým tlačítkem na příslušnou buňku a zadáním povelu ***Editovat*** z místní nabídky.

3) Definice oblasti za běhu programu

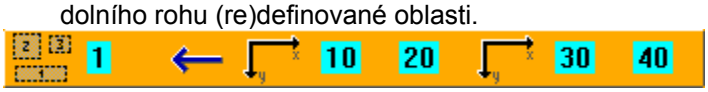
(Re)definice rozměrů

Oblasti můžete pomocí prvku ***Přiřaď*** zřizovat (definovat) a měnit (redefinovat) za běhu programu. Postup sestavení (re)definičního příkazu je následující:

1. Klepnutím na tlačítko ***Tabulka*** nad pracovní plochou otevřete okno ***Soubory***.

Tlačítko ***Tabulka***

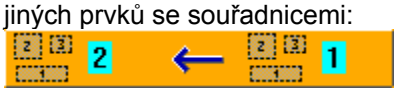
2. Ve sloupci ***Oblasti*** klepnete pravým tlačítkem na buňku příslušnou oblasti, kterou chcete (re)definovat. Tím otevřete místní nabídku, v níž zadáte ***Uchopit***. Po zadání příkazu se okno zavře.
3. Najedete kurzorem na příslušné místo programu a klepnete. Do programu se vloží prvek ***Oblast*** následovaný literálem s číslem (re)definované oblasti.
4. Za vložené prvky vložíte požadované souřadnice levého horního rohu a poté souřadnice pravého dolního rohu (re)definované oblasti.



Oblasti číslo 1 přiřaď souřadnice x10y20 a x30y40.

Pozn.: Nepoužijeme-li v definici číslo oblasti, bude výsledek stejný, jako kdybychom použili číslo **1**.

Oblasti můžeme přiřazovat souřadnice nejenom zadáním příslušných čísel, ale i předáním souřadnic jiných prvků se souřadnicemi:

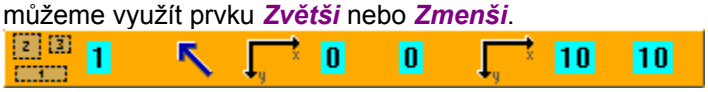


Oblasti číslo 2 přiřaď souřadnice oblasti číslo 1.

Levému hornímu rohu oblasti číslo 3 přiřaď bodové souřadnice Baltika, pravému dolnímu rohu přiřaď souřadnice bodu, na který ukazoval kurzor myši při stisku levého tlačítka.

Zvětšení či zmenšení oblasti

Nepotřebujeme-li definovat zcela nové souřadnice ale chceme pouze stávající souřadnice lehce upravit, můžeme využít prvku ***Zvětši*** nebo ***Zmenši***.

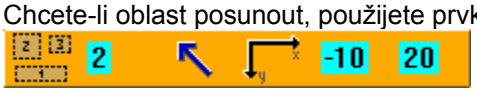


Pravý dolní roh oblasti číslo 1 posuň o 10 bodů na jihovýchod. Jinými slovy: Zvětši oblast o 10 bodů doprava a o 10 bodů dolů.

Můžete také některou ze souřadnic pravého dolního rohu vynechat. Tuto možnost použijete, pokud potřebujete na **obrazovku** zvětšit nebo zmenšit na určitou šířku (resp. výšku) tak, aby se nezadaná souřadnice přepočítala automaticky podle poměru.

Posunutí oblasti

Chcete-li oblast posunout, použijete prvky ***Zvětši*** nebo ***Zmenši*** jen s jednou sadou souřadnic:

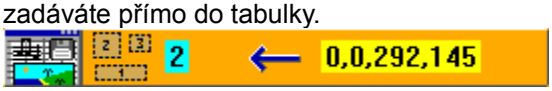


Posuň oblast číslo 2 o 10 bodů doleva a o 20 bodů dolů.

Trvalá (re)definice oblasti

Všechna předchozí přiřazení hodnot ovlivnila velikost oblasti pouze po dobu běhu programu. Chcete-li zadat **souřadnice oblasti pro všechna další spuštění programu**, předřaďte předchozím přiřazovacím příkazům prvek ***Tabulka***. Nové souřadnice však v tomto případě musíte zadat jako text - stejně, jako je

zadáváte přímo do tabulky.



Oblasti č. 2 přiřaď nastálo souřadnice x0y0 x292y145

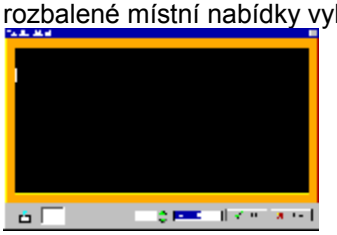
Pozn.: Tímto přiřazením se **změní zdrojový kód programu**. Aby se nastavení oblasti zachovalo pro příští spuštění programu, musíte program **uložit na disk**.

Chcete-li trvale přiřadit jedné oblasti současné tabulkové souřadnice jiné oblasti, použijte prvek tabulka na obou stranách přiřazení

Oblasti č. 3 přiřaď nastálo současné tabulkové souřadnice oblasti č. 2

Editor oblastí

Editor oblastí slouží k jednoduššímu zadání souřadnic **oblasti**. Editor oblasti můžete spustit tak, že v **tabulce** stisknete pravým tlačítkem myši na oblasti, kterou chcete zadat nebo změnit a z následně rozbalené místní nabídky vyberete možnost ***Editovat***. Otevře se okno editoru oblastí:

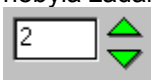


Okno editoru oblastí zvětšíte tak, že na něj klepnete myší

V tomto okně pak můžete pomocí levého tlačítka myši vyznačit oblast, případně ji potom měnit tažením za okraje či rohy nebo přesouvat tažením za vnitřek.

Všechny zadané oblasti jsou v okně vyznačeny šedým rámečkem. **Aktivní oblast**, tj. kterou právě nastavujete, je vyznačena bílým rámečkem.

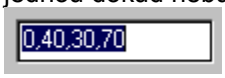
Tlačítkem ***Načti obrázek*** můžete orientačně načíst obrázek do aktivní oblasti. Pokud aktivní oblast nebyla zadána, obrázek se načte na obrazovku v nezměněné velikosti.



Číslo aktivní oblasti můžete měnit přímo přepsáním čísla v políčku nebo zvětšovat či zmenšovat příslušnými tlačítky.

Aktivní oblast můžete zadat i klepnutím pravým tlačítkem myši na danou oblast na pracovní ploše.

Klepnete-li v místě, kde se několik oblastí překrývá a nevybere se ta, kterou máte na mysli, klepněte ještě jednou dokud nebude vybrána požadovaná oblast.



Vpravo vedle čísla aktivní oblasti jsou zobrazeny **souřadnice aktivní oblasti**, které se při jakékoliv změně oblasti průběžně mění.

Souřadnice aktivní oblasti můžete změnit i klepnutím do pole a zadáním přímo z klávesnice.

Koš použijete v případě, že potřebujete oblast smazat, tzn. udělat z ní oblast nezadanou. Můžete stisknout tlačítko ***Koš*** nebo do něj odstraňovanou oblast přesunout.

Přehrávání CD a použití souborů z Tabulky

Tabulka oblastí a souborů

Následující výklad nebudeme tříštit podle jednotlivých druhů souborů, ale uspořádáme jej podle činností.

V této sekci si ukážeme, jak:

- Ü **spustit aplikaci,**
  přehrát multimediální soubor,
  zobrazit obrázek či
  pustit skladbu z CD,
- Ü **ovlivnit dobu přehrávání souboru či skladby,**
- Ü **umístit obrázek či animaci do zadané oblasti,**
- Ü **jak v programu zjistit a upravit text položky v tabulce**

Další informace o možnostech jednotlivých **typů souborů** najedete v sekci **Tabulka souborů a oblastí.**

1) Doporučený postup pro použití multimédií

1. Připravte si soubor s příponou **.wav**, **.mid**, **.rmi** nebo **.avi** se zvukem (hudbou) nebo videoanimací.
2. Do **tabulky souborů** vložte název připraveného souboru.
3. **Do programu vložte** jeden z multimediálních prvků s číslem, pod kterým máte soubor vložen v **tabulce souborů.**

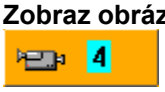
Příklady:

Spusť aplikaci č. 4

Přehraj zvuk č. 4

Přehraj hudbu MIDI č. 4

Zobraz obrázek č. 4



Přehraj videoanimaci AVI č. 4

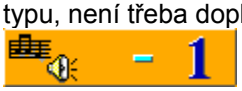
Přehraj čtvrtou skladbu z CD

Předchozí příkazy vykonají požadovanou akci (spusť, přehraj, zobraz) s obsahem souboru, jehož název je v tabulce souborů ve čtvrtém řádku příslušného sloupce.

Výjimkou je poslední příkaz k přehrání skladby z CD, který se na Tabulku neodvolává. Prvek ***Přehraj skladbu z CD*** je na příkazovém panelu mezi příkazy pro animaci a maskovacími komentáři. Jeho použití je však téměř totožné s použitím ostatních příkazů pro přehrávání zvuků, takže je vysvětlujeme pohromadě.

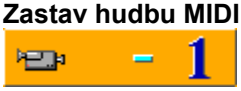
2) Doba přehrávání

Zvuk, hudbu nebo videoanimaci **ukončíte** buď spuštěním jiného souboru nebo zadáním čísla **-1** za příslušný příkaz. Vzhledem k tomu, že v daném okamžiku se může přehrávat pouze jeden soubor daného typu, není třeba doplňovat číslo zastavovaného souboru.



Zastav zvuk WAVE

Zastav hudbu MIDI



Zastav animaci AVI

Zastav skladbu z CD

Často je potřeba počkat na dokončení přehrávky - ať už proto, že po ní chcete spustit skladbu další, nebo proto, že s koncem programu zároveň končí přehrávání všech skladeb a vy chcete nechat přehrávku před koncem programu dojet do konce.

Má-li program **počkat na dokončení přehrávky**, zadejte za prvek charakterizující typ přehrávaného souboru číslo **-2**.

Počkej, až dohraje zvuk WAVE

Počkej, až dohraje hudba MIDI

Počkej, až dohraje animace AVI

Počkej, až dohraje skladba z CD

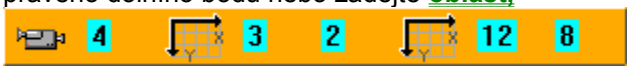
3) Umístění obrázku či animace

Chcete-li zobrazit obrázek nebo přehrát videoanimaci na zvolených **souřadnicích**, zadejte tyto souřadnice za příkaz pro uložení /přehrání:

Přehraj videoanimaci č. 4 na souřadnicích X3 Y2

Zobraz obrázek č. 4 na souřadnicích X3 Y2

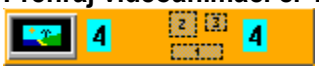
Chcete-li vložit obrázek či přehrát animaci v předem zadané oblasti, zadejte souřadnice levého horního a pravého dolního bodu nebo zadejte **oblast,**



Přehraj videoanimaci č. 4 v oblasti určené souřadnicemi X3 Y2 a X12 Y8.

Zobraz obrázek č. 4 v oblasti určené souřadnicemi X3 Y2 a X12 Y8.

Přehraj videoanimaci č. 4 v oblasti č. 4.



Zobraz obrázek č. 4 v oblasti č. 4.

Pozn.: Obrázek i videoanimaci můžete se stejným výsledkem zobrazit pomocí **přiřazení na obrazovku**.

Přehraj videoanimaci č. 4

Zobraz obrázek č. 4

4) Jak v programu zjistit a upravit text položky v Tabulce

Chcete-li při provádění programu použít nebo měnit názvy souborů v Tabulce, použijte prvek ***Tabulka souborů***, který se nachází v levém dolním rohu tabulky.

Prvek ***Tabulka souborů***

Za tímto prvkem následují v programu dva **parametry:**

Ü pořadové číslo nebo označení sloupce a

Ü číslo řádku.

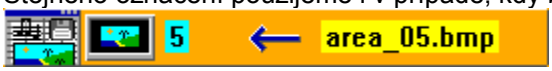
Pozn.: Parametr sloupec může mít hodnoty
1 - oblasti,
2 - obecné soubory,
3 - zvuky WAVE,
4 - hudba MIDI,
5 - obrázky nebo
6 - videoanimace AVI.
nebo příslušný prvek ***(Oblast, Soubor, Zvuk WAVE, Hudba MIDI, Obrázek, AVI)***
Parametr řádek může mít hodnotu 1 až 200.

Vypiš na obrazovku souřadnice oblasti (sloupec 1) č. 1

Číselný zápis sloupce naznačený v předchozím příkladu je výhodný pouze v případech, kdy se číslo sloupce v programu počítá. V ostatních případech je výhodnější použít prvek označující příslušný sloupec.

Vypiš na obrazovku název souboru s obrázkem č. 4.

Stejného označení použijeme i v případě, kdy budeme chtít nějaký údaj v tabulce změnit:



Zadej do Tabulky souborů pro 5. obrázek název souboru "area_05.bmp".

Baltíkův pomocník (procedura)

Tabulka pomocníků

Baltíkův pomocník je chlapík, který Baltíkovi pomáhá dělat různé věci.

Práce s pomocníky sestává ze dvou částí:

- Ü **definice** pomocníka, při níž určíte, co má pomocník udělat až jej z programu zavoláte, a
- Ü **volání** pomocníka v programu v okamžiku, kdy potřebujeme, aby vykonal předem naprogramovanou činnost.

V této sekci si povíme

kde se pomocníci používají,
jak se pomocníci definují,
jak se pomocníci z programu zavolají,
jak mohou pomocníci používat proměnné,
jak pomocníkovi předáme parametry,
jak pomocníkovi předáme proměnnou, v níž má něco vrátit,
jak činnost pomocníka předčasně ukončíme.

1) Použití pomocníků

Pomocníky použijeme s velkou výhodou ve chvílích, kdy potřebujeme **vykonat na několika místech programu stejnou činnost**. Není moudré naprogramovat tuto činnost tolikrát, kolikrát ji potřebujeme vykonat. Mnohem výhodnější je definovat pomocníka, tuto činnost jej "naučit" a v programu jej pak na příslušných místech zavolat, aby pro nás onu činnost vykonal.

Druhým důvodem, který nás vede k používání pomocníků, je **zpřehlednění programu**. V dlouhých programech bývají často chyby, které se špatně hledají. Daleko výhodnější je rozdělit celou úlohu na sadu úloh jednodušších a pro každou z nich definovat jejího vlastního pomocníka. Místo jednoho dlouhého programu tak získáme několik programů kratších, v nichž se lépe vyznáme a proto v nich také uděláme méně chyb. Navíc chyby, které v takovýchto přehledných programech náhodou uděláme, se mnohem lépe nacházejí a opravují.

V klasických programovacích jazycích se samostatně definovaná část programu, kterou počítač vykoná na požádání (tj. ekvivalent našeho pomocníka), nazývá **procedura** nebo **podprogram**.

Pozn.: Prvky ***Baltíkův pomocník*** a ***Zavolej Baltíkova pomocníka*** najdete v **tabulce pomocníků**:

Tabulka pomocníků

2) Definice Baltíkova pomocníka

Část programu, v níž definujeme, co má pomocník dělat, nazýváme **definice pomocníka**. Definice pomocníka musí **začínat prvkem** ***Nový Baltíkův pomocník***, za nímž následuje **označení pomocníka**.

Prvek ***Nový Baltíkův pomocník***
vložíte do programu z **Tabulky pomocníků.**

Baltík může mít mnoho pomocníků - jejich počet je omezen pouze velikostí paměti počítače. Aby se v

nich Baltík (i programátor) vyznal, musí mít každý pomocník svoje **označení**.

Pomocníka označíte tak, že za prvek ***Nový Baltíkův pomocník*** vložíte:

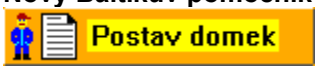
Ü **číslo,**

Ü **jméno** (tj. textový **řetězec**) nebo

Ü **předmět**

Nový Baltíkův pomocník ***1***

Nový Baltíkův pomocník ***předmět 4***



Nový Baltíkův pomocník ***Postav domek***

Z předchozích ukázek asi sami odhadnete, že nejvýhodnější je označovat pomocníky pomocí jmen a nebo alespoň pomocí předmětů, jejichž obrázek se nějak týká vykonávané činnosti.

Pozn.: Nezadání označení pomocníka má stejný efekt, jako zadání čísla 0. Doporučujeme ale této možnosti **nevyužívat**, protože v opačném případě při opravách nepoznáte, kde jste označení zapomněli záměrně a kde omylem.

Prvek ***Nový Baltíkův pomocník*** následovaný označením a případným komentářem bývá označován jako **hlavička pomocníka**.

Za hlavičku pomocníka vkládáte příkazy, které má váš nový pomocník provádět. Tyto příkazy tvoří **tělo pomocníka**.

Příkazy můžete vložit hned za jméno pomocníka nebo na libovolný další řádek. Nejdůležitějším kritériem je přitom maximální přehlednost programu

Tělo pomocníka (a tím i tělo celé definice) ukončí buď konec celého programu nebo počátek definice dalšího pomocníka.

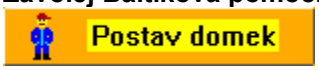
3) Volání Baltíkova pomocníka

Baltíkova pomocníka aktivujete tak, že na místo programu, kde budete chtít, aby pomocník vykonal to, co jste jej naučili, vložíte prvek ***Zavolej Baltíkova pomocníka*** následovaný **označením** volaného pomocníka. Po ukončení činnosti volaného pomocníka program pokračuje dalšími příkazy následujícími za příkazem volání.

Prvek ***Zavolej Baltíkova pomocníka*** vložíte do programu z **Tabulky pomocníků.**

Zavolej Baltíkova pomocníka ***1***

Zavolej Baltíkova pomocníka ***předmět 4***



Zavolej Baltíkova pomocníka *Postav domek*

Pomocník si může na pomoc **zavolat dalšího pomocníka** nebo dokonce může zavolat **sám sebe** - v tom případě mluvíme o tzv. **rekurzivním volání**.

4) Používání lokálních proměnných (košíků)

Baltíkův pomocník může kromě **globálních proměnných** (šuplíků), které jsou viditelné ze všech míst programu, používat také lokální proměnné (košíky), které vidí (a může používat) pouze ten **pomocník**, v jehož definici se daná proměnná vykytuje. Ostatní pomocníci ani samotný hlavní program k ní nemají přístup a mohou jim pouze **přiřadit počáteční hodnotu.**

Lokální proměnné (košíky)

Lokální proměnné jsou v Baltíkovi zobrazeny jako košíky. Jejich nedostupnost si můžete zdůvodnit tak, že si každý pomocník nosí své košíky pořád s sebou a nikomu nedovolí, aby se mu do nich díval nebo se v nich dokonce přehraboval.

Pozn.: Každý pomocník si na začátku své práce definuje (vytváří) své vlastní košíky. Používají-li dva pomocníci stejně nazvaný košík (lokální proměnnou), jedná se dva různé košíky, které se pouze shodou okolností stejně jmenují. Protože pomocníci o svých proměnných nevědí, nemůže tato shoda jmen vést k jakékoliv záměně.

Pomocník však programu, který jej žádá o pomoc, umožní, aby mu předal některé hodnoty či celé proměnné s doporučením, do kterého košíku si má každou z nich uložit.

Při **rekurzivním volání** (tj. volá-li pomocník sám sebe), se tato volání vůči sobě chovají stejně, jako by se chovala v případě, kdy volající pomocník volal zcela jiného pomocníka. V případě lokálních proměnných to tedy znamená, že

- **volající pomocník** může lokálním proměnným (košíkům) volaného pomocníka pouze **přiřadit počáteční hodnotu** a jinak jsou pro něj nedostupné,
- **volaný pomocník** si pořídí vlastní košíky (lokální proměnné) a košíky pomocníka, který jej zavolal, jsou pro něj neviditelné a tím i nedostupné, a to přesto, že volající pomocník je on sám.

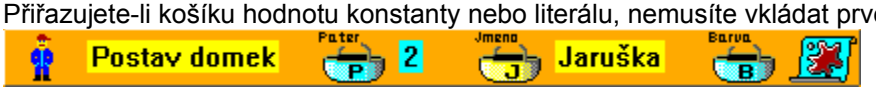
5) Předání parametrů

Baltíkův pomocník umožní volajícímu programu, aby při volání naplnil jeho košíky a přiřadil jim tak počáteční hodnotou (aby inicioval lokální **proměnné**) - odborně bychom řekli aby mu **předal parametry**.

Pozn.: Této možnosti však samozřejmě nemusíte využít nebo můžete přiřadit počáteční hodnotu pouze některým košíkům.

Zavolej pomocníka *předmět 4*, jeho košík *Klávesa* naplň klávesou *A* a jeho košík *Počitadlo* naplň hodnotou z šuplíku *Délka*.

Přiřazujete-li košíku hodnotu konstanty nebo literálu, nemusíte vkládat prvek ***Přiřaď***.



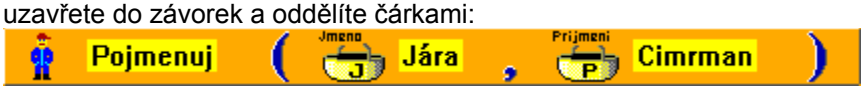
Zavolej pomocníka *Postav domek*. Jeho košík *Pater* naplň hodnotou *2*, košík *Jmeno* naplň hodnotou *Jaruška* a košík *Barva* naplň hodnotou *Kaštanová*.

POZOR!

Máte-li pomocníka označeného jménem, musíte při předávání řetězcových parametrů dát pozorna to, aby je Baltík nesloučit se jménem do jednoho řetězce jako v uvedeném příkladu (shválně jsme jej přeškrtali, abychom zdůraznili, že takto se program psát nemá).

Stejnému riziku se vystavujete v případě, kdy budete přiřazovat počáteční hodnoty dvěma řatězcovým košíkům za sebou.

Problémy s přiřazením hodnoty do řetězcové proměnné musíte obejít tak, že pomocníkovy parametry uzavřete do závorek a oddělíte čárkami:



6) Rozdíly mezi předáním parametrů hodnotou a odkazem

Máte-li počáteční hodnotu, kterou chcete naplnit některý z pomocníkových košíků, uloženu v proměnné, mohou nastat dvě situace:

- Ü chcete do košíku předat **pouze hodnotu** uloženou v proměnné a nechcete, aby volaný pomocník hodnotu v této proměnné jakkoliv měnil - pak hovoříme od **předání parametru hodnotou**, nebo
- Ü chcete pomocníkovi předat **celou proměnnou** s tím, že v průběhu své činnosti má do předané proměnné vložit aktualizovanou hodnotu - pak hovoříme o **předání parametru odkazem**. Pomocník pak bude místo své lokální proměnné používat předanou proměnnou (v lokální proměnné bude pouze odkaz na předanou proměnnou, kterou má použít místo ní).

Chceme-li předat parametr **hodnotou**, musíme přiřadit košíku tuto hodnotu (zde předáváme pouze peníze uložené v šuplíku),

chceme-li předat parametr **odkazem**, vynecháme znak přiřazení a tím předáme do košíku celou proměnnou (zde předáváme celý šuplík).

Pro názornost uvedeme **příklad** ze života: představte si, že rodiče posílají dítě (pomocníka) na nákup. Aby mohl pomocník vykonat požadovanou činnost (aby mohlo dítě nakoupit), potřebuje peníze - ty budou parametrem, jehož předání podmiňuje úspěšné splnění úkolu.

- Ü Dají-li rodiče dítěti peníze, za něž nakoupí, jedná se vlastně o předání parametru hodnotou.
- Ü Dají-li mu peněženku (proměnnou) s penězi (hodnota uložená v proměnné), předávají mu parametr odkazem (až budeš potřebovat peníze, podívej se do peněženky). Po ukončení požadované činnosti (nákupu) vrátí pomocník (dítě) svěřenou proměnnou (peněženku) s novou hodnotu.

Pokud pomocníka volá jiný pomocník, může přiřadit jeho lokální proměnné některou ze svých lokálních proměnných.

Příklad:

V následujícím příkladu je uveden program pomocníka, který počítá faktoriál (faktoriál musíte umět spočítat např. tehdy, když potřebujete zjistit pravděpodobnost své výhry ve sportce a podobných hrách).

Faktoriál se značí vykřičníkem a je definován následovně:

n! = 1× 2 × 3 × ... × n

Z uvedeného si jistě umíte odvodit, že **n!=n×(n-1)!** - tedy např. 5!=5×4!=120. Toho využívá i náš program, který pracuje v následujících krocích:

1. Pomocník nejprve zjistí, zda číslo, jehož faktoriál má zjišťovat (tj. obsah košíku ***N***), je větší než **1**. Pokud není, nemusí nic dělat (pro zjednodušení nepředpokládáme, že by se někdo ptal na faktoriál čísla menšího než **1**).
2. Do košíku ***Hmotnost*** uloží hodnotu o jedničku menší, než je hodnota čísla, jehož faktoriál počítá.
3. Požádá pomocníka (sebe sama), aby mu spočítal faktoriál tohoto čísla, tj. faktoriál čísla **N-1**. Parametr (číslo, jehož faktoriál má pomocník spočítat) předá odkazem, takže po skončení práce pomocníka najde v proměnné ***Hmotnost*** hodnotu zjišťovaného faktoriálu.
4. Vynásobí právě získanou hodnotu faktoriálu čísla o jedničku menšího (má ji v proměnné ***Hmotnost***) hodnotou čísla, jehož faktoriál zjišťuje (má ji v proměnné ***N***), neboli vynásobí **N×(N-1)!**. Výsledek (hledaný **N!**) uloží do proměnné ***N***.

Abyste si předchozího pomocníka vyzkoušeli, zkuste následující prográmek:

Pomocníkovi nemůžeme předávat přímo proměnnou ***Počitadlo***, protože je parametrem cyklu. Pomocník by její hodnotu změnil a nám by nefungoval cyklus. Proto jsme zavedli pomocnou proměnnou ***Hmotnost***.

5) Předčasné ukončení činnosti Baltíkova pomocníka

Prvek ***Opusť program nebo pomocníka***
ukončí provádění pomocníka.

Je-li příkaz **použit mimo pomocníka**, bude **ukončen celý program**.

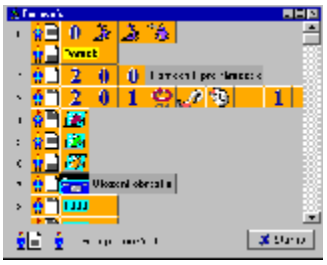
Chcete-li ukončit činnost více vnořených volání pomocníků, použijte prvek ***Opusť program nebo pomocníka*** s **číslem**:

Ukonči 2 úrovně
Použijete-li tento příkaz např. v pomocníkovi, kterého volá jiný pomocník, budou ukončeni oba; použijete-li jej v pomocníkovi, kterého volá přímo hlavní program, bude ukončen celý program.

Ukonči všechny úrovně
Použijete-li jako počet ukončovaných úrovní prvek ***Nekonečno***, ukončí se všechny úrovně, tedy i celý program.

Tabulka pomocníků

Baltíkův pomocník



Tabulka pomocníků umožňuje jednoduše **definovat nové** pomocníky, **vkládat volání** pomocníků do programu, **zobrazovat** definice jednotlivých pomocníků a **vracet se do definic** na místa, kde jsme v nich byli naposledy.

Ukázku tabulky pomocníků zvětšíte tak, že na ni klepnete myší

Tabulka pomocníků
Klepnutím na toto tlačítko tabulku otevřete.

Pomocníci jsou v tabulce seřazeni podle své pozice v programu.

Jak vložit definici nového pomocníka

Prvek ***Nový Baltíkův pomocník***
Klepnutím na toto tlačítko vložíte na konec programu novou **definici pomocníka.** Tu pak **označíte** a naprogramujete.

Jak do programu vložit volání pomocníka

Klepnutím na prvek definice pomocníka, změní se tato definice ve volání a vy uchopíte **volání** vybraného pomocníka, které pak můžete vložit do programu.

Druhou, méně výhodnou možností, je uchopit klepnutím prvek ***Volání pomocníka*** u spodního okraje okna, vložit jej na příslušné místo v programu a dopsat nebo vložit označení příslušného pomocníka.

Jak zobrazit definici pomocníka

Klepnete-li v tabulce na zobrazený první řádek definice pomocníka, zobrazí se definice příslušného pomocníka. Nevejde-li se definice na obrazovku, zobrazí se alespoň její počátek.

Jak se vrátit do definice na místo, kde jsem byl naposledy

Klepnete-li v tabulce **pravým** tlačítkem myši na zobrazený první řádek definice pomocníka, přejdete dovnitř definice vybraného pomocníka na místo, kde jste jej naposled opustili.

Pokud jste s tímto pomocníkem ještě nepracovali, přejdete na začátek jeho pomocníka stejně jako při klepnutí levým tlačítkem.

Pozn.: Po uzavření tabulky a ukončení práce s vybraným pomocníkem se můžete vrátit zpět do programu stiskem pravého tlačítka myši na tlačítku ***Tabulka pomocníků***.

Podmínky

Tato sekce obsahuje následující témata:

1) Podmínky a jejich použití

Podmínky se používají především při rozhodování o dalším postupu programu v **cyklu s počáteční podmínkou (while, do - while)** a v příkazu pro **rozhodování (if).** Tyto příkazy mají následující syntaxi:

while <Podmínka> <Příkaz>
<Příkaz> **do - while** <Podmínka>
if <Podmínka> <Příkaz>
if <Podmínka> <Příkaz> **else** <Příkaz>

Jako <Podmínka> zde může vystupovat

- Ü některá z **podmínek** uvedených dále nebo
- Ü **číselná hodnota** - v tom případě je nulová hodnota ekvivalentní logické hodnotě **NE** (Nesplněno) a nenulová hodnota logické hodnotě **ANO** (Splněno).

Ekvivalence podmínek a číselných výrazů platí i obráceně: **libovolnou podmínku můžeme (po případném uzávorkování) použít jako číslo.** Přitom

- Ü **splněná podmínka** v rací hodnotu **1**,
- Ü **nesplněná podmínka** vrací hodnotu **0**.

Hodnoty podmínek pak můžeme pro další použití uchovávat i v proměnných.

Proměnné *A* přiřaď Viditelný
Je-li Baltík právě viditelný, uloží se do **proměnné *A*** číslo **1**, není-li viditelný, uloží se do ***A*** číslo **0**.

2) Seznam použitelných podmínek

Baltík a jeho stav

Neviditelný

Podmínka je splněna, je-li Baltík neviditelný.

Viditelný
Podmínka je splněna, je-li Baltík viditelný.

Bez obláčku
Podmínka je splněna, je-li zapnuto čarování bez obláčku.

S obláčkem
Podmínka je splněna, je-li zapnuto čarování s obláčkem.

Rychlost
Podmínka se používá s číslem. Podmínka je splněna, pokud má Baltík rychlost rovnu tomuto číslu.

Rychlost 4
Podmínka je splněna, pokud má Baltík rychlost 4.

Východ
Podmínka je splněna, je-li Baltík otočen na východ.

Jih
Podmínka je splněna, je-li Baltík otočen na jih.

Západ
Podmínka je splněna, je-li Baltík otočen na západ.

Sever
Podmínka je splněna, je-li Baltík otočen na sever.

Předměty

Nějaký předmět
Podmínka je splněna, je-li před Baltíkem, případně na zadaných **souřadnicích** nějaký předmět.

Předmět

Použijete-li jako podmínku některý předmět z **banky předmětů**, podmínka bude splněna, pokud bude před Baltíkem právě tento předmět.

Zadáte-li za předmětem **souřadnice**, je podmínka splněna tehdy, když na daných souřadnicích je tento předmět.

Klávesy

Klávesa
Tento druh podmínek se do programu vkládá pomocí tlačítka ***Banky dat***,

Banky dat

kde si z **banky kláves** vyberete některou z kláves. Podmínka je splněna tehdy, když je tato klávesa stejná s klávesou, kterou Baltík přečetl příkazem ***Čti klávesu nebo tlačítko myši*** (viz **Jednoduché příkazy** a ***Čti klávesu***).

Pozn.: Pokud jako podmínku použijete klávesu ***Shift***, ***Ctrl*** nebo ***Alt***, je tato podmínka splněna, je-li daná klávesa právě držena (v poloze "stisknuto").
Použijete-li jako podmínku jednu z kláves ***CapsLock***, ***ScrollLock*** nebo ***NumLock***, je podmínka splněna, je-li daný přepínač zapnut, tzn. je-li indikátor daného přepínače na klávesnici rozsvícen.

Nějaká klávesa
Prázdná klávesa znamená, že podmínka je splněna, byla-li stisknuta libovolná klávesa nebo některé tlačítko myši.

Za prvkem ***Nějaká klávesa*** může následovat **číslo** v závorkách. Pak je podmínka splněna právě tehdy, byla-li stisknuta klávesa s tímto číslem. Číslo klávesy můžete zjistit v bance kláves nebo v programu použitím **Převodu na číslo**.

Ukázku banky kláves zvětšíte tak, že na ni klepnete myší

Byla stisknuta klávesa č. 6 (klávesa F)

Klávesa je držena
Kromě čtení stisknuté klávesy můžete také zjišťovat, jestli je klávesa nebo tlačítko myši drženo (tzn. že bylo stisknuto a ještě nebylo puštěno). Za tuto podmínku vložte klávesu, jejíž stav vás zajímá. Podmínka je splněna, je-li zadané tlačítko drženo. Jako klávesu můžete použít také některé z tlačítek **myši**.

Je držena klávesa *A*
Podmínka je splněna, držíte-li v daném okamžiku klávesu ***A*** stisknutou. Jako klávesu v této podmínce můžete použít i prvek ***Nějaká klávesa*** nebo tlačítka myši.

Pozn.: Pokud používáte tuto podmínku, je vhodné vždy **vyprázdnit frontu kláves**, aby se v této frontě stisknuté klávesy nehromadily.

Znaky

Nějaký znak
Tato podmínka je splněna, pokud byl přečten nějaký **znak (klávesa s nějakým znakem)**.

Pozn.: Stiskem ***šipek***, ***CapsLock***, ***Esc*** a dalších řídících kláves se nepřečte žádný znak.

Znak č. 65
Tato podmínka je splněna, pokud byl přečten znak s **kódem** **65**, tj. velké ***A***.

Oblasti a animované předměty

Oblast 

Leží v oblasti nebo animovaném předmětu (kontrola překrývání)

Podmínka kontrola překrývání je splněna, pokud se Baltík, **animovaný předmět**, bod nebo políčko dotýkají nebo překrývají s **oblastí** nebo jiným animovaným předmětem, případně v oblasti či na animovaném předmětu bylo stisknuto tlačítko **myši** nebo je tam právě ukazatel **myši**.

