

Tabulkový procesor 602Tab



602Tab	3
Základní manipulace s prostředím a soubory	4
Nastavení prostředí	4
Načtení sešitu	6
Vkládání ODBC dat	6
Uložení sešitu.....	7
Zobrazení v oknech.....	7
Měřítko zobrazení	8
Odvolání a obnova	8
Práce se schránkou	8
Vyhledávání a náhrady v listu	9
Fulltextové vyhledávání.....	10
Zápis dat do buněk a jejich editace	10
Zápis do buňky	10
Oprava hodnoty v buňce	11
Režim vkládání a režim editace	11
Oblasti a jejich označení	11
Typy dat vkládaných do buněk.....	12
Výpočetní vztahy – vzorce	14
Operandy vzorců	14
Operátory a jejich priorita	15
Odkazy na buňky	16
Podmíněný výraz	17
Manipulace s listy	18
Založení a vymazání listu.....	18
Přejmenování listu.....	18
Změna pořadí listů	18
Skrytí a opětné zobrazení listu.....	18
Uzamčení listu.....	19
Odkazy mezi jednotlivými listy.....	19
Nastavení rozměrů buněk.....	20

Šířka sloupců	20
Výška řádků	20
Vkládání prázdných buněk.....	21
Úpravy v listu	22
Kopírování a přesuny obsahu buněk	22
Vyplnění oblasti.....	22
Seřídění buněk.....	24
Formátování buněk.....	25
Zalamování textu v buňkách	25
Zarovnávání	25
Změna formátu buňky	26
Nastavení písma	26
Barvy, rámování a výplň buněk.....	28
Speciální atributy.....	28
Mazání a odstranění buněk	29
Další operace nad buňkami v listu	30
Filtrace dat	30
Zamykání částí listu	30
Ukotvení příček	31
Pojmenované buňky.....	31
Pohledy	32
Komentáře.....	32
Vkládání objektů.....	32
Vkládání funkcí.....	33
Výpočty	34
Optimalizace	35
Náhledy a tisk	36
Rozměry listu a rozměry tiskové stránky.....	36
Okraje a záhlaví	36
Vynucené zlomy stránek	37
Náhled před tiskem	37
Rozsah tisku.....	37
Vlastní tisk.....	37
Přehled funkcí	39
Matematické funkce.....	39
Logické funkce.....	48
Statistické funkce.....	50
Databázové funkce	73
Finanční funkce	77
Funkce pracující s datem.....	84
Textové funkce	87
Informační funkce	92
Vyhledávací funkce.....	94

602Tab

Co je program 602Tab

602Tab je tabulkový kalkulátor neboli *spreadsheet*. 602Tab načítá a ukládá soubory nejen ve svém interním formátu, ale je plně kompatibilní s formátem Microsoft Excel (XLS). Nabízí všechny obvyklé standardní funkce (matematické, statistické, finanční a další), které potřebujete pro výpočty v tabulkách. Umožňuje také zobrazení tabelovaných dat ve formě 2D a 3D grafů s využitím integrovaného modulu obchodních grafů MagicGraf. Kromě grafů lze do tabulek vkládat i obrázky a další objekty. Celou tabulku lze vložit např. do textového editoru. K dispozici je celá řada předem připravených výpočtových šablon, jejichž pomocí můžete vyplňovat nejrůznější formuláře s automatickým výpočtem potřebných údajů. Buňky i jednotlivé listy tabulek lze linkovat do výsledného výpočtu. 602Tab podporuje práci s rozhraním ODBC, takže data lze načítat a následně zpracovávat z libovolné databáze.

Kompatibilita 602Tab

602Tab je kompatibilní se základními formáty používanými různými tabulkovými procesory. Tak například můžete načítat a zpracovávat data pořízená programem Microsoft Excel 95/97/2000 a naopak, sešity vytvářené či upravované v 602Tab lze ukládat ve formátu Excelu (XLS). 602Tab podporuje i soubory formátů Lotus 1-2-3 (WKS), Quattro Pro (WQ1/WQ2), DBase (DBF) a také textové formáty (TXT/CSV). Vnitřním formátem 602Tab je formát (WLS).

HTML

Tabulkový kalkulátor 602Tab podporuje práci s HTML – umožňuje sešity s tabulkami ukládat bez dalších úprav do formátu HTML. Přitom vy nemusíte mít žádné znalosti HTML jazyka. Veškeré změny učiněné v tabulce vytvořené v 602Tab se projeví v konečném dokumentu HTML.

ODBC

602Tab podporuje práci s rozhraním ODBC, což umožňuje využívat data pocházející z jakéhokoli datového zdroje ODBC jako zdrojová data. Ta se tak mohou snadno stát obsahem Vašich tabulek, a tedy předmětem Vašich výpočtů.

Grafy a ostatní objekty 602Tab

Grafy

Řada čísel sama osobě zaujme jen málokoho z nás na první pohled. Pro přesvědčivé zobrazení výsledků je vhodná jejich interpretace ve formě grafu. K tomu využijete integrovaný modul MagicGraf; ten nabízí celkem 16 typů 2D i 3D grafů, které lze navíc různým způsobem modifikovat.

OLE objekty

Díky kompatibilitě s Drag&Drop a OLE 2 umožňuje 602Tab vkládat na libovolné místo v sešitu obrázky, grafy a další objekty. Do tabulek lze například vkládat obrázky upravené „na míru“ pomocí aplikace 602Photo nebo „umělecky ztvárněné“ textové objekty MagicText. Tabulkový procesor rovněž umožňuje snadné integrování tabulek a grafů do souborů jiných formátů. Celou tabulku včetně grafu je například snadné vložit do dokumentu vytvořeném v textovém procesoru 602Text.

Funkce

602Tab disponuje rozsáhlou nabídkou obvyklých interních funkcí – počínajíc matematickými přes funkce popisné statistiky, funkce finanční i databázové až po funkce pro práci s datem a časem, řetězcové a informační.

602Tab dovede linkovat buňky jednotlivých tabulek i listy do výsledných výpočtů. Výhody funkcí tabulkového kalkulátoru využívají i výpočtové šablony, s jejichž pomocí můžete vyplňovat některé formuláře, ve kterých se automaticky vypočtou potřebné údaje.

Ostatní

Vybrané oblasti buněk v tabulce 602Tab je možné volitelně pojmenovat. Při následných výpočtech tak odpadne nutnost opětovně definovat danou oblast.

Celý pracovní sešit či jen jeho vybrané listy můžete chránit heslem proti nepovolenému přístupu.

Základní manipulace s prostředím a soubory

Nastavení prostředí

Dialog s konfiguračními kartami otevřete příkazem **Předvolby** z menu **Pomůcky**.

- **Zobrazovat** – pro povolení nebo potlačení zobrazení částí listu (záhlaví, mřížka, lišty ...).
- **Obecné** – pro povolení nebo potlačení provádění některých automaticky probíhajících rutinních akcí (indexace) a přednastavení některých základních parametrů (písmo).
- **Soubory** – pro nastavení základních parametrů pro práci na úrovni souborů.
- **Seznamy** – umožňuje definovat a udržovat pojmenované předem naplněné seznamy položek.

Samostatné nastavení mají ještě tlačítkové lišty.

Karta Zobrazovat

- **Záhlaví řádek a sloupců** – při zaškrtnutí se budou zobrazovat tlačítka v záhlaví řádků (šedá čísla) a sloupců (šedá písmena)
- **Mřížku** – při zaškrtnutí se bude zobrazovat tečkovaný rastr ohraničující jednotlivé buňky. V opačném případě rastr zobrazen nebude, což je vhodné pro „hotové“ formuláře, zejména pro tisk.
- **Buňky s nulovou hodnotou** – při zaškrtnutí bude zobrazen obsah buněk, jejichž numerická hodnota je „nula“. V opačném případě budou takové buňky prázdné.
- **Komentáře** – při zaškrtnutí bude při přiblížení ke komentované buňce zobrazen komentář ve speciální bublině.
- **Označení buněk s komentářem** – při zaškrtnutí budou buňky s komentářem označeny červenou tečkou v levém horním rohu.
- **Vzorce namísto hodnot** – není-li čtverec zaškrtnutý, zobrazují se v buňkách hodnoty vypočtené pomocí vzorců. V opačném případě se místo hodnot zobrazuje znění vzorců.
- **Svislý posuvník** – při zaškrtnutí se bude zobrazovat svislý posuvník pro vertikální „skrolování“ listu v okně.
- **Vodorovný posuvník** – při zaškrtnutí se bude zobrazovat vodorovný posuvník pro horizontální „skrolování“ listu v okně.
- **Uši listů** – při zaškrtnutí se budou v levé dolní části okna zobrazovat záložky s názvy listů spolu s „převíjecími“ tlačítky.
- **Objekty** – při zaškrtnutí se bude zobrazovat obsah objektů (obrázky apod.). V opačném případě se zobrazí pouze tabulky na listu 602Tab.

Karta Obecné

- **Posun po Enter** – voličem můžete předvolit, co se provede, ukončíte-li zadávání obsahu buňky klávesou **Enter** (žádný posun – posun o jednu buňku dolů – posun o jednu buňku vpravo).
- **Počet listů v novém sešitu** – pomocí tohoto pole můžete určit, kolik prázdných listů má obsahovat každý nově založený sešit.
- **Standardní písmo** – voličem můžete z nabídky systémových písem Windows vybrat to, které bude použito jako základní pro zobrazení v listech.
- **Velikost** – voličem můžete vybrat velikost písma, které bude použito pro zobrazení v listech jako základní.
- **Povolit přetahování buněk myši** – při zaškrtnutí lze přetahovat obsah buněk a oblastí myši při stisknutém levém tlačítku (Drag & Drop).

Karta Soubory

- **Soubory ukládat jako** – pomocí voliče vyberte, zda sešity mají být do souborů ukládány v interním formátu 602Tab (WLS) nebo ve formátu Microsoft Excel (XLS).
- **Automaticky indexovat soubory** – při zaškrtnutí budou sešity při uložení automaticky indexovány pomocí nadstavby Agent602 pro usnadnění pozdějšího fulltextového vyhledávání.
- **Užívat Mail602 pro posílání souboru** – při zaškrtnutí bude soubor s pracovním sešitem odeslán prostřednictvím poštovního systému Mail602. V opačném případě bude použit poštovní systém nastavený v prostředí Windows.
- **Seznam posledních souborů** – při zaškrtnutí bude v dolní části menu soubor zobrazeno tolik názvů naposledy otevřených souborů se sešity, kolik jich specifikujete v poli vpravo od položky. Kliknutím na položku s názvem v menu se daný soubor otevře a sešit načte bez nutnosti použití příkazu **Otevřít**.
- **Kódování češtiny v textových a DBF souborech** – pomocí voliče lze vybrat implicitní způsob kódování češtiny při práci s textovými a databázovými soubory.
- **Oddělovač položek v textových souborech** – do pole запиšte smluvený znak, který bude reprezentovat implicitní oddělovač položek ve větě textového souboru, aby mohly být chápány jako databázové položky.

Karta Seznamy

Na kartě lze vytvářet pojmenované seznamy obsahující výčet několika standardně hromadně používaných položek (názvy dnů v týdnu, měsíců v roce apod.).

Lišty s tlačítky

Součástí okna 602Tab jsou i lišty s ovládacími tlačítky. Jejich pomocí můžete pouhým kliknutím myši ovládat většinu funkcí programu.

K úpravě tlačítkových lišt můžete využít příkazy plovoucího menu, které otevřete kliknutím pravým tlačítkem myši do oblasti lišty.

Zastoupení lišt

V horní části menu jsou příkazy ovládající zobrazení jednotlivých lišt:

- **Úpravy** – obsahuje tlačítka pro práci se soubory a se schránkou, pro odvolávání akcí, hledání a náhrady, ovládání funkcí a třídění.
- **Formát** – obsahuje ovladače pro výběr druhu a velikosti písma, tlačítka pro úpravy stylu písma, pro zarovnání a nasazení nejčastěji používaných formátů.

