

Obsah - Help k programu ScriptCad

- Úvod
- Menu programu
- Visuální návrhy výkresů
- Příkazy a jejich popis
 - Globální proměnné
 - Úsečka - příkaz Line
 - Kóta - příkaz DimLine
 - Text - příkaz Text
 - Bod - příkaz Point
 - Obdelník - příkaz Rectangle
 - Výplň - příkaz FillArea
 - Kružnice - příkaz Circle
 - Část kružnice - příkaz CirclePart
 - Elipsa - příkaz Ellipse
 - Obrázek - příkaz Picture
 - Práce s vrstvami
 - Vyčíslitelné výrazy
 - Procedury jako definice vlastních objektů
 - Vkládání skriptů do jiných skriptů - příkaz @include
- Prvky společné pro více příkazů
 - Barvy
 - Tloušťka čar
 - Typy čar
 - Typy šrafování
 - Typy kót
- Nastavení programu (ScriptCad.ini)
- Tiskový log a podpora tisku v přesném měřítku
- Registrace
- Licenční podmínky

Úvod

Program *ScriptCad* je kreslicí/rýsovací nástroj pro 2D návrhy. Pracuje v prostředí operačního systému Windows (Windows 95/98/98SE/ME/2000/XP) a byl naprogramován v Delphi 5 Professional.

Na rozdíl od většiny jiných CAD programů má program *ScriptCad* trochu odlišnou filozofii práce. V jiných CAD programech uživatel využívá vizuálního návrhu výkresu, tedy kliká v menu na piktogramy geometrických útvarů a následné kliknutí v kreslicí oblasti tento útvar zobrazí. I když i tento způsob práce je v programu *ScriptCad* také podporován (viz sekce Visuální návrhy výkresů), hlavní možnosti programu *ScriptCad* spočívají v kreslení pomocí příkazů (jejich úplný výčet a popis je v sekci Příkazy a jejich popis).

Menu programu

V levé části hlavního okna (skriptová oblast) píšeme kreslicí příkazy (jejich úplný výčet a popis je v sekci [Příkazy a jejich popis](#)). Po stisku tlačítka *Překreslit* (nebo z menu *Výkres-Překreslit*) je na základě příkazů vykreslen do pravé části hlavního okna výkres.

Menu Skript

Příkazy z levé části hlavního okna je možné ukládat do textových souborů (tzv. skriptů). Tyto skripty mají příponu *.vic*. Pro práci se skripty slouží menu *Skript* - umožňuje skripty nejen ukládat a otevírat, ale lze zvolit i font (čez písma), kterým budou skripty v levé části hlavního okna programu zobrazovány.

Menu Výkres

Další sekci menu je *Výkres*. Umožňuje (jak již bylo uvedeno výše) překreslení výkresu na základě příkazů skriptu (*Výkres-Překreslit*). Výkresovou část okna lze vymazat (*Výkres-Vymazat*). Pro zobrazení a skrývání vrstev výkresu slouží menu *Výkres-Zobrazení vrstev*. Dále můžeme výkres uložit jako obrázek (*Výkres-Uložit výkres jako*) nebo vytisknout na tiskárni (*Výkres-Vytisknout výkres*).

Obrázky výkresů lze ukládat ve třech formátech (BMP, JPG a GIF). Rastrový obrázek je vždy tak velký, jak velká je kreslicí plocha. Pokud chceme obrázek zvětšit, nezvětšujeme BMP (nebo JPG, GIF) v nijakém grafickém editoru (došlo by tím k zhoršení kvality obrázku)! Lepší je obrázek/výkres zvětšit rovnou v programu *ScriptCad* (na to stačí jediný příkaz *PixelsPerUnit* - viz sekce [Příkazy a jejich popis](#)) a uložit jej rovnou v takovém rastru (rozlišení), jaké potřebujeme. Je možné volit i barevnou hloubku ukládaných rastrových obrázků (blíže je popsáno v sekci [Nastavení programu](#) - parametr *BitsPerPixel*).

Při tisku na tiskárni je výkres **vždy** roztažen na celou tiskovou plochu (např. A4). Přitom se samozřejmě zachovávají poměry stran výkresu. Pokud výkres vytištěný na tiskárni je moc hrubý (moc jemný), lze experimentovat s parametrem *Stretch*, viz sekce [Nastavení programu](#).

Menu Nápověda

Z této části menu lze spouštět soubor nápovědy k programu (*Nápověda-Obsah*) a vyhledávat v něm (*Nápověda-Hledat*). Také je možno spustit dialogové okno *O programu* (*Nápověda-O programu*).

Důležitou součástí této sekce menu je možnost přepínat mezi jazyky (*Nápověda-Jazyk*). Lze si tedy vybrat, v jakém jazyce bude program *ScriptCad* komunikovat. Ve verzi *ScriptCad 1.00* je k dispozici angličtina a čeština.

Soubor nápovědy k programu *ScriptCad* je možné spouštět i klávesou *F1*.

Visuální návrhy výkresů

Pro visuální návrh/kreslení je určeno kontextové menu nad výkresem (pravou částí hlavního okna programu). Toto menu se objeví po kliknutí pravým tlačítkem myši nad zmiňovanou oblastí. U každé položky v tomto menu je také uvedeno, kolika kliknutími myši uvedený prvek vznikne. U některých prvků (např. komentář) vznikne uvedený prvek ihned (bez dalšího klikání myši). U jiných prvků (např. bod) program očekává ještě jeden klik na výkresovou oblast, aby viděl, kam má uvedený prvek umístit. U složitějších prvků (např. úsečka) program očekává dva kliky myši nad výkresovou oblastí. První klik vyznačí počátek a druhý klik vyznačí konec (úsečky).

Menu obsahuje prvky:

- **úsečka**

Do skriptové oblasti (levá část hlavního okna) přidá příkaz, který vykreslí úsečku. Pro umístění úsečky do výkresu je nutné 2x kliknout do výkresové oblasti (levým tlačítkem myši). První kliknutí vyznačí počáteční bod úsečky, druhé kliknutí vyznačí konec úsečky a celou operaci dokončí. Dokončením operace se rozumí automatické vypsání příkazu do skriptové oblasti a jeho vykonání ve výkresové oblasti.