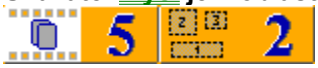
Pozn.: Prvek ***Oblast*** (...leží v oblasti...) najdete v **tabulce oblastí**.

Následující ukázky demonstrují nejčastější druhy podmínek:

Baltík je v oblasti 2.

Levé tlačítko myši bylo stisknuto v oblasti 2.

Ukazatel myši je v oblasti 2.



Animovaný předmět č. 5 leží v oblasti 2.

Bod [56;110] leží v oblasti 2.

Políčko [X Baltíka, Y myši] leží v oblasti 2.
Místo oblastí můžete používat také animovaný předmět, např.:

Bod [292;145] leží v animovaném předmětu č. 6

nebo:

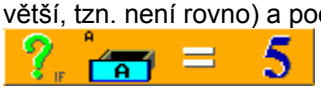
Animovaný předmět č. 1 je v kolizi s animovaným předmětem č. 6,
neboli neprůhledné body animovaného předmětu č. **1** se překrývají s neprůhlednými body animovaného předmětu č. **6**

Pozn.: Pokud číslo chybí, bere se oblast nebo animovaný předmět číslo **0**.

Čísla a řetězce

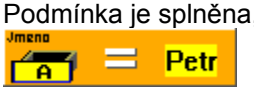
Operátory porovnání

Pomocí čísel , **řetězců**, případně **souřadnic**, a operátorů < "je menší, než", = "je rovno" a > "je větší, než" můžete vytvořit podmínku, která bude splněna, bude-li porovnání pravdivé. Například podmínka "2 < 5" bude vždy splněna, podmínka "X Baltíka = 14" bude splněna, když bude Baltík stát u pravého okraje scény. Můžete také vytvořit složené operátory, např. "<=" (je menší nebo rovno), "<>" (je menší nebo větší, tzn. není rovno) a podobně.



A je rovno 5
Podmínka je splněna, je-li v proměnné ***A*** hodnota **5**.

A je menší, než 5
Podmínka je splněna, je-li v proměnné ***A*** číslo menší, než **5**, např. **4**, **0**, **-20** apod.



Jmeno **je rovno Petr**
Tato podmínka je splněna, je-li v proměnné ***Jmeno*** uložen řetězec "Petr".

Pozn.: Při používání řetězců jsou relační operátory používány vzhledem k tabulce **ASCII**, takže např.
"Adam" < "Běta", ovšem "adam" > "Běta", protože malé a je v tabulce až za velkým B, podobně
"Čeněk" > "David", protože Č je až za D a
"Hana" > "Chrudoš", protože Ch je bráno jako dva znaky C a h, a C je v tabulce před H.

Souřadnice

xy **je rovno [15;13]**
Podmínka je splněna, jsou-li v proměnných ***xy*** uloženy souřadnice bodu **[15;13]**, tedy je-li ***x*****=15** a ***y*****=13**.

Pozn.: Při porovnávání souřadnic nemůžete používat relační operátory **<** a **>**.

Pozn.: Jsou-li alespoň jedny z porovnávaných souřadnic **souřadnicemi políčka**, pak se porovnávají souřadnice políčka

***xy* je rovno [1;1]**
Podmínka je splněna, jsou-li v proměnných ***xy*** souřadnice libovolného bodu políčka **[1;1]**, tedy je-li ***x*** mezi **39** a **77** a ***y*** mezi **29** a **57**, např. **[50;50]**.

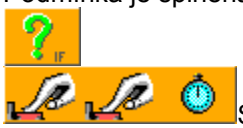
Číslo

Jak jsme již řekli v úvodu sekce, podmínka, kterou tvoří samotné **číslo**, je splněna, je-li toto číslo různé od nuly. Např. podmínka
"a % 2" je splněna, pokud číslo z proměnné a dává po dělení dvěma zbytek různý od nuly (tzn. je-li v proměnné a libovolné liché číslo).

Stopky


Stopky běží

Podmínka je splněna, pokud stopky právě běží.


Stopky zastaveny
Podmínka je splněna, jsou-li stopky zastaveny.

3) Použití logických operátorů

Podmínky můžete spojovat a za pomoci **závorek** a **logických operátorů** **vytvářet podmínky složitější**:

Operátor NOT (neplatí)
Operátor ***NOT*** obrací pravdivost podmínky, na níž je použit. Použijete-li jej před podmínkou, bude výsledná podmínka splněna tehdy, když původní podmínka splněna nebude a naopak. Například podmínka "**NOT Nějaký předmět"** je splněna tehdy, když před Baltíkem není žádný předmět.

Operátor AND (a zároveň)
Operátor ***AND*** použijete mezi dvěma podmínkami tehdy, chcete-li, aby výsledná podmínka byla splněna právě když jsou splněny obě podmínky **současně**. Například podmínka

"[D] AND Nějaký předmět"

je splněna, pokud byla stisknuta (a přečtena příkazem Čti klávesu) klávesa [D] a současně je před Baltíkem nějaký předmět.

Operátor OR (nebo)
Operátor ***OR*** použijete mezi dvěma podmínkami tehdy, chcete-li, aby výsledná podmínka platila právě když je splněna **alespoň jedna** z těchto podmínek. Například podmínka

"[End] OR [F10]"

je splněna, pokud byla stisknuta (a přečtena příkazem Čti klávesu) klávesa [End] nebo klávesa [F10].

Operátor XOR (výlučné nebo, nonekvivalence)
Operátor ***OR*** použijete mezi dvěma podmínkami tehdy, chcete-li, aby výsledná podmínka platila právě když je splněna **právě jedna** z těchto podmínek, tzn. první nebo druhá, ale ne obě současně. Například podmínka

"[D] OR Nějaký předmět"

je splněna, pokud byla stisknuta (a přečtena příkazem Čti klávesu) klávesa [D] a před Baltíkem není žádný předmět, nebo pokud nebyla stisknuta klávesa [D] a před Baltíkem je nějaký předmět.

Závorky
se používají tehdy, když chceme určit, která část složené podmínky se má vyhodnocovat přednostně. Bez použití závorek se **nejdříve** vyhodnotí operátor ***NOT***, pak operátor ***AND*** a nakonec operátor ***OR***. Například podmínka

"Nějaký předmět AND ([A] OR [B])"

je splněna, je-li před Baltíkem nějaký předmět (např. auto) a současně byla stisknuta jedna z kláves [A] nebo [B]. Naproti tomu tatáž podmínka zapsaná bez závorek:

"Nějaký předmět AND [A] OR [B]"

je splněna právě tehdy, platí-li alespoň jedno z následujících tvrzení:

1. Před Baltíkem je předmět a současně byla stisknuta klávesa [A].
2. Byla stisknuta klávesa [B] (v tomto případě už nezáleží na tom, co je před Baltíkem).

Stejnou podmínku bychom za pomoci závorek zapsali:

"(Nějaký předmět AND [A]) OR [B]"

Složitější podmínky budou vždy přehlednější, použijete-li závorky.

Cykly s podmínkou - cykly "while" a "do - while"

Prvek ***cykly while*** sdružuje prvky pro zápis cyklu s podmínkou.
Tento cyklus použijeme tehdy, když potřebujeme opakovaně provádět nějakou činnost tak dlouho, dokud nepřestane platit nějaká podmínka.

Po vložení prvku ***cykly while*** do programu se rozvine nabídka ze dvou variant tohoto prvku:

Prvek ***while*** uvozuje cyklus s podmínkou na začátku.

Prvek ***do - while*** uzavírá cyklus s podmínkou na konci.

Cyklus s podmínkou na začátku (while)

Tento cyklus použijeme v případě, že chceme testovat podmínku ještě před provedením těla cyklu.

Syntaxe:
while <Podmínka> <Příkaz>

Pozn.: Prvek ***while*** spolu s podmínkou označujeme jako **hlavička cyklu**, <Příkaz> označujeme jako **tělo cyklu**. Pamatujte, že tímto příkazem může být i **složený příkaz** nebo např. příkaz cyklu.

Zpracování:

1. Baltík otestuje, zda zadaná **podmínka** platí. Pokud neplatí, ukončí cyklus a pokračuje v provádění programu příkazem následujícím za tělem cyklu.
2. Pokud podmínka platí, provede <Příkaz> (tělo cyklu) a znovu testuje podmínku podle bodu **1**.

Dokud není před Baltíkem nějaký předmět, dělej Popojdi

Dokud není před Baltíkem nějaký předmět, opakuje příkazy { Vlevo vbok, Vyčaruj zeď, Vpravo vbok, Popojdi}

Pozn.: Před příkazem nebo blokem příkazů se ignorují mezery a konce řádků.

Cyklus s podmínkou na konci (do - while)

Podmínka na konci cyklu je vhodná pro případ, kdy k jejímu vyhodnocení potřebujeme provést nějaké operace v těle cyklu a proto k jejímu testování dochází až po provedení těla.

Syntaxe:
<Příkaz> **do - while** <Podmínka>

Pozn.: Prvek ***do - while*** spolu s podmínkou označujeme jako **hlavička cyklu**, <Příkaz> označujeme

jako **tělo cyklu**. Pamatujte, že tímto příkazem může být i **složený příkaz** nebo např. příkaz cyklu.

Zpracování:

1. Baltík provede <Příkaz> (tělo cyklu).
2. Pak otestuje, zda zadaná **podmínka** platí. Pokud neplatí, ukončí cyklus a pokračuje v provádění programu příkazem následujícím za tělem cyklu. Pokud podmínka platí, znovu provádí <Příkaz> podle bodu **1**.

Prováděj cyklus: čti klávesu, zobraz přečtenou klávesu, dokud není stisknuta klávesa [End]

Baltík nejprve přečte a zobrazí klávesu, pak teprve zjišťuje, zda byla stisknuta (a zobrazena) klávesa End. Pokud ano, cyklus je ukončen. Pokud ne, je znovu přečtena a zobrazena klávesa.

Pozn.: Před příkazem nebo blokem příkazů se ignorují mezery a konce řádků.

Cyklus s řídící proměnnou - cyklus "for"

Předčasné ukončení cyklu:

Prvek ***for*** uvozuje počítané opakování těla cyklu s řídící **proměnnou.**
Tento cyklus využijeme např. tehdy, když chceme několikrát opakovat příkaz nebo blok příkazů, jehož vykonávání však nějak závisí na počtu již provedených opakování, tedy na hodnotě řídící proměnné.

Syntaxe:

for <Parametr_cyklu> **[[,]** <Počáteční_hodnota> **[[,]** <Koncová_hodnota> **[[,]** <Krok> **]]]** <Příkaz>

Navíc platí:

- Ü Za nezadané parametry se dosazuje hodnota **1**.
- Ü Parametr cyklu musí být celočíselná nebo reálná **proměnná**.

Pozn.: Prvek ***for*** spolu s následujícími parametry označujeme jako **hlavička cyklu**, <Příkaz> označujeme jako **tělo cyklu**. Pamatujte, že tímto příkazem může být i **složený příkaz** nebo např. příkaz cyklu.

Zpracování:

1. Spočte hodnoty výrazů <Počáteční_hodnota>, <Koncová_hodnota> a <Krok>.
2. Řídící proměnné přiřadí počáteční hodnotu.
3. Otestuje, nepřekročila-li řídící proměnná koncovou hodnotu cyklu přičemž směr překročení je určen znaménkem parametru cyklu. Pokud překročila, ukončí cyklus a pokračuje v plnění programu prvním příkazem za cyklem.
4. Nebyla-li koncová hodnota překročena, provede tělo cyklu, tj. <Příkaz> (příkazem může být i **složený příkaz**).
5. Spočte výrazy<Koncová_hodnota> a <Krok>.
6. Přičte k řídící proměnné hodnotu výrazu <Krok> a pokračuje testováním podle bodu **3**.

Příklad:

Vyčaruje postupně všechny předměty z nulté banky.

Baltík přiřadí do proměnné ***Počitadlo*** jedničku a vyčaruje předmět číslo 1 (prázdné pole). Pak zvýší hodnotu počitadla o jedničku a vyčaruje předmět číslo 2. V této činnosti bude pokračovat tak dlouho, dokud nevyčaruje předmět číslo 150. Když pak zvýší hodnotu počitadla, zjistí, že překročila koncovou hodnotu, tj. hodnotu **systémové konstanty** ***Počet předmětů***, a cyklus proto ukončí. Po ukončení cyklu bude mít proměnné ***Počitadlo*** hodnotu 151.

Pozn.: Hlavičku cyklu můžete pro přehlednost uzavřít do kulatých závorek:

Pozn.: Nehrozí-li nejednoznačná interpretace, můžete čárky mezi parametry vynechat.


FOR
18
1
{
18
}

Větvení "if" a "if - else"

Prvek ***if*** uvozuje příkaz větvení.
Tento příkaz použijeme v případě, když potřebujeme rozhodnout o dalším postupu na základě platnosti či neplatnosti nějaké podmínky.

Prvek ***else*** uvozuje alternativní větev.
Alternativní větev se používá v případě, kdy se rozhodujeme mezi dvěma možnými akcemi, za nimiž následuje společné pokračování.

Syntaxe:

if <Podmínka> <Příkaz_1> **[else** <Příkaz_2> **]**

Pozn.: Mezi podmínkou a příkazem a před větví ***else*** se ignorují mezery a konce řádků.

Realizace

1. Baltík vyhodnotí podmínku
2a. Je-li podmínka splněna, provede Příkaz_1.
2b. Není-li podmínka splněna a obsahuje-li konstrukce větev ***else***, provede Příkaz_2.
3. Pokračuje v plnění programu příkazem, který následuje za příkazem ***if***.

Vnořování příkazů

Příkazy ***if*** do sebe můžeme vnořovat, tj. na místě Příkazu_1 nebo Příkazu_2 můžeme znovu použít příkaz ***if***. Využíváme toho zejména v případech, kdy se potřebujeme rozhodnout mezi více než dvěma možnostmi. Podobu takového pěticestného rozhodování bychom mohli zapsat následovně (ti bystřejší si ji jistě dokáží odvodit ze zápisu syntaxe):

if <podmínka_1> <příkaz_1>
else if <podmínka_2> <příkaz_2>
else if <podmínka_3> <příkaz_3>
else if <podmínka_4> <příkaz_4>
else <příkaz_5>

Baltík provádí předchozí program následovně:
Je-li splněna Podmínka_1, provede se Příkaz_1. Není-li splněna, testuje se Podmínka_2. Je-li splněna Podmínka_2, provede se Příkaz_2. Není-li splněna, testuje se Podmínka_3. Tak se pokračuje stále dál. Není-li splněna žádná z podmínek, provede se příkaz ve větvi else - v našem případě Příkaz_5.

Příklady

Jestliže je před Baltíkem nějaký předmět, proveď Vlevo vbok.

Je-li před Baltíkem nějaký předmět, vyčaruj místo něj trávu a popojdi, jinak vyčaruj strom.

Čekej na stisk klávesy a přečti ji.

Jestliže byla stisknuta šipka nahoru, popojdi

Jinak jestliže byla stisknuta šipka doleva, otoč se vlevo

Jinak jestliže byla stisknuta šipka doprava, otoč se vpravo

Jinak vyčaruj tulipán.

Větvení "switch - case"

Větvení "switch - case" slouží k zjednodušenému zápisu větvení programu do více větví podle hodnoty výrazu. Zapisuje se pomocí dvou prvků if:

Větvení typu "**switch - case**" uvozujeme zdvojeným prvkem ***if***.
Tohoto větvení využijeme k zjednodušenému zápisu v případech, kdy se podle hodnoty výrazu rozhodujeme pro vykonání jedné z několika akcí.

Syntaxe:

if if <Výraz> <Oddělovač> **(** <Oddělovač> <Případ> <Příkaz> **) ... [else** <Příkaz> **]**

Poznámky:

- Ü Všimněte si, že mezi Výrazem a prvním Případem musí být oddělovač, tj. mezera nebo konec řádku. V opačném případě by Baltík oba dva elementy sloučil a považoval je za jeden trochu složitější **výraz.**
- Ü Před každým případem s výjimkou prvního, mezi případem a příkazem a před větví else se ignorují mezery a konce řádků.
- **Ü** Hodnotou výrazu, která řídí celé větvení, může být jak celé nebo reálné **číslo,** tak **řetězec.**.
- Ü Jako řídící výraz i jako výrazy v případech můžete používat také **předměty** nebo **klávesy**.
- Ü Při používání řetězců jsou relační operátory používány vzhledem k **tabulce kódů znaků**, takže např. "Adam" < "Běta", ovšem "adam" > "Běta", protože malé a je v tabulce až za velkým B, podobně "Čeněk" > "David", protože Č je až za D a "Hana" > "Chrudoš", protože Ch je bráno jako dva znaky C a h, a C je v tabulce před H.

Realizace

1. Baltík nejprve vyhodnotí Výraz a výsledek si zapamatuje.
2. Prochází jednotlivé Případy v tom pořadí, v jakém jsou zapsány v programu, a testuje, zda zapamatovaná hodnota vyhovuje popisu.

3.a Najde-li Případ, jemuž zapamatovaná hodnota výrazu vyhovuje, vykoná příslušný Příkaz.

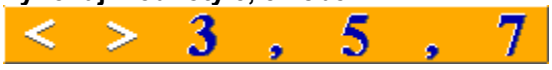
3.b Nehodí-li se zapamatovaná hodnota k žádnému případu a má-li příkaz větev ***else***, vykoná Příkaz v této větvi.

4. Pokračuje v plnění programu.

Příklady Případů:

Vyhovují všechny hodnoty větší nebo rovné 3

Vyhovují hodnoty 3, 5 nebo 7



Vyhovují všechny hodnoty s výjimkou hodnot 3, 5 a 7

Vyhovuje hodnota 7 a všechny hodnoty od 3 do 5 včetně (tj. např. hodnoty 3, 3.5, 4.99 nebo 5)

Vyhovují hodnoty menší nebo rovné 2 a hodnoty větší nebo rovné 8.

Vyhovují hodnoty větší než 3 a zároveň menší nebo rovné 5

Vyhovují hodnoty z intervalu 1 až 10 a hodnoty větší nebo rovné 20

Příklad programu

Podprogram *Značkuj* popojde s Baltíkem o jedno políčko ve směru, do nějž je natočen, a na políčku, které právě opustil, zanechá značku, podle níž se pozná, kam se vydal.

Přerušení opakování ("break")

počítané opakování nekonečný cyklus cyklus s řídící proměnnou cyklus while

Prvek ***Break*** (přerušení opakování)
použijeme v případě, kdy potřebujeme nestandardně opustit cyklus.

Za příkazem ***Break*** můžete zadat **číslo**, jež určuje počet vnořených cyklů, která chcete ukončit:

Použijete-li příkaz ***Break*** s prvkem ***Nekonečno***, budou ukončeny všechny vnořené cykly.

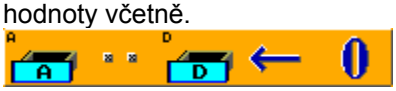
Pozn.: K předčasnému opuštění **pomocníka** nebo celého programu použijte příkaz **Ukonči provádění pomocníka**.

Příklady:

Opakuj donekonečna čtení klávesy. Ukonči cyklus po stisku klávesy *End* Jinými slovy: čekej na stisk klávesy ***End***.

Interval

Prvek ***Interval*** používáme ke zkrácenému zápisu seznamu všech hodnot od počáteční do koncové hodnoty včetně.

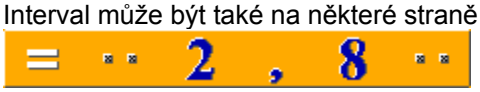


Vynuluj celočíselné proměnné A, B, C a D.

Vytiskni 3., 4. a 5. znak řetězce R.

Vygeneruj náhodně některé z čísel 1, 2, 3, 4, 5, 6.

Interval může být také na některé straně neomezený:



Případ v příkazu typu case nastane, je-li hodnota menší nebo rovna 2 nebo větší nebo rovna 8.

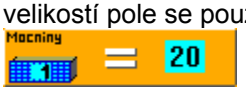
Pole

Pole je skupina proměnných (prvků) stejného typu, které můžete používat jako jeden objekt.

Polím je v bance **proměnných** věnován předposlední řádek.

Deklarace

Než začneme pole používat, musíme je nejprve **deklarovat**. V deklaraci Baltíkovi oznamujeme, že se chystáme používat dané pole a zároveň také specifikujeme, kolik bude toto pole mít prvků. Pro stanovení velikostí pole se používá prvek ***Rovno***:



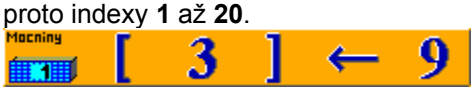
Deklarujeme celočíselné pole *Mocniny* s 20 prvky

Indexace

K jednotlivým proměnným přistupujeme pomocí indexů označujících pořadí dané proměnné v poli. **Prvky pole** proto někdy označujeme jako **indexované proměnné**.

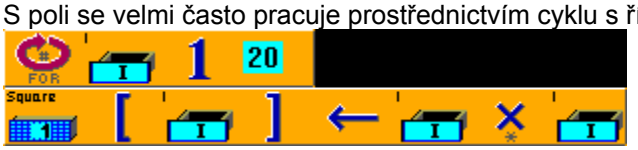
Indexy uzavíráme do hranatých závorek.

Proměnné v poli se indexují (číslují) od **1**. Proměnné v poli ***Mocniny***, které jsme deklarovali výše, mají proto indexy **1** až **20**.



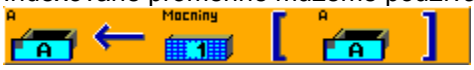
Třetímu prvku pole *Mocniny* přiřaď hodnotu 9

S poli se velmi často pracuje prostřednictvím cyklu s řídící proměnnou:



Prvkům pole *Mocniny* přiřaď druhé mocniny jejich indexů.
Úlohu vyřešíme pomocí cyklu s parametrem (řídící proměnnou), který bude nabývat postupně hodnoty indexů pole, tj. hodnoty od **1** do **20**. Při každém průchodu cyklem přiřadíme jedné indexované proměnné hodnotu rovnou druhé mocnině jejího indexu.

Indexované proměnné můžeme používat stejně jako obyčejné proměnné:



Proměnné *A* přiřaď druhou mocninu její hodnoty

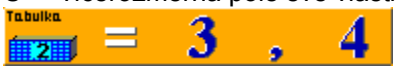
***Mocniny[2]* krát popojdi**
Baltík popojde o 4 políčka

Vícerozměrná pole

Prozatím jsme probírali pouze jednorozměrná pole. Pole ale může mít i více rozměrů.

Pole často nazýváme podle počtu rozměrů:

- Ü **jednorozměrné** pole nazýváme **vektor**,
- Ü **dvojrozměrné** pole nazýváme **tabulka** nebo **matice**,
- Ü vícerozměrná pole své vlastní názvy nemají.

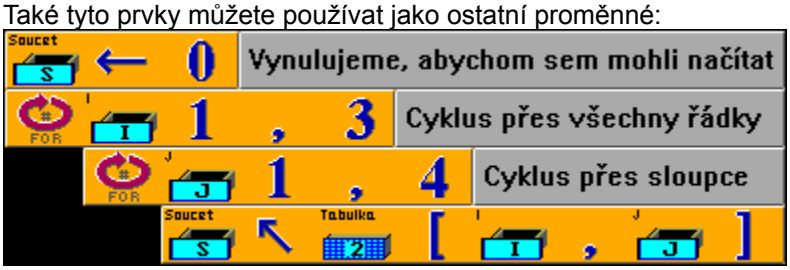


Deklaruj dvojrozměrné celočíselné pole *Tabulka* se třemi řádky a čtyřmi sloupci.

Toto pole má prvky:

Tabulka[1,1]	Tabulka[1,2]	Tabulka[1,3]	Tabulka[1,4]
Tabulka[2,1]	Tabulka[2,2]	Tabulka[2,3]	Tabulka[2,4]
Tabulka[3,1]	Tabulka[3,2]	Tabulka[3,3]	Tabulka[3,4]

Také tyto prvky můžete používat jako ostatní proměnné:



Sečti všechny prvky pole *Tabulka* a výsledek ulož do proměnné *Soucet*.

Pozn.: Těla cyklů nemusíme dávat do složených závorek, protože jsou tvořena jediným příkazem: tělo **vnějšího** cyklu je tvořeno příkazem cyklu a tělo **vnitřního** cyklu příkazem pro přičtení hodnoty prvku k hodnotě proměnné ***Soucet***.

Hromady

V této sekci budeme probírat následující témata:

Co je to hromada

Prvek ***Hromada***.
Hromada je speciální datová struktura, jejímiž prvky jsou **proměnné.**

Hromada může obsahovat **libovolné množství proměnných**, se kterými je možno pracovat jako s ostatními proměnnými. Výhodou proměnných v hromadě je, že **jsou uloženy na disku**, a proto zůstávají zachovány i po skončení programu.

POZOR! **Používáte-li velké hromady, může někdy jejich ukládání na disk zpomalit program.**

Hromady jsou číslovány čísly **1** až **99** a soubory hromad na disku pak mají názvy `jmeno.v01` až `jmeno.v99`, kde `jmeno` je jméno vašeho programu.

Pozn.: Pokud nezadáte číslo hromady nebo zadáte číslo **0**, budete pracovat se **základními proměnnými**, které jsou v paměti.

2) Vytvoření hromady

Hromadu vytvoříte pomocí prvku přiřaď a proměnných, které má obsahovat. V rámci definice se proměnným hromady zároveň přiřadí aktuální hodnoty příslušných proměnných.

Vytvoř hromadu č. *1* z globálních celočíselných proměnných *A..K*, lokální reálné proměnné *x1* a globálních řetězcových proměnných *S2* a *S3*

POZOR! **Novým přiřazením proměnných rušíte starou hromadu a zřizujete hromadu novou!**

Pozn.: To, zda při zřizování hromady použijeme lokální či globální proměnné je pouze otázka názvu. Oba druhy proměnných budou po zřízení hromady jenom její - budou v hromadě lokální. Budete se k nim moci obracet pouze prostřednictvím jejich hromady - viz pasáž **Používání proměnných z hromady.**

Vytvoření hromady z jiných hromad

Hromadu také můžete tvořit **z jiných hromad** nebo z některých proměnných jiných hromad:

Vytvoř hromadu *1* z hromady *2*
Hromada *1* nyní bude mít stejný obsah jako hromada *2*. Bude tedy obsahovat stejně pojmenované proměnné a jejich počáteční hodnota bude rovna hodnotě stejnojmenných proměnných z hromady *2* v době přiřazení.

Vytvoř hromadu *1* z proměnných *A*, *B* hromady *2*

Obsahuje-li hromada *2* proměnné *A* a *B*, bude hromada 1 tvořena proměnnými *A* a *B*. Jejich počáteční hodnoty budou rovny hodnotám stejnojmenných proměnných z hromady *2* v době přiřazení.

Použití množinových operací

Použitím prvků ***AND*** a ***OR*** můžete vytvořit průnik, resp. sjednocení dvou hromad do jiné hromady:

Vytvoř hromadu *1* z průniku hromad *2* a *3*

Hromada *1* nyní bude nyní obsahovat pouze ty proměnné, které jsou společné pro hromadu *2* a hromadu *3*. Pokud např. hromada *2* obsahuje proměnné *A*, *B* a *C* a hromada *3* obsahuje proměnné *B*, *C* a *D*, bude hromada *1* obsahovat proměnné *B* a *C*. Jejich hodnoty budou naplněny, pokud budou mít i v hromadách *2* a *3* stejnou hodnotu. Pokud budou mít různou hodnotu, nebudou tyto proměnné naplněny (budou neinicializované).

Vytvoř hromadu *1* ze sjednocení hromad *2* a *3*

Hromada *1* bude nyní obsahovat všechny proměnné, které jsou v hromadách *2* a *3*. Pokud např. hromada *2* obsahuje proměnné *A*, *B* a *C* a hromada *3* obsahuje proměnné *B*, *C* a *D*, bude hromada *1* obsahovat proměnné *A*, *B*, *C* a *D*. Hodnoty proměnných, které jsou pro obě zdrojové hromady společné, budou naplněny pouze v případě, budou-li mít odpovídající proměnné v hromadách *2* a *3* stejnou hodnotu.

3) Modifikace hromady

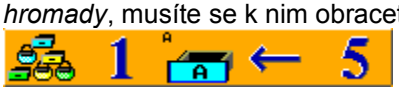
Pomocí prvků ***Zvětši*** a ***Zmenši*** můžete do hromady proměnné přidávat nebo je z ní ubírat:

Přidej k hromadě *1* proměnnou *B*
Hromada *1* bude kromě původních proměnných navíc obsahovat i proměnnou *B*.

Uber z hromady *1* proměnnou *B*
Obsahovala-li hromada *1* obsahovala proměnnou *B*, bude z ní ubrána.

4) Používání proměnných z hromady

Proměnné z hromady můžete používat jako ostatní proměnné. Jak jsme si ale již řekli v pasáži *Vytvoření hromady*, musíte se k nim obracet prostřednictvím jejich hromady.



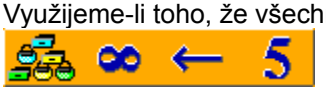
Proměnné *A* z hromady *1* přiřaď *5*

Nastav rychlost na hodnotu, která je v proměnné *A* z hromady *1*

Pomocí prvku ***Nekonečno*** můžete také přiřadit jeden výraz všem proměnným hromady najednou:

Všem proměnným z hromady *1* přiřaď hodnotu *5*

Využijeme-li toho, že všechny proměnné považujeme za součást hromady **0**, můžeme napsat:



Všem použitým proměnným v programu přiřaď *5*

5) Zrušení hromady

Chcete-li hromadu zrušit, tj. smazat z disku, použijte jako číslo hromady záporné číslo:

Zruš hromadu *1*

Číslo

V úrovni **Začátečník** můžete v místě, kde je požadováno číslo, používat **pouze celá čísla sestavená z prvků *Číslice.***

V úrovni **Pokročilý** můžete v místech, kde jsou požadována čísla, použít libovolný **číselný výraz.**

Výrazy

Výraz je předpis, jak ze zdrojových hodnot získat hodnotu požadovanou.

Baltík rozeznává tři druhy výrazů:

- Ü **číselné výrazy,** jejichž vyhodnocením získáme číslo,
- Ü **řetězcové výrazy,** jejichž vyhodnocením získáme textový řetězec a
- Ü **souřadnicové výrazy,** jejichž vyhodnocením získáme souřadnice.

Tyto druhy dat se nemohou míchat. Budete-li chtít použít v řetězcovém výrazu číslo, resp. v číselném výrazu řetězec, budete muset použít vhodnou **konverzní funkci** převádějící **řetězec na číslo,** resp. **číslo na řetězec.**

Číselné výrazy

Číselný výraz se skládá z jednodušších číselných výrazů a operací mezi nimi. V této sekci postupně probereme následující součásti číselných výrazů:

1) Konstanty a proměnné

Všude, kde ve výrazu použijete **konstantu** nebo **proměnnou,** se objeví její hodnota:

Šířka obrazovky v bodech
Výsledkem je šířka obrazovky v bodech, tj. 585.

Výsledkem je hodnota, která je uložena v proměnné A.

2) Aritmetické operátory

2.1 Unární operátory (operátory s jedním operandem)

Prvek ***Unární operátor*** slouží především k označení znaménka svého operandu. Po jeho vložení před číslo se prvek změní na prvek ***Unární minus***, který mění znaménko svého operandu.

Prvek ***Unární minus***

Pozn.: Uvnitř výrazů můžete ke specifikaci znaménka používat i obyčejné (binární) minus (viz níže). Toto zvláštní, unární minus slouží především k zadávání záporných **parametrů** bez nutnosti používat závorky.

Pozn.: Stiskem pravého tlačítka na unárním minus a výběrem volby Upravit můžete změnit toto znaménko na **unární plus**. Unární plus nemá žádný účinek a existuje pouze pro úplnost.

2.2 Binární aritmetické operátory (operátory se dvěma operandy)

Mezi binární aritmetické operátory počítáme operátory **sčítání**, **odčítání**, **násobení, dělení** a operátor **získání zbytku po dělení** (operátor **modulo**).

Sčítání, např. 1 + 1 = 2.

Odčítání, např. 130 - 85 = 45.

Násobení, např. 8 * 9 = 72.

Dělení, např. 36 / 9 = 4.

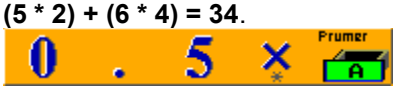
Zbytek po celočíselném dělení (modulo), např. 10 % 3 = 1.

Pro **typ výsledné číselné hodnoty** platí:

- Ü jsou-li oba operandy **celá čísla**, je výsledek celé číslo, např. hodnota výrazu **10 / 3** je **3**.
- Ü je-li jeden z operandů **reálné číslo**, výsledek je reálné číslo, např. hodnota výrazu **10.0 / 3** je **3.333333333333333**.

Je-li výraz složitější, vyhodnocuje se zleva doprava přičemž se respektuje **priorita operátorů.**

Hodnota výrazu bude
(5 * 2) + (6 * 4) = 34.



Hodnotou výrazu bude polovina hodnoty, uložené v proměnné ***Prumer***.

3. Logické operátory

Logické operátory použijete zejména tehdy, když potřebujete spojovat jednodušší **podmínky** do složitějších. Použijete-li logický operátor v číselném výrazu, je jeho hodnota:

- Ü **1** při splnění testované podmínky,
- Ü **0** při jejím nesplnění.

Konjunkce (AND, a zároveň)
je splněna jsou-li splněny oba operandy.

Disjunkce (OR, nebo)
je splněna, je-li splněn alespoň jeden operand.

Negace (NOT, neplatí)
obrací pravdivostní hodnotu operandu.

Nonekvivalence (OR, výlučné nebo)
je splněna, je-li splněn právě jeden operand.

Pozn.: Konjunkce bývá někdy označována za **logický součin** a disjunkce za **logický součet**.

Viditelný AND S obláčkem
Hodnotou tohoto výrazu je číslo **1**, je-li Baltík viditelný a čarování je zapnuto s obláčkem, jinak **0**.

4. Relační operátory

Relační operátory se používají k porovnání dvou čísel nebo řetězců. Jejich výsledek bývá nejčastěji používán jako **podmínka**. Použijete-li relační operátor v číselném výrazu, je jeho výsledkem nula (nepravda) nebo jednička (pravda).