Jedním kliknutím na položku příkazu se zobrazení příslušných tlačítek potlačí (zatržítka před položkou zmizí), dalším kliknutím se tlačítka obnoví (a zatržítka je opět na svém místě).

Posledním příkazem (je-li zobrazena alespoň jedna z lišt) je příkaz **Konfigurace**. Ten otevře komplexní dialog pro nastavení lišt. Dialog pro úpravu lišt také můžete otevřít příkazem **Lišty** z menu **Zobrazit**.

Nastavení lišt

Voličem **Skupina** určíte, kterou z lišt budete chtít nastavit.

Následuje dvojice sekcí **Použitelná tlačítka** a **Aktuální nastavení**. Mezi nimi jsou tlačítka **Přidat** a **Ubrat**. V levé sekci (Použitelná tlačítka) jsou zastoupena všechna tlačítka, která může lišta obsahovat. Ve druhé sekci jsou ta tlačítka, která jsou na liště v dané chvíli zastoupena. Chcete-li nějaké tlačítko přidat, označte jej kliknutím myši a stiskněte tlačítko **Přidat**. Pokud chcete nějaké tlačítko z lišty naopak vyjmout, klikněte na něj v seznamu pravé sekce a stiskněte tlačítko **Ubrat**.

Pořadí tlačítek na liště můžete nastavit podle vlastních požadavků. Kliknutím označte položku v seznamu **Aktuální nastavení** a pomocí tlačítek **Nahoru** a **Dolů** ji posuňte na požadované místo.

Velikost tlačítek je dána velikostí obrázků (ikon) na jejich čelní ploše. Pomocí voliče **Obrázky** můžete vybrat jednu ze tří variant: **Malé ikony** – **Střední ikony** – **Velké ikony**.

Každé tlačítko na liště obsahuje vedle obrázku také textový popis. Ten můžete podle nastavení voliče **Umístění textu** zobrazit buď dole (pod obrázkem), vpravo (vedle obrázku) či můžete text úplně vynechat.

Barevná tlačítka

K modifikaci zobrazení lišt můžete využít zaškrtačací čtverec **Všechny lišty kreslit barevně**. Při zaškrtnutí budou tlačítka zobrazena stále barevně. V opačném případě bude tlačítko barevné jen nastavíte-li nad něj kurzor myši.

Bublinová nápověda

Bublinová nápověda je důležitým informačním subsystémem, který vás pružně informuje o významu jednotlivých ovládacích prvků. Pokud je bublinová nápověda aktivní (máte zatržený čtverec **Zobrazovat bublinovou nápovědu**), zobrazí se při nastavení kurzoru myši nad tlačítko stručný nápovědný text uzavřený do rámečku, tzv. „bubliny“. Jestliže zaškrtnete i čtverec **Ukazovat v bublinách i klávesové zkratky**, bude informace v bublině doplněna klávesovým povelům vedoucím ke spuštění stejné akce jako stisk tlačítka (pokud takový povel existuje).

Poslední dosud nepopsané tlačítko je tlačítko **Reset**. Jeho stiskem se uspořádání tlačítek vrátí k implicitní podobě jakou mělo po instalaci programu.

Načtení sešitu

Načtení sešitu ze souboru

K načtení sešitu ze souboru je určen příkaz **Otevřít**. Ten otevře standardní dialog pro výběr souboru s možností načíst sešit, případně podle dalších voleb v seznamu **Typ zobrazených souborů** určit způsob importu. Dialog obsahuje obvyklé prostředky pro výběr souborové masky, volbu adresáře a specifikaci souboru.

Dialog pro otevření obsahuje v levé horní části čtveřici tlačítek pro volbu obsahu centrálního pole dialogu:

- **Adresáře** – běžné zobrazení tak, jak bývá zvykem v systémových dialogích Windows. Sešity jsou zobrazeny ve formě malých náhledů, které si můžete zvětšit kliknutím na symbol lupy v pravém dolním rohu.
- **Alba** – umožňuje načíst soubor katalogizovaný formou fotografického alba.
- **Web** – umožňuje načíst soubor ze stránky WEBu. Dialog se v tomto případě doplní o volič URL adresy (stránka se zobrazí v centrální části dialogu). Zaškrtnutím čtverce **Načíst originál** se bude načítat přímo z WEBu a nikoliv z vyrovnávací paměti, kde je obsah stránky na vašem počítači v dané chvíli uložen.
- **Skříň** – pro načtení kancelářským způsobem (skříň – šanon – složka).

Import dat z textového souboru

Příkaz **Importovat textový soubor** z menu **Data** umožní do 602Tab importovat obsah textového souboru.

- Ukazatel listu sešitu nastavte na buňku, od které se mají data načtená ze souboru vložit.
- Spusťte příkaz **Importovat textový soubor**.
- Pomocí standardního dialogu pro práci se soubory vyhledejte a požadovaný textový soubor.
- Soubor se načte a zároveň se otevře dialog pro volbu způsobu převodu řádků textu do jednotlivých sloupců.
- Vyberte vhodný oddělovač, v jehož místě budou řádky textu rozděleny do sloupců a stiskněte tlačítko **Převod**.

Vkládání ODBC dat

Funkci načítání dat přes ODBC rozhraní si předvedeme na importu databáze ve formátu dBASE do sešitu tabulkového procesoru.

Nejprve určíte lokalizaci souboru s databází:

- Stiskněte tlačítko **Start** a spusťte příkaz **Nastavení**. V submenu pak spusťte příkaz **Ovládací panely**.

- V okně **Ovládací panely** dvakrát klikněte na ikonu **32bit ODBC**.
- Na kartě **User DSN** dialogu **ODBC Data Source Administrator** nastavte ukazatel na položku **dBASE files** a stiskněte tlačítko **Configure**.
- V dialogu **ODBC dBASE Setup** zrušte zatržení přepínače **Use current directory** a stiskněte tlačítko **Select directory**.
- V dialogu **Select directory** určete přístupovou cestu k adresáři s vaší databází.

Specifikovanou databázi již lze bez problému načíst na list sešitu v tabulkovém procesoru:

- V menu **Data** spusťte příkaz **ODBC data**.
- V dialogu **Vybrat zdroj dat** otevřete kartu **Zdroj dat počítače** a dvakrát klikněte na položku **dBASE Files**.
- V dialogu **Výběr tabulky** se otevře seznam všech tabulek obsažených ve vybraném adresáři.
- Označením tabulky kliknutím myši se v sekci **Sloupce** otevře seznam názvů položek záznamu dané databáze.
- Stiskem tlačítka **Načíst data** se obsah databáze importuje do sešitu, počínajíc pozicí ukazatele na aktuálním listu.
- Pokud chcete vybrat jen některé položky, případně určit jejich pořadí či jinak určit způsob importu, klikněte na tlačítko [SQL dotaz](#).

SQL dotaz se zapisuje prostřednictvím tzv. SQL jazyka. Jeho popis přesahuje rámec této publikace; bližší informace naleznete např. na Internetu.

Tip – pole v dialogu je předvyplněno položkami záznamu databáze v pořadí, jak jsou v databázi uloženy. „Přerovnáním“ nebo vynecháním některých názvů polí určíte způsob importu do listu vašeho sešitu.

Uložení sešitu

Běžné uložení

Příkazem **Uložit** (klávesovým povelům **Ctrl+S**) uložíte sešit do souboru, ze kterého byl načten nebo do kterého byl naposledy uložen. Pokud sešit ještě uložen nebyl (nová nebo importovaná data), vyvolá se automaticky dialog **Uložit jako**.

Uložení se zadáním názvu (export)

Příkaz **Uložit jako** umožní uložit sešit do souboru specifikovaného jména (do vybraného adresáře a na zvolený disk). Příkaz použijete např. pro uložení sešitu do nového souboru (např. po modifikaci existujícího sešitu). Dialog se automaticky vyvolá při jakémkoliv požadavku na uložení sešitu, který dosud nebyla zapsán do souboru ve formátu 602Tab.

Sešity 602Tab se standardně ukládají do souborů s příponou **WLS**. Uložení nebo načtení v jiném formátu (a tím i s jinou implicitní příponou) se posuzuje jako export nebo import.

Mezi jednotlivými spuštěními programu se zachová informace o adresáři naposledy použitým pro uložení.

Uložení části listu do HTML formátu

Příkaz **Uložit jako HTML** uloží obsah buněk z označené oblasti do HTML souboru, jehož název zapíšete do pole **Jméno souboru**. Pokud soubor již existuje, můžete si při jeho vyhledání pomoci tlačítkem **Nalistovat**.

Přepínač **Orámovat buňky** zajistí zvýraznění obrysu buněk vytažením plnou čarou.

Zobrazení v oknech

Pokud pracujete s několika současně otevřenými okny, máte možnost měnit jejich uspořádání v rámci hlavního pracovního okna Windows. K tomu máte k dispozici příkazy sdružené v menu **Okno**.

- Příkaz **Mozaika** uspořádá okna s listy sešitů do dlaždicovité podoby (někdy se používá též termín mozaika). Všechna okna budou viditelná; čím více jich však bude, tím menší plochu budou mít k dispozici.
- Příkaz **Kaskáda** uspořádá okna do kaskádové podoby. Plně viditelné bude jen okno s prvním listem; ostatní budou podsunuta tak, aby bylo vidět jejich záhlaví s názvem a úzký svislý pruh v levé části.
- Příkaz **Minimalizovat vše** minimalizuje všechna okna s listy do tvaru ikon – spouštěcích tlačítek v systémovém okně Windows. Kliknutím na tlačítko se okno s listem zvětší do původní velikosti.
- Příkaz **Uzavřít vše** uzavře všechna otevřená okna, čímž se také ukončí činnost 602Tab. Pokud některý z listů byl upravován a nebyl dosud uložen, proces se pozastaví a vypíše se varovná zpráva.

V dolní části menu **Okno** je vypsán přehled právě otevřených oken 602Tab. Aktuální pracovní okno je zatrženo.

Měřítka zobrazení

Měřítka zobrazení se týká pouze způsobu zobrazení listu v pracovním okně. Jeho změna se neprojeví změnou fyzické velikosti vlastního listu.

Měřítka zobrazení nastavíte buď v dialogu příkazu **Zvětšení** (menu **Zobrazit**) nebo voličem na nástrojové liště. Měřítka se udávají v procentech; základ je 100%. Menší čísla znamenají zmenšení (na obrazovce uvidíte více buněk). Větší čísla naopak buňky zvětší.

Spuštěním příkazu **Normálně** bude list sešitů zobrazen běžným způsobem, který je základní pro práci s 602Tab. Tento způsob zobrazení se používá pro zápis dat, jejich formátování, filtraci a zpracování.

Příkazem **Konce stránek** bude list sešitů zobrazen v režimu náhledu na celé stránky. To je editovatelný pohled, při kterém se automaticky upraví měřítka zobrazení tak, že vidíte celé stránky, jak budou tištěny. Rozhraní stránek můžete v tomto pohledu měnit tahem myši ve všech čtyřech možných směrech.



Příkaz **Záhlaví a zápatí** otevře dialog **Formát stránky** na kartě **Okraje a záhlaví**. Zde máte možnost specifikovat horní a dolní záhlaví jednotlivých tiskových stránek, tedy objekty, které se budou opakovat na začátku a na konci každé stránky.

Odvolání a obnova

Provedené akce lze postupně odvolávat a tak se vrátet „do historie“ práce s 602Tab.

Pokud to s odvoláním přeženete je také možné se stejnou cestou vrátet směrem k „nejaktuálnějšímu stavu“ listu.