- **kóta**

Do skriptu přidá příkaz, který vykreslí úsečku s šipkami na konci (tzv. kóta). Pro umístění kóty do výkresu je nutné opět 2x kliknout do výkresové oblasti pro vyznačení počátečního a koncového bodu.

- **text**

Do skriptu přidá příkaz, který umístí text na místo, jež vyznačíme kliknutím myši (1x). Znění textu se upraví ve skriptové oblasti.

- **bod**

Do skriptu přidá příkaz na vykreslení bodu. Pro umístění bodu do výkresu je nutné 1x kliknout do výkresové oblasti.

- **výplň**

Vyšrafovuje zadanou oblast. Je nutné kliknutím myši (1x) vyznačit bod v nějaké uzavřené oblasti. Je vyšrafována celá takto vyznačená oblast (oblast uzavírají body jiné barvy, než má bod oblast vyznačující).

- **obdelník**

Do skriptu přidá příkaz, který vykreslí obdelník. Pro umístění obdelníku do výkresu je nutné 2x kliknout myši do výkresové oblasti. První kliknutí vyznačí levý horní roh obdelníku, druhé kliknutí vyznačí pravý dolní roh obdelníku.

- **kružnice**

Do skriptu přidá příkaz, který vykreslí kružnici. Pro umístění kružnice do výkresu je nutné 2x kliknout myši do výkresové oblasti. První kliknutí vyznačí střed kružnice, druhé kliknutí vyznačí jeden z bodů na kružnici.

- **část kružnice**

Do skriptu přidá příkaz, který vykreslí část kružnice (čtvrt oblouku). Pro umístění části kružnice do výkresu je nutné 2x kliknout myši do výkresové oblasti. První kliknutí vyznačí střed oblouku, druhé kliknutí vyznačí jeden z bodů oblouku. Přitom umístím bodu na oblouku určíme nejenom poloměr oblouku, ale je tím určen i kvadrant, v kterém je kruhový oblouk vykreslen.

- **elipsa**

Do skriptu přidá příkaz, který vykreslí elipsu. Pro umístění elipsy do výkresu je nutné 2x kliknout myši do

výkresové oblasti. Tato dvě kliknutí vyznačí obdelník (viz výše), do kterého je elipsa vepsána.

- **obrázek**

Do skriptu přidá příkaz pro vložení obrázku. Cesta k obrázku může být buď absolutní (např. "C:\SomeFolder\example.gif") nebo relativní (např. "example.gif"). Pokud je uvedena cesta relativní, tak soubor s obrázkem je hledán buď v adresáři s programem *ScriptCad.exe* (pokud skript ještě není uložen), nebo v adresáři s uloženým (a právě editovaným) skriptem (.viz soubor).

- **globální proměnné**

Do skriptu přidá sekci, která nastavuje nejdůležitější parametry pro (budoucí) výkres. **Každý výkres by měl začínat touto sekci** (jinak budou zvoleny defaultní parametry, které později nemusí být optimální). Pro umístění sekce nejsou vyžadovány dodatečné kliky myši na výkresovou oblast. Více informací o globálních proměnných získáte v sekci [Globální proměnné](#).

- **vrstva**

Do skriptu přidá příkaz pro přepnutí vrstvy. Číslo vrstvy je nutné napsat ručně přímo do skriptu. Přípustná čísla vrstev jsou celá čísla od 1 do 9.

- **procedura**

Do skriptu přidá šablonu pro vytvoření procedury. Poté je nutné ručně upravit název procedury a počet proměnných a je nutné definovat tělo procedury (tedy co se má při použití procedury kreslit). Více informací o procedurách získáte v sekci [Procedury jako definice vlastních objektů](#).

- **komentář**

Do skriptu přidá komentář, tedy řádek začínající znakem % (procento). Vše od tohoto znaku až do konce řádku bude při vykreslování výkresu ignorováno.

- **zruš kreslící akci**

Zruší rozpracovanou akci. Tuto volbu zvolíme, pokud uprostřed nějaké akce (např. kreslení úsečky - ještě před druhým kliknutím) zjistíme, že si jí nepřejeme dokončit.

POZN: Pokud po vykreslení objektu zjistíme, že jeho parametry nejsou správné (např. se nenachází úplně přesně na požadovaném místě), lze parametry doladit v levé části hlavního okna - tedy přímo ve vytvářeném skriptu. Ve skriptové oblasti lze příkazy i mazat nebo vytvářet tak, jak je popisováno v sekci [Příkazy a jejich popis](#). Po ruční editaci příkazu spustíme překreslení výkresu volbou *Výkres-Překreslit*.

POZN: Visuální způsob kreslení (pomocí kontextového menu nad výkresovou oblastí) je sice přímým, ale přece jen ne úplně plnohodnotným způsobem vytváření výkresu. Při vizuálním návrhu je problémem se přesně trefovat do míst, kam objekt skutečně chceme, a objekty je nutné v levé části hlavního okna stejně dolaďovat. Visuální způsob návrhu není tak flexibilní. Při přímém psaní příkazů lze v parametrech používat více voleb. Více informací o přímém psaní příkazů naleznete v sekci [Příkazy a jejich popis](#).

Příkazy a jejich popis

Přestože vizuální způsob tvorby výkresu (viz sekce [Visuální návrhy výkresů](#)) je snazší, neumožňuje tak přesné umístění objektů do výkresu ani takovou flexibilitu parametrů příkazů. Při vytváření výkresu přímým psaním příkazů (do skriptové sekce v levé části hlavního okna) budete mít nad kreslením plnou kontrolu.

Souřadný systém

Program *ScriptCad* je určen pro 2D kreslení. Souřadný systém proto vytvářejí dvě osy - x a y . **Počátek souřadného systému** (tedy bod $[0,0]$) je v **horním levém rohu** výkresové oblasti.

X -ová osa je (vodorovná) horní hrana výkresové oblasti a x -ové jednotky se zvětšují zleva do prava (tedy na pravé (svislé) hraně výkresové oblasti jsou maximální hodnoty x , na levé (svislé) hraně výkresové oblasti je $x=0$).

Y -ová osa je (svislá) levá hrana výkresové oblasti a y -ové jednotky se zvětšují shora dolů (tedy na dolní (vodorovné) hraně výkresové oblasti jsou maximální hodnoty y , na horní (vodorovné) hraně výkresové oblasti je $y=0$).