Menší

Rovno

Větší

Menší nebo rovno

Větší nebo rovno

Různé

Je-li hodnota *A* různá od *0*, Popojdi

5. Závorky

Kulaté závorky slouží k několika účelům:

- Ü ohraničení části výrazů, které se mají vyhodnotit dříve,
- Ü ohraničení složených parametrů funkcí,
- Ü zpřehlednění programu.

Postup při použití prvku ***Závorky*** pro uzávorkování výrazu je stejný jako použití složených závorek pro

označení ***bloku příkazů.***

Postup při označování výrazu kulatými závorkami:

1. Klepneme v panelu příkazů na prvek ***Závorky***.
2. Přesuneme vybraný prvek **před první** prvek uzávorkovávaného výrazu.
3. Klepnutím sem umístíme prvek s otevírací závorkou.
4. Přesuneme držený prvek **za poslední** prvek uzávorkovávaného výrazu.
5. Klepnutím sem umístíme prvek s uzavírací složenou závorkou.

Přiřaď do *A* hodnotu výrazu *5 * (2+6) * 4*

6. Funkce

Vždy, když Baltík narazí při vyhodnocování výrazu na funkci, zavolá ji. Funkce podle svého vnitřního předpisu zjistí příslušnou hodnotu, kterou Baltíkovi vrátí a ten ji v daném místě vyhodnocovaného výrazu použije.

Použití každé funkce je vysvětleno v sekci, která se zabývá příslušným tématem. Proto zde uvedeme pouze ukázky několika typických použití funkcí:

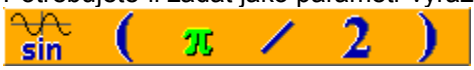
Přiřaď do *A* hodnotu zadanou z klávesnice

Řada funkcí potřebuje k výpočtu své hodnoty jeden či více parametrů, které se vkládají za volání funkci.

Přehraj zvuk č. 4

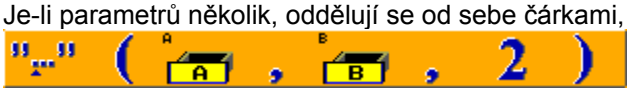
Souřadnice *y* myši při stisku levého tlačítka

Potřebujete-li zadat jako parametr výraz, uzavřete jej do závorek.



sinus (π / 2)

Je-li parametrů několik, oddělují se od sebe čárkami,



Zjisti pozici podřetězce A v řetězci B od pozice 2

7. Bitové logické operátory a bitové posuvy

Tyto operátory pracují s celými čísly o délce 32 bitů - přesněji s jejich bitovou reprezentací.

Prvek ***Bitové operace***
slouží k vkládání bitových operátorů a posunů

Po vložení prvku ***Bitové operace*** do programu se rozvine nabídka, ze které můžete vybrat příslušný operátor.

3.1 Bitové logické operátory

Bitové logické operátory jsou operátory, při jejichž vyhodnocování se provádějí logické operace nad každými dvěma odpovídajícími bity obou operandů.

Bitová konjunkce (AND, a zároveň)
Výsledný bit má hodnotu **1**, pokud odpovídající bity obou operandů mají hodnotu **1**.

Bitová disjunkce (OR, nebo)
Výsledný bit má hodnotu 1, pokud alespoň jeden z odpovídajících bitů jeho operandů má hodnotu 1.

Bitová negace (NOT, neplatí)
změní hodnoty všech bitů svého operandu.

Bitová nonekvivalence (XOR, výlučné nebo)
Výsledný bit má hodnotu 1, pokud je součet odpovídajících bitů roven 1.

Pozn.: Konjunkce bývá někdy označována za **logický součin** a disjunkce za **logický součet**.

Hodnotou výrazu je číslo vzniklé bitovým součinem **1**. Číslo **5** má ve dvojkové soustavě tvar **101B** (**B** na konci označuje zápis ve dvojkové soustavě), číslo **3** je ve dvojkové soustavě **11B**, výsledkem je tedy **1** (**001B**).

3.2 Bitové posuvy

Bitové posuvy slouží k posuvu všech bitů levého operandu doprava nebo doleva o počet, který je zadán pravým operandem.

Bitový posun vlevo

Výsledkem operace **a << b** je číslo **a** posunuté o **b** bitů vlevo. Vytlačené bity vlevo jsou ztraceny. Uvolněné bity vpravo jsou doplněny nulami

Pozn.: Je-li absolutní hodnota čísla dostatečně malá, je posun o jeden bit vlevo ekvivalentní násobení dvěma.

Hodnotou výrazu je číslo **5** (**101B**) posunuté o **3** bity vlevo, přičemž **3** bity vpravo budou doplněny nulami,

výsledek je tedy **101000B**, tzn. číslo **40**.

Aritmetický bitový posuv vpravo

Výsledkem operace **a >> b** je číslo **a** posunuté o **b** bitů vpravo. Vytlačené bity vpravo jsou ztraceny. Uvolněné bity vlevo jsou doplněny hodnotou znaménkového (tj. nejvyššího) bitu.

Pozn.: Aritmetický bitový posun kladného čísla vpravo je ekvivalentní dělení dvěma.

Hodnotou výrazu je číslo **9** (**1001B**) posunuté o **2** bity vpravo, přičemž **2** uvolněné bity vlevo budou doplněny znaménkovým (tj. nejvyšším) bitem - v tomto případě nulou. Výsledek je tedy **10B**, tzn. číslo **2**.

Hodnotou výrazu je číslo **-6** (**...1111010B**) posunuté o **2** bity vpravo, přičemž uvolněné bity vlevo budou doplněny znaménkovým bitem - v tomto případě jedničkou. Výsledek je tedy **...111110B** (celkem 31 jedniček a na konci nula), tzn. číslo -**2**.

Pozn.: Binární reprezentaci záporného čísla získáte tak, že invertujete všechny bity jeho kladného protějšku s výjimkou posledního bitu.

Bitový posuv vpravo bez znaménka

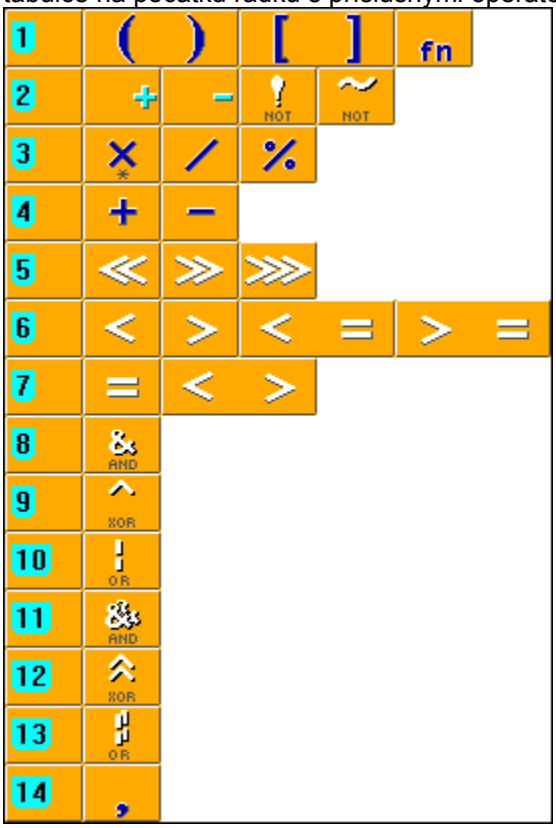
Výsledkem operace **a >>> b** je číslo **a** posunuté o **b** bitů vlevo. Přetékající bity vpravo jsou ztraceny. Uvolněné bity vlevo jsou doplněny nulami.

Hodnotou tohoto výrazu je číslo **9 (1001B)** posunuté o **2** bity vpravo, přičemž uvolněné bity vlevo budou doplněny nulami, výsledek je tedy **10B**, tzn. číslo **2**.

Hodnotou tohoto výrazu je číslo **-6** (**...1111010B)** posunuté o **2** bity vpravo, přičemž uvolněné bity vlevo budou doplněny nulami, výsledek je tedy **0011...11110** (2 nuly, 29 jedniček a nula), tzn. číslo **1073741822 (2^30 -2)**.

Priorita operátorů (pořadí vyhodnocování)

Priorita operátorů respektuje zvyklosti platné v jazycích **Java**, **C++** a **C**. Čím nižší číslo je v následující tabulce na počátku řádku s příslušnými operátory uvedeno, tím vyšší prioritu mají.



Výrazy se vyhodnocují postupně zleva doprava. Operátory s vyšší prioritou se při vyhodnocování výrazu provedou před operátory s nižší prioritou. Budeme-li mít např. výraz

a + b * c

vyhodnotí se nejprve součin **b*c** a výsledek se přičte k **a**.

Řetězce

Vedle číselných konstant proměnných obsahujících čísla může Baltík pracovat i s řetězcovými proměnnými obsahujícími texty.

V nápovědě jsou řetězce odlišeny od okolního textu tím, že jsou uzavřeny do uvozovek, např.: "Ahoj světe!", "Stiskni klávesu", "Petr", "123456", "***&^&...><<?" apod.

V této sekci se budeme věnovat následujícím tématům:

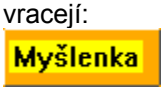
1) Použití řetězců řetězcové výrazy

Řetězec je chápan jako pole znaků. Z toho vyplývají možnosti práce s řetězcem:

- Ü můžeme s ním pracovat jako s celkem,
- Ü můžeme přistupovat k jeho jednotlivým znakům a
- Ü můžeme pracovat s jeho částmi - tzv. **podřetězci**.

V následujících příkladech využijeme toho, že **samostatně stojící řetězec** v programu **způsobí svoje vytištění na obrazovku** do prostoru před Baltíkem.

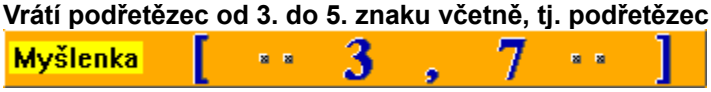
Zkuste postupně vložit do programu následující řetězcové výrazy a nechte si vytisknout hodnotu, kterou vracejí:



Vrátí text "Myšlenka"

Vrátí 3. znak řetězce - písmeno "š"

Vrátí podřetězec od 3. do 5. znaku včetně, tj. podřetězec "šle"



Vrátí podřetězec od počátku do 3. znaku spojený s podřetězcem od 7. znaku do konce řetězce, tj. slovo "Myška".

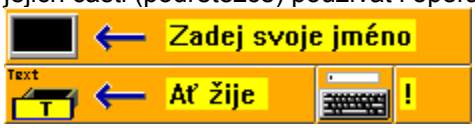
Jak vidíte v poslední ukázce, podřetězec můžeme konstruovat i z několika částí. Obecná syntaktická definice konstrukce podřetězce je následující:

<Řetězec> **[** <Interval> **[,** <Interval>**]...** **]**

Často budete využívat toho, že **vedle sebe stojící řetězce se automaticky spojují** do řetězce jediného.

Budu-li mít v řetězcové proměnné ***Začátek*** uloženo křestní jméno "Bohumír", v řetězcové konstantě ***Mezera*** uloženu mezeru a v řetězcové proměnné ***Konec*** uloženo příjmení "Soukup", tak se v uvedeném příkladu sousedící řetězce automaticky spojí a vytvoří řetězec "Bohumír Soukup".

V řetězcových výrazech můžeme vedle základních prvků, jakými jsou literály, konstanty, proměnné a jejich části (podřetězce) používat i operace s řetězci a funkce vracející řetězec.



Baltík se uživatele zeptá na jméno, jež pak použije v řetězci, který uloží do proměnné *Text*.

2) Funkce pracující s řetězci

Délka řetězce

Očekává jako parametr řetězec a vrací počet znaků v řetězci.

Bude-li mít konstanta ***Autor*** hodnotu "Lermontov", bude mít výraz hodnotu ***9***.

Převod řetězce na číslo

Očekává jako parametr řetězec a vrací číslo, na něž lze řetězec převést. Typ hodnoty záleží na převáděném čísle.

Není-li řetězec možno převést na číslo, ohlásí chybu ***Špatně zapsaný výraz***.

Vrátí celočíselnou hodnotu **314**

Vrátí reálnou hodnotu **3,14**

Vrátí chybu ***Špatně zapsaný výraz***, protože v převáděném textu je místo desetinné tečky zapsána desetinná čárka, takže jej Baltík neumí převést na číslo.

Převod čísla na řetězec

Očekává jako parametr číslo a vrací příslušný textový řetězec.

Používá se nejčastěji při výstupu do **souboru** a při výstupu na obrazovku, při němž nechceme příkaz výstup dělit na několik částí:

Zobraz na displeji zadání příkladu

Při převodu na řetězec můžeme zadat **formát** výsledného čísla:

Do řetězcové proměnné Datum ulož dnešní datum ve formátu den.měsíc.rok

Pozice podřetězce v řetězci

Očekává dva až čtyři parametry:

1. (povinný) **Hledaný podřetězec**
2. (povinný) **Prohledávaný řetězec**
3. (volitelný) **Startovací pozice**, tj. pozice, od které se prohledávání začne. Není-li zadána, začne se při dopředném směru prohledávání od začátku a při zpětném směru od konce.
4. (volitelný) **Směr prohledávání** (**1** = dopředu, **-1** = zpět). Není-li zadán, prohledává se odpředu.

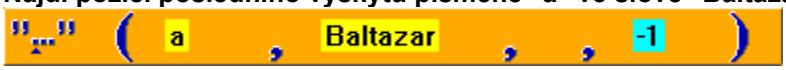
Syntaktický zápis funkce má tvar:

Pozice (<podřetězec>, <řetězec>, <od>, <směr>)

Funkce **vrátí pozici prvního znaku hledaného řetězce**, a to nezávisle na směru hledání - tj. vrátí stejné číslo, ať prohledáváme zepředu dozadu či naopak.

Nenalezne-li funkce v prohledávaném řetězci hledaný podřetězec, **vrátí nulu**.

Najdi pozici posledního výskytu písmene "a" ve slově "Baltazar"



Převod na velká nebo malá písmena

Pomocí prvku ***Převod na VELKÁ písmena*** můžete převést všechna písmena zadaného řetězce na velká.

Převeď řetězec "Martin Luther King" na velká písmena

Výsledkem funkce je řetězec "MARTIN LUTHER KING".

Pomocí prvku ***Převod na malá písmena*** můžete převést všechna písmena zadaného řetězce na malá.

Převeď řetězec "Martin Luther King" na malá písmena

Výsledkem funkce je řetězec "martin luther king".

3) Modifikace obsahu řetězcových proměnných

Přidání podřetězce

Chcete-li na konec obsahu řetězcové proměnné připojit řetězec, použijte prvek ***Zvětši***:

Je-li v proměnné A řetězec "piko" a v proměnné B řetězec "la", bude po provedení přiřazovacího příkazu v A řetězec "pikola".

Chcete-li vkládat řetězec dovnitř proměnné, zadejte do závorek číslo znaku, před který chcete vkládat.

Je-li v proměnné B slovo "Bazar", bude v ní po provedení příkazu slovo "Baltazar".

Vyjmutí podřetězce

Chcete-li z obsahu řetězcové proměnné ubrat nějakou část textu, použijte prvek ***Zmenši***:

Vyjmi z B 3 znaky počínaje 3. znakem
Proměnná ***B*** bude opět obsahovat slovo "Bazar".

Není-li zadána pozice, od níž se mají odebrat znaky, bude se odebírat **od počátku** řetězce.

Z proměnné můžeme odebírat nejen zadaný počet znaků, ale také přesně definovaný podřetězec.

Z textu v *B* vyjmi podřetězec "aza". Proměnná ***B*** bude obsahovat řetězec "Br".

I zde pak můžeme zadat pozici, od které se začne s jeho hledáním.

Grafické příkazy

Pomocí grafických příkazů můžete na kreslit kruhy, elipsy, čtverce, obdélníky, čáry, body, můžete používat sprej a vybarvovat uzavřené oblasti. Pomocí prvku **bod** můžete zjistit barvu zadaného bodu obrazovky.

Obdélník (čtverec)

Elipsa (kruh)

Čára

Bod - umožňuje i zjistit barvu

Vyplň

Sprej

Grafické příkazy mají jednotný zápis:

<Prvek> <Parametr> <Styl> <Barva> <Souřadnice dvou bodů>

Jednotlivé položky v příkazu mají následující význam:

<Prvek> je jeden výše vyjmenovaných grafických prvků

<Parametr> je **číslo**, které znamená:

- Ü pro prvky ***Kruh***, ***Čtverec***, ***Čára*** a ***Bod*** **tloušťku čáry**,
- Ü pro příkaz ***Vyplň*** **barvu hranice**, po kterou se má vybarvovat, případně barvu plochy, která se má přebarvit (viz styl),
- Ü pro příkaz ***Sprej*** **počet vykreslených bodů** (není-li zadán, vykreslí se **100** bodů).

<Styl> je kombinace prvků ***Režim XOR*** a ***Vyplň***

Prvek ***Režim XOR***

Jeho zadání způsobí, že se bude vykreslovat režimem XOR, tzn. že nová barva bude "vypočtena" aplikací logické funkce nonekvivalence na původní a vykreslovanou barvu.

Pro praktické využití je důležité to, že druhým vykreslením stejného obrazce na stejné místo v režimu XOR, je původně vykreslený obrazec smazán a podklad získává původní podobu. Toho lze s výhodou použít např. při animování vykreslovaného obrazce, protože není nutno obnovovat původní podklad.

Význam prvku ***Vyplň*** závisí na prvku na nějž je aplikován:

- Ü pro ***Kruh*** a ***Čtverec*** zadává kreslení obrazce, jehož vnitřek bude vyplněn,
- Ü pro prvek ***Vyplň*** zadává, že se bude vybarvovat do zadané hranice. Pokud styl neobsahuje

prvek ***Vyplň***, bude se vyplňovat souvislá plocha do hranice zadané barvy.

<Barva>

Prvek ***Barva*** určuje barvu, kterou se bude kreslit. Tuto barvu lze zadat dvěma způsoby (podrobnosti viz sekce **Barva**):

- Ü zadáním barvy v dialogovém okně,
- Ü zadáním čísla, které tuto barvu udává. Pro snadnější použití jsou v bance celočíselných konstant **systémové konstanty** seřazeny použitelné barvy. Pokud barva není zadána, bude se vykreslovat bílou barvou.

<Souřadnice dvou bodů>

Souřadnice dvou bodů udávají polohu a rozměry vykreslovaného obrazce.

Poznámky:

- **Ü** Pro příkazy ***Vyplň*** a ***Bod*** mají význam jen první z dvojice souřadnic.
- **Ü** ***Sprej*** je kulatý. Vykreslované **body** tedy nikdy **nebudou mimo elipsu** vepsanou do obdélníku zadaného dvojicí souřadnic.
- Ü Nezadáte-li souřadnice, bude se pracovat s políčkem před Baltíkem.

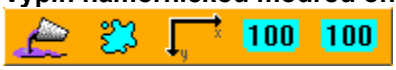
Příklady grafických příkazů:

Nakresli *Obdélník* na souřadnicích (20;20;500;40)

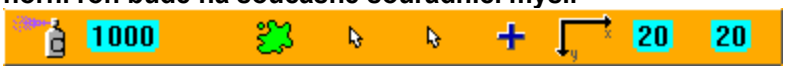
Nakresli stříbrnou čáru 10 bodů silnou z levého horního rohu svého prostoru (X0 Y0) do pravého dolního rohu [X(šířka obrazovky-1)Y(výška obrazovky-1)]

Do oblasti č. 1 nakresli v režimu XOR vyplněnou kaštanovou elipsu (bude-li oblast čtvercová, nakreslí se kruh)

Vyplň námořnickou modrou ohraničenou oblast kolem bodu X100 Y100



Nastříkej zelený sprej o hustotě 1000 bodů do obdélníkové oblasti velké 20×20 bodů jejíž levý horní roh bude na současné souřadnicí myši.



Barva bodu obrazovky

Prvek ***Bod*** můžete použít jako **číselný výraz**, který má hodnotu **barvy** bodu na zadaných souřadnicích

obrazovky.

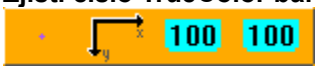
Prvek ***Bod*** umožňuje zjišťovat dva druhy barev:

Ü Nezadáte-li další parametr, získáte číslo od **0** do **16 777 216** (=2^24-1), což je přesné číslo barvy formátu **TrueColor**.

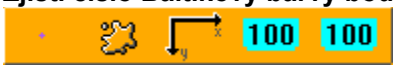
Ü Zadáte-li za prvek ***Bod*** ještě prvek ***Barva***, získáte číslo od **0** do **15**, které označuje pořadové číslo Baltíkovy barva - viz **systémové konstanty**.

Pozn.: Bude-li v druhém případě na daných souřadnicích bod jiné než Baltíkovy barvy, bude hodnotou výrazu nejbližší Baltíkova barva.

Zjisti číslo TrueColor barvy bodu na souřadnicích X100 Y100)



Zjisti číslo Baltíkovy barvy bodu na souřadnicích X100 Y100



Zadávání barvy

V této sekci se budeme postupně vysvětlovat následující činnosti:

- Přímé zadávání barvy
- Přímé zadání Baltíkovy barvy
- Číselné zadání Baltíkovy barvy
- Jak se v počítači tvoří a označují barvy
- Přímé zadání barvy TrueColor
- Zadání barvy prostřednictvím složek RGB
- Zadání barvy TrueColor jedním číslem
- Zadání barvy pozadí

1) Přímé zadávání barvy

Prvek ***Barva*** slouží v Baltíkových programech k zadávání barvy. Po jeho vložení do programu se otevře dialogové okno ***Výběr barvy***, které vám umožní požadovanou barvu jednoduchým způsobem zadat.

2) Přímé zadání Baltíkovy barvy

Ze záplavy barev, které je počítač schopen zobrazit, si Baltík vybral **16** svých oblíbených barev, jejichž zadávání maximálně zjednodušil. Budete-li chtít zadat jednu z Baltíkových oblíbených barev, stačí v paletě v levém horním rohu dialogové ho okna ***Výběr barvy*** kliknout na políčko s touto barvou.

3) Číselné zadání Baltíkovy barvy

Baltík má své oblíbené barvy očíslované od **0** do **15**. Chcete-li mít možnost zadat jednu z Baltíkových barev až při běhu programu, vložte do programu prvek ***Baltíkova barva*** (v dialogovém okně ***Výběr barvy*** jej najdete vlevo dole) a za něj **výraz,** jehož hodnota danou barvu definuje.

Pamatujte, že při zadávání hodnot Baltíkových barev můžete využít "barevných" **systémových konstant.**

Kolem oblasti č. 1 nakresli azurový obdélník

4) Jak se v počítači tvoří a označují barvy

Současné počítače již dokáží zobrazit více než 16 barev. Zobrazitelný počet barev závisí na řadě věcí a pohybuje se od 256 do 4 miliard. Povězme si proto nejprve, jak se barvy v počítači tvoří a jak bychom je mohli označovat.

Podíváte-li se na monitor svého počítače hodně zblízka nebo dokonce lupou, uvidíte, že každý bod obrazovky (říkejme mu **pixel**, což je zkratka z anglického picture element) je ve skutečnosti tvořen třemi

body: jeden svítí pouze červeně, druhý pouze zeleně a třetí pouze modře. Díky nedokonalosti vašeho oka se při pohledu z dostatečné vzdálenosti barvy těchto tří bodů slijí a vytvoří dojem jediného barevného bodu - pixelu.

Nesvítí-li žádný bod, má pixel barvu černou; svítí-li naopak všechny naplno, má barvu bílou; svítí-li všechny napůl, má barvu šedou. Svítí-li červená s modrou, bude mít pixel fialovou, svítí-li naplno červená se zelenou, má pixel barvu žlutou, svítí-li obě napůl, má barvu hnědou. Všechny barvy, které váš počítač dokáže zobrazit, vznikají jako kombinace intenzit jednotlivých "jednobarevných" bodů.

Při popisu barev se poslední dobou ustálil popis prostřednictvím trojice čísel od **0** do **255** popisující intenzitu jednotlivých jednobarevných bodů tvořících pixel.

Tato čísla se vždy uvádějí v pořadí Červená-Zelená-Modrá. Uvědomíte-li si, že v angličtině se tyto barvy jmenují Red-Green-Blue, bude vám hned jasné, proč se tento popis označuje zkratkou RGB.

Trojice (0,0,0) tak označuje černou, (255,255,255) bílou, (255,0,0) jasnou červenou, (128,0,0) tmavší červenou, (255,255,0) žlutou atd.

Výše popsaným způsobem můžeme popsat přibližně 16 milionů barev. Tato paleta se často označuje anglickým **TrueColor** (skutečné barvy).

5) Přímé zadání barvy TrueColor

Stisknete-li v dialogovém okně ***Nastavení barvy*** tlačítko ***Další***, otevře se dialogové okno ***Barva***. V něm vybrat jakoukoliv barvu z těch které nabízí vaše nastavení počítače.

Vybírat můžete čtyřmi způsoby:

- Ü klepnutím na políčko v paletě vlevo,
- Ü klepnutím na bod v paletě vpravo,
- Ü zadáním intenzity jednotlivých složek RGB,
- Ü zadáním hodnot jednotlivých složek v parametrizaci Odstín-Sytost-Světelnost

Po zadání získá oblast na ikoně zadanou barvu a na tlačítku se objeví malé spektrum symbolizující, že tento prvek nastavuje barvu mimo základní Baltíkovu šestnáctku.

Kolem oblasti č. 1 nakresli okrový (barva 4047551) obdélník

6) Zadání barvy prostřednictvím složek R, G, B

Chcete-li mít možnost barvu v průběhu programu vypočítat nebo ji zadat numericky prostřednictvím složek RGB, klepněte v okně ***Výběr barvy*** na ikonu ***Barva RGB*** a uchopený prvek vložte do programu. Za něj pak zadejte tři **číselné výrazy,** které budou definovat podíl jednotlivých složek.

Pozn.: Nezadáte-li všechny tři parametry, nastaví se barva, v níž má chybějící parametr hodnotu 0.

Kolem oblasti č. 1 nakresli 9 bodů široký okrový (barva 128,128,0) obdélník

7) Zadání barvy TrueColor jedním číslem

Číslo barvy TrueColor se dá jednoduše spočítat z čísel barvy jednotlivých složek pomocí vzorce:

65536*B +256*G +R

kde **B**, **G** a **R** jsou podíly jednotlivých barevných složek.

Této možnosti využijete zejména tehdy, budete-li chtít použít barvu, kterou jste dříve zjistili příkazem **barva bodu**:

Kolem políčka před Baltíkem nakresli 9 bodů široký obdélník v barvě bodu, na nějž ukazuje kurzor myši.

8) Zadání barvy pozadí

Při formátování písma v příkazech pro **zobrazení** textu a čísel a při zadávání výzvy pro **čtení z klávesnice** můžete specifikovat na jaké pozadí se bude vystupující text vypisovat.

Barva pozadí se zadává obdobně jako barva popředí. Jediný rozdíl je v tom, že v dialogovém okně ***Výběr barvy*** se nastaví přepínač ***Význam*** na hodnotu ***Pozadí***.

Prvky pro barvu pozadí mají příslušným způsobem vybarvené okolí namalované skvrny:

Baltíkova barva pozadí

Barva pozadí TrueColor

Barva pozadí RGB

Příklad:

Zobraz řetězec "INVERT" písmem Arial, černou barvou na bílém pozadí

Zobrazení

Narazí-li Baltík v programu na **samostatně stojící prvek**, který nikam nepatří, **zobrazí** jej na pozici před Baltíkem.

Jedinou **výjimkou** z tohoto pravidla **jsou čísla**, protože již od úrovně **Začátečník** platí, že samostatně stojící číslo oznamuje, kolikrát se má opakovat následující příkaz.

Chceme-li zobrazit číslo nebo chceme-li zobrazit jiný objekt rafinovanějším způsobem, přiřadíme zobrazovaný prvek prvku ***Obrazovka***.

Syntaktický zápis tohoto příkazu je (budete-li se chtít o kterémkoliv parametru dozvědět více, klepněte na něj):

Obrazovka <souřadnice> ***Přiřaď*** [<hodnota>] [<formát>] [<písmo>] [<barva>] [<Průhlednost>]

Rozeberme si jednotlivé parametry příkazu:

1) Souřadnice

Baltík umožňuje zadat **souřadnice,** na které bude zobrazovaný objekt umístěn. To, jakým způsobem bude vůči zadaným souřadnicím umístěn, však záleží na zarovnání nastaveném v rámci zadání **formátu**

- Ü je-li nastaveno zarovnání doleva, budou zadané souřadnice souřadnicemi levého horního rohu zobrazovaného objektu,
- Ü je-li nastaveno zarovnání doprava, budou zadané souřadnice souřadnicemi pravého horního rohu zobrazovaného objektu,
- Ü je-li nastaveno zarovnání na střed, budou zadané souřadnice souřadnicemi středu horního okraje zobrazovaného objektu.

Nezadáte-li souřadnice, bude objekt zobrazen na aktuálních souřadnicích znakového kurzoru, který je na počátku programu v levém horním rohu s při výpisu čísel a textů příslušně stěhuje tak, aby byl vždy za posledním zapsaným znakem.

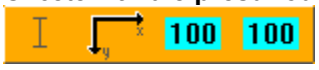
Pokud by se měl text vypisovat mimo obrazovku vpravo, aktuální souřadnice se automaticky přesunou na začátek dalšího řádku.

Zobraz současné datum a čas tak, že levý horní roh bude na souřadnicích ukazatele myši.

Potřebujete-li zjistit polohu znakového kurzoru, zadejte prvek souřadnice následovaný prvkem ***Kurzor pro výpis***.

Ulož aktuální souřadnice kurzoru pro výpis do proměnné *xy*.

Chcete-li **trvale přesunout znakový kurzor**, nastavte souřadnice prvku ***Souřadnice zobrazení***.



Souřadnice znakového kurzoru nastav na x100 y100.

Pozn.: Kromě příkazu přiřazení můžeme souřadnice znakového kurzoru upravit i pomocí příkazů ***Zvětši*** a ***Zmenši***.

Pozn.: Chcete-li vytisknout velikost některé souřadnice, musíte ji nejprve převést na číslo.

Vytiskni aktuální souřadnice Baltíka v hranatých závorkách

2) Hodnota

Hodnota, kterou chcete zobrazit, může být **číslo**, **datum a čas**, **řetězec**, řada **předmětů** nebo **soubor**.

Nezadáte-li **formát výpisu**, použije se standardní formát výpisu podle typu zadané hodnoty.

Zobraz A

Zobraz současné datum a čas

Pozn.: Nezadáte-li k zobrazení žádnou hodnotu, smaže se zadaný počet znaků podle formátu.

3) Formát

Požadovaný formát se zadává prvkem ***Formát*** následovaným textovým řetězcem s vlastním popisem formátu.

Abyste si nemuseli pamatovat význam jednotlivých "šifer", otevře Baltík po vložení prvku ***Formát*** **dialogové okno *Formát***, v němž jednoduše zadáte všechny potřebné parametry a na základě zadaných parametrů pak Baltík vygeneruje příslušný řetězec sám.

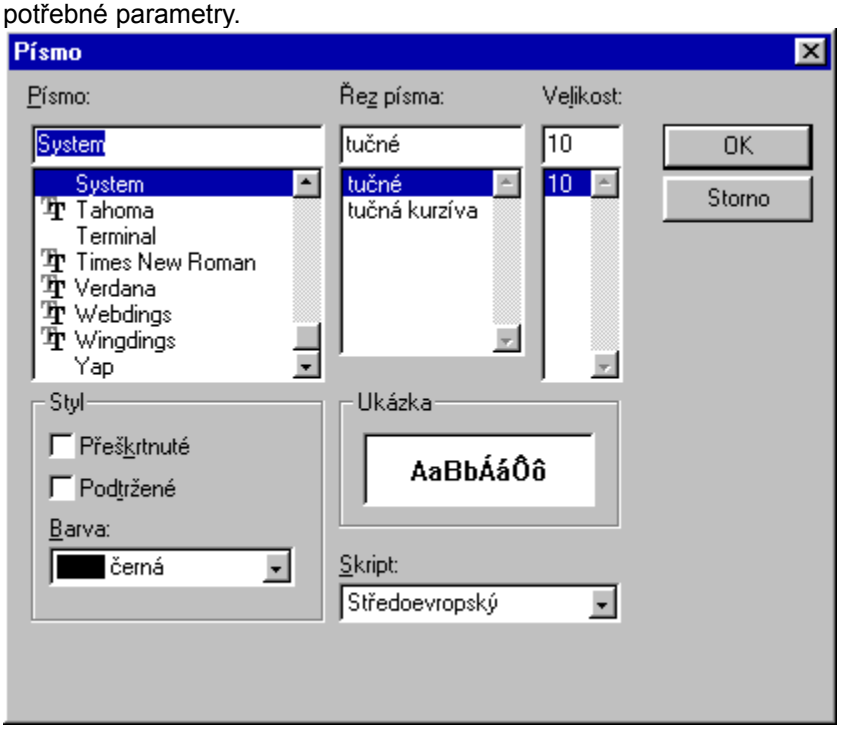
Protože toto okno se používá i při **nastavování formátu** při výstupu do textového řetězce, je jeho použití probráno v samostatné sekci.

Pozn.: Budete-li chtít ovlivnit vzdálenost mezi zobrazovanými předměty, musíte vložit prvek ***Písmo*** a nastavit dostatečnou velikost znakové sady.

4) Písmo

Pomocí prvku ***Písmo*** můžete zadat, jakým druhem, stylem, velikostí a barvou písma se má daný text vypisovat.

Po vložení nového prvku ***Písmo*** do programu se otevře dialogové okno výběru písma, v němž zadáte potřebné parametry.



Vlastní písmo (jen při výpisu čísla)

Vypisujete-li na obrazovku čísla, můžete místo číslic použít předměty. V takovém případě za prvek ***Písmo*** zadáte **předmět,** který se má použít jako nula. Baltík pak předpokládá, že za ním postupně následují předměty pro další číslice, předmět pro mezeru, znaménko mínus, desetinnou čárku, oddělovač času a oddělovač data. Celá posloupnost předmětů v bance tedy může vypadat např. takto:

Následuje-li za předmětem **číslo,** definuje vzdálenost jednotlivých předmětů. Této možnosti využijete např. tehdy, když si **nakreslíte** své vlastní číslice, které budou užší, než jeden předmět:

Zobraz *A*, jako nulu použij nulu z banky č. *2*, vzdálenost čísel zuž na 25 bodů.