K dispozici máte tyto příkazy, povely a tlačítka:

Potřebujete ...	menu Úpravy	povel	tlačítko
odvolat poslední akci	Odvolat	Ctrl+Z	
vrátit se k minulému stavu	Znovu	Ctrl+Y	

Práce se schránkou

Využití standardní schránky Windows umožňuje čtveřice příkazů, kterou najdete ve druhé sekci menu **Úpravy**.

- **Vymout (Ctrl+X)** – přenese obsah i formát označených buněk do schránky; původní buňky se vyprázdní.
- **Kopírovat (Ctrl+C)** – zkopíruje (duplikuje) obsah i formát označených buněk do schránky; původní buňky se zachovají.
- **Vložit (Ctrl+V)** – zkopíruje obsah schránky na pozici ukazatele. Při tom se použijí jak hodnoty, tak i formáty původních buněk.
- **Vložit jinak** – zkopíruje obsah schránky na pozici ukazatele s volitelnými parametry.

Dialog **Vložit jinak** dává možnost volby, zda se mají kopírovat:

- hodnoty i vzorce / pouze vzorce / pouze hodnoty
- obsah buněk včetně formátu / obsah buněk bez formátování (přepínač **Formáty**)
- poznámkové texty / bez poznámkových textů (přepínač **Poznámky**).

V sekci **Operace** je dále možnost volby numerické interakce vlepovaných buněk s případnými korespondujícími buňkami v oblasti vlepění – buď žádná nebo sčítání, odčítání, násobení či dělení.

Přepínačem **Transponovat** se vlepovaná oblast překlopí tak, že z řádků se stanou sloupce a naopak.

Přepínač **Vynechat prázdné** ponechá buňky cílové oblasti vlepění odpovídající prázdným vlepovaným buňkám beze změn.

Vyhledávání a náhrady v listu

Vyhledání určitého údaje

V listu je možné specifikovaným způsobem vyhledat zadaný vzor.

K vyhledání použijte dialog příkazu **Hledat** z menu **Úpravy** nebo klávesový povel **Ctrl+F**.

- Hledaný vzor zapište do vstupního pole **Najít**.
- Prohledávání je možné nastavením voliče **Jak hledat** směřovat po řádcích nebo po sloupcích.
- Vzor lze vyhledávat jen v buňkách nebo jen ve vzorcích; to vyberete pomocí voliče **Kde**.
- Při vyhledání je možné podle nastavení přepínače **Rozlišovat malá a VELKÁ** respektovat rozdíly mezi malými a velkými písmeny nebo je ignorovat.
- Pokud má vzor tvořit náplň celé buňky a ne třeba jen část slova, zaškrtněte přepínač **Pouze celé buňky**.

Vyhledávání s nastavenými parametry spustíte stiskem tlačítka **Najít další**. Pokud je shoda nalezena, ukazatel se nastaví na odpovídající buňku. V opačném případě se nestane nic. Pokud potřebujete stejné vyhledání opakovat, stiskněte znovu zmíněné tlačítko. Jestliže jste již dialog opustili, můžete pro opakování vyhledání použít tlačítko z nástrojové lišty.

Náhrady

Vedle vyhledání zadaného vzoru je možné aplikovat operaci náhrady. Při ní se zadaný vzor specifikovaným způsobem vyhledá a nahradí jiným řetězcem.

K vyhledání řetězce znaků a jeho náhradě použijte příkaz **Nahradit** z menu **Úpravy** nebo klávesový povel **Ctrl+H**, případně tlačítko z nástrojové lišty.

Příkaz zobrazí dialog **Nahradit**. Ovládací prvky jsou významově obdobné dialogu **Najít**; chybí zde pouze volič **Kde**. Řetězec, kterým má být nalezený vzor nahrazen zapište do pole **Nahradit čím**.

Náhradu vyhledaného vzoru vyžádáte stiskem tlačítka **Nahradit**. Pokud použijete tlačítko **Nahradit vše**, proběhne náhrada v rámci celého listu.

K dialogu pro náhrady můžete také přejít z dialogu pro vyhledání; k tomuto účelu je v rámečku k dispozici tlačítko **Nahradit**.

Fulltextové vyhledávání

K fulltextovému prohledávání indexovaných dokumentů je určena nadstavba Agent602. Její ovládání umožňují příkazy soustředěné v submenu **Fulltext**.

Indexace sešitů

Fulltextového prohledávání se mohou účastnit pouze indexované dokumenty. Podklady pro vyhledávání se ukládají do zvláštní databáze. Zde se ukládají (v závislosti na přednastaveních příslušných programů) údaje o dokumentech, zásilkách elektronické pošty i WWW stránkách. 602Tab nabízí automatickou i manuální indexaci.

- Automatickou indexaci ukládaných sešitů vyžádáte zaškrtnutím přepínače **Automaticky indexovat sešity**. Ten najdete na kartě **Soubory** v dialogu příkazu **Předvolby** (najdete ho v menu **Pomůcky**).
- K manuální indexaci určitých souborů je určen dialog příkazu **Indexovat** (submenu **Fulltext**). Zde nejprve vyberete adresář, specifikujete soubory a pak je stiskem tlačítka **Hledat** zpracujete pro fulltextovou databázi.

Vyhledávání

Vyhledávání inicializujete příkazem **Vyhledat** (submenu **Fulltext**). Otevře se dialog, který se skládá ze tří karet. Pro fulltextové vyhledávání použijete kartu **Hledání**.

Do vstupního pole na kartě запиšte slovo nebo skupinu slov, která chcete v indexovaných dokumentech vyhledat. Pokud se mají určitá slova v dokumentu vyskytovat současně, oddělte je znakem '&'; pokud má k nalezení shody postačovat výskyt alespoň jednoho ze slov, oddělte je znakem 'čárka'.

Při specifikaci dotazu lze využít speciální jazyk (*Verity Query Language*), který umožňuje mezi hledanými pojmy vytvářet logické vazby pomocí různých operátorů. Vlastní kompletní popis dotazovacího jazyka se vymyká rámci tohoto textu; pokud jej přesto budete vyžadovat, doporučujeme prostřednictvím Internetu kontaktovat firemní stránku Software602 (URL: <http://www.software602.cz>) nebo využít některou z vyhledávacích služeb Internetu (najdete zde několik tisíc relevantních odkazů).

Po nastavení podmínky stiskněte tlačítko **Hledat**.

Výsledek vyhledávání je zobrazen ve střední sekci okna jako seznam souborů seřazený podle tzv. skóre, což je stupeň „přiléhavosti“ k nastavené podmínce.

Zápis dat do buněk a jejich editace

Buňky se plní daty zapisovanými klávesnicí. Vložená data je možné dodatečně upravovat. Obsah buněk lze také kopírovat a přesouvat do jiných buněk, případně mazat. Vlastní manipulace při vkládání a opravách dat se odehrávají mimo rámec mřížky, ve vstupním poli. Buňky se plní buď v režimu vkládání nebo v režimu editace.

Zápis do buňky

Režim vkládání umožňuje naplnit prázdnou buňku hodnotou libovolného typu nebo nahradit obsah stávající buňky jinou hodnotou.

- Nastavte ukazatel na buňku, do které budete chtít něco vložit.
- Požadovanou hodnotu запиšte do vstupního pole.
- Stiskem libovolné znakové klávesy se aktivuje režim vkládání, jehož pomocí запиšete buď novou hodnotu nebo přepíšete hodnotu stávající. Hodnota zapsaná do vstupního pole se do buňky zafixuje teprve až stisknete tlačítko se symbolem zatržení.
- Posunete ukazatel na jinou buňku.
- Stisknete klávesu **Enter**.

Při posunu ukazatele po listu se do vstupního pole promítá skutečný obsah buněk pod ukazatelem. Proč skutečný ?

- Buňka zobrazuje jen viditelnou část čísla či textu, do vstupního pole se však kopíruje celé číslo či text.
- Při zobrazení čísel lze nastavit formáty, které viditelný obsah buněk upraví podle vašeho přání, případně doplní symbolem měny. Ve vstupním poli vidíte hodnoty bez úprav. To platí i pro zobrazení chyb, logických hodnot a kalendářního data.
- Pokud je k vytvoření obsahu buňky použit vzorec, vidíte v buňce výslednou hodnotu. Do vstupního pole se promítá vlastní vzorec.

Další postavení ukazatele záleží na klávese, kterou jste vloženu hodnotu akceptovali. Pokud použijete některou z kláves pro posun kurzoru, ukazatel se posune na sousední buňku ve směru šipky této klávesy; stisknete-li klávesu **Enter**, ukazatel zůstane nastaven na buňce, do které jste zapisovali.

Oprava hodnoty v buňce

Pokud se při zápisu zmýlíte, můžete chybu opravit.

Část vložených znaků až k chybě ze vstupního pole vymažte pomocí klávesy **Backspace** a zbytek zapište znovu.

Pokud je špatně "úplně všechno" (třeba jste omylem aktivovali klávesu **CapsLock**), pak se můžete během zadávání vrátit k původnímu obsahu buňky:

- Stiskem tlačítka s červeným proškrtnutím.
- Stiskem klávesy **Esc**.

Na závěr zdůrazněme, že v režimu vkládání hodnot nemůžete editovat s použitím šipkových kláves pro posun kurzoru; jejich stisk, jak již víte, vede k vložení hodnoty do buňky.

Režim vkládání a režim editace

Vedle režimu vkládání můžete alternativně zvolit režim editace. Ten se od předešlého liší především funkcí kláves pro vodorovný posun kurzoru, které zde slouží opravdu k editaci obsahu vstupního pole buněk a nikoliv k akceptování vložené hodnoty.

Režim editace aktivujte takto:

- Nastavte ukazatel na požadovanou buňku.
- Stiskněte klávesu **F2** nebo myší klikněte do vstupního pole.
- Ve vstupním poli opravte, co je třeba.

Opravenou hodnotu do buňky v tomto případě vložíte stiskem kláves **Enter**, **Up**, případně **Down**. Můžete také použít "odfajfkované" tlačítko vlevo od vstupního pole.

Při editaci obsahu buněk se ve vstupním poli nerozlišuje režim přepisu a vkládání, jak jej znáte z práce s textovými procesory. Znaky se zapisují na pozici kurzoru a zbývající text vpravo od něj se odsouvá.

Oblasti a jejich označení

Oblast je skupina buněk tvořících obdélníkový výřez listu. Na oblast jako na celek je možné aplikovat většinu operací přípustných pro jednotlivé buňky. Oblast lze také s výhodou plnit hodnotami generovanými pomocí řad.

Na buňku pod ukazatelem lze pohlížet jako na oblast o velikosti jedné buňky. Velikost oblasti je omezena pouze kapacitou sešitu; může se tedy pohybovat od jedné buňky až po celý list.

Označení oblasti

Oblast je možné označit klávesnicí i myší.

- Ukazatel nastavte do jednoho z rohů budoucí oblasti.
- Stiskněte klávesu **Shift** a tahem myši při stisknutém levém tlačítku oblast označte. Můžete ji také označit kurzorovými klávesami se šipkami.

K označení jedné oblasti lze použít i následující klávesové povely:

Shift+End	označí řádku do poslední obsazené buňky
Ctrl+Shift+End	označení celé tabulky do poslední obsazené buňky
Ctrl+Shift+mezerník	označení komplet celé tabulky
Ctrl+mezerník	označí celý sloupec
Shift+mezerník	označí celý řádek

Nespojitá oblast

Pokud místo klávesy **Shift** stisknete klávesu **Ctrl**, můžete tahem myši označit **nespojitou oblast** (několik vzájemně nenavazujících oblastí buněk).

Vyplnění oblasti

Pokud označíte oblast a vkládáte hodnotu, můžete klávesovým povelom **Ctrl+Enter** vyplnit celou oblast vloženou hodnotou či vzorcem.

Význam oblasti

Na oblasti se odvolávají odkazy používané ve výrazech vzorců. Vzorce mohou obsahovat libovolné množství odkazů na oblasti bez ohledu na jejich velikosti a umístění v listu. V rámci listu však může v daném časovém okamžiku figurovat **jediná označená oblast (ta však může být i nespojitá)**.