Při přejezdění myší nad výkresovou oblastí je aktuální hodnota (x,y) zobrazována ve spodní části hlavního okna. Souřadnicové hodnoty jsou měřeny v jednotkách, které udává proměnná *PixelsPerUnit* (více informací o této a dalších globálních proměnných naleznete v sekci [Globální proměnné](#)).

Globální proměnné

Každý výkres by měl začínat sekcí, kde jsou nastaveny nejdůležitější (obecné a pro všechny další příkazy společné) proměnné výkresu (globální proměnné). Tyto globální proměnné jsou velmi důležité a je jim věnována samostatná sekce [Globální proměnné](#).

Komentář

Pokud je kdekoli ve skriptu znak % (procento), vše od tohoto znaku až na konec řádku je při zpracování skriptu (tj. kreslení výkresu) ignorováno.

Příkazy pro geometrické útvary

- [Úsečka - příkaz Line](#)
- [Kóta - příkaz DimLine](#)
- [Text - příkaz Text](#)
- [Bod - příkaz Point](#)
- [Obdelník - příkaz Rectangle](#)
- [Výplň - příkaz FillArea](#)
- [Kružnice - příkaz Circle](#)
- [Část kružnice - příkaz CirclePart](#)
- [Elipsa - příkaz Ellipse](#)

Globální proměnné

Jedná se o proměnné nastavující velikost výkresové sekce (proměnné *MaximumX*, *MaximumY*, *PixelsPerUnit*), barvu pozadí výkresu (proměnná *BackgroundColor*), defaultní (přednastavené) hodnoty pro barvu tužky (proměnná *DefaultColor*), font písma textu (proměnná *DefaultFontName*), velikost písma (proměnná *DefaultFontSize*), velikost úhlů šipek a délek šipek u kót (proměnné *DefaultDimAngle* a *DefaultDimLength*) a posuvy výkresu (proměnné *ShiftX* a *ShiftY*).

MaximumX

Nastaví šířku výkresové oblasti v jednotce specifikované pomocí *PixelsPerUnit*.

MaximumY

Nastaví výšku výkresové oblasti v jednotce specifikované pomocí *PixelsPerUnit*.

PixelsPerUnit

Udává, kolik pixelů bude jedna jednotka. Např. *PixelsPerUnit=10* znamená, že za jednotku je považováno 10 px. V těchto jednotkách jsou potom udávány všechny rozměry (viz další geometrické obrázky). Proměnná *PixelsPerUnit* umožňuje kdykoliv (i dodatečně) zoomování výkresu. Např. po zdvojnásobení *PixelsPerUnit* se zdvojnásobí úplně všechny rozměry ve výkresu a výkresová plocha se zvětší.

BackgroundColor

Nastaví barvu pozadí výkresu (defaultní přednastavena bílá - *White*).

DefaultColor

Nastaví barvu kreslicího pera (defaultní přednastavena černá - *Black*). Touto barvou budou kresleny všechny objekty, u kterých nebude výslovně uvedeno jinak.

DefaultFontName

Nastaví font písma (defaultní písmo "*Arial*"). Název fontu musí být v uvozovkách.

DefaultFontSize

Nastaví velikost písma (defaultní 8). Při volbě *PixelsPerUnit=1* toto nastavení *DefaultFontSize* vyhovuje, při nastavení *PixelsPerUnit* na vyšší hodnotu je nutné *DefaultFontSize* úměrně zmenšit!

DefaultDimAngle

Při kreslení kót je nastaven úhel u šipky (defaultní 10 stupňů).

DefaultDimLength

Při kreslení kót je nastavena délka šipky (defaultní 10 jednotek). Při volbě *PixelsPerUnit=1* toto nastavení *DefaultDimLength* vyhovuje, při nastavení *PixelsPerUnit* na vyšší hodnotu je nutné *DefaultDimLength* úměrně zmenšit!

ShiftX

Posune výkres ve směru x-ové souřadnice (o daný počet jednotek - závisí také na *PixelsPerUnit*). Tímto způsobem lze výkres celý posunout bez toho, abychom museli měnit souřadnice ve všech objektech. **Je nutné ale upozornit, že po posuvech pomocí ShiftX a ShiftY nebudou souhlasit souřadnice, které jsou uváděny jako pomůcka pod výkresovou částí.**

ShiftY

Posune výkres ve směru y-ové souřadnice.

POZN: V dalších příkazech (jako Line, Circle apod.) lze globálních proměnných *DefaultColor*, *DefaultDimAngle*, *DefaultDimLength*, *DefaultFontName* a *DefaultFontSize* používat namísto konkrétních

hodnot. Je-li napø. *DefaultColor=Blue*, pak øíkaz *Point 124, 169, DefaultColor* je totéž jako *Point 124, 169, Blue*.

Pozor! Promínné *MaximumX*, *MaximumY*, *PixelsPerUnit* a *BackgroundColor* jsou zásadní. Po øeenastavení libovolné z těchto promínných dojde k vymazání výkresové èásti (pravá èást hlavního okna). Proto tyto promínné musí být uvedeny hned na zaèátku skriptu.

POZN: X-ový rozmír výkresu v pixelech je tedy $MaximumX * PixelsPerUnit$. Y-ový rozmír výkresu v pixelech je $MaximumY * PixelsPerUnit$.

POZN: Pokud rozmír výkresu v pixelech je malý, mohou (na pohled) vznikat defekty dané øíliš nízkým rozlišením rastru. Tyto (pouze vizuální) defekty zmizí, zvítší-li se rastr (napø. zvítšením hodnoty promínné *PixelsPerUnit*).

POZN: Promínnou *PixelsPerUnit* nastavujte vždy až za promínnými *MaximumX* a *MaximumY*!! Jinak může nastat situace, kterou nejlépe vysvitlíme na øíkladu: nech

```
PixelsPerUnit = 1
MaximumX = 500
MaximumY = 400
```

Tedy proti doporuèení je *PixelsPerUnit* hned na prvním místì. Co se stane? První bìh skriptu nastaví plochu na *500 px x 400 px*. Pak se rozhodneme zmínit jednotky, ve kterých bude výkres kreslen a napíšeme

```
PixelsPerUnit = 100
MaximumX = 5
MaximumY = 4
```

Vznikne sice také plocha *500 px x 400 px*, ale způsob, jakým se tak stane, je poněkud nevhodný. Skript je totiž provádín po jednotlivých øíkazech, takže první øíkaz *PixelsPerUnit=100* způsobí, že bude vytváøena plocha *50000 px x 40000 px* (protože je stále ještě *MaximumX = 500* a *MaximumY = 400*). Následní bude plocha sice zase zmenšena, ale k tomu už nemusí vůbec dojít, protože øedchozí neúmnírné zvítšení plochy vyèerpalo veškerou pamì operáèního systému a program ohlásí chybu!