5) Barva písma

Při určování písma můžete také zadat **barvu** popředí (tj. barvu písma) a pozadí (podkladu). Nastavování barev se podrobně věnuje sekce **Zadávání barvy.**

6) Průhlednost

Na konci příkazu zobrazení můžete zadat **průhlednost,** s níž má být výpis zobrazen.

Není-li průhlednost zadána, (chybí prvek ***Průhlednost***) nebo je zadána hodnota 0 (***Průhlednost 0***), bude výpis zobrazen neprůhledně.

Je-li zadána průhlednost bez čísla (***Průhlednost***) nebo s číslem **1** (***Průhlednost 1***), bude výpis zobrazen průhledně.

Bude-li zadána průhlednost s číslem větším (např. ***Průhlednost 2***), bude výpis zobrazen průhledně a animovaně, tzn. předchozí výpis bude nejdříve smazán.

Pro porovnání významu tohoto parametru si zkuste zadat a spustit následující program a různými hodnotami parametru průhlednosti (na obrázku je nastaven na 2, tj. na animované zobrazení):

Vypiš animovaně aktuální čas na pozici kurzoru myši

Formátování výstupu

Pro výstup čísel nabízí Baltík rozsáhlé možnosti ovlivňování jeho formátu. Umožňuje nám čísla tisknout či ukládat do textových řetězců v nejrůznějších podobách. K zadávání formátu slouží prvek ***Formát:***

Formát

V této sekci postupně probereme následující témata:

1) Vložení formátovacího příkazu do programu

Chceme-li ovlivnit formát, v němž vystupují data **na obrazovku** nebo v němž je ukládáme **do textového řetězce** či **do souboru**, vložíme za vystupující číslo prvek ***Formát***.

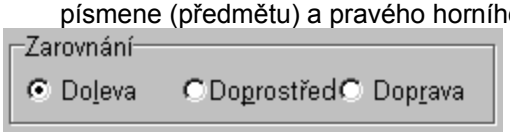
Po vložení prvku ***Formát*** do programu se otevře stejnojmenné dialogové okno, v němž je pro každý druh vystupujících dat vyhrazena karta, na níž můžete nastavit požadované parametry výpisu.

Po zadání požadovaných parametrů ze za prvek ***Formát*** vloží řetězcový literál s definicí zadaného formátu. Při následných úpravách můžete buď přímo editovat (upravovat) tento literál (musíte ovšem znát způsob kódování formátu), nebo můžete požádat o úpravu prvku ***Formát***. Tím znovu otevřete dialogové okno ***Formát***, v němž můžete nastavit všechny parametry podle nových požadavků.

2) Zarovnání výstupu na obrazovku

Všechny karty umožňují nastavit zarovnání vystupujícího textu či předmětů. Toto nastavení se však uplatní **pouze při výstupu na obrazovku**. Pak udává, která část vystupujícího řetězce bude umístěna na zadaných souřadnicích:

- Ü při zarovnání **vlevo** se na zadanou souřadnici umístí levý horní roh prvního písmene (předmětu),
- Ü při zarovnání na **vpravo** se na zadanou souřadnici umístí pravý horní roh posledního písmene (předmětu),
- Ü při zarovnání **na střed** se na zadanou souřadnici umístí střed spojnice levého horního rohu prvního písmene (předmětu) a pravého horního rohu posledního písmene (předmětu).



3) Výstup textu

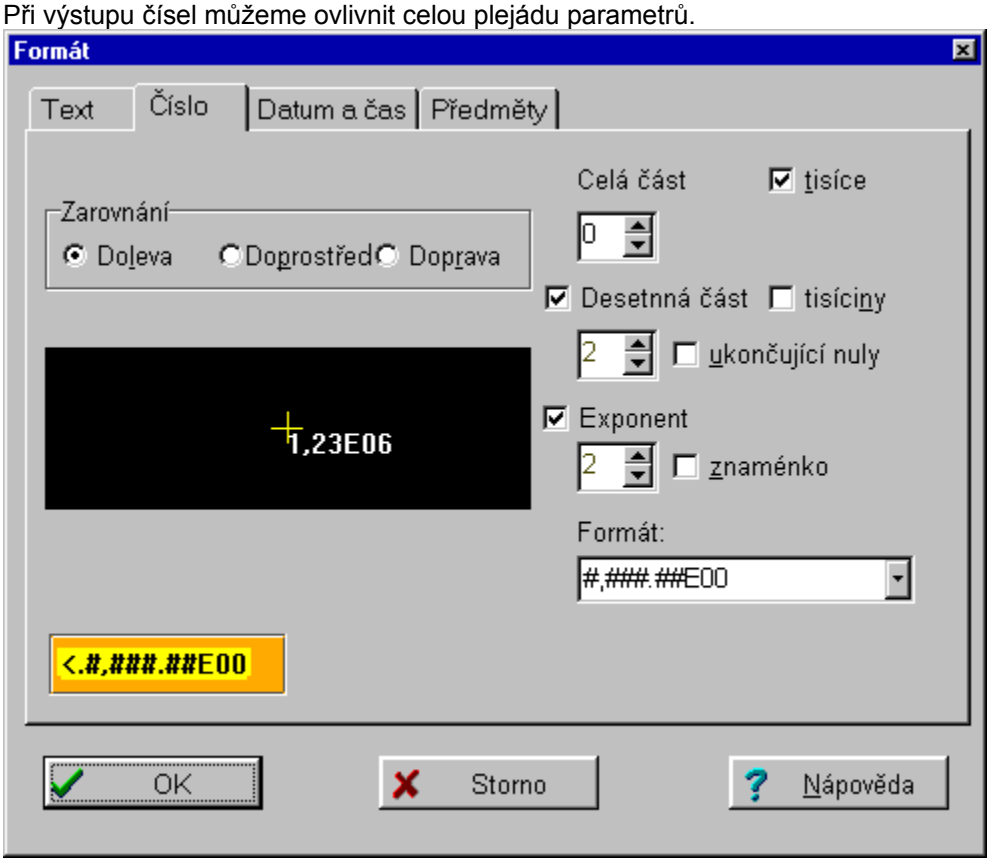
Zadání textového výstupu má smysl pouze pro čísla menší nebo rovna **255**. Místo nich totiž Baltík vypíše znak s příslušným kódem.

Daleko užitečnější je tato karta při výstupu textů na obrazovku, protože s její pomocí můžete nastavit

zarovnání vystupujícího textu, o němž jsme mluvili před chvílí.

4) Výstup čísel

Při výstupu čísel můžeme ovlivnit celou plejádu parametrů.



Celá část

V poli ***Celá část*** nastavujeme nejmenší počet čísel, které bude mít vystupující číslo před desetinnou čárkou. Bude-li jich potřebovat méně, doplní se chybějící číslice nulami.

Zadáme-li proto **3**, vytiskne se např. číslo **12** jako **012**.

Nechceme-li, aby se za chybějící číslice doplňovaly nuly, musím zadat buď **1** nebo **0** (pak se u čísel menších než jedna nezobrazí před desetinnou čárkou nic).

Tisíce

Zaškrtnutím pole ***Tisíce*** požádáme Baltíka, aby v celé části odděloval trojice číslic znakem, který je v operačním systému nastaven jako oddělovač tisíců.

Desetinná část

Zaškrtnutím pole ***Desetinná část*** aktivujeme číselné pole, v němž nastavujeme největší přípustný počet číslic za oddělovačem desetinné části a dvě zaškrtávací políčka, ovlivňující vzhled desetinné části.

Závěrečné nuly

Zaškrtnutím pole ***Závěrečné nuly*** požádáme Baltíka, aby desetinná část měla vždy tolik cifer, kolik je zadáno v poli ***Desetinná část***. Pokud by vystačila s menším počtem, doplní se číslo na konci nulami.

Tisíciny

Zaškrtnutím pole ***Tisíciny*** požádáme Baltíka, aby v desetinné části odděloval trojice číslic znakem, který je v operačním systému nastaven jako oddělovač tisíců.

Exponent

Zaškrtnutím pole ***Exponent*** aktivujeme číselné pole, v němž nastavujeme největší přípustný počet číslic v exponentu. Zároveň zadáváme požadavek na výstup čísla v exponenciálním tvaru, v němž se zvlášť udává tzv. **mantisa**, z níž se dozvíme, z jakých číslic se číslo skládá, a zvlášť **exponent**, který nám prozradí, jak je číslo velké - přesněji kterou mocninou deseti musím vynásobit mantisu, abychom obdrželi požadované číslo.

Příklady:

1,23E+02	1,23 * 10^2	123
12,3E+01	12,3 * 10^1	123
0,123E+3	0,123*10^3	123

Znaménko

Zaškrtnutím pole ***Znaménko*** požadujeme, aby se v exponentu uvádělo znaménko i před kladnými exponenty. není-li toto pole zaškrtnuto, bude se znaménko vypisovat pouze před záporný exponent.

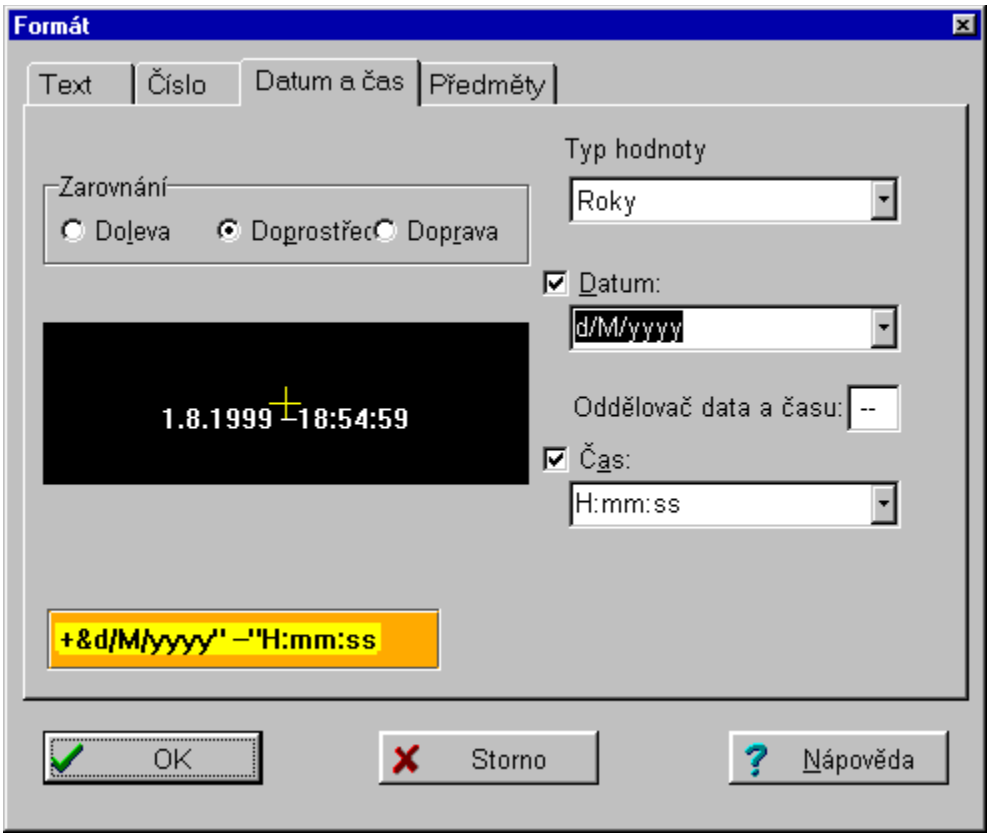
Formát

V poli formát je vypsán zadaný formát v zakódované podobě. V rozbalovacím seznamu máte navíc několik nejčastěji používaných formátů, které mohou zkušenější uživatelé zadat přímo aniž by museli vyplňovat jednotlivá pole. Ta se podle zadaného formátu nastaví sama.

Tip: Zadáte-li v poli ***Formát*** místo desetinné tečky hvězdičku, bude se desetinná tečka vypisovat pouze pro čísla, která mají desetinnou část, kdežto pro celá čísla se vypisovat nebude.

Datum a čas

Formát výstupu data a/nebo času můžete ovlivnit prostřednictvím řady parametrů. Na rozdíl od zadávání formátu čísel je však musíte zadávat ručně prostřednictvím formátovacích řetězců.



Typ hodnoty

V rozbalovacím seznamu ***Typ hodnoty*** zadáte nejvyšší jednotku, v níž bude údaj udán.

Zadáte-li např. při výstupu dnešního data jako typ hodnoty hodiny, vyšší jednotky se ve výstupu neobjeví a na místě hodin se objeví počet hodin uplynuvších od půlnoci z 30. na 31.12. roku 0. Chcete-li znát počet hodin uplynuvší od půlnoci ze včerejška na dnešek, musíte mít zadánu vyšší jednotku, než hodiny.

Datum

Zaškrtnutím políčka ***Datum*** aktivujete vstupní pole, v němž můžete zadávat formát data podle následujícího klíče:

yyyy	čtyřciferný letopočet,
yy	poslední dvě číslice letopočtu,
M	číslo udávající pořadí měsíce v roce,
MM	pořadí měsíce v roce zobrazované vždy pomocí dvou číslic (před číslicemi 1-9 se píše nula),
MMM	pořadí měsíce v roce římskými číslicemi,
MMMM	měsíc slovy,
MMMMM	měsíc třípísmennou zkratkou,
d	pořadí dne v měsíci,
dd	pořadí dne v měsíci zobrazované vždy pomocí dvou číslic (před číslicemi 1-9 se píše nula),

Čas

Zaškrtnutím políčka ***Čas*** aktivujete vstupní pole, v němž můžete zadávat formát času podle následujícího klíče:

h	hodiny při 12hodinové denní periodě (16 hodin se zobrazí jako 4 hodiny),
hh	hodiny při 12hodinové periodě zobrazované vždy pomocí dvou číslic (před číslicemi 1-9 se píše nula),
H	hodiny při 24hodinové periodě,
HH	hodiny při 24hodinové periodě zobrazované vždy pomocí dvou číslic (před číslicemi 1-9 se píše nula),
m	minuty,
mm	minuty zobrazované vždy pomocí dvou číslic (před číslicemi 1-9 se píše nula),
s	sekundy,
ss	zobrazované vždy pomocí dvou číslic (před číslicemi 1-9 se píše nula),
t	desetiny sekundy
tt	setiny sekundy
ttt	tisíciny sekundy
a	zkratka označující při 12hodinové periodě hodin, zda se jedná o dopoledne nebo dopoledne (označení dopoledne a odpoledne se přebírají z operačního systému).

Oddělovač data a času

Požadujete-li současný výstup data i času, aktivuje se pole ***Oddělovač data a času***, do nějž můžete zadat textový řetězec, který se na výstupu vloží mezi označení data a času.

Další vkládané znaky

Chcete-li kamkoliv do vystupujícího řetězce vložit nějaké další znaky, musíte je vložit do formátovacího řetězce příslušného údaje (data nebo času) na příslušné místo uzavřené do uvozovek.

Budeme-li chtít např. výpis aktuálního času doplnit slovním popisem zobrazované jednotky, může formátovací řetězec pro čas vypadat (všimněte si, že mezi uvozovky se musí vkládat i potřebné mezery):

h" hodin, "m" minut, "s" sekund"

Předmět

Poslední kartu použijete při kreslení předmětů na obrazovku. Chcete-li zobrazit číslo jako předmět, můžete za toto číslo přidat určení formátu specifikující, že se jedná o předmět.

Zobraz číslo 2 jako předmět - zobrazí zeď.

Vstup dat z klávesnice

Vstup znaků z klávesnice

Základním prvkem pro vstup dat je prvek ***Přečti číslo nebo řetězec z klávesnice***.

Po zadání tohoto příkazu se v Baltíkově prostoru okénko, do kterého můžete zadat číslo nebo řetězec. Poté, co zadáte požadovaná data, okénko z obrazovky zmizí.

Příkaz se chová jako funkce, která vrací zadanou hodnotu. Tuto hodnotu můžete buď přiřadit do proměnné, nebo ji použít jako parametr nějakého jiného příkazu.

Pozn.: Když je tento příkaz použit na místě čísla a místo čísla je zadán řetězec, který není možné převést na číslo, vrátí tento příkaz místo zadaného řetězce číslo **0**.

Příkaz ***Přečti číslo nebo řetězec z klávesnice*** nabízí programátoru rozsáhlé možnosti se dvěma možnými podobami syntaxe.

Ü Budeme-li chtít **zadávat čísla nebo obyčejné texty,** použijeme příkaz ve tvaru:

Klávesnice [***Převod na řetězec***] ***(*** <Počáteční_obsah> <Souřadnice> <Barva_písma> <Barva_pozadí> <Šířka_okénka> <Poloha_kurzoru> ***)***

Ü Budeme-li chtít zadat **název souboru** s možnostmi, které nabízí **standardní okno pro otevírání souborů,** použijeme příkaz ve tvaru:

Klávesnice Soubor (<složka> <maska> <jméno> <přípona>***)***

Pozn.: Všechny parametry jsou nepovinné.

Čtení standardních dat

Okénko se zobrazí před Baltíkem, bude černé s bílým řetězcem "William", 7 znaků široké, kurzor bude blikat před druhým písmenem "i".

Pozn.: S výjimkou parametrů <Šířka_okénka> a <Poloha_kurzoru> nezáleží na vzájemném pořadí jednotlivých parametrů.

Převod na řetězec

Použijete-li za prvkem ***Přečti číslo nebo řetězec z klávesnice*** prvek ***Převod na řetězec***, bude funkce vracet **řetězec** (jinými slovy nebude zadaný řetězec konvertovat na číslo), ať už je funkce použita kdekoliv.

Zobraz řetězec zadaný z klávesnice

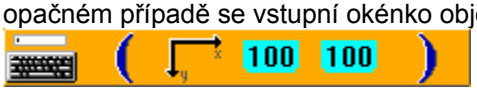
Počáteční obsah

Bude-li některý parametr **řetězec**, bude tento řetězec zobrazen v okénku jako původní hodnota. V opačném případě bude vstupní okénko na počátku prázdné.

Čti data z klávesnice, původní obsah: "123"

Souřadnice

Zadáte-li jako jeden z parametrů **souřadnice**, zobrazí se vstupní okénko na těchto souřadnicích. V opačném případě se vstupní okénko objeví uprostřed obrazovky



Čti data z klávesnice na souřadnicích x100y100

Barva písma a pozadí

Baltík umožňuje **zadat barvu** písma i podkladovou barvu vstupního pole. Nezadáte-li barvu **písma**, bud **černé**, nezadáte-li barvu **pozadí**, bude **šedé**.

Čti data z klávesnice, barva písma azurová

Čti data z klávesnice, barva písma bílá, barva pozadí černá

Šířka okénka

Číslo,, které zadáte jako parametr, určuje šířku okénka, která ovlivňuje, kolik znaků může uživatel do okénka vložit. Nebude-li zadána, nastaví se **15**.

Čti data z klávesnice do okénka pro max. 50 znaků

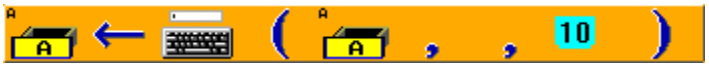
Pozn.: Je-li zadaný **počáteční obsah** delší než zadaná šířka, okénko se rozšíří.

Poloha kurzoru

Případné **druhé** **číslo** určuje, před který znakem bude blikat kurzor. Aby mělo toto nastavení smysl, musíte samozřejmě zadat také **počáteční obsah.**

Okénko se zobrazí před Baltíkem, bude černé s bílým řetězcem "William", 7 znaků široké, kurzor bude blikat před druhým písmenem "i".

Pozn.: Chcete-li zadat pozici kurzoru, musíte zároveň zadat i šířku pole. Nechce-li se vám zadávat šířka pole, musíte tento parametr zadat alespoň prázdný prostřednictvím dodatečné čárky



Zadej nový obsah proměnné *A* přičemž kurzor umísti na počátku před 10. znak jejího původního zobrazeného obsahu.

Při zadání pozice kurzoru již není možno dodržet konvenci, že při otevření okna má být původní text

vybrán. Pokud se vám tato konvence nehodí, můžete se počátečního vybrání textu zbavit právě tím, že zadáte úvodní polohu kurzoru. Přitom můžete využít toho, že

zadáte-li počáteční pozici kurzoru
větší než je počet znaků počátečního obsahu,
umístí se kurzor za poslední znak.

Pokud polohu kurzoru nezadáte, bude celý text označen a vstup se bude chovat podle konvencí Windows.

Čtení a zpracování jednotlivých znaků

Vstup dat z klávesnice

Baltík může načítat nejen celé texty či čísla, ale je schopen reagovat i na jednotlivé stisky kláves nebo tlačítek myši.

Pozn.: Neřekneme-li jinak, budeme v následujícím textu pod pojmem **stisk klávesy** rozumět jak stisk některé klávesy na klávesnici, tak případný stisk tlačítka myši.

V této pasáži si postupně ukážeme:

- **Vyprázdnění fronty kláves**
- **Čtení klávesy s čekáním**
- **Čtení klávesy bez čekání**
- **Zákaz přerušení programu stiskem Esc**
- **Vypnutí vstupu z klávesnice a tlačítek myši**
- **Zpracování znaků**
  - **Zjištění zadaného znaku**
  - **Zjištění kódu zadaného znaku**
  - **Převod kódu na odpovídající znak**
  - **Zjištění kódu klávesy**

Vyprázdni frontu kláves

Některé klávesy zpracovává počítač rychleji, některé pomaleji. Je totiž rozdíl, má-li počítač v reakci na stisk klávesy pouze vypsat znak na obrazovku nebo provést nějakou složitější akci. Při běhu programu proto často nastávají situace, že uživatel stiskne další klávesu dříve, než počítač zpracuje klávesu předchozí.

Aby uživatel nemusel myslet na to, kterou klávesu počítač zpracovává rychle a kterou pomalu, staví se v počítači jednotlivé stisknuté klávesy do fronty, v níž čekají, až si jich program všimne.

Z fronty lze převzít následující informace:

- Ü **znak**, který byl zadán,
- Ü **kód znaku**, který byl zadán (bývá někdy označován jako ASCII kód, i když to není přesné),
- Ü kódové číslo klávesy, která byla stisknuta (tzv. **scan-kód**),
- Ü **klávesu z banky kláves** odpovídající zadanému znaku.

Často se stane, že potřebujete zapomenout na všechny předchozí stisknuté klávesy a zaručit, že program bude reagovat až na klávesu, která byla stisknuta po nějakém definovaném okamžiku. Tento úkol řeší prvek ***Vyprázdni frontu kláves***, který frontu kláves vyčistí.

Čti klávesu nebo tlačítko myši - čekej na stisk
Čti klávesu s čekáním

Příkaz předá další informaci čekající ve frontě kláves. Nebylo-li od posledního **vyprázdnění fronty kláves** ještě nic stisknuto, program se zastaví a čeká, až stisknete nějakou klávesu nebo tlačítko myši a

vložíte tím další informaci do fronty.

Nekonečněkrát opakuj: { Přečti klávesu s čekáním, vyčaruj před Baltíkem odpovídající předmět z banky kláves, zobraz otevírací hranatou závorku, zadaný znak, středník, kód zadaného znaku, středník, scan-kód stisknuté klávesy, zavírací hranatou závorku a mezeru.}

Pozn.: Čísla jsou v ukázce převáděna na řetězce proto, abychom vystačili s jediným příkazem tisku.

Předchozí prográmek očekává vaše stisky kláves a tlačítek myši a zobrazuje po každém stisku vyžádanou sadu informací.

Pozn.: Pokud za příkazem ***Čti klávesu s čekáním*** zadáte číselný parametr, bude příkaz čekat na stisk klávesy nejvýše zadaný počet milisekund. Poté program pokračuje dál, jako by nepřečetl nic.

Čti klávesu nebo tlačítko myši - nečekej na stisk
Čti klávesu bez čekání

Příkaz ***Čti klávesu bez čekání*** použijete tehdy, když nepotřebujete, aby váš program čekal na stisk další klávesy a zajímá vás jen, jestli ve frontě čeká nějaká dosud neošetřená klávesa.

Program se po tomto příkazu nezastaví. Pouze se podívá do fronty kláves, zda tam nějaká nečeká a pokud tam nějakou najde, zapamatuje si o ní vše potřebné stejně, jako dříve popsaný příkaz pro čtení klávesy s čekáním.

To, že je při testu byla fronta kláves **prázdná**, poznáte podle **nulového scan-kódu**.

Abyste viděli rozdíl mezi čtením klávesy bez čekání a s čekáním, zkuste zadat tento prográmek, který s jednosekundovou periodou nahlíží do fronty kláves a vypisuje na obrazovku, co ve frontě našel. Zkuste napsat rychle za sebou několik kláves a podívejte se, jak je bude Baltík postupně zpracovávat.

Zákaz přerušení stiskem Esc

Zakaž ukončení klávesou *Esc*

Po tomto příkazu již nebude možné ukončit program stiskem klávesy ***Esc*** a bude naopak možno číst a zpracovávat klávesu ***Esc*** příkazy pro čtení z klávesnice.

Obnov možnost ukončení programu stiskem *Esc.*

Prvek ***Esc*** můžete použít také samotný jako **číslo**. Toto číslo bude mít hodnotu aktuálního nastavení možnosti ukončení klávesou ***Esc*** a použijete je např. tehdy, když chcete po provedení určité části programu obnovit nastavení, které bylo předtím.

Zapamatuj si aktuální nastavení, zakaž přerušení stiskem *Esc*, zavolej pomocníka *Program* a obnov původní nastavení možnosti přerušení stiskem *Esc*

Vypnutí vstupu z klávesnice a tlačítek myši

Vypni vstup z klávesnice a tlačítek myši

Po použití tohoto příkazu Baltík přestane číst klávesy a stisky tlačítek myši v příkazech pro čtení. Program bude klávesy i tlačítka myši ignorovat. Toho můžete využít např. tehdy, pokud chcete, aby se v určitém úseku programu neměnila fronta událostí z klávesnice.

Zapni vstup z klávesnice a tlačítek myši

Použitím tohoto prvku s číslem nula můžete přijímání klávesových a tlačítkových vstupů opět zapnout.

Zpracování přečtených znaků

Jak jsme již uvedli v pasážích o ***vyprázdnění fronty kláves,*** znaky mohou vystupovat v několika podobách. Podle toho, který druh informace vás v danou chvíli zajímá, použijete odpovídající příkaz.

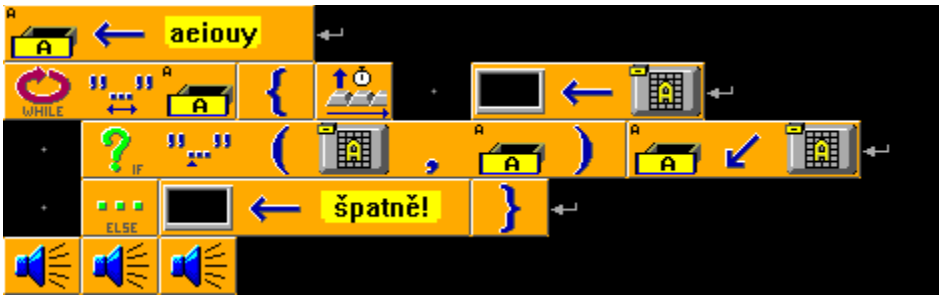
Přečtením se zapamatovaná informace nevymaže. Můžete se na ni proto ptát opakovaně do příštího zadání některého z příkazů:

- Ü ***Čti klávesu s čekáním***
- Ü ***Čti klávesu bez čekání***
- Ü ***Vyprázdni frontu kláves***

Podívejme se nyní na jednotlivé zjistitelné informace podrobněji.

Přečtený znak

Prvek ***Přečtený znak*** vrací poslední znak přečtený z fronty kláves. Tento znak můžete v programu použít jako jednoznakový řetězec.



Vyzkoušej uživatele ze znalosti samohlásek
Program požádá uživatele, aby napsal všechny samohlásky. Každý přečtený znak vypíše na obrazovku. Je-li znak samohláska, odebere ji z proměnné ***A***, v níž udržuje seznam dosud nezadaných samohlásek. Není-li to samohláska, pronese varovné hlášení. Po zadání všech samohlásek zatroubí fanfáru.

Znak se zadaným kódem

Inverzní funkcí je funkce, které předáte číslo a ona vrátí znak, jehož kódem je dané číslo. Jedinou omezující podmínkou je, že číslo musí být menší nebo rovno **255**.

Vytiskni kódovou tabulku písmem *Arial Narrow CE*.
Program vytiskne do Baltíkova prostoru kódovou tabulku, v níž vždy vykreslí znak a za následující rovnítko vypíše jeho kód.

Pozn.: Pro úsporu místa program používá znakovou sadu ***Arrial Narrow***. Pokud ji na svém počítači nemáte, nahraďte ji jinou sadou. Budete ale asi muset zmenšit velikost znaků.

Jako relativně praktickou ukázku využití převodu znaků na jejich kódy a zpět se podívejte na následující program.

Šifruj zadávaný text tak, že místo zadaného znaku vytiskneš znak, jehož kód vznikne operací XOR kódu právě zadaného znaku s kódem předchozího zadaného znaku

Program je realizován jako nekonečný cyklus, který bude ukončen stiskem klávesy ***Enter***. Tělo cyklu je naprogramováno následovně:

1. Počká na zadání znaku a ověří, zda se nemá cyklus ukončit.
2. Provede operaci XOR (a XOR b = ~a&b | ~b&a) kódu právě zadaného znaku s kódem minulého znaku, který si pamatuje v proměnné ***A*** (ta je na počátku nulová).
3. Z výsledku ponechá pouze 8 spodních bitů (& 255), protože znaky jsou kódovány pouze 8 bity.

4. K výslednému kódu připočte 32 (kód mezery), aby tento kód reprezentoval tištitelný znak (kódy menší než 32 jsou přiřazeny řídícím znakům, jako konec řádku, přechod na novou stránku, tabulátor, pípnutí apod.).
5. Takto získaný kód převede na znak, který vytiskne na obrazovku.
6. Zapamatuje si kód naposledy zadaného znaku v proměnné ***A***.

Pozn.: Závěrečné přičítání čísla 32 je zařazeno pouze proto, aby výsledné znaky byly tisknutelné na obrazovce. Dokud budete zadávat znaky ze spodní poloviny kódové tabulky, tj. písmena anglické abecedy a běžné znaky vyskytující se na klávesnici, nepřekročí zašifrovaný kód 255 a výsledek bude snadno dešifrovatelný. Nebudete-li na závěr přičítat 32, můžete šifrovat všechny znaky.

Pozn.: Zašifrovaný text dešifrujete téměř stejně, jako jste jej zašifrovali: nejprve odečtete 32 (pokud jste je přičítali), pak provedete XOR s kódem předchozího dešifrovaného znaku a nakonec ponecháte pouze spodních 8 bitů.

Kód zadaného znaku

Kód zadaného znaku závisí na použité jazykové mutaci programu ošetřujícího klávesnici. Baltík ví, jakou klávesnici máte nastavenu a ze stisknuté klávesy si dokáže odvodit, který znak jste do počítače poslali.

Kód zadaného znaku závisí na použité **kódové tabulce,** Ta se však mezi jednotlivými jazykovými mutacemi liší. Jiná kódová tabulka se používá v západní Evropě, jiná ve střední a východní Evropě a jiná např. v Jugoslávii (ta má dokonce svoji vlastní tabulku).

Pozn.: Přehled znaků, které zná vámi používaná kódová tabulka zjistíte zadáním povelu ***Možnosti → Tabulka kódů znaků***.

Byl-li přečten stisk tlačítka myši, pak tento příkaz vrátí následující kódy:

1 při stisku levého tlačítka myši,

2 při stisku pravého tlačítka myši a

4 při stisku prostředního tlačítka myši, resp. kolečka u myši s kolečkem.

POZOR! Tyto kódy neurčují jednoznačně, že bylo stisknuto tlačítko myši, protože stejné kódy dostanete i po stisku klávesových kombinací ***Ctrl+A***, ***Ctrl+B*** a ***Ctrl+D***.

Scan kód

Vedle kódu znaku lze v programu také zjistit kódové číslo stisknuté klávesy. Toto číslo je pro danou klávesu na všech počítačích stejné a pamatuje si je sama klávesnice.

Scan-kódy jsou výhodné zejména při ovládání her, protože toto ovládání má být pokud možno nezávislé na konvencích dané jazykové mutace ovladače klávesnice.

Scan kód zabírá normálně **7** bitů (hodnoty **0** až **127**). Osmý bit je vyhrazen pro informaci o tom, zda klávesa byla stisknuta či puštěna.

Baltík navíc používá i devátý bit. Ten nese informaci o tom, zda stisknutá klávesa patří mezi klávesy přidané na klávesnici v roce 1986 a umožňuje tak např. rozlišit, zda byla šipka nahoru zadána z numerického nebo kurzorového pole (spodních 8 bitů scan-kódu je u obou kláves shodných).

Byl-li přečten stisk tlačítka myši, pak tento příkaz vrátí následující kódy (spodních 8 bitů je nulových):

256 při stisku levého tlačítka myši,

512 při stisku prostředního tlačítka myši, resp. kolečka u myši s kolečkem a

768 při stisku pravého tlačítka myši.

Myš

Ve vašich programech můžete používat myš.

Můžete zjišťovat,
- kde se nyní nachází ukazatel myši,
- které tlačítko bylo naposledy stisknuto,
- kde bylo stisknuto,
- byl-li při tom kurzor myši nad danou oblastí či animovaným předmětem,
- jestli je tlačítko právě drženo,
- je-li nyní kurzor myši nad definovanou oblastí či animovaným předmětem

S ukazatelem myši můžete manipulovat. Můžete
- schovat kurzor a opět jej ukázat,
- ovládat z programu jeho polohu.

K zjišťování stavu myši a jejím ovládání slouží následující prvky:

1) Které tlačítko bylo stisknuto

V podmínkách můžete použít **prvek symbolizující tlačítko** myši (levé, prostřední, pravé, některé, žádné) ke zjištění, zda bylo příslušné tlačítko stisknuto a přečteno.

V příkazu větvení pak takovou podmínku můžete použít např. takto:

Čti klávesu s čekáním. Jestliže bylo stisknuto levé tlačítko myši, popojdi

Program čeká na stisk klávesy nebo tlačítka myši. Pokud stisknete levé tlačítko myši, Baltík popojde, jinak neudělá nic.

Stejně jako každá klávesa má i každé tlačítko myši své číslo. Toto číslo vidíte buď ve spodním stavovém řádku, máte-li aktuální ukazatel myši nad prvkem levé (prostřední nebo pravé) tlačítko myši v seznamu příkazů, nebo se toto číslo dozvíte použitím Převodu klávesy na číslo:

Číslo levého tlačítka myši, tj. 76

Číslo tlačítka myši také vracejí funkce číslo stisknuté klávesy a číslo stisknutého tlačítka myši, pokud bylo toto tlačítko stisknuto.