Označená oblast je z funkčního hlediska významově blízká bloku textu. Zatímco blok v kontextu textového procesoru operuje s pracovní jednotkou na bázi znaku, oblast operuje s celými buňkami, bez ohledu na to, zda jsou prázdné či plné a pokud jsou plné, co v nich je uloženo.

Zdůrazněme si význam **tlačítek v záhlaví sloupců a řádků** pracovní plochy; ty označí jako oblast celý sloupec (nebo řádek). Celý list jako oblast označíte myší stiskem tlačítka bez označení v průsečíku záhlaví řádků a sloupců.

Platnost označení oblasti

Oblast zůstává označena do nejbližšího stisku libovolné kurzorové klávesy nebo kliknutí myši na pracovní plochu.

Typy dat vkládaných do buněk

602Tab pracuje s několika typy dat, které automaticky identifikuje podle způsobu zápisu.

Typ datum není v 602Tab typ v pravém slova smyslu, je to číselný údaj. Za samostatný typ je také možné považovat i prázdný typ; obsah prázdné buňky.

Typ číselný (numerický)

Numerická data se mohou podle potřeby vkládat v těchto podobách:

Celočíselné – záporná čísla se zapisují s unárním operátorem "minus" před číslem. Příпустné jsou jen číslice a znaménko. Znaménko "plus" lze sice do vstupního pole zapsat, při přenosu do buňky se však ignoruje.

164646
-4464

Reálné s desetinným oddělovačem – záporná čísla se zapisují s unárním operátorem "minus" před číslem. Příпустné jsou jen číslice, desetinný symbol a znaménko. Znaménko "plus" lze zapsat do vstupního pole, při přenosu do buňky se ignoruje.

646,466

-9999,11

Reálné v exponenciálním tvaru ±#,#####E±## – přípustné jsou jen číslice, desetinný oddělovač, znaménko před číslem, znaky 'E' nebo 'e' ke specifikaci exponentu a samotné znaménko exponentu. Znaménko "plus" před číslem sice zapsat do vstupního pole lze, při přenosu do buňky se ignoruje. Na místě znaku '#' se může vyskytovat libovolná číslice, celý zápis čísla $-1,234 \cdot 10^{-12}$ pak vypadá $-1,234E-12$.

6,63E+14
6,63e14
6,63E14

602Tab nerozlišuje samostatný finanční datový typ. Peněžní hodnoty vkládejte jako čísla, ale bez mezer mezi tisíci apod. Položka vložená jako **25000** bude posuzována (správně) jako číslo. Pokud ji však napíšete takto: **25.000** nebo s mezerou mezi tisíci: **25 000** – bude chápána jako textová. Zobrazení v korektním tvaru, včetně symbolu měny, zajistíte přiřazením vhodného formátu buňce.

Typ logický

Položky **logického typu** mohou nabývat pouze dvou hodnot:

- PRAVDA – logický výraz je pravdivý (do listu se zapisuje jako **TRUE**).
- NEPRAVDA – jako nepravda se posuzuje nepravdivý logický výraz (do listu se zapisuje jako **FALSE**).

Typ datum

Položky typu datum jsou určeny k záznamu hodnot kalendářních dat. Pro jejich zpracování je k dispozici řada speciálních funkcí. Pořadí položek pro zadávání (dny, měsíce a roky) a jejich oddělovače určuje nastavení národního prostředí Windows. Zobrazení kalendářních dat v pak lze modifikovat volbou jednoho z celé řady formátů.

Textový (znakový) typ

Textový typ slouží k popisným a komentářovým vsuvkám a k popisům sloupců. Jako textový je interpretován každý řetězec znaků, který nelze interpretovat jako jiný přípustný typ. Textový tvar obsahu buňky lze při formátování vynutit přiřazením typu **Text** v dialogu **Formát buněk**.

Rotační uňuchadlo
2.25E+12

Pozor – při nastavení desetinného oddělovače „čárka“ je řetězec 2,25 interpretován jako číslo, zatímco řetězec 2.25 je textová konstanta.

Nezaměňujte způsob zobrazení v buňkách, který závisí na nastaveném formátu se způsobem zápisu do vstupního pole!

Označení chyb

Datový typ chyba slouží k identifikaci chybně zadaných hodnot v buňkách. Skládá se ze dvou částí:

- dvou znaků „vykřičník“
 - identifikačního řetězce.
- | | |
|----------------|---|
| !!chCis | (nepřípustná hodnota – např. logaritmus či odmocnina záporného čísla) |
| !!ch/0 | (dělení nulou) |
| !!chHod | (nepřípustný typ argumentu – např. je zadán textový řetězec a očekává se číslo) |
| !!chOdk | (nepřípustný nebo chybný typ odkazu) |
| !!chJm | (nepřípustné nebo chybné jméno) |

Při vložení hodnoty do buňky se ihned testuje její správnost a přípustnost. Je-li zobrazena chybová zpráva, zarovnává se doprostřed buňky

Textovy_znakovy_typ

Výpočetní vztahy – vzorce

Pomocí vzorců se do buněk vkládají hodnoty vzniklé jako výsledek operací nad konstantami a hodnotami, které jsou obsahem jiných buněk.

Vzorce se zapisují stejným způsobem jako textové řetězce, tj. do vstupního pole po nastavení ukazatele na buňku, kam má být vzorec vložen. To, zda bude na řetězec znaků pohlíženo jako na vzorec nebo jako na textový objekt, určuje znak „rovnítko“. Je-li tento znak uveden jako první znak, bude řetězec interpretován jako vzorec.

Pravidla pro zápis vzorců:

- Nejprve zapište znak '='.
- Za tímto znakem následuje tělo vzorce složené z označení buněk, operátorů, funkcí a dalších prvků. Priorita jednotlivých částí vzorce se může určit ozávkováním.

Zápis vzorce se objevuje pouze ve vstupním poli, v odpovídající buňce se zobrazují výsledky vzniklé jeho vyčíslením.

Příklady:

```
=(A10 + A11)/C16
=SUM(A1:J10)
=TRIM(PROPER("Jana NOVÁ"))
```

Mezery mezi označením buněk a operátory jsou přípustné, při uložení vzorce do buňky se však eliminují. Mezery v zápisu označení oblasti přípustné nejsou.

Vzorce se vyhodnocují okamžitě po zadání (602Tab pracuje s automatickou rekalkulací).

Příklad

Do buněk **A1** a **B1** vložte určité hodnoty; řekněme 3 a 5. Do buňky **C1** napište posloupnost znaků =A1+B1. Tím jste vytvořili vzorec, který zajistí sečtení obsahu buněk **A1** a **B1** a zobrazení výsledku (číslo 8) v buňce **C1**, tedy v té buňce, kde je uveden.

Vzorec je tvořen dvěma **operandy** (**A1** a **B1**) a **operátorem** sčítání. Pokud změníte hodnotu libovolné z buněk **A1** a **B1**, bude výsledek v buňce **C1** okamžitě aktualizován.

Operandy a operátory

Výrazy mohou obsahovat konstanty, odkazy na buňky a oblasti a také funkce. To jsou operandy výrazu. Vztah mezi jednotlivými operandy specifikují operátory. Operátory určují, jaké operace se mají s operandy provádět. Prioritu vyhodnocení různých částí výrazu lze stanovit ozávkováním.

Zápis vzorce na první pohled spíše připomíná zápis formule v programovacím jazyku.

Operandy vzorců

Vzorec může být tvořen operandy několika typů:

Číselné (numerické) konstanty. Mají stejný tvar jako při zadávání do buněk.

```
=100+D8*25
=1,005E-12/12,66+3,1415926
```

Textové konstanty. Jsou tvořeny řetězcem znaků ohraničeným uvozovkami.

```
=“Franta”
=“mývalovec”&“kuní”
```

Logické konstanty. Jsou dvě: **TRUE** (pravda) a **FALSE** (nepravda).

```
=TRUE
=FALSE
```

Konstanty typu datum. Používají se ke vkládání kalendářních dat.

21.6.97
31.12.2001

Odkazy na buňky lze ve vzorcích kdykoliv použít na stejných místech jako konstanty. Při vyhodnocení vzorce bude místo odkazu do vzorce dosazena skutečná hodnota buňky. Odkazy na buňky mohou být relativní, absolutní nebo smíšené (podrobněji probereme dále).

=A1
=BB5+E15+1e12
=\$B1+\$A1+\$B\$3

Odkazy na oblasti se používají v argumentech některých funkcí určených k hromadnému zpracování skupin buněk.

=SUM(B2:G13)

Standardní tabulkové funkce jsou vnitřní funkce programu 602Tab. Zapisují se obecně ve tvaru:

=JménoFunkce(Argumenty)
=AVG(D4:F18)

Operátory a jejich priorita

Operátory určují vztahy mezi operandy vzorce, to znamená, že určují, co se má s těmito operandy provést.

Aritmetické operátory

Předpokládejme, že v buňce **A1** je hodnota 5, v buňce **B1** hodnota 2 a v buňce **C1** hodnota 10.

Aritmetické operátory mají pevně stanovenou prioritu. Ta určuje pořadí vyhodnocení operací mezi operandy. Nejprve budou vyhodnoceny operace s operátory s vyšší prioritou. Prioritu vyhodnocení lze změnit ozávkováním částí výrazu. Pokud jsou ve vzorci závorky, vyhodnotí se přednostně části výrazu uvnitř závorek. Podívejte se na příklady v tabulce:

	operátor	příklad	výsledek
Sčítání	+	=A1+B1	7
		=A1+B1+C1	17
Odčítání	-	=B1-A1	-3
		=C1-A1+B1	7
Násobení	*	=A1*B1	10
		=(A1+B1)*C1	70
		=A1*(B1+C1)	100
Dělení	/	=C1/B1	5
		=C1/B1*A1	25
		=C1/(A1*B1)	1
Umocňování	^	=C1+A1^B1	35
		=(C1+A1)^B1	225

=-A1-A2 výsledek je -7
=-(A1-A2) výsledek je -3

Pokud je ve vzorci uveden tzv. **unární operátor** + nebo - (přiřazuje operandu znaménko), má prioritu nejvyšší. I zde lze aplikovat ozávkování.

Logické operátory

Logické operátory jsou AND a OR a používají se pro logické operace AND (a současně) a OR (nebo).

Řetězcový operátor

Operátor **&** spojí obsah dvou řetězců, mezi něž je vložen, do řetězce jediného.

Tak například vyhodnocením výrazu "haf "&"haf "&"haf" vznikne trojitě zaštekání.

Relační operátory

Relační operátory porovnávají dvě hodnoty. Jsou-li ve vztahu určeném operátorem, vrací se logická hodnota „pravda“, v opačném případě „nepravda“.

Jak vypadají jednotlivé operace, jsou-li **X** a **Y** obecně hodnoty číselného, textového, datového či logického typu:

$X = Y$	X se rovná Y
$X <> Y$	X se nerovná Y
$X < Y$	X je menší než Y (znakově: Y předchází před X)
$X > Y$	X je větší než Y (znakově: X je před Y)
$X <= Y$	X je menší nebo rovno Y
$X >= Y$	X je větší nebo rovno Y

Předpokládejme, že v buňce **A1** je číslo 6, v buňce **B1** je řetězec "cd" a v buňce **C1** je řetězec "xyz".

<u>výraz</u>	<u>výsledek</u>
A1+A1>10	pravda
A1=111	nepravda
A1<B1	chyba typu TYPE!
1.1.1910<1.1.1995	pravda
TRUE<FALSE	nepravda

Odkazy na buňky

Ve vzorcích se vyskytují odkazy na buňky či oblasti. V této kapitole se podíváme na to, co se stane při kopírování nebo posunech vzorců.