Úsečka - příkaz Line

Obecný tvar: Line x1, y1, x2, y2, TypĚáry, TloušťkaĚáry, BarvaĚáry

Minimální verze příkazu: Line x1, y1, x2, y2

Konkrétní příklad: Line 87, 71, 320, 226, Solid, 1, Blue

(x1, y1) ... počáteční bod úsečky (1234)

(x2, y2) ... koncový bod úsečky (1234)

TypĚáry ... viz Typy čar

TloušťkaĚáry ... viz Tloušťka čar

BarvaĚáry ... viz Barvy

Pokud některé parametry chybí (srovnej obecný tvar příkazu s minimální verzí příkazu), je vykreslena úsečka, kde TypĚáry je *Solid*, TloušťkaĚáry je *1* a BarvaĚáry je *DefaultColor* z globálních proměnných.

Kóta - příkaz DimLine

Obecný tvar: DimLine x1, y1, TypŠipky1, x2, y2, TypŠipky2, TypĚáry, TloušťkaĚáry, BarvaĚáry, ÚhelŠipky, DélkaŠipky

Minimální verze příkazu: DimLine x1, y1, TypŠipky1, x2, y2, TypŠipky2

Konkrétní příklad: DimLine 104, 232, Inner, 310, 52, Inner, Solid, 1, Black, 10, 8

(x1, y1)	... počáteční bod kóty (<u>1234</u>)
TypŠipky1	... <u>Typ kóty</u> u počátečního bodu
(x2, y2)	... koncový bod kóty (<u>1234</u>)
TypŠipky2	... <u>Typ kóty</u> u koncového bodu
TypĚáry	... viz <u>Typy čar</u>
TloušťkaĚáry	... viz <u>Tloušťka čar</u>
BarvaĚáry	... viz <u>Barvy</u>
ÚhelŠipky	... je úhel (ve stupních) mezi hlavní úsečkou a šipkou (v koncovém bodě kóty)
DélkaŠipky	... je délka (v jednotkách daných promínnou <i>PixelsPerUnit</i>) šipky kóty (<u>1234</u>)

Pokud některé parametry chybí, je vykreslena kótovací čára, kde TypĚáry je *Solid*, TloušťkaĚáry je *1*, BarvaĚáry je globální promínná *DefaultColor*, ÚhelŠipky je globální promínná *DefaultDimAngle* a DélkaŠipky je globální promínná *DefaultDimLength*.

Text - příkaz Text

Obecný tvar: Text x, y, Vita, IsBold, IsItalic, IsUnderline, IsStrikeOut, JménoFontu, BarvaTextu, VelikostPísma, ÚhelTextu

Minimální verze příkazu: Text x, y, Vita

Konkrétní příklad: Text 179, 225, "Text je zde", 0,0,0,0, "Arial", Red, 12, 90

(x, y) ... bod, na kterém se nachází počátek textu (1234)
Vita ... text, který bude zobrazen (celý text musí být v uvozovkách) (texty lze i spojovat "ah" + "oj")
IsBold ... udává, zda je text tučný (IsBold=1) nebo obyčejný (IsBold=0)
IsItalic ... udává, zda je text kurzívou (IsItalic=1) nebo obyčejný (IsItalic=0)
IsUnderline ... udává, zda je text podtržený (IsUnderline=1) nebo obyčejný (IsUnderline=0)
IsStrikeOut ... udává, zda je text přeškrtnutý (IsStrikeOut=1) nebo obyčejný (IsStrikeOut=0)
JménoFontu ... jméno fontu, kterým bude text vysázen. Jméno fontu musí být uzavřeno do uvozovek

(např. "MS Sans Serif"). Font musí být přítomen v operačním systému

(Windows).

Seznam všech dostupných fontů získáte prostřednictvím menu *Skript-Font*.

BarvaTextu ... viz Barvy
VelikostPísma ... velikost písma (v jednotkách daných proměnnou *PixelsPerUnit*) (1234)
ÚhelTextu ... natožení textu. Texty lze psát nejen vodorovně (ÚhelTextu=0), ale i svisle (ÚhelTextu=90) nebo pod jakýmkoliv celočíselným úhlem (ve stupních).

Pokud některé parametry chybí, je zobrazen text, kde IsBold=0, IsItalic=0, IsUnderline=0, IsStrikeOut=0, JménoFontu je globální proměnná *DefaultFontName*, BarvaTextu je globální proměnná *DefaultColor*, VelikostPísma je globální proměnná *DefaultFontSize* a ÚhelTextu je 0 stupňů.

Bod - příkaz Point

Obecný tvar: Point x, y, Barva

Minimální verze příkazu: Point x, y

Konkrétní příklad: Point 201, 182, Green

(x, y) ... souřadnice bodu (1234)

Barva ... barva bodu

Pokud parametr Barva chybí, je jako barva použita globální proměnná *DefaultColor*.

Obdelník - příkaz Rectangle

Obecný tvar: Rectangle x1, y1, x2, y2, TypĚáry, TloušřkaĚáry, BarvaĚáry

Minimální verze příkazu: Rectangle x1, y1, x2, y2

Konkrétní příklad: Rectangle 101, 233, 185, 293, Dot, 1, Gray

(x1, y1) ... levý horní vrchol obdelníku (1234)
(x2, y2) ... pravý dolní vrchol obdelníku (1234)
TypĚáry ... viz Typy ěar
TloušřkaĚáry ... viz Tloušřka ěar
BarvaĚáry ... viz Barvy

Pokud některé parametry chybí, je vykreslena obdelník, kde TypĚáry je *Solid*, TloušřkaĚáry je 1 a BarvaĚáry je *DefaultColor* z globálních proměnných.