2) Je tlačítko ještě drženo?

Chcete-li vědět, zda je tlačítko myši právě drženo, použijte v **podmínce** následující zápis:

Levé tlačítko myši drženo

Tato podmínka je splněna, je-li v dané chvíli stisknuto a drženo levé tlačítko myši.

3) Souřadnice myši

Při zjišťování souřadnic myši máte několik možností.

Částečné souřadnice

Můžete např. získat souřadnici ***x*** nebo ***y***, na které je právě ukazatel myši:

x **myši**

y **myši**

Chcete-li zjistit jednu souřadnici (x nebo y), na které bylo stisknuto tlačítko myši, použijte tyto prvky společně s daným tlačítkem:

x myši při stisku levého tlačítka

Tento výraz vrátí hodnotu souřadnice x, na které bylo naposledy stisknuto levé tlačítko myši.

Kompletní souřadnice

Další možností je zjištění celých souřadnic ukazatele myši nebo zjištění souřadnic, na kterých bylo stisknuto některé tlačítko myši:

Současné souřadnice ukazatele myši

Souřadnice posledního stisku levého tlačítka myši

Souřadnice y myši v okamžiku posledního stisku prostředního tlačítka myši

Souřadnice x myši v okamžiku posledního stisku pravého tlačítka myši

Souřadnice y myši v okamžiku posledního stisku libovolného tlačítka myši

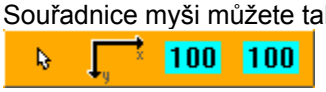
Tyto prvky můžete použít všude tam, kde se používají **souřadnice.**.

Následuje-li za souřadnicemi stisku tlačítka myši **řetězec**, obsahující znak "u" (případně "U"), tedy **událost** "puštěna klávesa", budou výsledné souřadnice souřadnicemi uvolnění stisku tlačítka myši:

Souřadnice y myši v okamžiku posledního puštění libovolného tlačítka myši

4) Nastavení souřadnic myši

Souřadnice myši můžete také nastavit:



Nastav souřadnice ukazatele myši na x100y100

5) Myš v oblasti nebo v animovaném předmětu

V **podmínkách** můžete také sledovat, zda bylo stisknuto (a příkazem ***Čti klávesu*** přečteno) tlačítko myši v oblasti, zadané v **tabulce oblastí**:

Levé tlačítko stisknuto v oblasti 2
Tato podmínka je splněna, bylo-li poslední přečtenou klávesou levé tlačítko myši, a zároveň bylo toto tlačítko stisknuto nad oblastí č. 2 z tabulky oblastí.

Žádné tlačítko nebylo stisknuto, ukazatel myši v oblasti 2
Tato podmínka je splněna, pokud poslední přečtenou klávesou nebylo žádné z tlačítek myši, a zároveň se ukazatel myši právě nachází nad oblastí č. 2 z tabulky oblastí.

Ukazatel myši v oblasti 2
Tato podmínka je splněna, pokud se ukazatel myši právě nachází nad oblastí č. 2. Přitom vůbec nezáleží na stavu tlačítek myši.

Místo oblasti můžete použít také **animovaný předmět**:

Levé tlačítko stisknuto v animovaném předmětu č. 4
Tato podmínka je splněna, pokud poslední přečtenou klávesou bylo levé tlačítko myši, a zároveň bylo toto tlačítko stisknuto nad animovaným předmětem č. 4.

6) Schování a zobrazení ukazatele myši

Baltík umožňuje měnit viditelnost myši zadáním čísla za prvek ukazatele. Pro snadnější použití jsou v bance celočíselných **konstant** umístěny konstanty pro pravdivou (TRUE) a nepravdivou (FALSE) logickou hodnotu.

Schovej ukazatel myši

Ukaž ukazatel myši

Pozn.: Na začátku programu je ukazatel myši viditelný.

Události

Práce s událostmi patří k vyšší škole programátorského umění. Události vám umožňují reagovat nejen na zadání znaku, ale na každý stisk či puštění klávesy či tlačítka myši.

Chcete-li hlídat jednotlivé události, vložte Příkaz pro sledování události vypadá tak, že za prvkem čtení klávesy následuje řetězec, který obsahuje písmena (můžete je nahradit příslušnými řetězcovými **konstantami**), označující události, které chcete hlídat (na velikosti písmen nezáleží):

D klávesa byla stisknuta (**D**own),

U klávesa byla puštěna (**U**p)

C byl přečten znak (**C**haracter)

T vypršel čas časovače (**T**imer).

Čekej na událost *Klávesa puštěna*

Při čtení události může řetězec obsahovat více druhů událostí, pak budou čteny všechny tyto události. Pokud např. jako řetězec zadáte ***DUC***, budou čteny všechny klávesové události.

Druh události, která byla právě přečtena se ukládá do řetězcové proměnné ***Událost***:

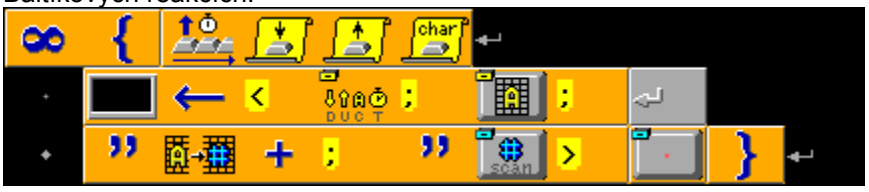
Po přečtení události jsou naplněny zvláštní proměnné:

Ü ***U*** a ***D*** naplní proměnné ***Klávesa*** a ***Scan-kód***,

Ü ***C*** naplňuje proměnnou ***Znak***.

Ostatní zvláštní proměnné přečtení události vynuluje.

Následující prográmek by vám mohl dát představu o možném naprogramování hlídání událostí a o Baltíkových reakcích.



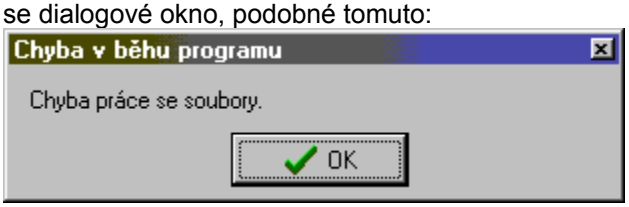
V nekonečné smyčce čekej na klávesové události a vypisuj obsahy zvláštních proměnných

Předchozí program vždy počká na následující klávesovou událost a pak ve špičatých závorkách postupně vypíše typ přečtené události (proměnná ***Událost***), přečtený znak (proměnná ***Znak***), kód přečteného znaku a scan-kód stisknuté klávesy. Poté ještě vytiskne příslušnou klávesu z banky kláves.

Chyby při běhu programu

Chyba při běhu programu

Při provádění programu může dojít k chybě, kterou programátor nemůže předvídat. K takovým chybám patří např. chyby práce se soubory (soubor pro čtení neexistuje, do souboru pro zápis nelze zapisovat apod.), chyby při převodu řetězce na číslo, chyby při výpočtech. Pokud k takovéto chybě dojde, zobrazí se dialogové okno, podobné tomuto:



Pokud však chcete, aby se takovéto okno nezobrazovalo, použijte prvek ***Chyba při běhu programu*** s číslem nula (FALSE):

Nezobrazuj chyby při běhu programu

Od této chvíle nebudou chyby při běhu programu zobrazovány a nebudou ukončovat běh programu. Při každém příkazu, který by mohl takovouto chybu způsobit, se naplní vnitřní proměnná ***Chyba*** číslem chyby. Obsah této proměnné zjistíte použitím prvku ***Chyba při běhu programu***:

Do proměnné ***A*** přiřaď číslo chyby.

V proměnné ***A*** bude číslo chyby posledního kritického příkazu, tj. příkazu, který mohl tuto chybu vyvolat. Mezi ně patří:

- Příkazy pro práci se soubory a složkami,
- příkazy pro práci s multimédii a obrázky,
- převod řetězce na číslo,
- dělení, modulo, operace s reálnými čísly.

V následující tabulce vidíte seznam všech možných hodnot proměnné ***Chyba***:

0	OK
-1	Chyba při převodu řetězce na číslo
-2	Dělení nulou
-3	Nesprávná matematická operace (např. log(-3))
-4	Přetečení
-101	Nelze vytvořit soubor
-102	Nelze otevřít soubor
-103	Soubor neexistuje
-104	Nesprávné číslo souboru z tabulky
-105	Nelze zapisovat do souboru
-106	Nelze zjistit vlastnosti souboru
-107	Dočasná složka nebo soubor jsou nedostupné

Pozn.: Číslo chyby je vždy záporné

Chcete-li zobrazování chyb opět zapnout, použijte prvek ***Chyba při běhu programu*** samotný nebo s

číslem 1 (TRUE):

Zobrazuj chyby při běhu programu

Příklad:

Nezobrazuj chyby, vytvoř soubor "j:text.txt", vypiš na obrazovku číslo chyby
Pokud váš počítač neobsahuje disk **J**, zobrazí se číslo chyby -101, nelze vytvořit soubor.

Převod na číslo

Nějaký předmět

Většina objektů, s nimiž baltík pracuje, má svoji číselnou reprezentaci. K zjištění hodnot této reprezentace slouží prvek ***Převod na číslo.*** Pomocí tohoto prvku můžeme zjistit:

- Ü **číslo předmětu** před Baltíkem nebo na zadaných souřadnicích,
- Ü **číslo stisknuté klávesu nebo tlačítka myši,**
- Ü **rychlost Baltíka**
- Ü **hodnotu čísla zadaného jako řetězec.**

1) Číslo předmětu

Je-li parametrem **předmět**, vrátí příkaz ***Převod na číslo*** číslo daného předmětu

Vrátí číslo azurové pětky - 6. předmět v bance .b02 => 2 x 1000 + 6 = 2006.

Toto číslo můžete použít stejně jako ostatní čísla v příkazech, které to dovolují:

Je-li obsah proměnné *A* větší než číslo předmětu před Baltíkem, nastav čarování bez obláčku.

Pokud za prvkem ***Převod na číslo*** nenásleduje žádný parametr, vrací číslo předmětu, který je před Baltíkem:

Číslo předmětu před Baltíkem

Pokud za prvkem ***Převod na číslo*** zapíšete **souřadnice**, vrátí číslo předmětu na těchto souřadnicích:

Číslo předmětu v levém horním rohu obrazovky

Pozn.: Číslo **prázdného políčka** je **0**, číslo **okraje** je **151**.

Jako předmět mohou vystupovat i klávesy z banky kláves nebo tlačítka myši. Funkce pak vrátí jejich pořadové číslo v bance kláves:

Vrať číslo klávesy *F* (= 6)

Vrať číslo levého tlačítka myši (=76 - viz dále)

Pozn.: Pokud potřebujete znát skutečné číslo předmětu - klávesy (např. pro čarování na obrazovku), použijte prvek ***Převod na číslo*** zdvojeně:

Vrať číslo předmětu - klávesy *F* (=201006)

2) Číslo klávesy nebo tlačítka myši

Vrať číslo stisknuté klávesy v bance kláves nebo tlačítka myši

Tento výraz vrací číslo stisknuté klávesy v nebo tlačítka myši, přečteného příkazem pro čtení kláves.

Tlačítka myši sice v bance kláves své obrázky nemají, mají však přiřazena svá čísla:

- Ü levé tlačítko = 76
- Ü prostřední tlačítko = 77
- Ü pravé tlačítko = 78

Nebyla-li přečtena žádná klávesa nebo byla přečtena klávesa, která není v bance kláves, funkce vrací **0**.

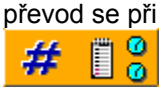
5) Rychlost Baltíka

Bude-li parametrem prvek ***Rychlost***, funkce ***Převod na číslo*** vrátí rychlost, na kterou je Baltík nastaven, tj. číslo v rozmezí **0** do **10**. Číslo **10** se přitom vrací v případě, kdy byla nastavena rychlost nekonečno.:

Je-li nastavená rychlost větší než 1, nastav rychlost o jedničku menší

6) Převod data a času na číslo

Pokud za prvkem Převod na číslo použijete datum a čas, bude převedeno na číslo (počet dní). Tento převod se při operacích s čísly provádí automaticky.



Převeď současné datum a čas a na číslo

7) Převod řetězce na číslo

Toto téma již bylo probráno v pasáži o řetězcích.

Pokročilejší práce s předměty

Převod předmětu na číslo

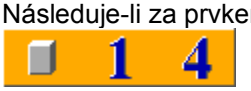
V režimu Pokročilý nabízí Baltík daleko větší možnosti práce s předměty. Můžete nejen čarovat předem dané předměty, ale můžete čarovat i předměty určené až v průběhu programu, zjišťovat, který předmět je na zadaném poli vyčarován a řadu další akcí.

Klíčovým rpvkem pro pokročilou práci s předměty ja prvek ***Nějaký*** předmět, který nabízí dva druhy služeb:

- Ü **převod čísla na předmět**, který pak můžete vyčarovat nebo testovat jeho výskyt použít v **podmínkách**
- Ü **zjištění předmětu** před Baltíkem nebo na zadaných **souřadnicích.**

1) Převod čísla na předmět

Následuje-li za prvkem ***Nějaký předmět*** nějaký **číselný výraz**, převede se jeho hodnota na **předmět**.



Předmět číslo 14 (zvonek)

2) Pravidla pro přiřazení čísla předmětu

Číslo předmětu se skládá z čísla banky a čísla předmětu v bance zkombinovaných podle vzorce:

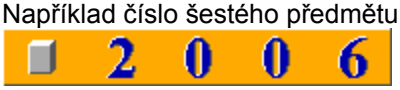
<Číslo_předmětu> = <Číslo_banky> x 1000 + <Číslo_předmětu v bance>

Pozn.: Číslo prázdného políčka je **0**, číslo okraje je **151**.

Banky jsou číslovány takto:

- Ü banky **.b00** až **.b99** mají čísla **0** až **99**,
- Ü banky **.c00** až **.c99** mají čísla **100** až **199**.

Například číslo šestého předmětu z banky **.b02** je **2×1000 + 6**, tj. **2006**.



Předmět číslo 6 z banky .b02
Tento zápis můžete použít stejně jako samotný předmět číslo **2006**:

Získáte-li výše popsaným převodem z čísla předmět, můžete jej vyčarovat před Baltíka nebo na zadané **souřadnice**:

Čaruj předmět 1102 (zlatou kouli) na souřadnice X7 Y4 (přibližně doprostřed scény)

Takto získaný předmět můžete použít v **podmínkách** jako obyčejný předmět, chcete-li zjistit, zda je daný

předmět před Baltíkem nebo na zadaných **souřadnicích**:

Když je před Baltíkem předmět číslo 13 (třešně), čaruj strom

2) Zjištění čísla předmětu před Baltíkem nebo na zadaných souřadnicích

Není-li za prvkem ***Nějaký předmět*** číslo, funkce ani číselný výraz v závorkách, funguje prvek jako funkce vracející číslo předmětu před Baltíkem, případně na **souřadnicích**, zadaných za tímto prvkem.

Předmět před Baltíkem

Předmět, který je na souřadnicích X5 Y6

Průhlednost pro pokročilé

Průhlednost - využití v programech

1) Nastavení průhlednosti souborů z tabulky

Chcete-li aby se obrázek z **tabulky souborů** zobrazoval průhledně, musíte u daného názvu souboru v tabulce zapnout průhlednost. Průhlednost nastavíte:

- Ü Klepnutím pravým tlačítkem na názvu souboru v tabulce (tj. vyvoláním místní nabídky) a zadáním volby ***Průhledný*** nebo
- Ü vložením dvojtečky (:) před název souboru.

Průhledná barva je pak dána barvou bodu v **levém dolním** rohu obrázku. **Celý dolní řádek** obrázku se nebude zobrazovat.

2) Nastavení průhledné barvy

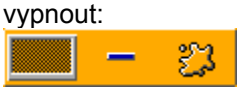
Chcete-li, aby se při průhledném zobrazování použila jako průhledná **jiná barva**, než je určeno v daném předmětu nebo obrázku, případně chcete-li zobrazit s použitím průhledné barvy obrázek **původně neprůhledný**, použijte prvek ***Průhlednost*** s prvky s **barvou:**

Průhledná barva = Baltíkova námořnická modrá

Po použití tohoto příkazu budou všechny předměty, banky, scény a obrázky vykresleny tak, že v místech, kde byla Baltíkova námořnická modrá, bude vidět pozadí.

Pozn.: Nastavená průhledná barva se používá pouze tehdy, je-li zapnuto průhledné zobrazování (viz Přepínání průhledného a neprůhledného zobrazování).

Chcete-li, aby se předměty (obrázky) vykreslovaly opět se **svou vlastní** průhlednou barvou a neprůhledné předměty (obrázky) neprůhledně, můžete pomocí prvku ***Mínus*** nastavení průhledné barvy vypnout:



Vypni nastavení průhledné barvy

Po použití tohoto příkazu se bude k zobrazování předmětů, bank, scén a obrázků používat jejich vlastní průhledná barva.

Nezobrazovat

Prvek ***Nezobrazovat*** slouží k zákazu a opětovnému povolení zobrazování.
Použijete-li tento příkaz samotný nebo s **číslem** různým od nuly, nebude se od této chvíle nic zobrazovat (Baltíkův pohyb, vyčarované předměty, animace, načtené scény, banky, obrázky). Všechny zobrazovací příkazy se budou provádět skrytě a se svým zveřejněním budou čekat až na povolení zobrazování.

Pozn.: Příkazem ***Nezobrazovat*** obrazovku nezhasnete. Pouze zmrazíte její dosavadní podobu do chvíle, kdy zobrazování opět povolíte.

Použitím příkazu ***Nezobrazovat*** s **číslem** **0**, zobrazíte najednou všechna skrytě prováděná zobrazení a od této chvíle se bude opět všechno zobrazovat průběžně.

Tyto příkazy s výhodou použijete ve chvíli, když potřebujete, aby se všechno, co nakreslíte, objevilo najednou.

Příklad 1:

Představte si, že chcete posunout autíčko. Nejprve je musíte smazat ze současné pozice (nemohou tam být dvě současně) a pak autíčko znovu vykreslit na novou pozici. Jenomže, jakmile autíčko smažete, chvilku tam není nic, než se objeví nové. Autíčko proto při svém posunu blikne.

Uvedený problém můžete řešit tak, že:

1. příkazem ***Nezobrazovat*** vypnete zobrazování (jinými slovy: zmrazíte její současný stav),
2. smažete původní autíčko (na obrazovce se nic neděje),
3. dáte zobrazit nové (na obrazovce se stále nic neděje) a
4. příkazem ***Nezobrazovat 0*** zapnete zobrazování - na obrazovce se objeví všechny provedené změny.

Příklad 2:

Při pohybu obrázku složeného z více předmětů (více animací) může vzniknout nežádoucí chvění obrázku, protože jednotlivé animace jsou prováděny jedna po druhé. Pokud však před každým krokem použijete příkaz Nezobrazovat a po každém kroku obrazovku Obnovíte, pohyb se stane plynulejším.

Matematické funkce

Matematické funkce počítají ze zadaného **parametru** výslednou hodnotu.

V této sekci postupně probereme následující matematické funkce:

Absolutní hodnota

Absolutní hodnota čísla je vzdálenost tohoto čísla od nuly na číselné ose. Absolutní hodnota kladného čísla je toto číslo, absolutní hodnota záporného čísla je číslo opačné (tedy kladné). Absolutní hodnota nuly je nula

Absolutní hodnota čísla (-3.8) = 3.8

Obecná mocnina a odmocnina

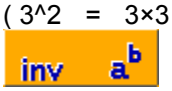
Pro celočíselné **mocniny** platí tvrzení, že **n-tá** mocnina parametru **z** je číslo, které získáme tak, že **n**-krát vzájemně vynásobíme parametr **z**.

Číslo **z** nazýváme **základ** mocniny a číslo **n** nazýváme **exponent.**

Třetí mocnina čísla 2
(2^3 = 2×2×2 = 8)

Není-li u mocniny uveden exponent, počítá se druhá mocnina.

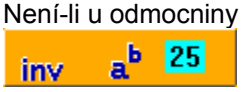
Druhá mocnina čísla 3
(3^2 = 3×3 = 9)



Inverzní funkcí k **n**-té mocnině je **n**-tá **odmocnina**, což je číslo, jehož **n**-tá mocnina je rovna parametru.

Třetí odmocnina čísla 8
(3√8 = 2 protože 2^3 = 8)

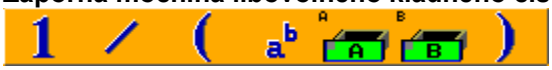
Není-li u odmocniny uveden exponent, počítá se druhá odmocnina.



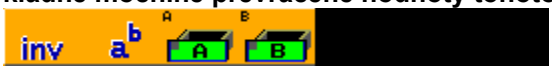
Druhá odmocnina čísla 25
(√25 = 5 protože 5^2 = 25)

Ve vyšších ročnících se dozvíte (nebo jste se již dozvěděli), že pojem mocniny lze rozšířit i na neceločíselné exponenty. Dosazením libovolných hodnot do použitých proměnných a vytisknutím výsledku na obrazovku můžete ověřit platnost následujících tvrzení.

Záporná mocnina libovolného kladného čísla (parametru) je rovna



kladné mocnině převrácené hodnoty tohoto čísla.



B-tá odmocnina libovolného kladného čísla

je rovna 1/B-té mocnině téhož čísla

Přirozený logaritmus

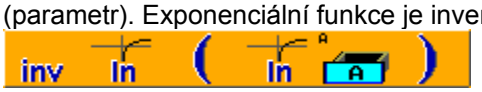
Přirozený logaritmus je číslo, na které bychom museli umocnit **Eulerovu konstantu**, abychom dostali původní číslo (parametr této funkce).

POZOR! Parametrem této funkce musí být kladné číslo.

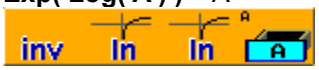
Přirozený logaritmus hodnoty (e*e) = 2

Exponenciální funkce

Exponenciální funkce vrací číslo, které získáme umocněním **Eulerovy konstanty** na dané číslo (parametr). Exponenciální funkce je inverzní funkce k funkci ***Přirozený logaritmus***.



Exp(Log(A)) = A



Předchozí výraz by bylo možno se stejným výsledkem použít i bez závorek.

Převody stupňů na radiány a zpět

Většina z nás je zvyklá vyjadřovat úhly ve stupních. Goniometrické funkce, o nichž pojednávají následující pasáže, naopak předpokládají vyjadřování velikosti úhlů v úhlových mírách - tzv. radiánech (značí se **rad**).

Velikost úhlu v radiánech je rovna délce oblouku kruhové výseče jednotkové kružnice.

Z předchozí věty můžeme odvodit, vztahy:

(x) ° = (x * π / 180) rad
(x) rad = (x * 180 / π) °

Sinus a arkussinus

Sinus je poměr délky odvěsny protilehlé k zadanému úhlu k délce přepony pravoúhlého trojúhelníka. Úhel se zadává v radiánech.

Pozn.: Hodnota funkce sinus je vždy v rozmezí **-1** až **1**.

Vypiš na obrazovku jak dlouhé prkno musím přistavit, aby nájezd na 0,5 m vysokou zídku nebyl strmější než 15°?

Arkussinus je funkce, která vrací úhel, jehož sinus jsme zadali jako parametr.

Pozn.: Hodnota funkce arkussinus je vždy v rozmezí **0** až **π/2**.

POZOR! Parametrem této funkce musí být číslo z intervalu v rozmezí -1 až 1

Vypiš pod jakým úhlem bude stoupat 3 m dlouhé prkno opřené o 1 m vysokou zeď?

Kosinus a arkuskosinus

Kosinus je poměr délky odvěsny přilehlé k zadanému úhlu k délce přepony pravoúhlého trojúhelníka. Úhel se zadává v radiánech.

Pozn.: Hodnota funkce sinus je vždy v rozmezí **-1** až **1**.

cos(π/3) = 0.5

Arkuskosinus je funkce, která vrací úhel, jehož kosinus jsme zadali jako parametr.

Zobraz
3 * arccos(0.5) = π

Tangens a arkustangens

Tangens úhlu je poměr délky protilehlé odvěsny ku délce odvěsny přilehlé k zadanému úhlu.

POZOR! Parametr funkce tangens **nesmí** být násobkem -π/2.

Spočti vzdálenost 100 m vysokého stožáru, který vidíme pod úhlem 5°.

Arkustangens je funkce, která vrací úhel, jehož tangens jsme zadali jako parametr.

Pozn.: Hodnota funkce arkustangens je vždy v rozmezí -π/2 až π/2.

Kolikaprocentní stoupání má silnice stoupající pod úhlem 15°?

Celá část

Celá část reálného čísla je část před desetinnou čárkou (při zápisu bez exponentu). Vznikne odtržením desetinné části.

Pozn.: Typ výsledku této funkce je celé číslo

Celá část (-3.2) = -3

Převod na reálné číslo

Tato funkce pouze mění typ výsledného výrazu na reálné číslo. Tato konverze je potřebná např. při dělení a pro formát **výpisu** čísla.

15 / 4 = 3 (celočíselné dělení)

Reálné číslo(15) / 4 = 3.75 (obyčejné dělení)

Generická funkce

Tato funkce očekává (na rozdíl od dříve popsaných matematických funkcí) dva číselné parametry:

- Ü první z nich určuje typ funkce,
- Ü druhý je jejím argumentem.

Tato funkce je vlastně generickou funkcí, která vyvolává z balíku připravených funkcí tu správnou na základě hodnoty prvního argumentu. V následujícím výkladu budou proto funkce z balíku seřazeny podle hodnoty argumentu, který je vyvolá:

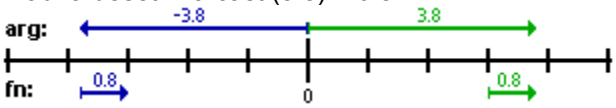
1 - Desetinná část (frac) - vrací desetinnou část argumentu:

Desetinná část (-3.8) = -0.8
Desetinná část (3.8) = 0.8

2 - Kladná desetinná část (posfrac) -
vrací desetinnou část, zbavenou znaménka:

fn 2 (− 3 . 8)

Kladná desetinná část (-3.8) = 0.8
Kladná desetinná část (3.8) = 0.8



3 - Vzdálenost k menšímu celému číslu -
vrací vzdálenost k nejbližšímu menšímu celému číslu

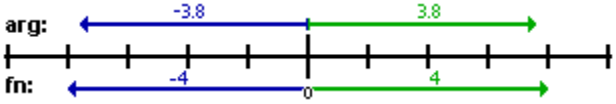
fn 3 (− 3 . 8)

Vzdálenost k menšímu celému číslu (-3.8) = 0.2
vzdálenost k menšímu celému číslu (3.8) = 0.8

4 - Zaokrouhlení (round) -
vrací nejbližší celé číslo.

fn 4 (− 3 . 8)

Zaokrouhleno (- 3.8) = - 4
Zaokrouhleno (3.8) = 4



Pozn.: V případě, že nelze rozhodnout (např. ***round(1,5)***), je výsledkem vyšší číslo (**2**).

Pozn.: Chcete-li zaokrouhlovat na určitý počet desetinných míst (např. na setiny nebo na tisíce), použijte nejdříve funkci **nastavení zaokrouhlení**. Chcete li zaokrouhlovat na určitý počet platných číslic, použijte nejdříve funkci **nastavení zaokrouhlení na platné číslice**.

5 - Nejblíže menší celé číslo (floor)

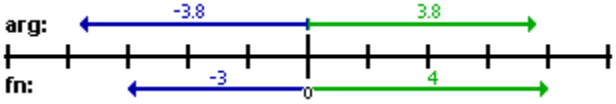
fn 5 (− 3 . 8)

Nejblíže menší celé číslo (- 3.8) = - 4
Nejblíže menší celé číslo (3.8) = 3

6 - Nejblíže větší celé číslo (ceiling)

```
fn 6 ( - 3 . 8 )
```

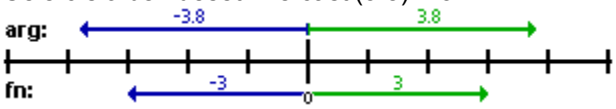
Nejblíže větší celé číslo (- 3.8) = - 3
Nejblíže větší celé číslo (3.8) = 4



7 - Celé číslo (int, trunc) - usekne desetinnou část

```
fn 7 ( - 3 . 8 )
```

Celé číslo bez desetinné části(- 3.8) = - 3
Celé číslo bez desetinné části(3.8) = 3



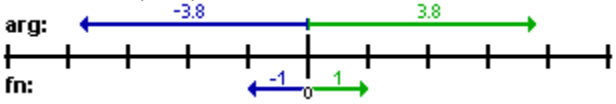
8 - Znaménko (sign)

Vrací:

- Ü **-1** pro záporný argument,
- Ü **0** pro nulu a
- Ü **1** pro kladný argument.

```
fn 8 ( - 3 . 8 )
```

Znaménko (-3.8) = -1



9 - Nastavení zaokrouhlování - Funkce nastaví zaokrouhlování na zadaný počet desetinných míst:

```
fn 9 ( 2 )
```

Nastav zaokrouhlování na 2 desetinná místa (na setiny)

Tímto příkazem nastavíte zaokrouhlování na setiny, takže potom bude mít výraz

Zaokrouhleno (1234.5678)

hodnotu 1234.57. Pokud jako argument nastavení zaokrouhlení použijete záporné číslo, nastaví se zaokrouhlování na (kladnou) mocninu deseti:

Nastavení zaokrouhlování na -3 desetinná místa (na tisíce)

Tímto příkazem nastavíte zaokrouhlování na tisíce, takže potom bude mít výraz

Zaokrouhleno (1234.5678)

hodnotu 1000.

Pozn.: Funkce nastavení zaokrouhlování vrací hodnotu minulého nastavení. Na začátku je zaokrouhlování nastaveno na 0 (zaokrouhlování na jednotky).

10 Zaokrouhlování na zadaný počet platných cifer - Funkce nastaví zaokrouhlování na zadaný počet platných cifer

Tato funkce nastaví zaokrouhlování pro funkci č. 4: **zaokrouhlení.** Parametr funkce určuje, kolik platných číslic bude obsahovat výsledné číslo funkce zaokrouhlování. Platná číslice je každá nenulová číslice nebo nula, za níž je alespoň jedna nenulová číslice. Např. číslo **103,456** zaokrouhleno na 4 platné číslice dává výsledek **103,5**.

Nastav zaokrouhlování na 2 platné číslice

Tímto příkazem nastavíte zaokrouhlování na 2 platné číslice, takže potom bude mít výraz

Zaokrouhleno (1234.5678)

hodnotu 1200.

Parametr může být jednoduchý výraz (**číslo**, **konstanta**, **proměnná**, **funkce**) nebo složený **výraz**, uzavřený v **závorkách.**

Soubory a složky

Práce s celými soubory/složkami Práce s obsahem souborů Práce s obsahem složek

Baltík umožňuje pracovat se soubory a složkami na disku. Vedle práce s daty uloženými v souboru nabízí i možnost práce s celými soubory a složkami. V následujících sekcích spolu postupně probereme tato témata:

Práce s celými soubory/složkami

Práce s obsahem souborů Práce s obsahem složek

Složka není nic jiného, než zvláštní typ souboru, jehož obsahem jsou informace o jiných souborech. Proto mají složky a "obyčejné" soubory řadu společných vlastností. V této sekci postupně probereme operace, které jsou společné "obyčejným" souborům a složkám. Budeme se věnovat následujícím tématům:

1) Název souboru/složky

V této verzi Baltíka musí názvy souborů vyhovovat konvenci označované zkratkou **8.3**. Musí tedy splňovat následující kritéria:

Ü Název souboru musí sestávat z nejvýše 8znakového **jména** a z nejvýše 3znakové **přípony**, které jsou od sebe odděleny tečkou.

Ü Název souboru smí obsahovat pouze **písmena** (včetně písmen s diakritikou), **číslice** a znaky **"!"** (vykřičník), **"#"** (mříž), **"~"** (vlnovka), **"^"** (stříška), **"$"** (dolar), **"&"** (et, and), **"@"** (šnek, zavináč), **","** (čárka), **"("**, **")"** (kulaté závorky), **"{"**, **"}"** (složené závorky)

Jsou-li v názvu souboru použity možnosti dlouhých názvů souborů, je třeba používat název zkrácený podle konvencí Windows.

Součástí názvu souboru může být i celá cesta. Pro každou složku na cestě platí výše uvedené konvence.

Za názvem souboru je možno uvést parametry popisující jeho vnitřní strukturu. Podrobnosti viz pasáž ***Datové soubory - záznamy, položky.***

2) Nastavení aktuálního souboru/složky

Baltík má vždy nastaven jeden soubor a jednu složku jako aktuální. K tomuto souboru/složce se pak můžete obracet přímo bez bližšího **určení**

Aktuální soubor/složku nastavíte pomocí prvku **=**:

Nastav soubor *prog.bpr* jako aktuální

Nastav složku *C:\WINDOWS* aktuální

Po provedení tohoto příkazu můžete používat místo zápisu

kratší zápis

Pozn.: Na začátku je aktuální soubor nastaven na soubor č. 1 z **tabulky souborů,** aktuální složka je nastavena na složku, ze které jste spustili program (POZOR! To nemusí být složka, v níž je program uložen!)

3) Určení souboru/složky

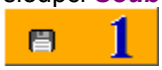
Úvodním prvkem příkazů pro práci se soubory je prvek ***Soubor***

Úvodním prvkem příkazů pro práci se složkami je prvek ***Složka***

Nechcete-li pracovat s **aktuálním souborem (složkou),** musíte v příkazu určit soubor (složku), s nímž chcete pracovat. Určení se vkládá za prvek ***Soubor***, resp. ***Složka***. Možné způsoby určení jsou popsány v následujících podkapitolách.

Číslo souboru v tabulce souborů

Nejedná-li se složku, ale pouze o "obyčejný" soubor a máte-li jeho název uveden v **tabulce souborů** ve sloupci ***Soubory***, můžete pro určení souboru použít číslo řádku s názvem souboru v tabulce.



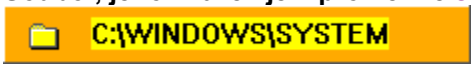
Soubor č. *1* z tabulky souborů, sloupce *Soubory*

POZOR! Abyste mohli použít určení souboru řádkem v tabulce souborů, nesmíte je doplnit určením cesty - viz dále pasáž ***Určení složky a pořadí ve složce***.