Platí dvě základní pravidla:

- Při **přesunu** vzorce se odkazy na buňky nemění.
- Při **kopírování** vzorce se nemění absolutní odkazy, zatímco relativní odkazy ano.

Připomeňme si postup kopírování a přesouvání pomocí myši:

- Ukazatel nastavte na buňku se vzorcem.
- Kurzor myši posuňte do oblasti orámování buňky tak, aby se změnil z kříže na šipku.
- Tahem myši při stisknutém levém tlačítku vzorec přesuňte na cílové místo. Pokud v okamžiku uvolnění tlačítka myši podržíte klávesu **Ctrl**, vzorec se zkopíruje.

Vzorec lze také kopírovat do okolních buněk tahem za nástroj pro vyplnění (pravý dolní roh ukazatele).

V následujících kapitolách se podrobněji zaměříme na relativní a absolutní odkazy.

Relativní odkazy

Relativní odkazy se při kopírování přizpůsobují tak, aby ukazovaly na stále stejně vzdálené buňky.

Absolutní odkazy

Absolutní odkazy se při kopírování vzorců nemění a ukazují na stále stejné buňky.

Při zápisu absolutních odkazů se používá znak 'dolar'.

\$B12 – Pokud dolar zapíšete jen před označení sloupce, bude sloupcová souřadnice absolutní a řádková relativní.

B\$12 – Jestliže dolar zapíšete jen před označení řádku, bude absolutní řádková souřadnice, zatímco sloupcová zůstane relativní.

\$B\$12 – aby byly obě souřadnice posuzované jako absolutní, je třeba dolar zapsat před označení řádku i sloupce.

Při vytváření vzorců lze samozřejmě také používat smíšené odkazy, kdy řádková část je relativní a sloupcová absolutní či naopak.

Automatické vkládání odkazů

Při vytváření vzorců, které se odkazují na buňky nebo oblasti, je možné využít **automatického vkládání odkazů**. Při automatickém vkládání se využívá dynamická "projekce" pozice ukazatele nebo označení oblasti do vstupního pole při vkládání vzorce.

Pro režim automatického vkládání odkazů je charakteristické, že pokud je kurzor vstupního pole nastaven na místo, kam je možné vložit odkaz na buňku nebo oblast, pak se při pohybu ukazatelem jeho okamžitá pozice zobrazuje ve vstupním poli. Pokud se v tuto chvíli označí oblast, promítne se stejným způsobem i její označení.

Pokud se chcete po vzorci pohybovat kurzorovými klávesami, musíte přejít do editačního režimu stiskem klávesy **F2**.

Podmíněný výraz

Podmíněný výraz může na základě definované podmínky vložit do buňky jednu ze dvou alternativních hodnot. První je použita, je-li podmínka splněna. V opačném případě je do buňky dosazena druhá hodnota.

Obecný zápis podmíněného výrazu je:

IF(*podmínka*; *true_výraz*; *false_výraz*)

Vzorec si rozepíšeme po jednotlivých částech a vysvětlíme si, co která část znamená:

IF(začátek vzorce, klíčové slovo IF s levou kulatou závorkou; odpovídající pravá kulatá závorka podmíněný výraz ukončí
<i>podmínka</i> ;	„je-li splněna podmínka daná logickým výrazem <i>podmínka</i> “
<i>true_výraz</i> ;	„dosad' do buňky hodnotu výrazu <i>true_výraz</i> “
<i>false_výraz</i>)	„v opačném případě dosad' do buňky hodnotu výrazu <i>false_výraz</i> “.

Manipulace s listy

Každý sešit se může skládat z jednoho i více listů. Listy si můžete představit v klasickém slova smyslu jako tabulky na samostatných papírech, které jsou položeny na sebe a seřaty svorkou.

Seznam listů najdete na záložkách při levém dolním okraji okna. Pracovní list zvolíte kliknutím myši na záložku s požadovaným názvem. Pokud je listů více než je možné v určitém okamžiku zobrazit záložek (nebo jsou jejich názvy velmi dlouhé), můžete využít převíjecí tlačítka vlevo od seznamu záložek.

Aktuální pracovní list zvolíte kliknutím myši na jeho záložku.

Založení a vymazání listu

Implicitní počet listů v sešitu můžete nastavit v konfiguračním dialogu na kartě **Obecné**. Další listy můžete přidávat podle vašeho přání a nepotřebné listy můžete kdykoliv vymazat.

- Nový list vložíte příkazem **List** z menu **Vložit**. Do sešitu se list zařadí pod implicitním názvem **List n**, kde *n* je nejbližší volné pořadové číslo listu.
- List ze sešitu vymažete příkazem **Odstranit list** z menu **Úpravy**.

Nezapomeňte, že odstraněním listu zároveň nevratně zmizí data, která zde byla uložena!

Přejmenování listu

Jednotlivé listy sešitu jsou standardně pojmenovány generovaným názvem, který se skládá z označení **List** a pořadového čísla.

Listy můžete přejmenovat tak, aby jejich názvy lépe vystihovaly obsahovou náplň.

- Klikněte pravým tlačítkem myši na záložku listu, který chcete přejmenovat.
- V plovoucím menu spusťte příkaz **Přejmenovat**.
- Nový název zapište do vstupního pole dialogu **Jméno listu**.

Ten můžete také otevřít příkazem **Přejmenovat** ze submenu menu **List** a menu **Formát**.

Změna pořadí listů

Příkaz **Přesunout list** z menu **Úpravy** otevře dialog, jehož pomocí můžete právě otevřený list „přesunout“ před list vybraný pomocí voliče v dialogu.

Skrytí a opětné zobrazení listu

Jednotlivé listy je možné dočasně opticky skrýt potlačením zobrazení jejich záložek. Příslušné příkazy najdete v menu **Formát** a submenu **List**.

- Příkaz **Skrýt** skryje záložku aktuálně otevřeného listu. List je možné znovu zviditelnit příkazem **Zobrazit**.
- Příkaz **Zobrazit** umožňuje zviditelnit listy skryté příkazem **Skrýt**. Zobrazí se dialog se seznamem skrytých listů. Kliknutím myši označte listy, které budete chtít zviditelnit a volbu potvrďte.

Příkazy pro práci s listem najdete také v plovoucím menu, které otevřete pravým kliknutím na jeho záložku.

Uzamčení listu

Příkaz **Zamknout list** z menu **Pomůcky** umožní zablokovat přístup k buňkám vybraného listu. V dialogu **Zamknutí listu** zadáte (nepovinné) heslo, na které bude zobrazení listu vázáno.

Zamknutí se týká jen těch buněk, které mají v dialogu příkazu **Buňky** (menu **Formát**) zaškrtnutý přepínač **Zamknout buňky** (zaškrtnutí je implicitní).

Odkazy mezi jednotlivými listy

Běžné odkazy na buňky a oblasti skládající se z písmene (označení sloupce) a číslice (označení řádku) ukazují na buňky téhož listu. Data je ovšem možné sdílet i mezi jednotlivými listy.

Pro odkaz na buňku ležící na jiném než aktuálním listu se před označení sloupce předradí název listu a znak „vykřičník“.

Mějme třeba listy pojmenované **1995**, **1996** a **1997** a v buňce **B130** na každém z nich úhrnnou roční sumu čehosi. Součet za poslední tři roky (na listu **1997**) pak může vypadat takto:

$$=B130+'1995'!B130+'1996'!B130$$

Přípustný je i takovýto zápis:

$$=Sum('List1:List4'!A1)$$

Ten znamená, že se sečte obsah buněk A1 na listech List1, List2, List3 a List4 a celkovou sumu uloží do aktuální buňky (např. na listu List5).

Nastavení rozměrů buněk

Šířka sloupců

Změnou šířky jedné buňky se změní šířka celého sloupce.

Změna šířky sloupce myší

Šířku sloupce změníte nejsnáze myší:

- Kurzor myši nastavte na pravý okraj tlačítka v záhlaví sloupce, jehož buňky chcete rozšířit nebo zúžit. Kurzor se změní na vodorovnou obousměrnou šipku.
- Tahem myši při stisknutém levém tlačítku nastavte potřebnou šířku sloupce. Při tom se zobrazuje pomocná čára udávající okamžitou šířku sloupce, jaká bude po uvolnění tlačítka myši.

Tento způsob lze aplikovat pro současnou změnu šířky několika sloupců. Sloupce nejprve označte stiskem tlačítek v jejich záhlaví. Pak můžete kurzorem táhnout za mezeru mezi libovolnými dvěma tlačítky. Šířka všech označených sloupců bude po uvolnění myši stejná.

Přizpůsobení šířky sloupce položkám

Pokud nemáte zapnuté zalamování textu v buňkách a potřebujete přizpůsobit šířku sloupce zapsaným položkám, spusťte příkaz **Přizpůsobit** ze submenu **Sloupec** (menu **Formát**). Šířka sloupce se přizpůsobí nejdelší položce.

Stejného efektu dosáhnete dvojným kliknutím na pravý okraj tlačítka v záhlaví sloupce, který chcete šířkově přizpůsobit jeho položkám.

Přesné nastavení šířky

Přesné nastavení šířky sloupce nabízí dialog příkazu **Šířka** (submenu **Sloupec**, menu **Formát**). Do vstupního pole zapíšete požadovanou šířku sloupce. Číslo reprezentuje počet znaků v normálním stylu, které lze do sloupce zapsat vedle sebe tak, aby byly všechny viditelné. Nulová hodnota sloupec v listu opticky skryje.

Skrytí a obnova zobrazení sloupců

Zúžením šířky sloupce na nulovou délku lze dosáhnout stavu, kdy sloupec z okna opticky zmizí. Obsah všech buněk je však zachován a po opětovném rozšíření sloupce se znovu zobrazí.

Zúžení lze dosáhnout myší nebo výše uvedeným příkazem **Šířka**. Znovuzobrazení sloupce dosáhnete opačným postupem. V menu **Formát** a submenu **Sloupec** také pro skrytí a znovuzobrazení sloupců najdete dvojici příkazů **Skryt** a **Zobrazit**.

Výška řádků

Změnou výšky jedné z buněk se nastaví výška celého řádku.

Změna výšky řádku myší

Výšku řádku změníte nejsnáze myší.

- Kurzor myši nastavte na dolní okraj tlačítka v záhlaví řádku, jehož výšku chcete změnit. Tvar kurzoru se změní na obousměrnou svislou šipku.
- Tahem myši při stisknutém levém tlačítku nastavte potřebnou výšku řádku.

Tento způsob lze aplikovat pro současnou změnu výšky několika řádků, pokud řádky označíte stiskem tlačítek v jejich záhlaví. V tomto případě můžete kurzor nastavit mezi libovolná dvě tlačítka. Výška takto upravených řádků bude stejná.

Přizpůsobení výšky řádku položkám

Pokud potřebujete automaticky přizpůsobit výšku řádku zapsaným položkám, použijte příkaz **Přizpůsobit** (submenu **Řádek**, menu **Formát**).

Stejného efektu dosáhnete dvojím kliknutím na dolní okraj tlačítka v záhlaví sloupce, který chcete výškově přizpůsobit jeho položkám.

Přesné nastavení výšky

Přesné nastavení výšky řádku nabízí dialog příkazu **Výška** (submenu **Řádek**, menu **Formát**). Do vstupního pole zapíšete požadovanou výšku řádku v bodech.

Skrytí a obnova zobrazení řádků

Snížením výšky řádku na nulovou hodnotu lze dosáhnout stavu, kdy řádek z okna opticky zmizí. Obsah všech buněk je však zachován a po opětovném zvýšení se znovu zobrazí.

Skrytí lze dosáhnout myší nebo výše uvedeným příkazem **Výška**. Znovuzobrazení řádku dosáhnete opačným postupem. V menu **Formát** také pro skrytí a znovuzobrazení řádků najdete dvojici příkazů **Skrýt** a **Zobrazit**.