Výplň - příkaz FillArea

Obecný tvar: FillArea x, y, TypŠrafování, BarvaŠrafování

Minimální verze příkazu: FillArea x, y

Konkrétní příklad: FillArea 179, 75, BDiagonal, Gray

TypŠrafování ... viz Typy šrafování

BarvaŠrafování ... viz Barvy

(x, y) ... souřadnice bodu, který je uvnitř uzavřené oblasti, která se má vyšrafovat.

Oblast uzavírají body jiné barvy, než má bod (x, y), který oblast vyznačuje (1234).

Pokud některé parametry chybí, je za TypŠrafování použit Solid a BarvaŠrafování je *DefaultColor* z globálních proměnných.

Kružnice - příkaz Circle

Obecný tvar: Circle x, y, Poloměr, TypČar, TloušťkaČar, BarvaČar

Minimální verze příkazu: Circle x, y, Poloměr

Konkrétní příklad: Circle 269, 293, 79, Solid, 1, NavyBlue

(x, y) ... souřadnice středu kružnice ([1234](#))

Poloměr ... poloměr kružnice (v jednotkách daných promínnou *PixelsPerUnit*) ([1234](#))

TypČar ... viz [Typy čar](#)

TloušťkaČar ... viz [Tloušťka čar](#)

BarvaČar ... viz [Barvy](#)

Pokud některé parametry chybí, je vykreslena kružnice, kde TypČar je *Solid*, TloušťkaČar je *1* a BarvaČar je *DefaultColor* z [globálních proměnných](#).

Část kružnice - příkaz CirclePart

Obecný tvar: CirclePart x, y, Polomir, OdÚhlu, DoÚhlu, TypČáry, TlouškaČáry, BarvaČáry

Minimální verze příkazu: CirclePart x, y, Polomir, OdÚhlu, DoÚhlu

Konkrétní příklad: CirclePart 69, 239, 60, 90, 180, Solid, 1, Black

(x, y) ... souřadnice středu kružnice ([1234](#))
Polomir ... polomir kružnice (v jednotkách daných promínnou *PixelsPerUnit*) ([1234](#))
OdÚhlu, DoÚhlu ... udává ve stupních, jak velká část kružnice bude vykreslena ([1234](#)).
1. kvadrant odpovídá rozmezí úhlů 0 až 90 stupňů,
2. kvadrant je 90-180 stupňů,
3. kvadrant je 180-270 stupňů,
4. kvadrant je 270-360 stupňů.
Zadávat lze pouze celá čísla.
TypČáry ... viz [Typy čar](#)
TlouškaČáry ... viz [Tloušťka čar](#)
BarvaČáry ... viz [Barvy](#)

Pokud některé parametry chybí, je vykreslena část kružnice, kde TypČáry je *Solid*, TlouškaČáry je *1* a BarvaČáry je *DefaultColor* z [globálních proměnných](#).

Elipsa - příkaz Ellipse

Obecný tvar: Ellipse x1, y1, x2, y2, TypČáry, TloušťkaČáry, BarvaČáry

Minimální verze příkazu: Ellipse x1, y1, x2, y2

Konkrétní příklad: Ellipse 125, 197, 191, 232, Solid, 1, Blue

(x1, y1) ... levý horní vrchol obdelníku, který elipsu vymezuje (1234)

(x2, y2) ... pravý dolní vrchol obdelníku, který elipsu vymezuje. Elipsa je do obdelníku vepsána (1234).

TypČáry ... viz Typy čar

TloušťkaČáry ... viz Tloušťka čar

BarvaČáry ... viz Barvy

Pokud některé parametry chybí, je vykreslena elipsa, kde TypČáry je *Solid*, TloušťkaČáry je *1* a BarvaČáry je *DefaultColor* z globálních proměnných.

Barvy

V programu *ScriptCad* jsou k dispozici následující barvy geometrických objektů:

Aqua, Black (černá), Blue (modrá), DarkGray, Fuchsia, Gray (šedá), Green (zelená),
LimeGreen, LightGray, Maroon, NavyBlue, OliveGreen, Purple, Red (červená), Silver (stříbrná), Teal,
White (bílá), Yellow (žlutá).

POZN: Namísto konkrétní barvy (např. Black) lze používat i globální proměnnou *DefaultColor*. Je-li např. *DefaultColor=Blue*, pak příkaz *Point 124, 169, DefaultColor* je totéž jako *Point 124, 169, Blue*.

Tloušťka čar

Tloušťka čar je jako jediná v programu *ScriptCad* udávána v absolutních jednotkách (pixels), nikoliv v jednotkách relativních (daných proměnnou *PixelsPerUnit*). Základní tloušťkou čar je *1 px* - pouze u této nejtenější tloušťky jsou k dispozici i čary čárkované nebo tečkované. U větších tloušťek čar jsou k dispozici jenom plné čary. Více informací o typech čar naleznete v sekci [Typy čar](#).

Typy čar

V programu *ScriptCad* jsou k dispozici následující typy čar:

Solid	... plná čára
Dash	... čárkovaná čára
Dot	... tečkovaná čára
DashDot	... čára, kde se střídají čárky a tečky
DashDotDot	... čára, kde se střídají čárky a dvojité tečky

POZN: Jiný typ čáry než plná (tedy typy Dash, Dot, DashDot, DashDotDot) je možné použít jen tehdy, když tloušťka čáry je 1!

Typy šrafovaní

U výplní uzavřených oblastí lze používat tyto typy šrafovaní:

Solid	... plná výplň
BDiagonal	... diagonální šrafovaní (šrafy stoupají)
FDiagonal	... diagonální šrafovaní (šrafy klesají)
Cross	... dlaždice
DiagCross	... dlaždice diagonální
Horizontal	... horizontální šrafy
Vertical	... vertikální šrafy

Typy kót

U kót lze používat tři typy šipek:

Inner	... vnitřní šipky u kót
Outer	... vnější šipky u kót
None	... kótovací čára bez šipky (na příslušném konci)

Nastavení programu (ScriptCad.ini)

Program si ukládá svoje nastavení do souboru *ScriptCad.ini*. Tento soubor se nalézá ve stejném adresáři jako program *ScriptCad.exe*. Soubor musí být k dispozici i pro zápis. Pokud zápis do souboru *ScriptCad.ini* nebude umožněn (např. program *ScriptCad* bude umístěn rovnou na CD nebo DVD), nastavení programu se neuloží a při dalším spuštění programu budou nastaveny opět defaultní hodnoty parametrů.