Řetězec s názvem souboru/složky

Soubor/složku můžete určit řetězcem s názvem určovaného souboru/složky. Součástí řetězce může, ale nemusí být popis cesty.

Soubor, jehož název je v proměnné *Jmeno*



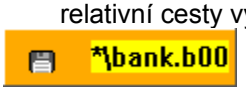
Složka *C:\WINDOWS\SYSTEM*

Pozn.: Pokud zadaná složka neexistuje, je výsledkem aktuální složka.

Není-li v názvu uvedena cesta, hledá se zadaný soubor v aktuální složce.

V zápisu cesty můžete použít následující konvence:

- Ü nezadáte-li v zápisu cesty název disku, použije se aktuální disk,
- Ü nezačíná-li zápis cesty obráceným lomítkem, je považován za popis relativní cesty vycházející z aktuální složky,
- Ü začíná-li zápis cesty symbolem "**.**" (tečka), je považován za popis relativní cesty vycházející z aktuální složky,
- Ü začíná-li zápis symbolem "**..**" (dvě tečky), je považování za popis relativní cesty vycházející z rodičovské složky aktuální složky,
- Ü začíná-li zápis symbolem "*****" (hvězdička následovaná obráceným lomítkem), je považování za popis relativní cesty vycházející ze složky, v níž se nachází spuštěný program,

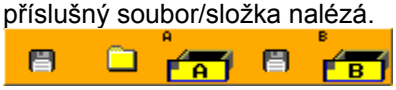


Soubor *bank.b00* ze složky spuštěného programu

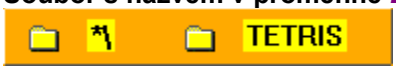
Složka, ze které byl program spuštěn

Určení složky a pořadí ve složce

Není-li v názvu souboru/složky specifikována **absolutní** cesta, můžete ji zadat pomocí složky, ve které se příslušný soubor/složka nalézá.



Soubor s názvem v proměnné *B* uložený ve složce s názvem v proměnné *A*.



Podsložka *TETRIS* složky, ze které byl program spuštěn

Zadáte-li určení složky, v níž se soubor/složka nachází, můžete se pak na její soubory/podsložky odvolávat nejenom názvem, ale také pořadím v seznamu souborů/podsložek v dané složce.

POZOR! Číslování souborů uvnitř složky je vnitřní záležitostí složky a nemusí respektovat (a většinou ani **nerespektuje**) **žádné řazení**.

Třetí soubor ve složce *C:\WINDOWS*

První podsložka druhé podsložky aktuální složky

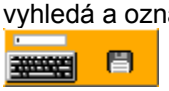
První podsložka složky *A:\IMPORT*

Možnost určit soubor/složku prostřednictvím jeho pořadí v seznamu souborů/podložek využijete zejména tehdy, budete-li chtít projít všechny soubory/podsložky dané složky.

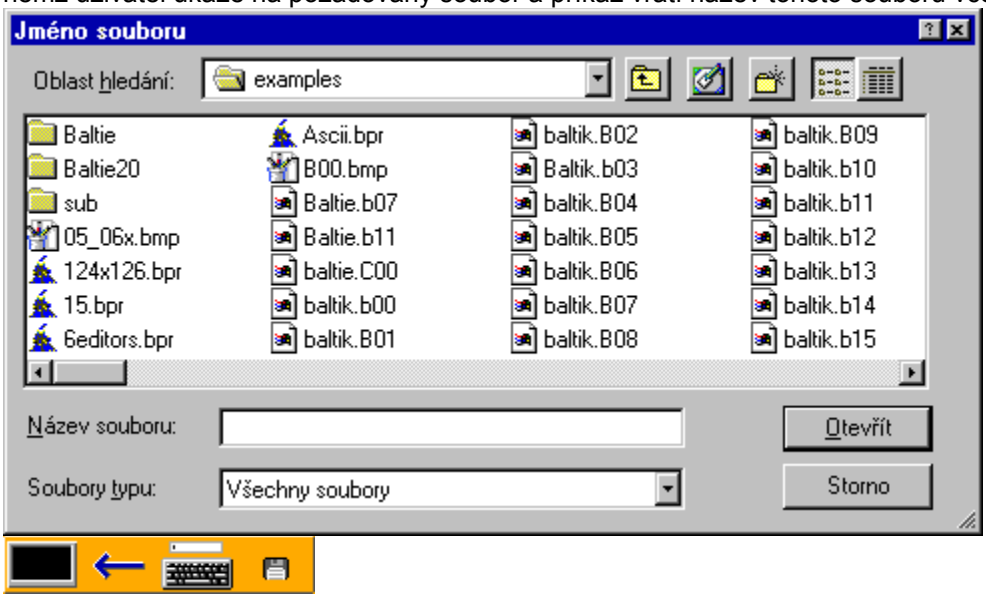
4)Vyhledání existujícího souboru

Často potřebujeme získat od uživatele název existujícího souboru, s nímž bude náš program pracovat. Pro současného programátora je nedůstojné nutit uživatele, aby psal název exitujícího souboru znak za znakem. Baltík nám naštěstí umožňuje nabídnout uživateli prostředky, pomocí nichž soubor na disku

vyhledá a označí.



Budete-li chtít, aby uživatel označil soubor, s nímž se bude dále pracovat, zadejte prvek ***Čtení z klávesnice*** následovaný prvkem ***Soubor***. Baltík v danou chvíli otevře dialogové okno ***Najít soubor***, v němž uživatel ukáže na požadovaný soubor a příkaz vrátí název tohoto souboru včetně úplné cesty.



Název souboru zadaného uživatelem vytiskni na obrazovku.

5) Zjištění názvu souboru/složky

Prvek ***Složka*** vrací řetězec s kompletním názvem (tj. s názvem včetně absolutní cesty) svého parametru.

Ulož do proměnné A1 kompletní název prvního souboru v aktuální složce

Na rozdíl od běžných řetězců se však tento řetězec nevypisuje na obrazovku automaticky, ale musíte jeho vypsání Baltíkovi přikázat.

Vytiskni na obrazovku kompletní název aktuální složky

Nepotřebujete-li kompletní název souboru, ale potřebujete-li naopak pouze název samotný (tj. bez určení cesty), musíte jej z kompletního názvu vypreparovat prostřednictvím řetězcových operací:

Vrať samotný název souboru, jehož kompletní název je v proměnné *A.*

Pozn.: Pokud zadaná složka neexistuje, je výsledkem prázdný řetězec.

6) Vytvoření souboru

Pokud **za určením** souboru/složky použijete pouze prvek ***Přiřaď***, vytvoří se nový, prázdný soubor/složka se zadaným jménem.

Vytvoř nový soubor *readme.now*

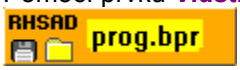
Vytvoř složku *C:\BALTIE\OLD*

Při zadávání vytvářeného souboru můžeme s výhodou využít okno ***Najít soubor,*** v němž nastavíte správnou složku a zadáte požadovaný název vytvářeného souboru.

Vytvoř soubor se jménem a umístěním zadaným z klávesnice.

7) Vlastnosti složky

Pomocí prvku ***Vlastnosti souboru/složky*** můžete zjišťovat a nastavovat vlastnosti souborů a složek.



Zobraz vlastnosti souboru *prog.bpr*

Zobraz vlastnosti složky *C:\BALTIE*

Výše uvedené příkazy zobrazí vlastnosti soubor ***prog.bpr*** resp. složky ***C:\BALTIE*** jako **řetězec**. Pokud soubor/složka danou vlastnost má, zobrazí se příslušné písmeno na své pozici. V opačném případě se na daném místě zobrazí tečka.

Jednotlivé vlastnosti jsou znázorněny následujícími písmeny:

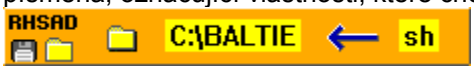
- **R** soubor/složka je určena jen pro čtení (Read only)
- **H** soubor/složka je skrytá (Hidden)
- **S** soubor/složka je systémová (System)
- **A** soubor/složka je archivní (Archive)
- **D** soubor je složkou (Directory)

Pozn.: Vlastnost *složka* (*D*) nelze měnit.

Zobrazí-li se např. řetězec **R..AD**, znamená to, že se jedná o složku určenou jen pro čtení, která není skrytá ani systémová a je archivní.

Nastavení všech vlastností

Chcete-li nastavit vlastnosti souboru/složky, použijte prvek ***Přiřaď*** následovaný řetězcem obsahujícím písmena, označující vlastnosti, které chcete nastavit. Na velikosti písmen a na jejich pořadí nezáleží.



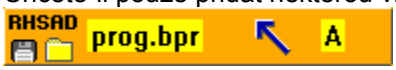
Nastav vlastnosti složky *C:\BALTIE* na *Shift*

Tento příkaz nastaví vlastnosti složky ***C:\BALTIE*** na ***systémová, skrytá***, takže její vlastnosti budou jako

řetězec vypadat:.***HS.D***.

Přidání vlastnosti

Chcete-li pouze přidat některou vlastnost a ostatní nechat netknuté, použijte prvek ***Přidej***:



K vlastnostem souboru *prog.bpr* přidej *A*

Provedením tohoto příkazu se k vlastnostem souboru ***prog.bpr*** přidá vlastnost ***archivní***, ostatní vlastnosti se nezmění. Bylo-li tedy před provedením příkazu nastavení vlastností ***.HS...***, změní se provedením příkazu na ***.HSA.***.

Odebrání vlastnosti

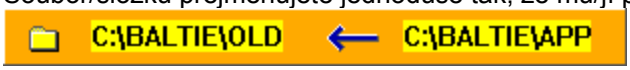
Z vlastností složky *C:\BALTIE* uber *aH*

Provedením tohoto příkazu se odstraní vlastnosti ***archivní, skrytá***. Nastavení ***.HSAD*** se provedením příkazu změní na ***..S.D***.

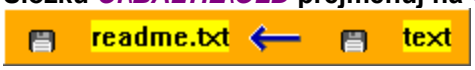
Byla-li odebíraná vlastnost před provedením příkazu nastavena, nic se nezmění.

8) Přejmenování souboru/složky

Soubor/složku přejmenujete jednoduše tak, že mu/jí přiřadíte řetězec s novým jménem:



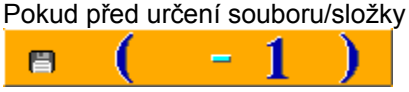
Složku *C:\BALTIE\OLD* přejmenuj na *C:\BALTIE\APP*



Soubor *readme.now* přejmenuj na *readme.txt*

9) Smazání souboru/složky

Pokud před určení souboru/složky vložíte prvek ***Mínus***, bude tento soubor/složka smazán.



Smaž soubor č. *1*

Smaž složku *H:\Pokusy*

Pozn.: Při mazání složky budou smazány také v všechny soubory a podsložky této složky.

POZOR! Mazané soubory se nepřesouvají do koše, ale nenávratně se odstraňují.

Pozn.: Baltík umožňuje zadat pouze skupiny souborů/složek, které se budou mazat Podrobnosti viz **Specifikace skupin souborů/složek**

10) Kopírování souboru/složky

Chcete-li obsah souboru/složky zkopírovat do jiného souboru/složky, přiřaďte jeden soubor/složku druhému:

Do souboru *config.sys* zkopíruj obsah souboru *config.old*

Pozn.: Pokud cílový soubor (tj. soubor, do nějž se kopíruje) neexistuje, vytvoří se. Existuje-li, Baltík se nejprve zeptá, jestli jej může přepsat.

Do složky *C:\BALTIE\APP* zkopíruj obsah složky *A:\IMPORT*
Tento příkaz přidá obsah složky ***A:\IMPORT*** včetně všech podsložek a jejich obsahů do složky ***C:\BALTIE\APP***. Existují-li ve složce ***C:\BALTIE\APP*** soubory nebo složky se stejnými názvy, budou přepsány. Ostatní složky a soubory zůstanou zachovány.

Chcete-li obsah cílové složky (v našem případě složky ***C:\BALTIE\APP***) **nahradit** obsahem zdrojové složky, tzn. smazat původní obsah cílové složky ***C:\BALTIE\APP***, použijte před označením cílové složky mínus:

Obsah složky *C:\BALTIE\APP* nahraď obsahem složky *A:\IMPORT*

Po provedení tohoto příkazu budou mít obě složky totožný obsah.

Pozn.: Baltík umožňuje zadat pouze skupiny souborů/složek, které se budou kopírovat. Podrobnosti viz **Specifikace skupin souborů/složek**

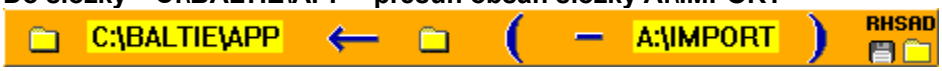
11) Přesun souboru/složky

Upravíte-li příkaz pro kopírování tak, že před určení zdrojového souboru/složky vložíte prvek ***Mínus*** a tuto dvojici prvků uzavřete do závorek, původní soubor/složka se smaže - půjde tedy o přesun:

Přesuň obsah souboru *C:\config.old* do souboru *C:\ARCHIV\config.old*

Po provedení tohoto příkazu soubor ***C:\config.old*** nebude existovat.

Do složky " C:\BALTIE\APP " přesuň obsah složky A:\IMPORT



Tento příkaz přesune obsah složky ***A:\IMPORT*** včetně všech podsložek a jejich obsahů do složky ***C:\BALTIE\APP***. Pokud ve složce ***C:\BALTIE\APP*** existují soubory nebo složky se stejnými názvy, budou přepsány. Ostatní složky a soubory zůstanou zachovány. Po provedení tohoto příkazu nebude složka A:\IMPORT existovat.

Chcete-li obsah cílové složky (v našem případě složky ***C:\BALTIE\APP***) **nahradit** obsahem zdrojové složky, tzn. smazat původní obsah cílové složky ***C:\BALTIE\APP***, použijte před označením cílové složky

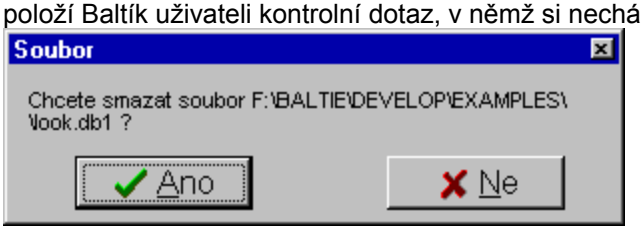
mínus:

Obsah složky *C:\BALTIE\APP* nahraď obsahem složky *A:\IMPORT* a složku *A:\IMPORT* smaž.

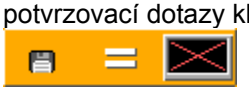
Pozn.: Baltík umožňuje zadat pouze skupiny souborů/složek, které se budou přesouvat. Podrobnosti viz **Specifikace skupin souborů/složek.**

12) Potlačení kontrolního dotazu

Kdykoliv, když při operacích se soubory hrozí nějaké poškození některého souboru (smazání, změna), položí Baltík uživateli kontrolní dotaz, v němž si nechá potvrdit souhlas uživatele s chystanou akcí.



V některých programech je kladení těchto kontrolních dotazů nevýhodné. Chcete-li je ve svém programu zakázat, zadejte prvek ***Soubor*** následovaným rovnítkem a prvkem ***Nezobrazuj***. Od této chvíle Baltík potvrzovací dotazy klást nebude.



Přestaň klást potvrzovací dotazy při operacích se soubory.

Kladení potvrzovacích dotazů opět zapnete zadáním prvku ***Soubor*** následovaným prvkem ***Obrazovka***. Prvek ***Obrazovka*** můžete nahradit prvkem ***Nezobrazuj*** následovaným číslem **0**.

Zapni znovu kladení potvrzovacích dotazů při operacích se soubory.

Zapni znovu kladení potvrzovacích dotazů při operacích se soubory.

Budete-li chtít vypnout, resp. zapnout kladení kontrolního dotazu pouze pro jeden příkaz, uveďte v něm prvek ***Nezobrazuj***, resp. prvek ***Obrazovka*** jako parametr.

Smaž soubor "prog.bpr", neptej se na potvrzení.

Práce s obsahem souborů

Práce s celými soubory/složkami Práce s obsahem složek

V této sekci postupně probereme následující témata:

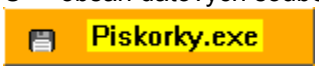
1) Spustitelné × datové soubory

Baltík rozděluje soubory do dvou základních kategorií:

- Ü spustitelné soubory, tj. soubory s příponou ***bat***, ***com***, ***exe*** a ***pif***, které obsahují kód programu, jenž je možno spustit,
- Ü datové soubory (ty ostatní), které obsahují údaje, jež lze pomocí vhodného programu interpretovat jako smysluplná data.

Na tom, do které kategorie soubor patří, závisí Baltíkova reakce na výskyt samotného souboru v programu:

- Ü spustitelné soubory Baltík rovnou spouští,
- Ü obsah datových souborů se snaží vypsat na obrazovku jako by se jednalo o textové soubory.



Spusť soubor *piskorky.exe* z aktuální složky

Vypiš obsah souboru *manual.txt* z aktuální složky na obrazovku

Spustitelné soubory jsou často schopny přebírat parametry z příkazového řádku. Tyto parametry je možno napsat za název spouštěného souboru:

Spusť program *command.com* a nech jej vypsat seznam všech souborů ve všech adresářích disku *C:* do souboru *C:\dir.txt*

2) Datové soubory - záznamy a položky

O datových souborech se předpokládá, že mají předem definovanou jednoduchou strukturu:

- Ü datové soubory jsou chápány jako posloupnost stejně strukturovaných **záznamů**,
- Ü každý záznam sestává ze stejné posloupnosti položek předem definovaných typů.

Pro popis typu jednotlivých položek se používají následující znaky (na velikosti písmen nezáleží):

B bajt - může nabývat hodnoty **0** až **255**

I krátké celé číslo (2 bajty) - může nabývat hodnoty **-32768** až **32767**
L dlouhé celé číslo (4 bajty)
R reálné číslo (6 bajtů)
S řetězec zakončený znakem NUL, tj. znakem s kódem **0**
T řetězec zakončený koncem řádku (jako konec řádku se používá dvojice znaků s kódy **13** a **10**)
W slovo = krátké celé číslo bez znaménka (2 bajty) - může nabývat hodnoty **0** až **65535**

Nejedná-li se o klasický textový soubor, musíte za názvem souboru (a následující mezerou) uvést posloupnost znaků označujících typ jednotlivých položek tvořících jeden záznam.

Příklady:

Rozhodneme-li se ukládat do souboru **hodnoty.max** posloupnost reálných čísel získaných jako výsledky měření, uvedeme jeho název ve tvaru **hodnoty.max R**.

Chceme-li ukládat výsledky nějakého závodu do souboru **výsledky** (bez přípony) tak, že vždy nejprve uložíme číslo závodníka (krátké celé číslo bez znaménka), pak jméno závodníka (řetězec ukončený znakem NUL), hodnoty jeho tří pokusů (reálná čísla) a nakonec jeho pořadí v celkové klasifikaci (krátké celé číslo bez znaménka), uvedeme jeho název ve tvaru **výsledky WSRRRW**.

V prvním z uvedených příkladů je každý záznam tvořen jedinou položkou - reálným číslem, v druhém příkladu je tvořen celkem šesti položkami - kladným číslem, řetězcem, třemi reálnými čísly a dalším kladným číslem.

Pozn.: Nezadáte-li typ jednotlivých položek záznamů, bude soubor považován za textový, tj. za soubor typu **T** (jednopoložkové záznamy tvořené textovými řetězci ukončenými koncem řádku).

3) Přístup k datovému souboru a jeho položkám

K datovému souboru můžete přistupovat přímo tak, že určíte záznam, se kterým chcete pracovat:

Pozn.: První záznam v souboru má číslo **1**.

Soubor č. *1*, záznam č. *2*
Je-li soubor č. 1 textový, bude tento zápis označovat druhý řetězec, tj. druhý řádek v tomto souboru.

Jsou-li záznamy souboru tvořeny několika položkami, můžete dalším parametrem určit pořadí položky, s níž chcete pracovat:

Soubor č. *1*, záznam č. *2*, položka č. *3*

Je-li položkou záznamu řetězec, můžete přistupovat dokonce k jeho jednotlivému znaku:

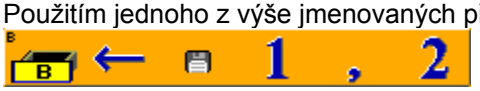
Soubor č. *1*, záznam č. *2*, položka č. *3*, znak č. *4*

Pozn.: Nemůže-li dojít ke spojení sousedních prvků, označujících indexy v souboru (např. používáte-li literály), nemusíte tyto prvky oddělovat čárkou:

Soubor č. *1*, záznam č. *2*, položka č. *3*, znak č. *4*

4) Čtení ze souboru

Použitím jednoho z výše jmenovaných přístupů k souboru můžete přímo používat hodnoty ze souboru:



Proměnné *B* přiřaď obsah *2.* záznamu souboru č. *1*.

5) Zápis hodnoty do souboru

Obdobným způsobem můžete do souboru přiřazovat:

3.* položce v *2.* záznamu souboru č. *1* přiřaď hodnotu proměnné *B

***3.* položku v *2.* záznamu souboru č. *1* zmenši o hodnotu proměnné *B*.**

***3.* položku v *2.* záznamu souboru č. *1* zvětši o hodnotu proměnné *B*.**

6) Zobrazení obsahu souboru

Jak jsme již několikrát řekli, vyskytne-li se v programu samotný prvek ***Soubor*** následovaný určením souboru, bude obsah tohoto souboru vypsán na obrazovku.

Zobraz soubor č. *1*

Budete-li chtít vypsat soubor s využitím jednoduchého (přesněji jednotného) formátování, v němž můžete nastavit souřadnice, druh písma, barvu, průhlednost či animovaný výstup, použijte příkaz **Zobraz.**

Zobraz obsah souboru *Pokus.txt* 12bodovým tučným písmem Arial.

Budete-li chtít nastavit zvláštní formátování pro každou položku v záznamu, musíte použít cyklus přes všechny záznamy, v jehož těle vytisknete řízeně každou položku jednotlivého záznamu.

Zobraz obsah souboru 1, přičemž první položka každého záznamu se bude tisknout jako text a druhá položka jako datum. Mají-li záznamy souboru více položek, ostatní se netisknou.

7) Tisk obsahu souboru na tiskárnu

Chcete-li soubor vytisknout, zadáte obdobný příkaz jako při jeho přiřazení na obrazovku, místo obrazovky použijete obrázek s konstantou "tiskárna".

Vytiskni obsah souboru "list.txt" na tiskárnu

8) Délka souboru

Chcete-li zjistit délku souboru, musíte si nejprve rozmyslet, zda vás zajímá počet bajtů, které data v souboru zabírají, nebo zda vás zajímá spíš počet záznamů, které soubor obsahuje.

Prvek ***Délka souboru v bajtech*** slouží ke zjištění velikosti souboru v bajtech,

Prvek ***Délka souboru v záznamech*** slouží naopak ke zjištění počtu záznamů, které soubor obsahuje.

Vrať délku souboru č. *1* v bajtech

Vrať počet záznamů v souboru č. *1*

Práce s obsahem složek

Práce s celými soubory/složkami Práce s obsahem souborů

V této sekci postupně probereme následující témata:

Specifikace skupin souborů a složek
Zjištění počtu souborů/složek ve složce a velikost složky

1) Specifikace skupin souborů a složek

Při mazání, kopírování a přesouvání složek a souborů ve složkách potřebujeme často nějak jednoduše určit množinu souborů, na které se bude zadávaná akce vztahovat.

Baltík nabízí dvě možnosti specifikace takovéto podmnožiny:

- Ü specifikací hodnot atributů (vlastností) souborů,
- Ü specifikací masky pro názvy souborů.

Specifikace skupiny souborů podle vlastnostmi

Nechcete-li smazat (přesunout, zkopírovat) všechny soubory a podsložky dané složky, můžete za určení mazané (přesouvané, kopírované) složky zadat vlastnosti souborů, které chcete smazat (přesunout, zkopírovat).

Požadované vlastnosti souborů se zadávají pomocí prvku ***Vlastnosti složky*** následovaného řetězcem, obsahujícím s označením **vlastností** prvků.

U každé vlastnosti můžete zadat, zda má být u vybíraných souborů nastavena či nikoliv. Požadavek na nastavení dané vlastnosti zadáváte znakem **+**, požadovanou nepřítomnost vlastnosti znakem **-** před příslušným písmenem vlastnosti.

Smaž ve složce *H:\Pokusy* všechny podsložky (vlastnost D) které nejsou určeny pouze pro čtení (-R).

Specifikace skupiny souborů vyhovujících dané masce

Často je potřeba aplikovat požadovanou operaci (mazání, kopírování, přesun) na řadu souborů, které mají v názvu nějaký charakteristický podřetězec, jenž je odlišuje od souborů, které mazat nechceme. Pro určení takovýchto souborů nabízí operační systém dva metaznaky (znaky, které zastupují jiné znaky):

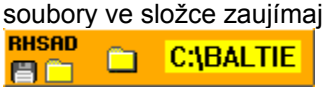
- Ü **otazník** zastupuje libovolný znak,
- Ü **hvězdička** zastupuje skupinu znaků od dané pozice až do konce dané části názvu - jména či přípony.

Chcete-li proto vybrat skupinu souborů, na něž budete danou operaci aplikovat, můžete na konec názvu složky přidat obrácené lomítko následované maskou popisující danou skupinu souborů.

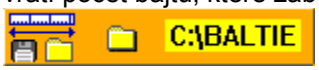
Všechny soubory s příponou *bmp* přesuň ze složky *..\OBR* do aktuální složky.

2) Velikost složky

Začnete-li se pídit po velikosti složky, musíte si nejprve rozmyslet, zda vás zajímá počet bajtů, které soubory ve složce zaujímají, nebo zda vás zajímá spíš počet souborů, které složka obsahuje.



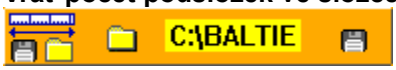
Vrať velikost složky *C:\BALTIE* v bajtech
vrátí počet bajtů, které zabírají soubory ve složce BALTIE.



Vrať počet souborů a složek ve složce *C:\BALTIE*
vrátí počet podsložek a souborů ve složce BALTIE.

Chcete-li zjistit pouze počet podsložek nebo pouze počet souborů, použijte na konci výrazu odpovídající prvek:

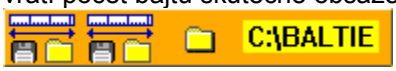
Vrať počet podsložek ve složce *C:\BALTIE*



Vrať počet souborů ve složce *C:\BALTIE*

Chcete-li zjistit velikost složky nebo počet položek ve složce včetně vnořených složek, použijte daný prvek dvakrát za sebou:

Vrať celkovou velikost složky *C:\BALTIE* v bajtech
vrátí počet bajtů skutečně obsazených celou složkou včetně všech jejích podsložek.



Vrať celkový počet souborů ve složce *C:\BALTIE*
zjistí, kolik souborů se nachází ve složce ***BALTIE*** a jejích podsložkách.

Práce s obrazovkou

Pozn.: Budeme-li v následujícím textu hovořit o obrazovce, budeme tím mít vždy na mysli **pracovní scénu.**

Prvek ***Obrazovka*** můžete použít ke smazání obrazovky, k načtení nebo uložení banky, scény nebo **obrázku**, zvětšování, zmenšování, k zobrazení čísla, data, času, souboru, řetězce, předmětů.

1. Smazání obrazovky

Samotný prvek ***Obrazovka*** smaže obrazovku - přesněji smaže všechny předměty, obrázky a grafiku na **pracovní scéně.** Animované předměty a Baltík zůstanou beze změn.

Smaž obrazovku

2. Jednoduché načtení scény, banky nebo obrázku a spouštění videoanimací

Chcete-li na obrazovku jednoduše načíst nějaký obrázek, použijte prvek ***Obrázek***, ***Banka*** nebo ***Scéna*** s určením daného obrázku. Stejně postupujte při spouštění videoanimace.

Pozn.: Za příkazy pro načtení obrázku můžete určit **průhlednost** s jakou bude obrázky načteny:

Načti průhledně obrázek "farao.bmp"

2.1 Obrázek z tabulky souborů

Obrázek z **tabulky souborů** můžete načíst použitím prvku ***Obrázek*** následovaným číslem **1** až **200**.

Načti na obrazovku obrázek, jehož název je v tabulce souborů ve sloupci obrázků v řádku č. ***1***.

Pozn.: Místo prvku ***Obrázek*** můžete použít prvek ***Obrazovka*** s číslem **2001**-**2200**.

Pozn.: Zvolený obrázek můžete do programu umístit přímo z **tabulky** stiskem pravého tlačítka myši na příslušném políčku tabulky a výběrem volby ***Uchopit***.

2.2 Scéna spuštěného programu

Scénu můžete načíst použitím prvku ***Scéna*** následovaným číslem **0** až **99** reprezentujícím scénu **.s00** až

.s99.

Načti scénu ***s01***

Název souboru se scénou je totožný s názvem programu. Jmenuje-li se váš program např. **prog**, bude tímto příkazem načtena scéna ze souboru **prog.s01**.

Pozn.: Místo prvku ***Scéna*** můžete použít prvek ***Obrazovka*** s číslem **0**-**99**.

Pozn.: Zvolenou scénu můžete do programu umístit jednoduše přepnutím na tuto scénu v režimu **Skládat scénu** nebo **Čarovat scénu** a stiskem tlačítka ***Scéna***. Tím příslušné označení scény uchopíte a můžete je umístit do programu.

2.3. Banka spuštěného programu

Banku můžete načíst použitím prvku ***Banka*** následovaným číslem **0** až **99** pro banky **.b00** až **.b99**, resp. **100** až **199** pro banky **.c00** až **.c99**.

Načti banku ***b01***

Název souboru s bankou je totožný s názvem programu. Jmenuje-li se váš program např. **prog**, bude tímto příkazem načtena banka ze souboru **prog.b01**.

Pozn.: Místo prvku ***Banka*** můžete použít prvek ***Obrazovka*** s číslem **1000**-**1199**.

Pozn.: Zvolenou banku můžete do programu umístit jednoduše přepnutím na **výběr předmětu,** výběrem zvolené banky a stiskem tlačítka Banka. Tím příslušné označení banky uchopíte a můžete je umístit do programu.

2.4 Videoanimace AVI z tabulky souborů

Videoanimaci z **tabulky souborů** můžete přehrát použitím prvku ***Videoanimace*** následovaným číslem **1** až **200**.

Přehraj animaci č. ***1***

2.5 Načtení z obecného souboru

Nenachází-li se soubor s obrázkem v tabulce souborů nebo není-li banka či scéna v souboru se stejným názvem, jaký má soubor se spuštěným programem, musíte použít prvek ***Obrázek*** následovaný **řetězcem** s názvem souboru.

Načti obrázek ***farao.bmp***

Načti banku ***c:\net\nibbler.c07***

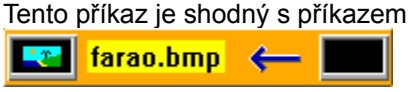
Načti scénu ***a:\program2.s00***

3. Prosté přiřazení

K načítání, ukládání, mazání obrázků můžete použít prvek ***Přiřaď***. Celý příkaz se skládá ze stanovení místa, kam chcete obrázek vložit, prvku ***Přiřaď*** a určení zdroje odkud chcete obrázek vzít.

Zobraz obrázek v souboru ***farao.bmp***

Tento příkaz je shodný s příkazem "Načti obrázek farao.bmp" z minulého odstavce.



Obsah obrazovky (pracovní scény) vlož do souboru ***farao.bmp*****.**

Pozn.: Tento příkaz obsah obrazovky skutečně **vloží**. Pokud tedy byl obrázek **farao** původně větší, než obrazovka, jeho okraj zůstane nezměněný. Chcete-li, aby byl v souboru skutečně jen obsah obrazovky, použijte před příkazem prvek ***Obrazovka***, který před provedením příkazu smaže místo určení:

Do prázdného (vyprázdněného) souboru ***farao.bmp*** **ulož obsah obrazovky**

Podobně budete postupovat, chcete-li načíst obrázek, ale před tím chcete obrazovku smazat:

Na smazanou obrazovku načti obrázek ze souboru ***farao.bmp***

Tímto příkazem můžete i tisknout obrázky na tiskárně. Jako cílový obrázek na levé straně přiřazení zvolte Obrázek č. 0:

Vytiskni obsah obrazovky na tiskárně

Pozn.: Při tisku můžete používat souřadnice, oblasti i přizpůsobení velikosti, o němž budeme hovořit vzápětí.

Kromě načítání ze souboru a ukládání do souboru můžete také přiřazovat z jednoho souboru do druhého:

Do obrázku ***bak.bmp*** **přiřaď obrázek** ***farao.bmp***

nebo z obrazovky na obrazovku

Na obrazovku přiřaď obrazovku

Pozn.: Tato varianta má smysl pouze s použitím souřadnic nebo oblastí při přesouvání, zmenšování, zvětšování obsahu obrazovky, o němž budeme hovořit za chvíli.

4. Přiřazení s úpravou velikosti

Použijete-li místo prvku ***Přiřaď*** (resp. ***Rovno***) prvek ***Menší***, upraví se velikost původní grafiky na velikost místa určení:

Na obrazovku přiřaď obrázek *farao.bmp* v upravené velikosti

Je-li obrázek ***farao.bmp*** **menší** než obrazovka (585 x 290 bodů), bude **zvětšen** na celou obrazovku. Je-li **větší**, bude příslušně **zmenšen**. V každém, případě bude načtený obrázek zabírat právě celou plochu obrazovky (přesněji pracovní scény).

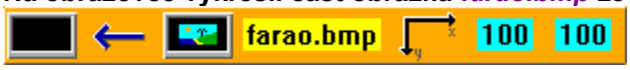
5. Přiřazení se zadáním souřadnic

Následují-li za určením místa určení nebo zdroje souřadnice jednoho bodu, označují levý horní roh oblasti, z níž má být obrázek vzat, resp. kam má být vložen.

Vykresli obrázek ze souboru *farao.bmp* na souřadnicích *x100 y100*

Tentokrát bude obrázek ***farao.bmp*** vložen na obrazovku tak, že jeho levý horní roh bude na souřadnicích **x100 y100**.

Na obrazovce vykresli část obrázku *farao.bmp* ze souřadnic *x100 y100*



Tímto příkazem vložíte obrázek ***farao.bmp*** na obrazovku tak, že bod, který měl v obrázku souřadnice ***x100 y100*** bude v levém horním rohu obrazovky.