Vkládání prázdných buněk

Smyslem akce je získat uprostřed existujících dat volný prostor; ať již k vytvoření dalšího řádku či sloupce, s nímž se původně nepočítalo nebo jen k optickému oddělení částí listu.

Vkládání buněk

Komplexní dialog pro nastavení směru a rozsahu vložení najdete pod příkazem **Vložit buňky** (menu **Vložit**). Dialog se nezobrazí, je-li jako oblast označen celý řádek nebo sloupec.

- Nastavením přepínače na **Posunout buňky vpravo** se data vpravo od oblasti odsunou tak, aby na místě oblasti vznikl volný prostor.
- Při nastavení přepínače na **Posunout buňky dolů** dojde k odsunu dat o výšku oblasti směrem dolů. Data se odsouvají jen v aktuální šířce označené oblasti.

Vložení celého řádku

Příkaz **Řádek** (z menu **Vložit**) vloží nad pozici ukazatele tabulky prázdný řádek. Protože tato akce není vzhledem ke spojitému označení řádků a sloupců možná fyzicky, odsune se celý zbytek obsahu listu pod ukazatelem o jeden řádek dolů.

Jestliže chcete vložit více řádků, označte potřebný počet buněk přímo pod místem, kam chcete vložit nové řádky. Označte nad sebou tolik buněk, kolik řádků chcete vložit.

Vložení celého sloupce

Příkaz **Sloupec** (z menu **Vložit**) vloží vpravo od pozice ukazatele tabulky prázdný sloupec. Protože tato akce není vzhledem ke spojitému označení řádků a sloupců možná fyzicky, odsune se celý zbytek obsahu listu pod ukazatelem o jeden sloupec vpravo.

Jestliže chcete vložit více sloupců, označte potřebný počet buněk napravo od místa, kam chcete vložit nové sloupce. Označte vedle sebe tolik buněk, kolik sloupců chcete vložit.

Úpravy v listu

Při úpravách obsahu tabulek dat můžete vycházet z aktuální buňky či označené oblasti. Na tuto buňku nebo oblast se aplikuje požadovaná operace.

Kopírování a přesuny obsahu buněk

K snadnému přesouvání a kopírování částí listu nepotřebujete nic jiného než myš.




- Označte oblast buněk, které budete chtít přesunout nebo kopírovat.
- Kurzor myši nastavte k okraji orámování oblasti, aby se ze silného kříže změnil v bílou šipku.
- Tahem myši při stisknutém levém tlačítku přesuňte obrys oblasti na požadované místo.
- Uvolněním tlačítka se obsah buněk přesune na dané místo.

Budete-li chtít oblast zkopírovat, podržte v okamžiku uvolnění tlačítka klávesu **Ctrl**. Kurzor se v tu chvíli doplní rámečkem se znakem „plus“.

Využití schránky

K přenosu dat v rámci listu i mezi různými listy lze využít standardní schránku Windows. Údaje vložené do schránky lze vlepít na jiné místo téhož listu nebo na jiné její listy.

K práci se schránkou se používají obvyklé příkazy a povely:

Akce:	příkaz z menu Úpravy	povel	tlačítko
vystříhnutí	Vymout	Ctrl+X	
zkopírování	Kopírovat	Ctrl+C	
vlepení	Vložit	Ctrl+V	

Dejte pozor, pokud vkládáte oblast do listu, kde jsou již některé z buněk naplněny. Vlepovaná oblast v cílovém místě přepíše původní obsah buněk! Pokud nechcete o cílovou část listu přijít, musíte ji předem zabezpečit překopírováním nebo odsunutím.

Pozor na skryté řádky či sloupce – před vložením se raději ujistěte, že v cílové oblasti jsou všechny řádky a sloupce viditelné!

Pokud budete chtít přesouvat nebo kopírovat tam, kam to není možné, změní se kurzor myši v symbol známé dopravní značky “zákaz stání”.

Změna způsobu vlepení

Výše uvedené povely pro vlepení přenášejí informace obsažené v buňkách komplexně, tj. včetně formátování, vzorců, poznámek a dalších komponent.

Budete-li potřebovat vlepít např. jen výsledné hodnoty bez vytvářejících vzorců či zcela vynechat formátování, podívejte se na dialog příkazu **Vložit jinak** (příkaz najdete také v plovoucím menu).

Vyplnění oblasti

Vyplnění je akce, při které se obsah řádků (sloupců) oblasti naplní hodnotami vloženými do jejího krajního sloupce (řádku), případně zvolenou řadou hodnot.

Pokud chcete všechny buňky v oblasti vyplnit stejnou hodnotou, pak nejprve označte oblast, pak do vstupního pole запиšte požadovaný údaj a zápis ukončete klávesovým povelům **Ctrl+Enter**.

Vyplnění hodnotami

Oblast se vyplní hodnotami načteného z jednoho z krajních řádků nebo sloupců.

Postupujte takto:

- Označte oblast, kterou budete chtít vyplnit. Výplňový vzor musí být v krajním (levém či pravém) sloupci nebo řádku (horním nebo dolním).
- V menu **Úpravy** spusťte příkaz **Vyplnit**. Podle toho, kde je umístěn vzor, kterým chcete oblast vyplnit, zvolte jeden z příkazů následujícího submenu.

Obdobně by se oblast vyplnila aplikací příkazů **Dolů** (vzorem v horním řádku) a **Nahoru** (vzorem v dolním řádku).

Vyplnění pomocí myši

Velice pružně lze oblast vyplnit myší, tahem za pravý dolní roh ukazatele (označení oblasti).

Vyplnění řádku nebo sloupce kopírováním hodnoty

- Ukazatel nastavte na buňku, jejíž obsah budete chtít kopírovat.
- Kurzor myši přesuňte na malou čtvercovou značku u pravého dolního rohu označené buňky, kde se změní v malý černý symbol ve tvaru „plus“.
- Tahem myši při stisknutém levém tlačítku vyplňte řádek nebo sloupec v potřebné délce (výšce).

Pokud je v buňce, která má být použita jako vzor pro vyplnění vzorec, bude se kopírovat do dalších buněk podle pravidel platných pro **kopírování** vzorců.

Vyplnění řadou

Tento způsob umožňuje vyplnění oblasti hodnotami generovanými podle určitého konkrétního předpisu.

Může se jednat o vyplnění:

Aritmetickou řadou. Číslo v každé následující buňce vznikne přičtením určené konstanty k číslu v buňce jí předcházející.

10, 20, 30, ... , 100, 110, ...
0, 2, 4, 6, 8, ... , 10, 12, ...

Geometrickou řadou. Číslo v každé následující buňce vznikne vynásobením čísla v buňce jí předcházející určenou konstantou.

1, 2, 4, 8, 16, 32, 64, ...
1, 10, 100, 1000, 10000, ...

Datovou řadou vycházející z určitého kalendářního data. Obsah každé další buňky je datum zvětšené (zmenšené) o určený počet dní, měsíců nebo let.

1.1.98, 1.2.98, 1.3.98, ...
1.6.97, 2.6.97, 3.6.97, ...

Řady se zásadně vkládají do **oblastí**. Oblast může být řádková, sloupcová nebo může obecně pokrývat i několik řádků a sloupců. U dvojrozměrných oblastí se řady generují ve všech řádcích (sloupcích) – podle směru zvoleného v dialogu.

Druh řady a její parametry určíte v dialogu příkazu **Řady** (v submenu příkazu **Vyplnit** spuštěného z menu **Úpravy**).

- Orientaci řady v dvourozměrné oblasti vyberete přepínačem **Řady tvoří**. Podle jeho nastavení budou za počáteční hodnoty řad brány obsahy buněk v horním řádku nebo levém sloupci oblasti.
- Typ řady určíte pomocí přepínače v sekci **Typ**.
- Pokud jste vybrali kalendářní typ, můžete v sekci **Časová jednotka** vybrat časový přírůstek (po dnech, pracovních dnech, po měsících nebo letech).
- Počáteční hodnota řady se přebírá z výchozí buňky. U řádkových (sloupcových) oblastí je to buňka zcela vlevo (nahore). U dvourozměrných oblastí je to podle nastavení přepínače **Řady tvoří** buď horní řádek nebo levý sloupec oblasti.

Hodnota v poli **Krok** určuje přírůstek, pomocí kterého se vytvoří vztah pro následující hodnoty v řadě. U aritmetických řad je to přírůstek mezi dvěma členy, u geometrických řad je to číslo, kterým je třeba člen vynásobit, aby byl získán člen následující.

Sestupnou aritmetickou řadu získáte zadáním záporného kroku. Sestupnou geometrickou řadu vytvoříte použitím převrácené hodnoty kroku. To znamená, že má-li být následující člen dvakrát menší než předchozí, zadejte krok 0,5.

Údaj v poli **Konečná hodnota** určuje hranici, jejímž dosažením se série přestane generovat. Údaj je nepovinný. Pokud chybí, vyplní se podle daného předpisu celá oblast. V oblasti tak může část buněk zůstat prázdná.

Setřídění buněk

Obsah buněk v označené oblasti můžete setřídít. Jako třídící kritérium lze brát jeden, dva nebo tři sloupce.

Komplexním třídícím nástrojem pro oblasti buněk je dialog příkazu **Seřadit** (menu **Data**). V dialogu lze vybrat jeden až tři sloupce, podle kterých bude oblast postupně setříděna. U každého lze samostatně určit vzestupný či sestupný směr třídění.

V sekci **Směr** lze určit, zda má třídění probíhat shora dolů nebo zleva doprava.

Přepínačem **Rozlišovat malá a VELKÁ** je možné určit, zda mají být malá a velká písmena posuzována jako stejná či nikoliv. Pokud ano, pak jsou při vzestupném třídění velká písmena před malými (tj. ve sloupci nahoře).

Při zaškrtnutí čtverce **První řádka je záhlaví** bude první (horní) řádka oblasti považována za záhlaví a anebude do třídění zahrnuta.

Formátování buněk

Na úrovni buněk lze nastavit řadu parametrů, které se týkají jak zobrazení hodnot v buňkách uložených, tak tisku. Formátovací operace související se zobrazením lze aplikovat od úrovně jedné buňky výše. Zde je například možné nastavit typ písma, jeho velikost a styl a zvolit způsob zarovnání.

Způsob interpretace hodnot v listu je určen mnoha faktory: typem písma, způsobem zarovnání položek v buňkách, orámováním buněk, nastavením šířky sloupců a výšky řádek.

Zalamování textu v buňkách

Až dosud jsme uvažovali, že kapacita buňky se „dlouhé“ položce přizpůsobí rozšířením sloupce. Je zde však ještě další možnost.

Na kartě **Zarovnání** dialogu příkazu **Buňky** najdete přepínač **Zalomit text**.

Chcete-li začít nový řádek textu v určitém bodě, klikněte v řádku vzorců (nebo v buňce, upravujete-li text v ní) na místo, ve kterém chcete řádek přerušit. Potom spusťte klávesový povel **Alt+Enter**.

Zalomení vkládaného textu do sloupců

Rozdělení kopírovaných textových dat do sloupců umožňuje příkaz **Text do sloupců** z menu **Data**.

- Přejděte do aplikace a dokumentu, ze kterého chcete data kopírovat.
- Označte text, který chcete kopírovat a zkopírujte jej do systémové schránky Windows (Clipboard).
- Přejděte do sešitu 602Tab a kliknutím myši nastavte ukazatel na levou horní buňku oblasti vložení; pak vložte obsah schránky do listu (klávesovým povel **Shift+Ins**, tlačítkem **Vložit** nebo příkazem **Vložit** z menu **Úpravy**).
- Označte oblast buněk, která obsahuje právě vložená data. Oblast může obsahovat jakékoli množství řádků, ale ne více než jeden sloupec.
- Spusťte příkaz **Text do sloupců**.
- Pomocí dialogu **Převod textu do sloupců** vyberte oddělovač pro rozdělení do sloupců a stiskněte tlačítko **Převod**.