ScriptCad.ini obsahuje 4 sekce.

V sekci **[MainForm]** se ukládá velikost a poloha hlavního okna programu *ScriptCad* na ploše Windows (proměnné *Height*, *Width*, *Left* a *Top*), program si pamatuje i velikost levé části (se skriptem) hlavního okna (proměnná *Splitter*). Pro uložení jazykového nastavení slouží proměnná *Language* (1000 pro angličtinu a 2000 pro češtinu). Proměnná *DemoScript* indikuje, zda již byl při prvním spuštění programu *ScriptCad* spuštěn i demonstrační skript *demo.vic*.

V sekci **[ScriptFont]** jsou uloženy informace o fontu (čezu písma), kterým jsou zobrazovány skripty v levé části hlavního okna.

V sekci **[Layers]** jsou uloženy informace o zobrazení/skrytí vrstev výkresu.

V sekci **[Print]** lze vyladit tiskové parametry.

Proměnná *BitsPerPixel* udává, v jaké barevné hloubce budou ukládány obrázky/výkresy při volbě *Výkres-Uložit výkres jako*. Defaultní je nastaveno *BitsPerPixel=4*, což znamená 4-bitovou barevnou hloubku (tj. 16 barev). Vzhledem k možnostem nastavení barev ve výkresu (viz sekci Barvy) je takové množství barev optimální. Je ale možné nastavit *BitsPerPixel* i na jiné hodnoty:

BitsPerPixel=8 pro 256 barev,
BitsPerPixel=16 pro 65536 barev
BitsPerPixel=24 pro 16777216 barev a
BitsPerPixel=32 pro 4294967296 barev.

Proměnná *Stretch* udává, kolikrát bude zoomován výkres při tisku v tiskárně.

Situaci vysvětlíme na příkladu: nechť x-ová velikost výkresu v pravé části hlavního okna programu *ScriptCad* je 100 pixelů a x-ové rozlišení tiskárny 6000 pixelů (závisí na formátu stránky v tiskárně - např. A4 - a dále závisí na kvalitě (dpi) tisku). Y-souřadnice výkresu ani tiskárny pro jednoduchost neuvažujeme.

Při defaultním nastavení *Stretch=4* bude napřed výkres interní v programu *ScriptCad* zoomován z výchozí hodnoty x-ové velikosti obrázku 100 px na hodnotu 1500 px, a to vektorově (bez ztráty kvality), podobně jako je tomu při nastavování proměnné *PixelsPerUnit* (viz sekce Globální proměnné). Hodnota 1500 px je proto, že je 4x menší (*Stretch=4*) než x-ové rozlišení tiskárny (zde 6000 px). Takto zvětšený výkres (1500 px) je pak v tiskárně natažen ještě 4x (*Stretch=4*) a bude vytištěn s co možná největším využitím plochy tisku (zde A4). Hodnotu *Stretch* lze volit v rozmezí 1 (obrázek bude příliš jemný, především šrafovaní) až 10 (obrázek bude příliš hrubý - "zubatý").

Do sekcí *[MainForm]*, *[ScriptFont]* a *[Layers]* je zapisováno automaticky poté, co příslušnou vlastnost uživatel v programu *ScriptCad* změní, zatímco sekci *[Print]* je nutno nastavovat ručně přímo v *ScriptCad.ini*. Je tomu tak proto, že autor programu považoval nastavení sekce *[Print]* za natolik nízkourovňové, že raději neumožnil změnu rovnou z programu *ScriptCad*. Nové hodnoty proměnných *BitsPerPixel* a *Stretch* nastavené v sekci *[Print]* se projeví při novém startu programu *ScriptCad*.

Registrace

Program *ScriptCad* je sharewareová aplikace. Zkušební lhůta je 30 dní. Pokud se rozhodnete ponechat si program i po zkušební době, musíte si program zaregistrovat, a to na e-mailové adrese **vincze@ji.cz**. Jedna licence (licence pro jeden počítač) stojí 10 USD (nebo 250 Kč). Registrační klíč, který Vám bude následně e-mailem zaslán, má podobu souboru *license.dll*. Zasláný soubor nakopírujete do adresáře k programu *ScriptCad.exe*. Registrační klíč bude platný i pro všechny vyšší verze programu *ScriptCad* (tedy nákup upgrade programu nebude nutný). Nové verze programu si lze stáhnout z webových stránek autora na adrese **<http://vincze.czweb.org>**.

Licenční podmínky

Software je majetkem autora Ing. Romana VINCZE. Uživatel bere na vědomí, že software je autorským dílem a je chráněn zákony České republiky o autorském právu, ustanovením mezinárodních smluv a všemi dalšími odpovídajícími zákony. Uživatel je povinen dbát autorské ochrany softwaru a jeho doplňků, jako např. dokumentace, zejména do ní nezasahovat a neměnit její obsah. Nesmíte provádět zpětný překlad a analýzu, dekompileovat nebo disasemblovat software. Uživatel je oprávněn pořídit si kopie díla výhradně pro jeho zálohování a ochranu před ztrátou.

ŽÁDNÉ ZÁRUKY.

Autor se úmyslně vzdává jakéhokoli ručení za program *ScriptCad*. Program *ScriptCad* je poskytován "tak jak je" bez jakékoli záruky. Veškeré riziko vyplývající z používání či fungování programu *ScriptCad* nese jeho uživatel.

ŽÁDNÁ ODPOVĚDNOST ZA PŘÍPADNÉ ŠKODY.

Autor programu neponese za žádných okolností odpovědnost za jakékoli škody (a to včetně a bez omezení za škody v podobě ztrát na zisku, obchodního úpadku, ztráty obchodních informací či jakékoli jiné peněžní ztráty) vyplývající z používání či nesprávného používání tohoto produktu.