Na obrazovku na souřadnice *x100 y100* vykresli obrázek *farao.bmp* ze souřadnic *x100 y100*

Tímto příkazem vložíte obrázek ***farao.bmp*** na obrazovku tak, že bod, který měl v obrázku souřadnice ***x100 y100*** bude na obrazovce na souřadnicích ***x100 y100***, zbylá část obrázku vlevo a nahoře nebude zobrazena (obrázek se ořeže).

6. Oblasti

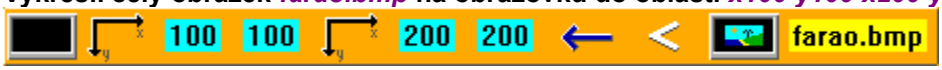
Následujíc-li za určením obrázku **souřadnice dvou bodů (oblast),** označují oblast, odkud se má obrázek vzít, resp. oblast, kam se má obrázek umístit.

Vykresli obrázek *farao.bmp* na obrazovku do oblasti *x100 y100 x200 y200* v nezměněné velikosti

Tímto příkazem se umístí levý horní roh obrázku ***farao.bmp*** na souřadnice **x100 y100** a bude-li větší, než zadaná oblast, bude přečnívající část oříznuta.

Použijete-li pro přiřazení prvek ***Menší***, bude celý obrázek zmenšen (popř. zvětšen) tak, aby se právě vešel do zadané oblasti:

Vykresli celý obrázek *farao.bmp* na obrazovku do oblasti *x100 y100 x200 y200*

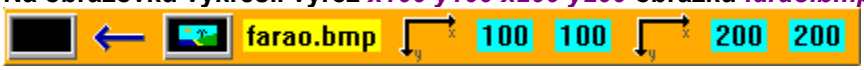


Po provedení tohoto příkazu bude na obrazovce zobrazen celý obrázek ***farao.bmp*** tak, že jeho levý horní roh bude na souřadnicích **x100 y100** a pravý dolní roh na souřadnicích **x200 y200**. V případě potřeby bude jeho velikost upravena

Pozn.: Pokud by byla oblast zadána obráceně, tj. **x200 y200 x100 y100**, bude obrázek převrácen - levý horní roh bude na **x200 y200**, pravý dolní roh na **x100 y100**.

Zadáte-li oblast u zdroje, použije se místo celé grafiky jen výřez:

Na obrazovku vykresli výřez *x100 y100 x200 y200* obrázku *farao.bmp*



Z obrázku se vyřízne zadaná oblast a vloží se do levého horního rohu obrazovky.

Na celou obrazovku vykresli výřez *x100 y100 x200 y200* obrázku *farao.bmp* zvětšený tak, aby ji právě pokryl

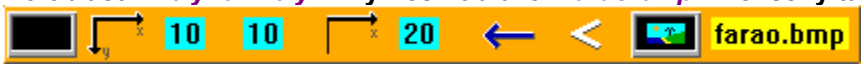
V tomto případě se zadaný výřez zvětší na celou obrazovku.

Do oblasti *x10 y10 x20 y20* vykresli výřez *x100 y100 x200 y200* obrázku *farao.bmp* zmenšený tak, aby ji právě pokryl

Tento příkaz nejdříve vyřízne z obrázku ***farao.bmp*** oblast **x100 y100 x200 y200** a pak ji zmenší na obrazovku do oblasti **x10 y10 x20 y20**.

Pozn.: Nezadáte-li jednu ze souřadnic pravého dolního rohu oblasti, do které má být obrázek vykreslen, dopočítá se tato souřadnice automaticky podle poměru stran obrázku:

Do oblasti *x10 y10 x20 y??* vykresli obrázek *farao.bmp* zmenšený tak, aby ji právě pokryl



Tento příkaz zobrazí celý obrázek ***farao.bmp*** tak, že jeho šířka bude od bodu 10 k bodu 20 a jeho výška se dopočítá tak, aby poměr šířky a výšky původního obrázku zůstal zachován.

8. Zvláštní případy

8.1 Z obrazovky na obrazovku

Na obrazovku přiřaď obsah obrazovky od souřadnic *x100 y100*

Tento příkaz zkopíruje do levého horního rohu obrazovky obsah obrazovky od souřadnic **x100 y100** do pravého dolního rohu obrazovky.

POZOR! Kopírovat se bude **pouze obraz**, tedy to, co skutečně vidíte, nikoliv předměty, které byly na obrazovku umístěny příkazy čaruj předmět nebo načti scénu, příp. banku. Obrazovka proto bude na některých místech lhát, protože na ní uvidíte jiné předměty, než v daných pozicích opravdu budou.

Chcete-li kopírovat předměty z jednoho místa obrazovky na jiné, použijte na pravé straně přiřazení prvek ***Nějaký předmět***:

Na obrazovku přiřaď předměty ze souřadnic *X2 Y3*

Tento příkaz zkopíruje předměty z pravé dolní části obrazovky počínaje políčkem **X2 Y3** do levého horního rohu obrazovky.

Abyste se přesvědčili o výše popisované skutečnosti, vytvořte následujícího pomocníka, který na požádání napíše na každý předmět na obrazovce jeho číslo:

Na každé políčko napiš číslo předmětu, který na políčku stojí.
V cyklu přes všechny řádky a sloupce vypíše pro každé "navštívené" políčko požadovanou informaci.

Pak spusťte následující program, který provede následující kroky:

1. Načti nultou banku předmětů, očísluj její předměty a počkej na stisk klávesy.
2. Zkopíruj obrazec od pole **X10 Y5** do levého horního rohu obrazovky, znovu očísluj předměty a počkej na stisk klávesy.
3. Zkopíruj předměty od pole **X10 Y5** do levého horního rohu obrazovky, znovu očísluj předměty a počkej na stisk klávesy.

Sledujte po každém kroku čísla předmětů v levém horním rohu obrazovky.

8.2. Ukládání banky

Do banky č. *1* přiřaď obrazovku
Tento příkaz uloží obsah obrazovky do banky **b01**. Tato banka však bude uložena jako šestnáctibarevná s použitím **16** Baltíkových barev (viz **konstanty**). Pokud byly na obrazovce i jiné barvy, budou převedeny na nejbližší Baltíkovu barvu.

Chcete-li v běhu programu používat **předměty s více než 16 barvami** (např. ***TrueColor***), postupujte takto:

1. Vytvořte obrázek (**.bmp**), ze kterého chcete vytvořit banku předmětů.
2. Uložte tento obrázek jako banku (např. **.b01**)

Do banky č. *1* přiřaď obrázek *farao.bmp*

Od této chvíle můžete v programu používat předměty č. **1001** až **1150** v takových barvách, v jakých byly v obrázku ***farao.bmp***. Po skončení programu se však barvy ztratí a zůstane Baltíkových 16 barev. Chcete-li **uchovat změny**, které jste v bance provedli v plných barvách, uložte obrázek znovu jako **.bmp**.

8.3. Scény, ikony

Scény (soubory s příponou **.s00** až **.s99**) a ikony (soubory s příponou **.ico**) nelze zvětšovat ani zmenšovat a budou zobrazeny v původní velikosti.

8.4. 256barevné obrázky

Chcete-li načíst obrázek s příponou **.bmp** v **256** barvách, musíte mít obrazovku nastavenu nejméně na ***HiColor*** (**16** bitů), jinak se vám obrázek nenačte.

Pokročilá práce se souřadnicemi

Prvky *Souřadnice bodu* a *Souřadnice políčka*

Souřadnice Baltíka, animovaného předmětu, myši, oblasti nebo zobrazení můžete zadat také pomocí prvku souřadnice:

Souřadnice políčka

Souřadnice bodu

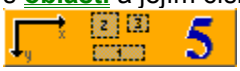
Použijete-li jeden z těchto prvků samostatně, znamená Baltíkovy souřadnice. Použití těchto prvků je tedy shodné s delším zápisem:

Souřadnice políčka, kde stojí Baltík

Souřadnice můžete použít také s animovaným předmětem a jeho číslem, případně **číselným výrazem** v závorkách:

Souřadnice animovaného předmětu číslo 5

s **oblastí** a jejím číslem, případně **číselným výrazem** v závorkách:



Souřadnice levého horního rohu oblasti číslo 5

nebo s některým prvkem **myši**:

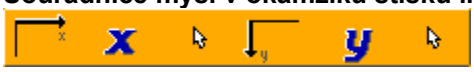
Souřadnice myši v okamžiku stisku levého tlačítka myši

Souřadnice políčka, v němž bylo stisknuto levé tlačítko myši

apod. Souřadnice **myši** můžete zadat také použitím samotného prvku myši. Následující zápisy jsou tedy shodné:

Souřadnice myši v okamžiku stisku libovolného tlačítka myši

Souřadnice myši v okamžiku stisku libovolného tlačítka myši



Souřadnice myši v okamžiku stisku libovolného tlačítka myši

Jako souřadnice můžete použít také zvláštní dvojice a čtveřice **konstant** nebo **proměnných**:

Dvojice proměnných X,Y - souřadnice políčka

Dvojice proměnných x,y - souřadnice bodu

Dvojice proměnných x1,y1 - souřadnice bodu

Dvojice proměnných x2,y2 - souřadnice bodu

Číslované dvojice proměnných - souřadnice bodu

Čtveřice proměnných x1,y1,x2,y2 - **souřadnice dvou bodů**

Číslované čtveřice proměnných - **souřadnice dvou bodů**

Vodorovná a svislá složka souřadnic

Pokud chcete vodorovnou (x) nebo svislou (y) část souřadnic Baltíka, políčka před Baltíkem, animovaného předmětu, myši, oblasti nebo zobrazení, použijte modré prvky ***x, y, X, Y.***

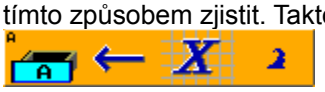
Vodorovná souřadnice bodu

Svislá souřadnice bodu

Vodorovná souřadnice políčka

Svislá souřadnice políčka

Po vložení jednoho z těchto prvků do programu se rozvine nabídka objektů, jejichž souřadnice je možno tímto způsobem zjistit. Takto vzniklé výrazy můžete používat jako čísla, např.



Do proměnné *A* přiřaď vodorovnou souřadnici políčka Baltíka

V proměnné A bude hodnota 0 až 14 podle toho, ve kterém sloupci Baltík stojí

Zobraz svislou souřadnici bodu animovaného předmětu č. 1

Na obrazovce bude zobrazena souřadnice ***y*** animovaného předmětu č. 1.

Zobraz bílý bod náhodně vlevo nahoru od ukazatele myši

Bod bude zobrazen na souřadnicích, které nebudou v žádném směru větší než současné souřadnice ukazatele myši.

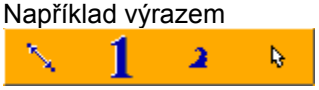
Vzdálenost

Vzdálenost

Za tímto prvkem následuje číslo, které udává, zda chcete zjišťovat vzdálenost políček (číslo je 0), nebo bodů (číslo je různé od nuly). Vzdálenost se určuje jako druhá odmocnina součtu druhých mocnin rozdílů jednotlivých souřadnic, tedy podle vzorce

vzdálenost = odmocnina ((x1-x2)^2 + (y1-y2)^2)

Například výrazem



Vzdálenost bodů Baltíkovy souřadnice a souřadnice myši při stisku některého tlačítka myši

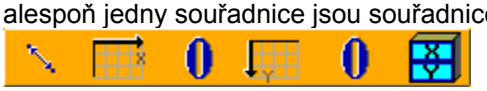
zjistíte, kolik bodů od Baltíka bylo naposledy stisknuto některé tlačítko myši.

Výrazem

Vzdálenost políček Baltíkovy souřadnice a souřadnice myši při stisku některého tlačítka myši

zjistíte tuto vzdálenost v políčcích.

Pokud číslo nezadáte, vzdálenost je zjišťována podle toho, jaký typ souřadnic zadáte. Budou-li obě zadané souřadnice souřadnicemi políček, bude výsledné číslo vzdáleností políček, jinak (tzn. pokud alespoň jedny souřadnice jsou souřadnice bodu), bude výsledné číslo vzdáleností bodů:



Vzdálenost políčka X0 Y0 a políčka se souřadnicemi v proměnných (X,Y)

Vzdálenost bodu x0y0 a políčka se souřadnicemi v proměnných (X,Y)

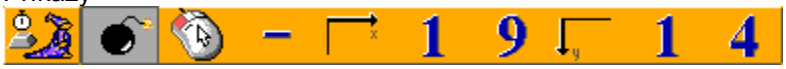
Počítání se souřadnicemi

Souřadnice můžete také **sčítat, odčítat, násobit** a **dělit**, případně **počítat zbytek po dělení**. Například příkaz

Proměnné xy zvětši o x1y1 krát x2y2

přidá k souřadnicím v šuplíku x,y součin souřadnic x1,y1 a x2,y2. Do šuplíku x se uloží x plus x1 krát x2, do šuplíku y se uloží y plus y1 krát y2.Tento konkrétní výraz můžete použít např. při pohybu předmětu, máte-li v šuplících x,y jeho současné souřadnice, v šuplících x1,y1 jeho směr (doprava:x1= 1, doleva:x1= -1, dolů y1= 1, nahoru y1= -1) a v šuplících x2, y2 jeho rychlost. Jedno provedení příkazu pak odpovídá jednomu kroku předmětu.

Příkazy



Čti klávesu s čekáním, vyčaruj bombu na (souřadnice myši při stisku levého tlačítka - x19 y14)

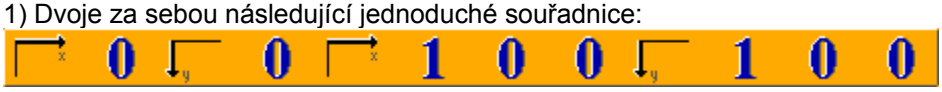
můžete použít k vyčarování předmětu (v našem případě bomby) tam, kde bylo stisknuto levé tlačítko myši. Odečtením souřadnic (x19, y14) zajistíte, že tam, kde jste stiskli tlačítko, bude střed předmětu (19 je přibližně polovina šířky předmětu, 14 přibližně polovina výšky předmětu).

Souřadnice dvou bodů (políček)

Souřadnice dvou bodů nebo políček budete potřebovat při ořezávání nebo změně velikosti **obrázku**, při změně hodnot **oblastí**, při prácí s **grafickými příkazy** a podobně.

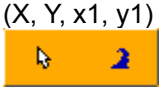
Jako dvojice souřadnic můžete použít:

1) Dvoje za sebou následující jednoduché souřadnice:



(0, 0, 100, 100)

(X, Y, x1, y1)



(souřadnice ukazatele myši, souřadnice Baltíka)

2) **Oblast**:

Oblast č. 1

3) Čtveřici **konstant** nebo **proměnných**:

celočíselná čtveřice konstant x1y1x2y2

reálná čtveřice číslovaných lokálních proměnných

= = AŽ SEM JE TO SNAD READY = = 99-06-11 = =

Datum a čas

Pro práci s datem a časem slouží především tyto prvky:

Datum a čas

Převod čísla na datum a čas

V této sekci probereme následující témata:

1) Vnitřní reprezentace data a času

Je-li prvek ***Datum a čas*** použit samostatně jako **číselný výraz,** má hodnotu reálného čísla, které udává, kolik dní uplynulo od půlnoci ze 29. na 30. prosince roku 1899:

Ü celá část udává počet celých uplynulých dní,

Ü desetinná část udává uplynulý zlomek daného dne.

Příklady:

Číslo	Význam
1,0	**31. prosince roku 1899, 00:00 hodin,**
2,5	**1. ledna roku 1900, 12:00 hodin,**
36 526,75	**1. ledna roku 2000, 18:00 hodin**

Pozn.: Počet uplynulých hodin ze zadaného dne získáte vynásobením desetinné části číslem 24.

2)Zadávání data a času

Datum a čas zadáváme postupně od největší jednotky do nejmenší tj. v pořadí rok - měsíc - den - hodina - minuta - sekunda - milisekundy.

Zadáváme-li datum jako **literál,** oddělujeme jednotlivé složky následovně:

Ü rok od měsíce a měsíc ode dne oddělujeme lomítkem (**/**),

Ü datum od času, tj. dny od hodin oddělujeme lomítkem (**/**), dvojtečkoku (**:**) nebo ampersandem (**&**),

Ü hodiny od minut, minuty od sekund a sekundy od milisekund oddělujeme dvojtečkou (**:**).

2) Zjištění aktuálního data a času

Samostatně stojící prvek ***Datum a čas***
vrací aktuální datum a čas

Zajímá-li vás pouze některý údaj, musíte si pomoci prostřednictvím funkcí.

3) Zjištění některé složky data a času

Zajímá-li vás pouze některá ze složek data, zadejte za prvkem ***Datum a čas*** některý z následujících řetězcových parametrů:

- **&** úplné datum a čas
- **/** počet dní od 30.12.1899 (tj. celá část data a času)
- **:** čas uplynulý od půlnoci ve zlomku dne (tj. desetinná část data a času)
- **y** rok (**y**ear)
- **M** měsíc (**M**onth)
- **d** den (**d**ay)
- **w** den v týdnu (day of **w**eek: 1=neděle, 2=pondělí...7=sobota)
- **H** hodiny (**H**our) 0 až 24
- **h** hodiny (**h**our) 0 až 12
- **a** polovina dne (0 - dopoledne, 1 - odpoledne)
- **m** minuty
- **s** sekundy
- **t** milisekundy

Zadáte-li jako další parametr datum a čas, vrátí funkce příslušnou složku zadaného data a času.

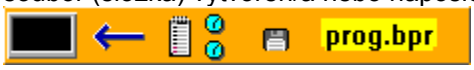
Nezadáte-li druhý parametr, vrátí požadovanou složku aktuálního data a času.

Vrátí současný rok - např. 1999.

Vrátí počet minut uplynulých od počátku poslední hodiny. Je-li např. 12:34, vrátí 34.

4) Datum a čas souboru nebo složky

Použitím prvku Datum a čas s označením **souboru** nebo **složky** můžete zjistit datum a čas, kdy byl soubor (složka) vytvořen/a nebo naposledy změněn/a.



Zobraz datum a čas souboru "prog.bpr"

5) Převod čísla na datum a čas

Chcete-li převést určité množství času v dané časové jednotce (např. 5 hodin) na číslo, použitelné jako datum a čas, použijte prvek ***Převod čísla na datum a čas*** s řetězcem, udávajícím jednotku a číslem, udávajícím množství. Jako jednotku můžete použít jeden z následujících jednoznakových řetězců:

- **y** rok (**y**ear)
- **M** měsíc (**M**onth)
- **d** den (**d**ay)
- **H** hodiny (**H**our)
- **h** hodiny (**h**our)

m minuty
s sekundy
t milisekundy

Zobraz 5 hodin jako čas

Tento příkaz zobrazí **5:00:00,** tedy 5 hodin ve standardním formátu času

Pozn.: Pokud převádíte na datum a čas počet měsíců, bude číslo převedeno na datum, odpovídající danému počtu účetních měsíců, tedy měsíců, které mají všechny 30 dní. Převádíte-li na datum a čas počet let, převede se číslo na datum odpovídající danému počtu účetních let, které mají 12 účetních měsíců, tedy 360 dní. Toto opatření je zavedeno kvůli nestejné délce měsíců v roce a nestejné délce let.

Stopky

Stopky

Stopky slouží k přesnému měření času od určitého okamžiku. Stopky můžete spouštět a zastavovat, přečíst z nich čas ,nastavit je na určitý čas a zjišťovat, jestli stopky běží v **podmínkách**.

Stiskni tlačítko na stopkách

Tímto příkazem stisknete tlačítko stopek: pokud byly stopky zastaveny, spustíte je; pokud běžely, zastavíte je. Na začátku programu jsou stopky vynulovány a zastaveny.

Zastav, vynuluj a spusť stopky

Tímto příkazem vždy zastavíte, vynulujete a spustíte stopky, ať už byly zastaveny nebo běžely. Je to, jako byste dvěma stisky tlačítka vymazali všechny stavy a třetím stiskem stopky spustili.

Pozn.: Tuto posloupnost můžete použít také jako podmínku. Ta je splněna, pokud stopky právě běží.

Zastav a vynuluj stopky

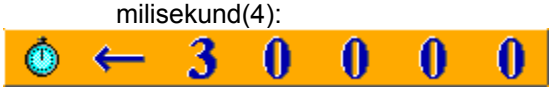
Tímto příkazem vždy stopky zastavíte a vynulujete, ať už byly zastaveny nebo běžely. Je to, jako byste dvěma stisky tlačítka vymazali všechny stavy.

Pozn.: Tuto posloupnost můžete použít také jako podmínku. Ta je splněna, jsou-li stopky zastaveny.

čas stopek

Tato funkce vrací uplynulý čas na stopkách v milisekundách

Pokud zadáte **číslo**, můžete zjistit uplynulý počet hodin(1), minut(2), sekund(3) nebo milisekund(4):



Nastav stopky na 30 sekund (30 000 milisekund)

Pokud nyní spustíte stopky, bude na nich 30 sekund .

Samostatné - PopUp

Klávesové zkratky:

Všeobecné

F1	Nápověda
Ctrl+N	Nová scéna, nový program
Ctrl+O	Otevřít (scénu, program)
Ctrl+S	Uložit (scénu, program)
Alt+F4	Konec programu

Režim Programovat:

F2	Uprav objekt nebo parametry
Ctrl+F	Najdi posloupnost příkazů
F3	Najdi další posloupnost příkazů
Shift+F3	Najdi předchozí posloupnost příkazů
F6	Nastav značku a vrať se k minulé značce
F9	Spusť program
Mezera	Zobraz objekt (náhled)
Shift+levé tlač. myši	Označ blok
Shift+klepnutí levým	Označ řádek
Shift+poklepání levým	Označ všechny řádky na stejné úrovni

Skrz body s průhlednou barvou prosvítá pod přesouvaným či kopírovaným obrázkem obrázek původní.

Oddělovač indexů pole

Oddělovač čísel složek v souboru

Oddělovač parametrů při práci s řetězci

Závorky v číselných výrazech

Závorky v podmínkách

Příkaz je tvořen jedním nebo několika prvky (ikonami) poskládanými v přesně určeném pořadí.

Rozsah jednoho příkazu poznáte nejlépe tak, když poklepáte na některý z prvků, které jsou součástí příkazu. Baltík totiž po poklepání na kterýkoliv z prvků příkazu vybere celý příkaz do bloku.

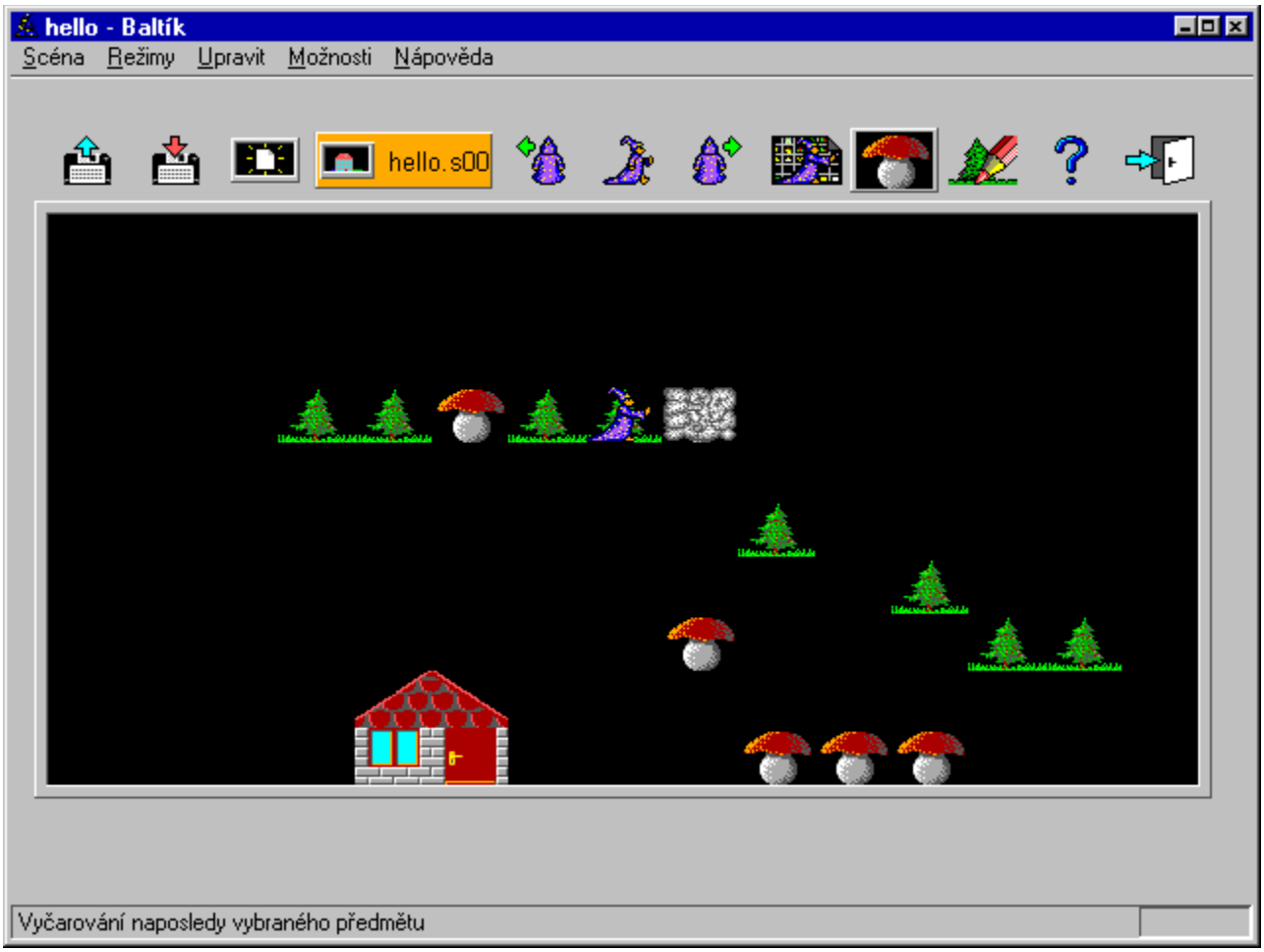
Máte-li v ***Možnosti → Prostředí*** zapnuto ***Příkazy vcelku***, tak mezi jednotlivými prvky tvořícími příkaz nebudou dělící čáry - příkazy proto budou zobrazeny vcelku jako jeden velký prvek.

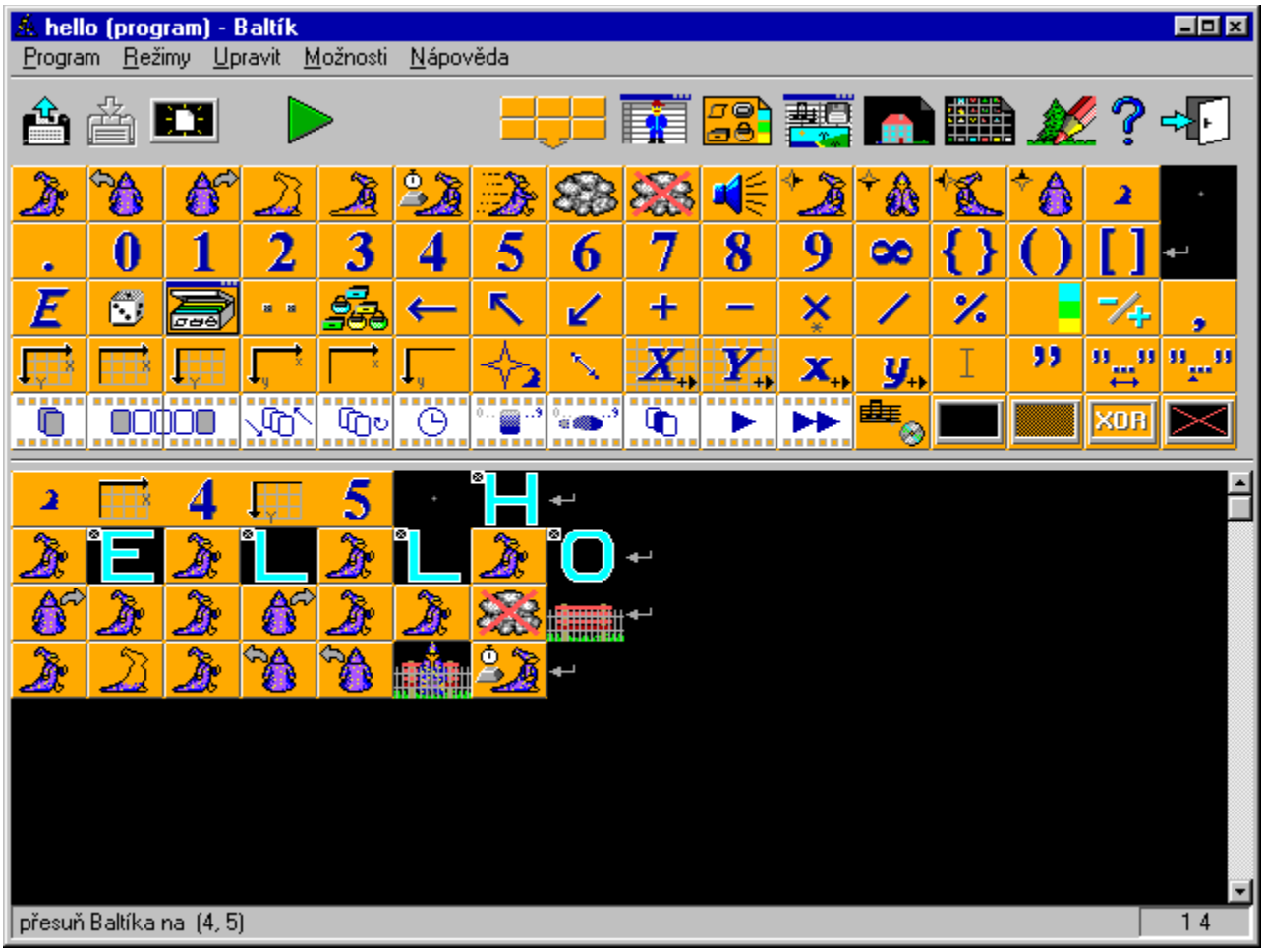
Pracovní scéna je scéna, kterou program zobrazuje, na kterou vkládáte předměty a po které se Baltík pohybuje.