Zarovnávání

Vodorovné zarovnání

Zarovnání určuje, kde bude položka v buňce zobrazena; zda vlevo, vpravo či uprostřed. Změna zarovnání nezmění typ dat.

Standardně je obsah buněk v listu zarovnan:

- textové položky k levým okrajům buněk
- položky všech ostatních typů k pravým okrajům buněk.

Standardní zarovnání je možné změnit a obsah vybraných buněk zarovnat specifikovaným způsobem bez ohledu na typ obsažených hodnot.

Nejsnáze určíte způsob zarovnání v buňce nebo v buňkách oblasti pomocí tlačítek na nástrojové liště. Tato tlačítka umožňují levé, centrované a pravé zarovnání.

Komplexní dialog pro určení zarovnání najdete na kartě **Zarovnání** v dialogu příkazu **Buňky** (z menu **Formát** nebo plovoucího menu). Přepínač v sekci **Horizontální** nabízí sedm různých způsobů zarovnání.

Nastavení **Obecně** ponechá způsob zarovnání na typu hodnoty v buňce.

Možnosti **Vlevo**, **Doprostřed** a **Vpravo** jsou identické s výše zmíněnými tlačítky na liště.

Při nastavení na **Vyplnit opakováním** se buňka vyplní opakováním vložené hodnoty v celé její šířce. Opakování je jen optické; hodnota vložená do buňky se nezmění.

Svislé zarovnání

Svislé zarovnání udává, zda položka bude v buňce umístěna dole, nahoře či uprostřed.

Způsob svislého zarovnání určuje přepínač v sekci **Vertikální** na kartě **Zarovnání** dialogu příkazu **Buňky**.

Orientace textu

Hodnoty zobrazené v jednotlivých buňkách lze pootočit o 90° nebo 270°.

K nastavení orientace použijete přepínač v sekci **Orientace textu** na kartě **Zarovnání** dialogu příkazu **Buňky**.

Změna formátu buňky

Formát je předpis, který určuje, jakým způsobem bude zobrazen obsah buňky. Formát se přiděluje buňce a nikoliv jejímu obsahu.

Někdy je např. zapotřebí čísla zobrazovat v obecném tvaru, jindy ve tvaru vědeckotechnickém. Manipulace s financemi zase vyžadují, aby se hodnoty zobrazovaly zaokrouhleně a s přesností na dvě desetinná místa, symbolizující haléře. Možností zápisu kalendářního data je tolik, že se o nich raději zde ani nebudeme rozepisovat.

Formát zobrazení v buňce se nastavuje na kartě **Formát** v dialogu příkazu **Buňky** (z menu **Formát** nebo plovoucího menu). Volič **Typ** umožňuje zvolit formát položky v rámci buňky nebo všech buněk označené oblasti.

Identifikace typu vložené hodnoty je automatická. Pokud potřebujete, aby se obsah buňky bez ohledu na typ choval jako prostý textový řetězec, zvolte formát **Text**. Vložením časového údaje se automaticky nastaví časový formát.

K rychlému nastavení některých často používaných formátů můžete použít tlačítka na nástrojové liště.

Nastavení písma

602Tab umožňuje v používat všechny druhy písma, které jsou v dané chvíli instalované ve vašich Windows.

Všechny nabízené typy písma jsou viditelné na obrazovce ve tvaru, v jakém se budou tisknout. Změna typu písma je možná na úrovni obsahu celé buňky (oblasti buněk). Uvnitř jedné buňky písmo měnit nelze.

Dialog pro výběr parametrů písma najdete na kartě **Písmo** v dialogu příkazu **Buňky** (z menu **Formát** nebo plovoucího menu).

Pro rychlý výběr typu písma použijte volič na začátku formátovací lišty.

Velikost písma můžete nastavit dalším voličem vpravo.

K nastavení (nebo potlačení) tučného písma, kurzívy a podtržení je určena trojice tlačítek se symboly **B**, *I* a U.

Styl

Příkaz **Styl** z menu **Formát** umožňuje vytvářet a spravovat pojmenované styly buněk podobné stylům odstavců u textových procesorů.

Styl buňky zahrnuje mimo názvu:

- informaci o tom, zda buňka má obsahovat číslo či jiný datový formát
- typ zarovnání
- písmo (font) a jeho velikost
- způsob ohraničení buněk
- způsob vystínování plochy buněk
- informaci, zda má být buňka implicitně uzamčena.

Jak styl zavedete:

- V menu **Formát** spusťte příkaz **Styl**.
- Do pole seznamu **Název stylu** zapište název nového stylu. Pokud chcete změnit formát existujícího stylu, klikněte v seznamu na název stylu, který chcete změnit.
- Klikněte na tlačítko **Upravit**.
- Na kartách dialogu nastavte požadované parametry a klikněte na tlačítko **OK**.
- V zaškrťovacích čtvercích zrušte zaškrtnutí u těch parametrů, které styl nemá zahrnovat.
- Kliknutím na tlačítko **OK** styl uložíte a můžete ho začít používat

Automatický formát

Příkaz **Automatický formát** z menu **Formát** umožňuje zformátovat oblast buněk (nejméně 3x3) do podoby tabulky graficky zvýrazněné podle jednoho z předem připravených formátů. Vedle vlastního typu formátu, který vyberete podle vzorníku, je dále možné pomocí zaškrťovacích čtverců vybrat, zda se má zohlednit:

- formát čísel
- zarovnání
- písmo (typ, velikost)
- okraje
- výška a šířka
- barva pozadí.

Chcete-li použít pouze vybrané části automatického formátu, zrušte zaškrtnutí ve čtvercích označujících formáty, které nechcete použít.

Štěteček

Příkaz **Štěteček** z menu **Formát** aktivovuje štěteček, který shromažďuje a (nebo) modifikuje atributy zobrazení obsahu buňky. Tyto údaje lze pak jako celek aplikovat přejížděním štětečku nad textem v jiných buňkách při stisknutém levém tlačítku myši. Pokud je při vyvolání příkazu označena oblast, načtou se do štětečku všechny jeho jednoznačné atributy, v opačném případě je štěteček prázdný.

Chcete-li použít pouze vybrané části automatického formátu, zrušte v sekci **Použít formáty** zaškrtnutí ve čtvercích označujících formáty, které nechcete použít.

Barvy, rámování a výplň buněk

Barva písma

Barvu písma nastavíte výběrem z nabídkové palety na kartě **Písmo** v dialogu příkazu **Buňky** (z menu **Formát**) nebo z plovoucího menu.

Barevná paleta pro „vybarvení“ písma se také otevře tlačítkem z lišty.

Barva pozadí a výplňový vzor

Barvu pozadí (podkladu buňky) nastavíte voličem **Barva** v sekci **Pozadí** na kartě **Pozadí**. Tu najdete v dialogu příkazu **Buňky** (z menu **Formát**) nebo z plovoucího menu.

Výplňový vzor buňky a jeho barvu nastavíte voliči **Vzorek** a **Barva vzorku**.

Barvu pozadí můžete rychle vybrat také z palety otevřené tlačítkem z lišty.

Rámování buněk

Na úrovni jednotlivých buněk můžete volit způsob a barvu jejich orámování. Potřebné voliče nastavíte v sekcích **Ohraničení** a **Typ** na kartě **Pozadí**. Karta je součástí dialogu příkazu **Buňky** (z menu **Formát** nebo z plovoucího menu).

V sekci **Ohraničení** vyberete, ze kterých stran má být buňka orámována. Volič **Okolo** se týká vnějšího obrysu označené oblasti.

Ke každému voliči v sekci **Ohraničení** je v sekci **Typ ohraničení** možné samostatně přiřadit styl a barvu rámujičící čáry.

Speciální atributy

Zobrazení mřížky

Zobrazení mřížky na obrazovce lze potlačit. Tuto možnost využijete např. u "odladěných" sešitů, které předáváte k rutinnímu využití.

Odstranění mřížky (a její opětné zobrazení) dosáhnete pomocí zaškrtačacího čtverce **Mřížku** v dialogu **Předvolby** (menu **Pomůcky**, příkaz **Předvolby**) na kartě **Zobrazovat**.

Popsané zobrazení nebo potlačení zobrazení mřížky se týká jen obrazovky. Otázku tisku mřížky řeší zaškrtnutí čtverce **Tisknout mřížku**; ten najdete v dialogu **Formát stránky** na kartě **Stránka** (příkaz **Formát stránky** z menu **Soubor**).

Zobrazení záhlaví řádků a sloupců

Zobrazení šedých tlačítek v záhlaví řádků a sloupců na obrazovce lze potlačit. Také tuto možnost využijete zejména při zobrazení hotových sešitů.

Odstranění tlačítek v záhlaví (a jejich opětné zobrazení) dosáhnete pomocí zaškrtačacího čtverce **Záhlaví řádek a sloupců** v dialogu **Předvolby** (menu **Pomůcky**, příkaz **Předvolby**) na kartě **Zobrazovat**.

Také nastavení nebo potlačení zobrazení nadpisů se týká jen obrazovky. Otázku tisku tlačítek řeší zaškrtnutí čtverce **Tisknout záhlaví řádků a sloupců**, který najdete v dialogu **Formát stránky** na kartě **Stránka** (příkaz **Formát stránky** z menu **Soubor**).

Mazání a odstranění buněk

Je možné:

- Smazat pouze okamžitý obsah buňky.
- Vymazat obsah buňky včetně formátování.
- Vymazat komentář vztahující se k buňce.
- Odstranit buňky jako celek (zbytek listu se přisune).

Vymazání obsahu a formátování buněk

Obecný postup mazání na úrovni buněk nebo označené oblasti (bez odstranění buněk) je obecně následující:

1. Pomocí ukazatele vyberte buňku nebo označte oblast, kterou chcete mazat.
2. V menu **Úpravy** spusťte příkaz **Vymazat** a v jeho submenu pak jeden z následujících příkazů:
 - **Vše** – vymaže obsah, formátování i komentáře.
 - **Obsah** – vymaže obsah buněk a zachová formátování. Stejný účinek má také klávesa **Del**.
 - **Formáty** – vymaže pouze formátování, konstanty a vzorce zůstanou zachovány.
 - **Komentáře** – vymaže pouze komentáře jednotlivých buněk.

Odstranění buněk

Odstraněním se aktuální sloupec nebo řádek (v případě označení oblasti i několik řádků či sloupců) nejen smaže, ale i zruší. Obsah ostatních buněk se přisune na uvolněné místo buď zdola nebo zprava:

1. Ukazatel nastavte na buňku, kterou chcete zrušit, případně označte oblast.
2. V menu **Úpravy** spusťte příkaz **Odstranit buňky**. V dialogu příkazu vyberte způsob odstranění buněk (dialog se nezobrazí, je-li jako oblast označen celý řádek nebo sloupec).

Význam přepínačů v dialogu:

- **Posunout buňky vlevo** – přisune buňky na místo vymazané oblasti zprava.
- **Posunout buňky nahoru** – přisune buňky na místo vymazané oblasti zdola.

Další operace nad buňkami v listu

Filtrace dat

Automatický filtr

Příkaz **Automatický filtr** z menu **Data** zajistí selektivní zobrazení podmnožiny řádků v seznamu pomocí filtrů.

Filtry můžete používat v daném okamžiku pouze u jednoho seznamu v listu.

- Označte oblast buněk – tabulku, jejíž obsah budete chtít filtrovat (typicky sloupce položek s názvy sloupců v nejhořejším řádku).
- V menu **Data** spusťte příkaz **Automatický filtr**.
- Buňky v prvním řádku oblasti budou nyní obsahovat tlačítka pro aktivaci výběrových podmínek filtru pro daný sloupec.
- Stiskem tlačítka se otevře seznam umožňující výběr jedné z předvolených třídících podmínek nebo sestavení podmínky vlastní.