Práce s vrstvami

Program *ScriptCad* podporuje kreslení ve vrstvách. Vrstev může být maximální 9. Kreslení ve vrstvi se zahajuje příkazem *Layer = CisloVrstvy*, např. *Layer = 2*. *CisloVrstvy* musí být celé číslo mezi 1 a 9 včetně. Příklad skriptu:

příkaz (kreslí v 1. vrstvě)

Layer = 2

příkaz (kreslí v 2. vrstvě)

příkaz (kreslí v 2. vrstvě)

příkaz (kreslí v 2. vrstvě)

Layer = 3

příkaz (kreslí v 3. vrstvě)

Layer = 1

příkaz (kreslí v 1. vrstvě)

příkaz (kreslí v 1. vrstvě)

Příkazy, které nastavují hodnoty globálních proměnných, jsou platné napříč všemi vrstvami. Např. nastavení proměnné *MaximumX* je platné pro celý zbytek výkresu (všechny vrstvy). Naproti tomu příkazy jako Line nebo Rectangle apod. se týkají jen jediné vrstvy (té, ve které jsou uvedeny).

Pro zobrazování/skrývání vrstev slouží menu *Výkres-Zobrazení vrstev* (nebo checkboxy *Vrstvy* na toolbaru). Lze zobrazit libovolnou kombinaci z 9 vrstev.

POZN: Příkaz pro přepnutí vrstev lze přidat nejen ručním napsáním do skriptu, ale i pomocí kontextového menu nad výkresovou oblastí (číslo vrstvy je ale nutné mít ručně).

POZN: Při vizuálním návrhu (kreslení pomocí kontextového menu nad výkresovou oblastí) se objekt vykreslí (ve výkresové oblasti) bez ohledu na vrstvy. Teprve po zadání *Výkres-Překreslit* je objekt zařazen (vykreslen) do příslušné vrstvy.

Obrázek - příkaz Picture

Obecný tvar: Picture x, y, Cesta, Šířka, Výška

Minimální verze příkazu: Picture x, y, Cesta

Konkrétní příklad: Picture 15, 100, "C:\SomeFolder\example.gif", 19, 19

(x, y) ... souřadnice levého horního rohu obrázku (v jednotkách daných proměnnou *PixelsPerUnit*) (1234)

Cesta ... cesta k obrázku na pevném disku (musí být v uvozovkách)

Šířka ... šířka obrázku (v jednotkách daných proměnnou *PixelsPerUnit*) (1234)

Výška ... výška obrázku (v jednotkách daných proměnnou *PixelsPerUnit*) (1234)

V parametru *Cesta* se může vyskytovat buď cesta absolutní (např. "C:\SomeFolder\example.gif") nebo relativní (např. "example.gif"). Pokud je uvedena cesta relativní, tak soubor s obrázkem je hledán buď v adresáři s programem *ScriptCad.exe* (pokud skript ještě není uložen), nebo v adresáři s uloženým (a právě editovaným) skriptem (.vic soubor).

Lze vkládat obrázky ve formátech *bmp*, *gif*, *jpg* a *jpeg*.

Parametry *Šířka* a *Výška* mohou chybět. Potom je obrázek vložen ve skutečném rozlišení a jeho šířka a výška na výkrese se nebude měnit ani při změně proměnné *PixelsPerUnit*. Pokud jsou parametry *Šířka* a *Výška* zadány, pak je obrázek do vyhrazené plochy rovnoměrně roztažen.

POZN: Vkládané obrázky by neměly mít více jak 16 barev, protože kreslicí plocha v programu *ScriptCad* má (defaultně) nastavenou 4-bitovou barevnou hloubku. Pokud bude do kreslicí plochy se 4-bitovou barevnou hloubkou vložen obrázek, který má barev více, budou barvy zredukovány. Pokud požadujete vložení obrázku s větší než 4-bitovou barevnou hloubkou se současným zachováním barev i na výkrese, je nutné v souboru *ScriptCad.ini* v sekci *[Print]* nastavit proměnnou *BitsPerPixel* - více informací najdete v sekci Nastavení programu.

Vyčíslitelné výrazy

Pod pojmem vyčíslitelný výraz rozumíme výraz, jehož hodnotu lze vyjádřit číselně. Může to být číslo (např. 5 nebo -3.2), ale může to být i složenina čísel a jednoduchých matematických operací, např.

$$3 + 5$$

$$3 * 2.3$$

$$3 - 2 * 5$$

$$3 / (1 + 2)$$

Jednoduchými matematickými operacemi rozumíme sčítání, odčítání, násobení a dělení. Přitom je možné použít (kulaté) závorky.

Příklad: Zapišeme-li

*Line 100 + 50, 50 * 2, 300 - 100, 500/10, Solid, 1, DefaultColor*

je to jako bychom zapsali:

Line 150, 100, 200, 50, Solid, 1, DefaultColor

Vyčísitelnost výrazů je důležitá pro Procedury jako definice vlastních objektů. V této dokumentaci jsou vyčíslitelné parametry u jednotlivých příkazů vyznačeny značkou (1234).

Procedury jako definice vlastních objektů

Pomocí procedur lze vytvořit vlastní objekty pro kreslení. Po vytvoření procedury je možné tuto využívat jako by to byl nativní příkaz programu *ScriptCad*.

Deklarace procedury

Procedura začíná klíčovým slovem *Procedure*, za kterým následuje název procedury a výčet proměnných. Proměnné jsou odděleny čárkami a **názvy proměnných musí začínat znakem #**. Na dalších řádcích se uvádí obsah (tílo) procedury, což jsou příkazy, které se vykonávají (vykreslují) při použití procedury. Definice procedury končí klíčovým slovem *End*.

Příklad deklarace procedury

Použití procedur ukazuje demonstrační skript *demo2.vic*. Tento skript mimo jiné obsahuje následující definici procedury (procedura nakreslí baterii (elektrický článek)):

```
Procedure Battery #x, #y
  Line #x, #y-1.5, #x, #y+1.5, Solid, 1, DefaultColor
  Line #x + 0.3, #y - 3, #x + 0.3, #y + 3, Solid, 1, DefaultColor
  Text #x + 0.2, #y - 4.9, "+", 0,0,0,0, DefaultFontName, DefaultColor, DefaultFontSize, 0
  Line #x + 0.3, #y, #x + 1.5, #y, Solid, 1, DefaultColor
End
```

Použití procedury

Jednou napsanou proceduru je možné používat, jako by to byl obyčejný kreslicí příkaz, takže například příkazy ...

```
Battery 18, 7
Battery 19.5, 7
Battery 21, 7
Battery 22.5, 7
```

... nakreslí el. zdroj o čtyřech článcích. Přitom proměnné *#x* a *#y* mají funkci souřadnic, na kterých je baterie vykreslena. Aby nebylo nutné deklarace stejných procedur uvádět znovu a znovu v každém výkresu, je možné vytvořit textový soubor s deklaracemi procedur a vkládat jej do ostatních skriptů pomocí příkazu [@include](#).