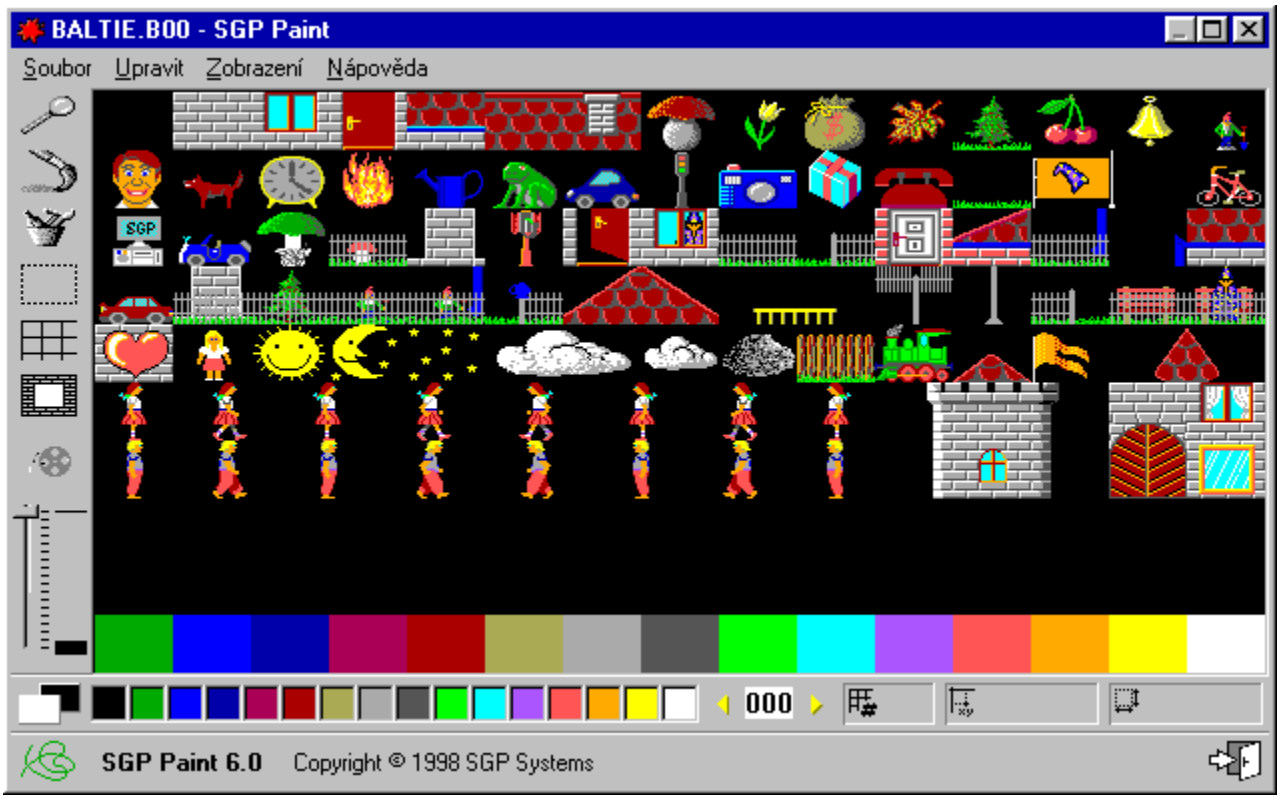
Mezi **Zvláštní banky** zařazujeme banky kláves, **proměnných** a **konstant.**

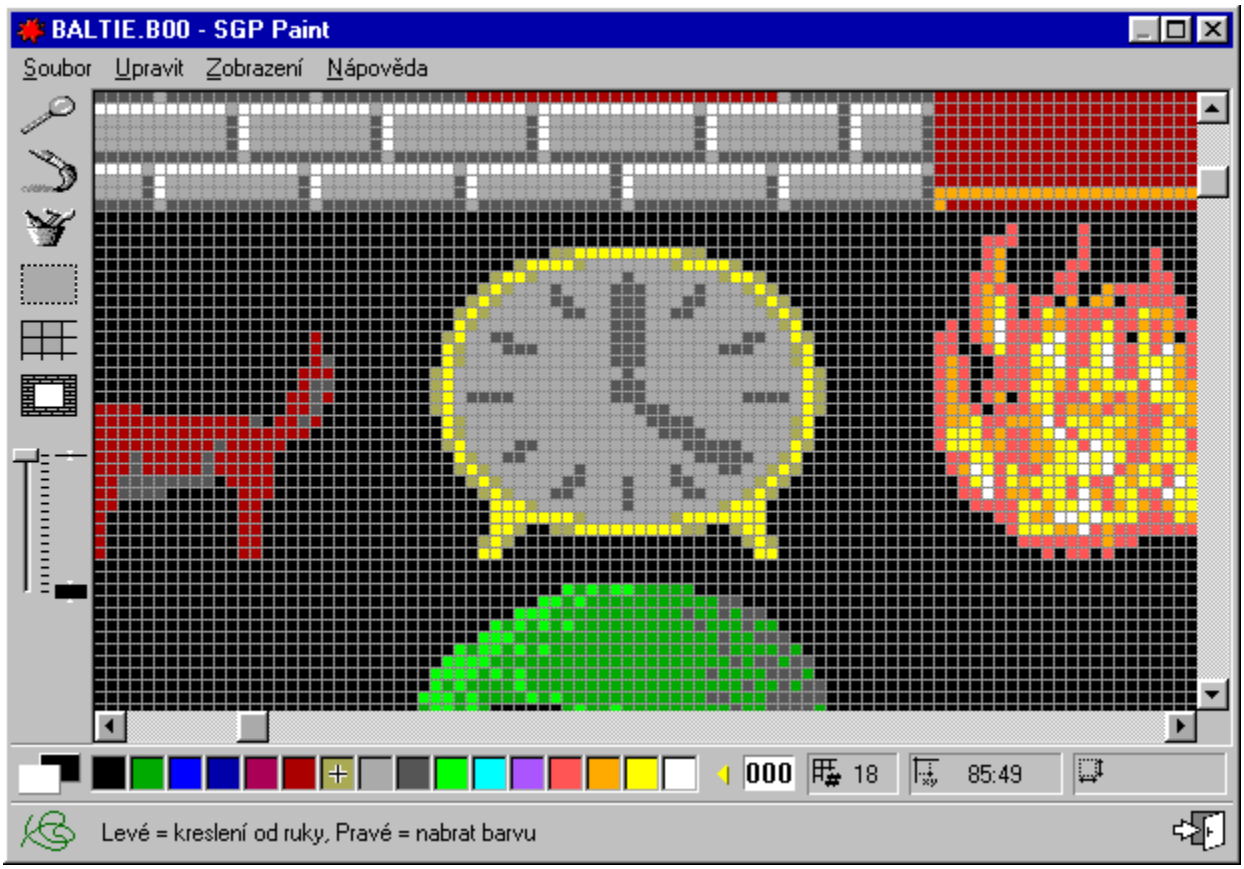

Předměty
SGP
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
baltie.b00


hello - Baltík
Scéna Režimy Upravit Nápověda
hello.s 00
Předmět 53 (levé tlačítko myši = přesun, pravé tlačítko = kopie)


hello - Baltík
Scéna Režimy Upravit Možnosti Nápověda
hello.s00
Vyčarování naposledy vybraného předmětu

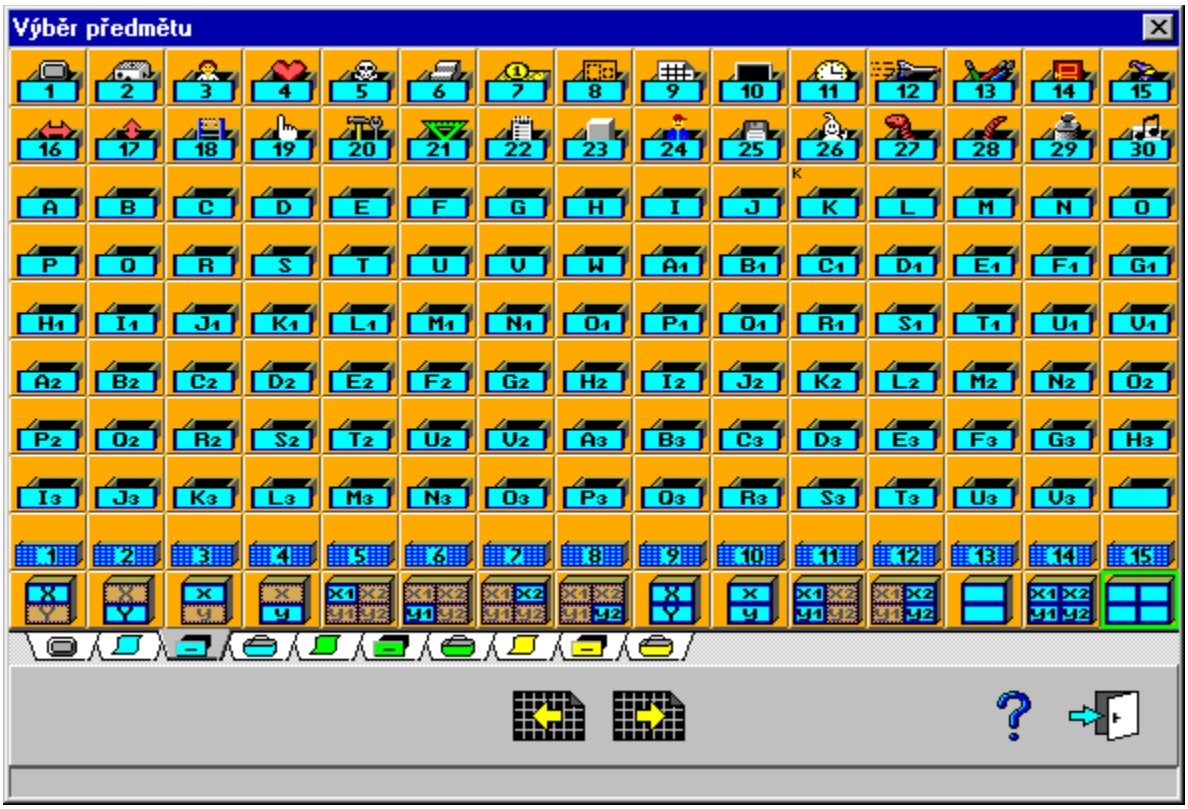

hello (program) - Baltík
Program Režimy Upravit Možnosti Nápověda
XOR
přesuň Baltíka na (4, 5)
1 4

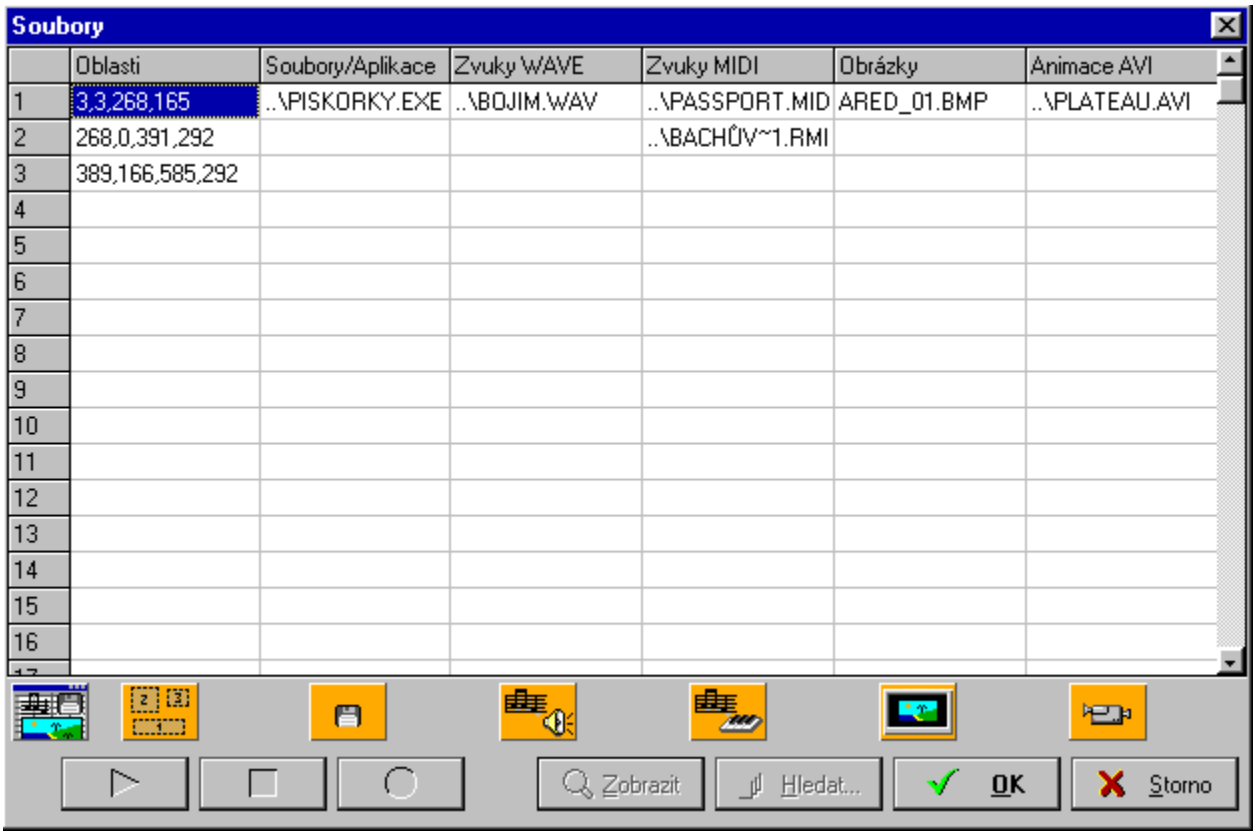

BALTIE.B00 - SGP Paint
Soubor
Upravit
Zobrazení
Nápověda
SGP
000
SGP Paint 6.0
Copyright © 1998 SGP Systems

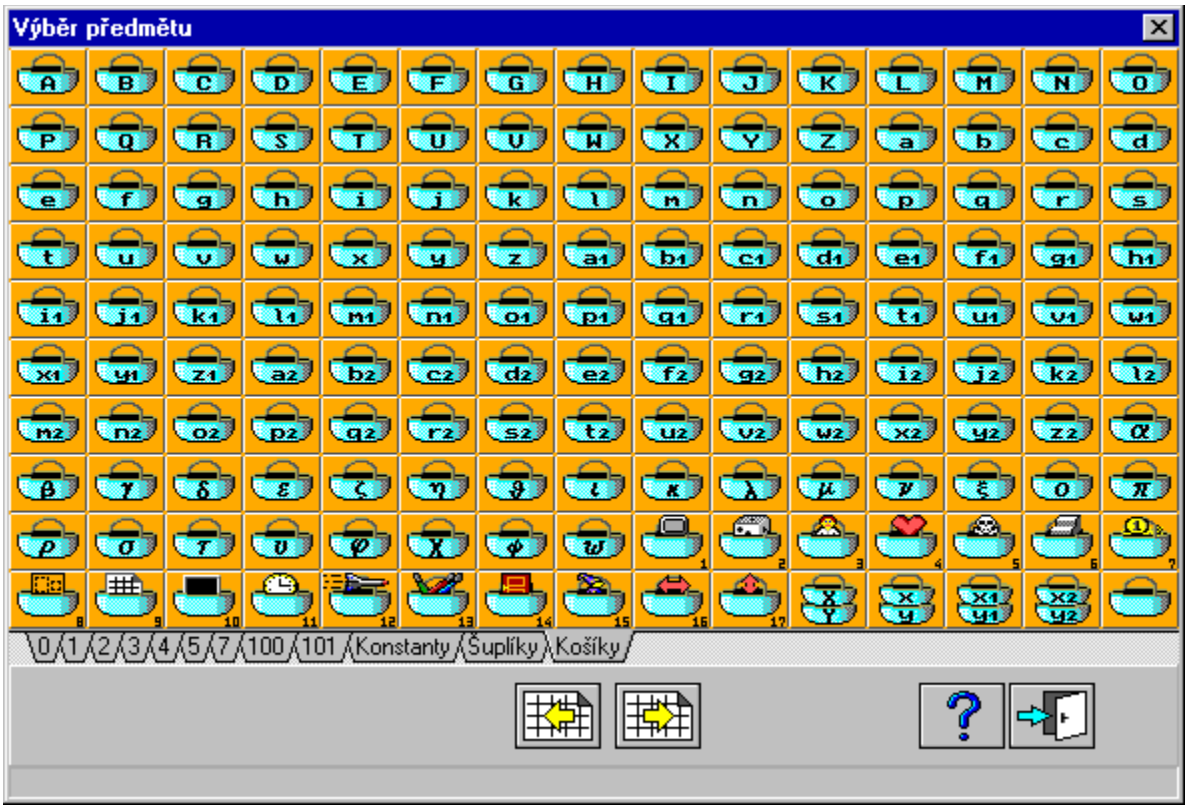

BALTIE.B00 - SGP Paint
Soubor
Upravit
Zobrazení
Nápověda
000
18
85:49
Levé = kreslení od ruky, Pravé = nabrat barvu

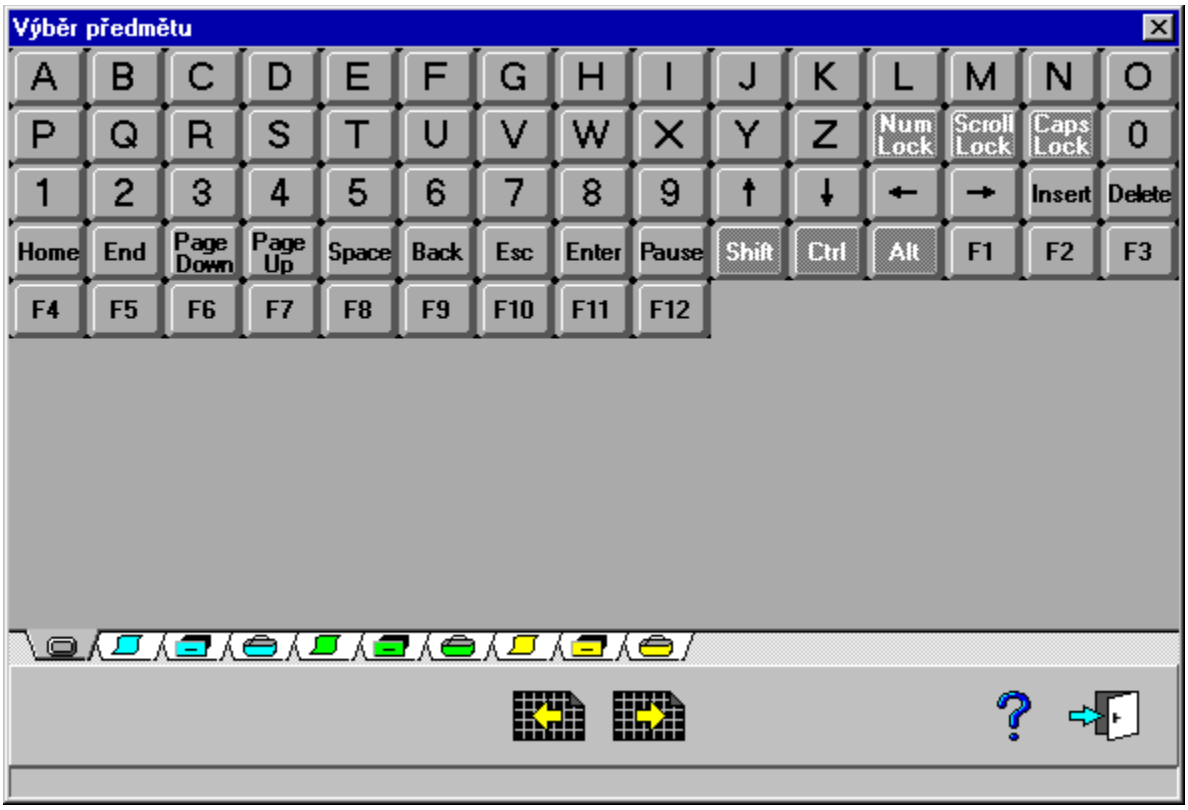
Tabulka

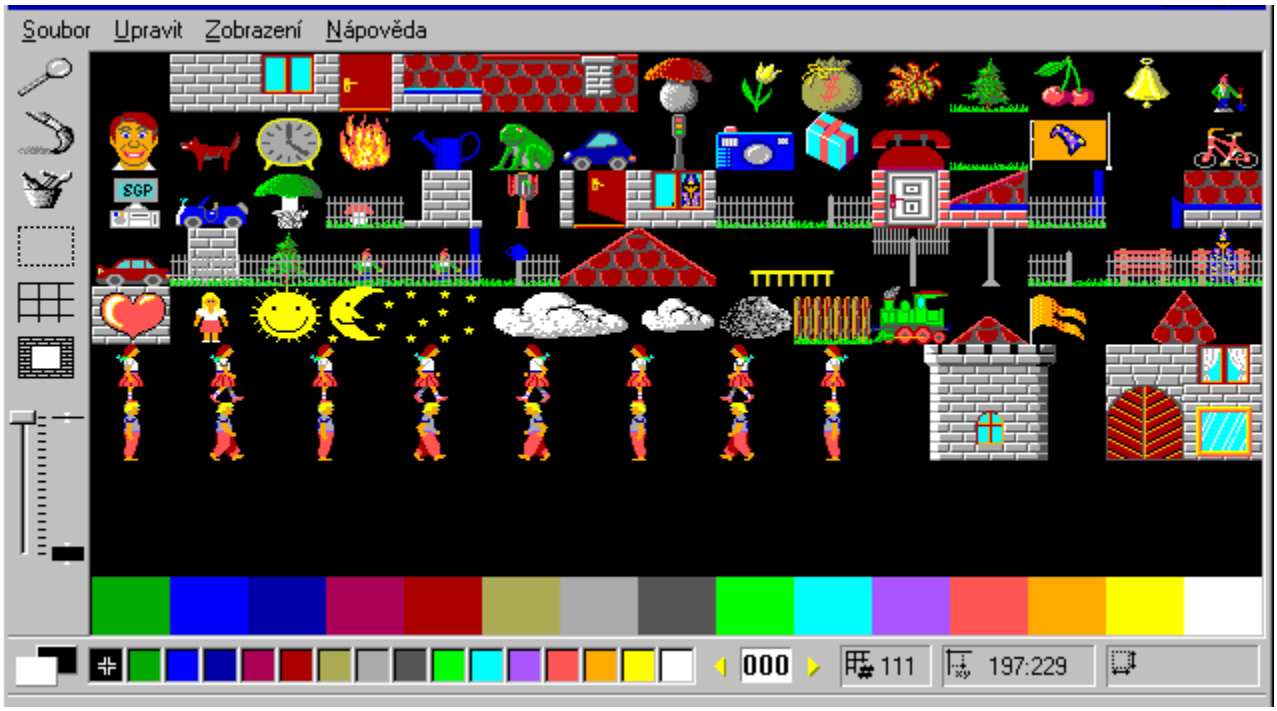
	Oblasti	Soubory/Aplikace	Zv
1	10,10,150,220	..\PISKORKY.EXE	..\
2	-50,-50,50,50		
3	0,0,584,289		
4			
5	0,29,1000,58		
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			


Výběr předmětu

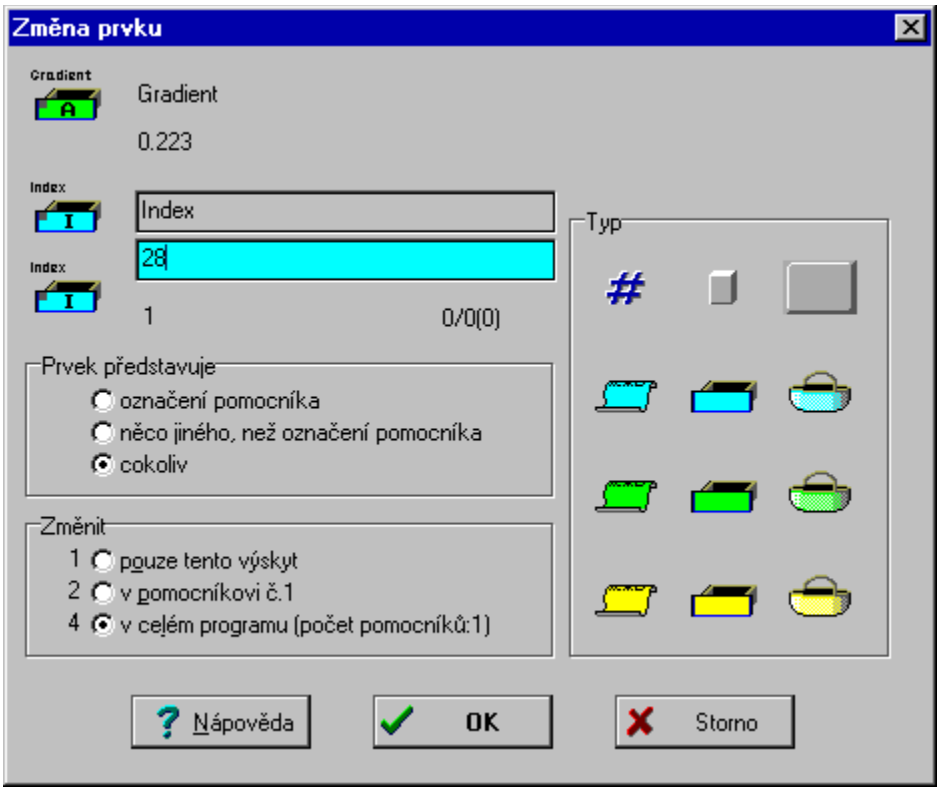

Soubory
Oblasti
Soubory/Aplikace
Zvuky WAVE
Zvuky MIDI
Obrázky
Animace AVI
1
3,3,268,165
..\PISKORKY.EXE
..\BOJIM.WAV
..\PASSPORT.MID
ARED_01.BMP
..\PLATEAU.AVI
2
268,0,391,292
..\BACHŮV~1.RMI
3
389,166,585,292
4
5
6
7
8
9
10
11
12
13
14
15
16
Zobrazit
Hledat...
OK
Storno

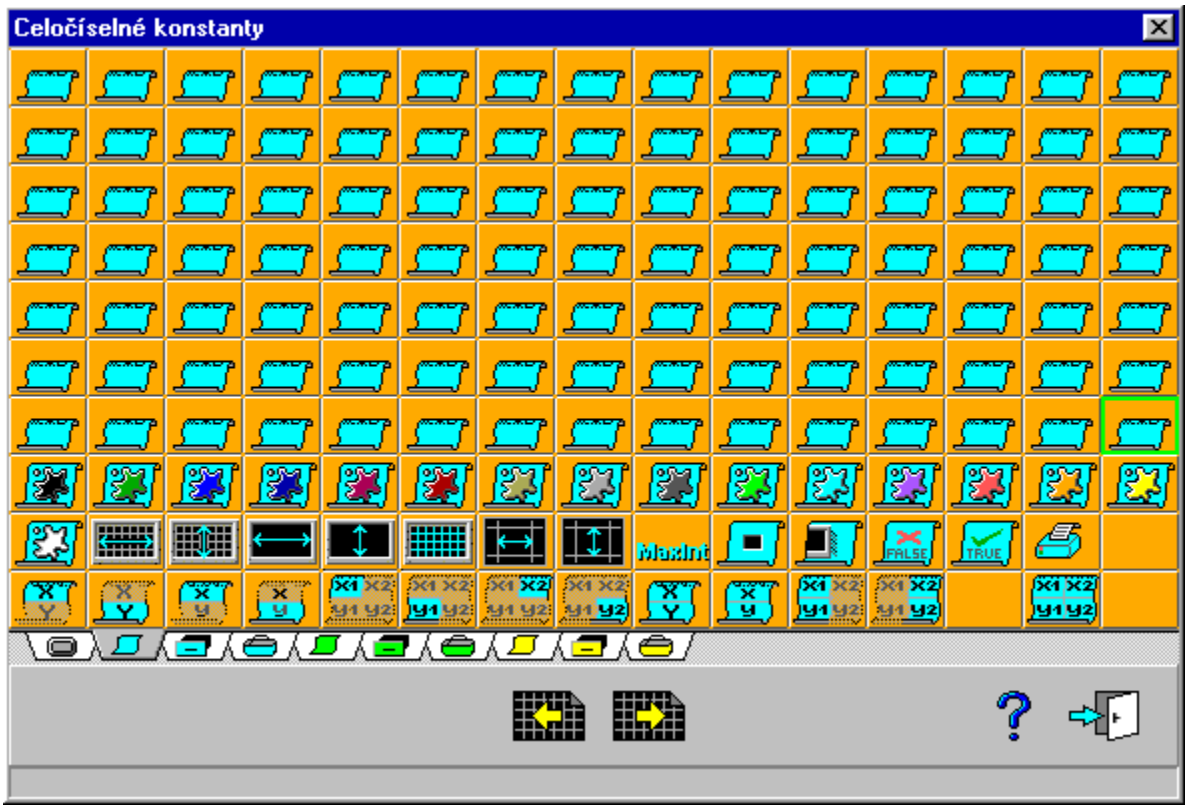

Výběr předmětu
0
1
2
3
4
5
7
100
101
Konstanty
Šuplíky
Košíky


Výběr předmětu
A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z Num Lock Scroll Lock Caps Lock 0
1 2 3 4 5 6 7 8 9 Insert Delete
Home End Page Down Page Up Space Back Esc Enter Pause Shift Ctrl Alt F1 F2 F3
F4 F5 F6 F7 F8 F9 F10 F11 F12

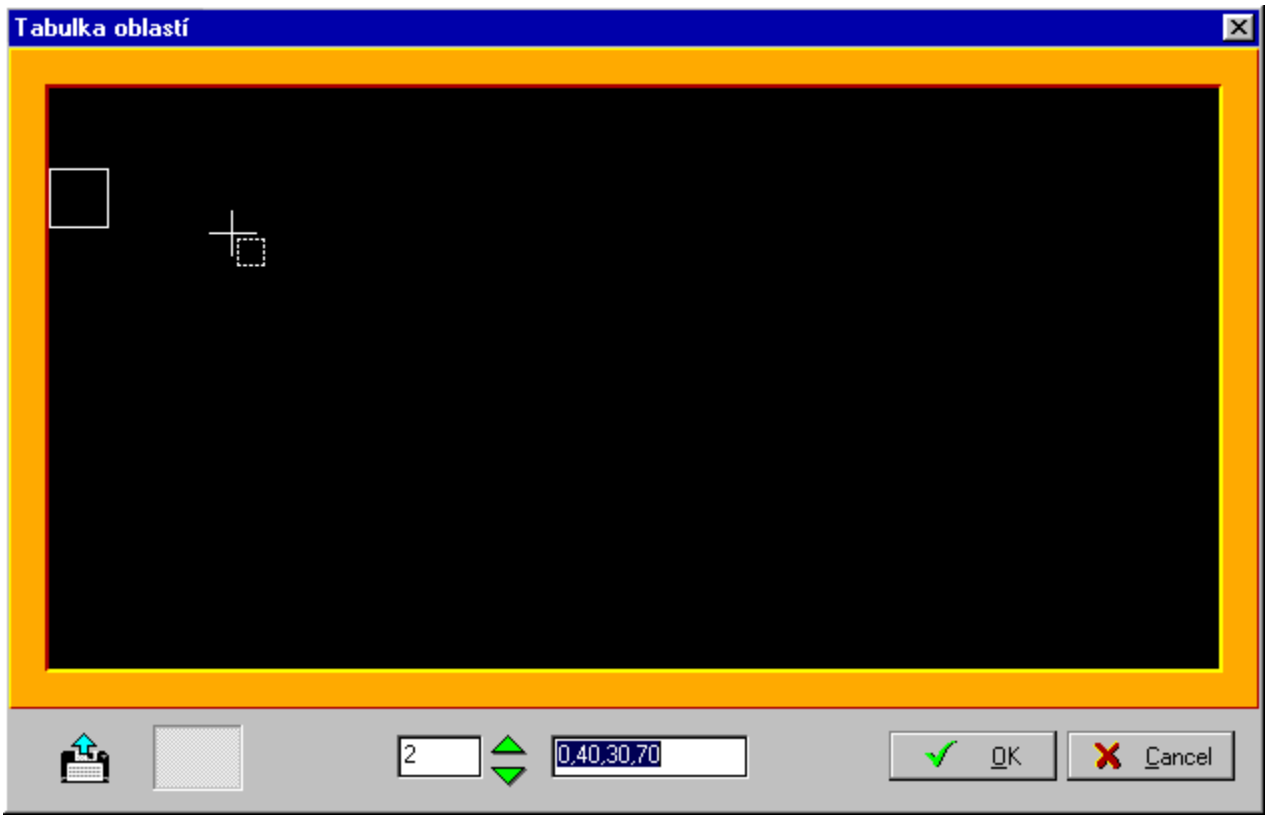

Soubor
Upravit
Zobrazení
Nápověda
SGP
000
111
197:229

	= 32	@ = 64	` = 96	□ = 128	= 160	Ŕ = 192	ŕ = 224
□ = 1	! = 33	A = 65	a = 97	□ = 129	ˇ = 161	Á = 193	á = 225
□ = 2	" = 34	B = 66	b = 98	‚ = 130	˘ = 162	Â = 194	â = 226
□ = 3	# = 35	C = 67	c = 99	ƒ = 131	Ł = 163	Ă = 195	ă = 227
□ = 4	$ = 36	D = 68	d = 100	„ = 132	¤ = 164	Ä = 196	ä = 228
□ = 5	% = 37	E = 69	e = 101	… = 133	Ą = 165	Ĺ = 197	Í = 229
□ = 6	& = 38	F = 70	f = 102	† = 134	¦ = 166	Ć = 198	ć = 230
□ = 7	' = 39	G = 71	g = 103	‡ = 135	§ = 167	Ç = 199	ç = 231
□ = 8	(= 40	H = 72	h = 104	ˆ = 136	¨ = 168	Č = 200	č = 232
□ = 9	) = 41	I = 73	i = 105	‰ = 137	© = 169	É = 201	é = 233
□ = 10	* = 42	J = 74	j = 106	Š = 138	Ş = 170	Ę = 202	ę = 234
□ = 11	+ = 43	K = 75	k = 107	‹ = 139	« = 171	Ë = 203	ë = 235
□ = 12	, = 44	L = 76	l = 108	Ś = 140	¬ = 172	Ě = 204	ě = 236
□ = 13	- = 45	M = 77	m = 109	Ť = 141	- = 173	Í = 205	í = 237
□ = 14	. = 46	N = 78	n = 110	Ž = 142	® = 174	Î = 206	î = 238
□ = 15	/ = 47	O = 79	o = 111	Ź = 143	Ż = 175	Ď = 207	ď = 239
□ = 16	0 = 48	P = 80	p = 112	□ = 144	° = 176	Đ = 208	đ = 240
□ = 17	1 = 49	Q = 81	q = 113	‘ = 145	± = 177	Ń = 209	ń = 241
□ = 18	2 = 50	R = 82	r = 114	’ = 146	˛ = 178	Ň = 210	ň = 242
□ = 19	3 = 51	S = 83	s = 115	“ = 147	ł = 179	Ó = 211	ó = 243
□ = 20	4 = 52	T = 84	t = 116	” = 148	´ = 180	Ô = 212	ô = 244
□ = 21	5 = 53	U = 85	u = 117	• = 149	µ = 181	Ő = 213	ő = 245
□ = 22	6 = 54	V = 86	v = 118	– = 150	¶ = 182	Ö = 214	ö = 246
□ = 23	7 = 55	W = 87	w = 119	— = 151	· = 183	× = 215	÷ = 247
□ = 24	8 = 56	X = 88	x = 120	˜ = 152	¸ = 184	Ř = 216	ř = 248
□ = 25	9 = 57	Y = 89	y = 121	™ = 153	ą = 185	Ů = 217	ů = 249
□ = 26	: = 58	Z = 90	z = 122	š = 154	ş = 186	Ú = 218	ú = 250
□ = 27	; = 59	[= 91	{ = 123	› = 155	» = 187	Ű = 219	ű = 251
□ = 28	< = 60	\ = 92	| = 124	ś = 156	Ľ = 188	Ü = 220	ü = 252
□ = 29	= = 61	] = 93	} = 125	ť = 157	˝ = 189	Ý = 221	ý = 253
□ = 30	> = 62	^ = 94	~ = 126	ž = 158	ľ = 190	T = 222	t = 254
□ = 31	? = 63	_ = 95	□ = 127	ź = 159	ż = 191	ß = 223	˙ = 255

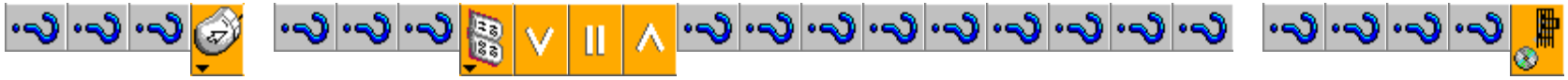

Změna prvku
Gradient
Gradient
0.223
Index
Index
Index
28
1
0/0(0)
Prvek představuje
označení pomocníka
něco jiného, než označení pomocníka
cokoliv
Změnit
1 pouze tento výskyt
2 v pomocníkovi č.1
4 v celém programu (počet pomocníků:1)
Typ
Nápověda
OK
Storno


Celočíselné konstanty
MaxInt
FALSE
TRUE


Pomocníci
0
0
1
Domek
2
2
0
0
Pomocník pro rámeček
3
2
0
1
WHILE
1
4
5
6
7
Uložení obrázku
8
1000
Počet pomocníků: 11
Storno


Tabulka oblastí
2
0,40,30,70
OK
Cancel

Vyberte příkaz, pro který chcete vidět nápovědu:



/*/
inv
a b
tan

Nápověda k tomuto tématu není ještě dokončena

Odpad

Následuje-li za tímto příkazem **číslo**, zapnou se nebo vypnou zvuky při Baltíkově chůzi a čarování. To, které zvuky budou zapnuté a které vypnuté, ovlivníte zadaným číslem tak, že sečtete čísla přiřazená zvukům, které chcete mít zapnuty, a výsledek zadáte za prvek ***Pípni***.

Jednotlivé zvuky jsou uloženy ve formátu **.wav** v souborech ***Baltie.w00*** až ***Baltie.w05***. V následující tabulce jsou uvedeny přípony příslušné jednotlivým zvukům spolu s čísly, která daný zvuk aktivují.

Číslo	Přípona	Příkaz - akce
1	w00	Popojdi
2	w01	Vlevo vbok, Vpravo vbok
4	w02	Čaruj předmět
8	w03	Náraz do zdi
16	w04	Neviditelný
32	w05	Viditelný

Budete-li tedy chtít zapnout zvuky pro chození, otáčení a případné nabourání do zdi, zadáte číslo **11** (**1+2+8**), budete-li chtít zapnout všechny zvuky, zadáte číslo **63** (součet všech čísel), budete-li chtít mít všechny zvuky vypnuty, zadáte **0**.

KONEC