Vkládání skriptů do jiných skriptů - příkaz @include

V sekci Procedury jako definice vlastních objektů je popsáno vytváření procedur. Aby nebylo nutné deklarace stejných procedur uvádět znovu a znovu v každém výkresu, je možné vytvořit textový soubor s deklaracemi procedur a vkládat jej do ostatních skriptů.

Obecný tvar: @include Cesta
Konkrétní příklad: @include "C:\SomeFolder\myfile.vic"

Cesta je cesta k souboru na pevném disku (musí být v uvozovkách). Pokud je tedy v nějakém skriptu uveden příkaz @include "myfile.vic", tak na toto místo jako by byl vložen celý obsah souboru "myfile.vic".

V parametru *Cesta* se může vyskytovat buď cesta absolutní (např. "C:\SomeFolder\myfile.vic") nebo relativní (např. "myfile.vic"). Pokud je uvedena cesta relativní, tak soubor je hledán buď v adresáři s programem *ScriptCad.exe* (pokud skript ještě není uložen), nebo v adresáři s uloženým (a právě editovaným) skriptem (.vic soubor).

POZN: Je možné zanožovat skripty pouze do první úrovně. Tedy ve skriptech, které jsou includovány příkazem @include již další include být nemohou.

Tiskový log a podpora tisku v přesném měřítku

Tiskový log LastPrint.log

Při tisku výkresu na tiskárnu vytváří program *ScriptCad* textový soubor s detailními informacemi o tisku. Tento soubor se jmenuje *LastPrint.log* a je vytvářen v adresáři s programem *ScrpCad.exe*. Při každém tisku je vytvářen znovu (tedy informace o starších tiscích se nearchivují). Příklad obsahu souboru *LastPrint.log* a vysvětlení položek:

ScriptName:	C:\MyFolder\demo.vic	... <i>název skriptu</i>
Date:	10/10/2006 16:06:34	... <i>datum tisku</i>
Printer:	Samsung ML-1510_700 Series	... <i>název tiskárny</i>
PixelsPerInch:	600 dpi x 600 dpi	... <i>rozlišení (bodů na palec)</i>
PageWidth:	4756	... <i>šířka tiskové strany v bodech</i>
PageHeight:	6810	... <i>výška tiskové strany v bodech</i>
PageRes:	4756 x 6810	... <i>šířka x výška tiskové strany v bodech</i>
PageRes (cm):	20.13 x 28.83	... <i>šířka x výška tiskové strany v centimetrech</i>
PhysPageSize:	4960 x 7014	... <i>velikost stránky v tiskárně včetně okrajů - v bodech</i>
PhysPageSize (cm):	21.00 x 29.69	... <i>velikost stránky v tiskárně včetně okrajů - v cm</i>
Orientation:	Portrait	... <i>orientace tisku (na výšku nebo na šířku)</i>

Při tisku program *ScriptCad* využívá celou tiskovou plochu (uvedenou v položce *PageRes*). Způsob roztahování výkresu na tiskovou plochu a další tisková nastavení (barevná hloubka) jsou popsány v sekci [Nastavení programu \(ScrpCad.ini\)](#).

Tisk v přesném měřítku

Řekneme, že nějaký výkres (např. stavební) chceme vytisknout v přesném měřítku 1:50. Protože program *ScriptCad* roztahuje výkresy na celou tiskovou plochu a protože poměr mezi vzdálenostmi na obrazovce a na tiskárně závisí na rozlišení tisku (300 dpi, 600 dpi, ...) na velikosti tiskové strany (A4, A3, ...) i na velikosti okrajů (srovnej *PageRes* a *PhysPageSize*), je nutné, aby si uživatel hned při vytváření výkresu ujasnil, v jakém měřítku bude chtít plánec vytvářet.

Další postup si vysvětlíme na konkrétním příkladu. Máme výkres domu v souboru *myhouse.vic*. Skutečný dům má mít (nijakou) délku 8 metrů a my chceme měřítko výkresu 1:50. Takže na výkrese z tiskárny chceme, aby tako délka byla $8:50 = 0.16$ metru = 16 cm. Naše tiskárna tiskne na A4 a program *ScriptCad* výkres maximálně roztahuje tak, aby kresba vyplňovala co největší rozsah tiskové plochy. Řekneme, že budeme tisknout na výšku, takže limitujícím faktorem je šířka výkresu (výkres zabere celou šířku A4, kdežto jen část - třeba polovinu - výšky A4). Takže plocha vymezená pro výkres se natáhne na 20.13 cm (šířka A4 bez okrajů - viz *PageRes (cm)*). Abychom měli na **vytištěném** výkrese zeď dlouhou 16 cm, stačí **například** zvolit *MaximumX = 20.13* a kreslíme zeď jako ...

Line 3, 5, 19, 5, Solid, 1, DefaultColor

... (protože $19 - 3 = 16$ - a to je délka zdi, co bude zdi). Ta 16-ka je relativní vůči 20.13 - prostě když se X-ová souřadnice daná jednotkou *MaximumX = 20.13* natáhne na 20.13 cm (protože tiskárna), bude každá jednotka ve výkresu (daná proměnnou *PixelsPerUnit*) odpovídat jednomu centimetru na tiskárně a proto i naše zeď bude mít na papíře délku 16 cm, tj. 1:50 vůči skutečnosti (8 metrů).

Hotový výkres a změna tiskárny

Často může nastat situace, že výkres je již hotový (nelze tedy měnit *MaximumX* a *MaximumY*). Potom namísto výpočtu *MaximumX* z *PageRes* (viz konkrétní příklad výše) počítáme obráceně - z *MaximumX* a

požadovaného množství vypočítáme *PageRes*. Je ale vysoce pravděpodobné, že vypočítané *PageRes* nebude odpovídat žádné standardní velikosti stránky (A4, A3, ...) a bude nutné při tisku volit (v ovladači tiskárny) uživatelem definovanou velikost tiskové strany. Např. pokud je *MaximumX* = 12 a chceme, aby šířka výkresu odpovídala dvanácti centimetrům, musíme při tisku zvolit v ovladači tiskárny šířku stránky 12 cm (**bez okrajů**).