Pokud použijete filtr v jednom sloupci, můžete ostatní sloupce filtrovat pouze podle hodnot, které jsou v přefiltrovaném seznamu viditelné.

Příkaz **Automatický filtr** umožňuje použít v jednom sloupci maximálně dvě podmínky. Pokud chcete v jednom sloupci použít tři a více podmínek, použít vypočtené hodnoty jako vlastní kritéria nebo kopírovat záznamy na jiné místo, můžete použít rozšířené filtry.

Ukončení filtrace

Příkazem **Zobrazit vše** opět zobrazíte všechny řádky filtrovaného seznamu.

Pokročilý filtr

Příkaz **Pokročilý filtr** z menu **Data** vám poslouží pro filtrování seznamu pomocí rozšířených kritérií.

V listu je vhodné ponechat nad seznamem alespoň tři prázdné řádky, které mohou sloužit jako oblast kritérií. Seznam musí obsahovat popisky sloupců.

- Zkopírujte ze seznamu popisky sloupců, ve kterých se nacházejí hodnoty, které chcete filtrovat.
- Vložte popisky sloupců do prvního prázdného řádku v oblasti kritérií.
- Do řádků pod názvy kritérií napište popisky kritérií, která chcete použít pro filtrování seznamu. Přesvědčte se, že mezi hodnotami kritérií a seznamem zůstal alespoň jeden prázdný řádek.
- Klepněte na buňku v seznamu.
- V menu **Data** spusťte příkaz **Pokročilý filtr**.
- Filtrování seznamu zkopírováním řádků, které splňují kritéria, do jiné oblasti listu, provedete klepnutím na přepínač **Zopírovat do** a na levý horní roh oblasti vložení.
- Zadejte do vstupního pole **Oblast s podmínkou** odkaz na oblast kritérií včetně popisků kritérií.

Tip: Pokud list obsahuje oblast s názvem Kritéria, objeví se odkaz na danou oblast ve vstupním poli **Oblast s podmínkou** automaticky.

Zamykání částí listu

Uzamknutím se znemožní přepis hodnot uložených v buňkách.

V menu **Pomůcky** máte k dispozici dva příkazy:

- **Zamknout list** – uzamkne právě otevřený list.
- **Zamknout sešit** – uzamkne celý sešit (všechny listy).

V obou případech je při zamknutí možné zadat uživatelské heslo. Bez jeho znalosti není možné list ani sešit znovu odemknout.

Standardně jsou listy i sešit odemčené. Po zamknutí se příslušný příkaz změní na **Odemknout list (sešit)**.

Na kartě **Formát** dialogu příkazu **Buňky** (menu **Formát**) je přepínač **Zamknout buňky**. Přepínač je standardně zakřížkovaný; tím se vyjadřuje, že aktuální buňka (oblast) se po uzamčení listu (sešitu) zneprístupní pro editaci.

Uvedené příkazy a přepínač je možné zkombinovat tak, aby list fungoval de facto jako formulář s konkrétně specifikovanými editovatelnými oblastmi.

- Na kartě **Formát** odemkněte buňky, které chcete zpřístupnit.
- Zamkněte list (sešit). Od této chvíle bude list fungovat jako formulář. Po odemčených buňkách se budete moci pohybovat pomocí klávesových povelů **Tab** (na následující) a **Shift+Tab** (na předchozí).

Ukotvení příček

Pod tímto pojmem budeme rozumět vytvoření oblasti, která bude na obrazovce zobrazena vždy, bez ohledu na posun (rolování) dat. Může se jednat o jeden i více řádků nahoře a jeden i více sloupců vlevo.

Kotvení příček má význam pouze pro prohlížení dat na obrazovce.

- Ukazatel nastavte na buňku tak, aby řádky, které mají zůstat pevné byly nad ukazatelem a sloupce vlevo od ukazatele.
- V menu **Zobrazit** spusťte příkaz **Ukotvit příčky**.
- Ukotvení se vyznačí zesílenými čarami na rozhraní rolovatelné a nerolovatelné části listu.

Ukotvení příček zrušíte příkazem **Uvolnit buňky**, který v menu **Zobrazit** vystřídal původní příkaz vedoucí k ukotvení. Na pozici ukazatele v tuto chvíli nezáleží.

Pojmenované buňky

V rámci listu je možné pojmenovat buňku nebo oblast buněk popisným názvem.

Práce s pojmenovanými buňkami a oblastmi může výrazně zpřehlednit a zrychlit manipulace se sešitem. Na pojmenované buňky se lze odkazovat stejně jako na buňky určené řádkovou a sloupcovou souřadnicí.

Mějme list obsahující čtyři dílčí tabulky, které vyčíslují zisky firmy za jednotlivá čtvrtletí. Jednotlivé kvartální zisky jsou uloženy v buňkách **D20**, **D35**, **D50** a **D65**. Roční zisk vyčíslíte vzorcem $=D20+D35+D50+D65$. Buňky s dílčími výpočty však můžete označit popisnými jmény; například buňku **D20** jako **KVARTAL_1**, buňku **D35** jako **KVARTAL_2**, buňku **D50** jako **KVARTAL_3** a buňku **D65** jako **KVARTAL_4**. Formule pro vyčíslení ročního zisku pak může vypadat takto:

$$=KVARTAL_1+KVARTAL_2+KVARTAL_3+KVARTAL_4.$$

Jak buňku či oblast pojmenujete?

Spusťte příkaz **Definovat jméno** z menu **Vložit**; tím se otevře dialog **Definovat jméno**. Ten umožňuje zapsat definici pojmenování libovolné buňky či označené oblasti a také vytvářet pojmenované konstanty.

V dialogu také můžete libovolnou z definic pomocí tlačítka **Vymazat** kdykoliv zrušit.

Pro vkládání dříve vytvořených názvů je k dispozici dialog příkazu **Vložit**. Ten obsahuje seznam všech existujících názvů a umožňuje vložení vybraného názvu do aktuální buňky.

Názvy buněk je nutno vytvářet podle určitých pravidel

- Prvním znakem musí být písmeno nebo podtržítka. Dalšími znaky mohou být písmena, číslice, podtržítka a tečky.

- Název nesmí být označením jiné buňky, např. **D20** nebo **D\$65**.
- Mezery nejsou povoleny. Jako oddělovač lze použít tečku a podtržítka.
- Při rozlišení jmen není brán zřetel na malá a velká písmena. Máte-li definované jméno Odchylka a později vložíte další jméno ODCHYLKA, pak druhý zápis přepíše první.
- Celková délka jména je omezena na 255 znaků.

Pojmenování oblasti

Oblasti lze pojmenovat naprosto stejně jako jednotlivé buňky. Při pojmenování je lhostejné, zda se jedná o oblast sousedících buněk nebo oblast skládající se z více pravoúhlých výřezů listu.

Pohledy

Pro každý list lze definovat několik „pohledů“. V jejich formě se ukládá kolekce údajů o způsobu, jakým má být obsah listu zobrazen a jakým má být tištěn.

Součástí pohledu jsou:

- Údaje o skrytých řádcích a sloupcích.
- Nastavení procentuálního zvětšení.
- Ukotvení příček.
- Přednastavení způsobu zobrazení – zda zobrazovat či nezobrazovat mřížku, záhlaví řádků a sloupců a buňky s nulovou hodnotou.

Dialog pro vytváření a správu pohledů otevřete příkazem **Pohledy** z menu **Zobrazit**.

Dialog platí pro výše uvedená nastavení aktuálně otevřeného listu.

- Tlačítkem **Přidat** uložíte sadu údajů o pohledu pod názvem, který zapíšete do vstupního pole v horní části dialogu.
- Tlačítkem **Vymazat** vymažete pohled pod ukazatelem seznamu pohledů.

Komentáře

Komentář je stručný vysvětlující a popisný text, který se zobrazí v bublině při nastavení kurzoru myši nad buňku.

Dialog pro vytváření komentářů otevřete příkazem **Komentář** z menu **Vložit**.

- Označte buňku, ke které budete chtít vložit komentář.
- Do pole **Komentář** zapíšete text komentáře. Jednotlivé řádky oddělte klávesou **Enter**.
- Stiskněte tlačítko **Přidat**.

Existující komentář můžete odstranit. V sekci **Komentáře v listu** nastavte ukazatel na jeho označení a stiskněte tlačítko **Odstranit**.

Při třídění položek na listu se umístění komentářů mění spolu s umístěním položek v tříděných sloupcích nebo řádcích.

Vkládání objektů

Do listu je možné vložit graf, obrázek, objekt programu 602Text nebo obecný OLE objekt.

K vložení uvedených objektů slouží příkazy menu **Vložit – Graf, Obrázek, MagicText** a **OLE objekt** nebo čtveřice tlačítek na nástrojovém panelu:

- Příkaz **Graf** spustí nad označenou oblastí dat server MagicGraf, který z dat vytvoří odpovídající graf.
- Příkaz **Obrázek** umožní načíst z grafického souboru obrázek a vložit jej do listu.
- Příkaz **MagicText** spustí stejnojmenný server pro grafické úpravy textu.

- Příkaz **OLE objekt** umožní do listu vložit obecný OLE objekt.

Vkládání funkcí

Ve vzorcích mohou figurovat standardní tabulkové funkce. Najdete mezi nimi základní matematické funkce, funkce pro práci s řetězci i speciální funkce pro tabulkové aplikace.

Název funkce a její argumenty můžete do vzorce zapsat ručně, pomocí klávesnice. Můžete také využít nabídku funkcí, která je k dispozici v seznamu pod tlačítkem se symbolem funkce (**f()**) na liště.

Seznam můžete modifikovat pomocí skupiny přepínačů vpravo a vybrat si jen funkce jednoho nebo několika vybraných typů. V dolní části seznamu je nápověda týkající se činnosti a argumentů funkce pod ukazatelem seznamu.

Po volbě jedné z položek:

- Aktuální buňka se přepne do režimu vkládání.
- Do vzorce na pozici kurzoru vloží název funkce a první levou závorku. Vkládá-li se funkce jako první položka ve vzorci, uvede se před ní znak 'rovnítka'.
- Posunem ukazatele se na místo prvního argumentu promítá označení aktuální buňky (případně oblasti).
- Pokud jde o funkci s jedním argumentem, stiskem klávesy **Enter** se uzavře. Pravá závorka a vložení funkce je ukončeno s argumentem podle nastavení ukazatele nebo oblasti.
- Jde-li o funkci s více argumenty, vložte oddělovač (zde je to v souladu s nastavením Windows středník) a pokračujte ve výběru dalších argumentů.

Rychlá sumace

Požadavek na vložení sumačního vzorce patří mezi nejčastější. Proto na nástrojovém panelu najdete tlačítko se symbolem sumy, které vám celý problém usnadní.



Jak budete postupovat:

- Nastavte ukazatel na buňku, kam budete chtít vložit výsledek sumace.
- Stiskněte tlačítko se symbolem sumy.
- Označte oblast buněk, jejichž obsah se má sečíst.
- Stiskněte klávesu **Enter** (stiskněte "odfajfkované" tlačítko); vzorec se uzavře a v buňce se objeví výsledek.

Výpočty

Souhrny

602Tab pomocí tohoto příkazu dokáže na listu kalkulovat mezisoučty a celkové součty. Označte oblast, kde budete chtít kalkulovat a v dialogu příkazu vyberte funkci, které má být použita (SUM, MAX, MIN, ...).

- Seřadíte seznam podle sloupce, ve kterém chcete vypočítat souhrny. Chcete-li například sumarizovat výrobky prodané jednotlivými prodejci v seznamu prodejců, objemů prodeje a počtů prodaných jednotek, seřadíte seznam podle sloupce prodejců.
- Klepněte na buňku v seznamu.
- V menu **Data** spusťte příkaz **Souhrny**.
- Ve voliči **Při změně ve sloupci** vyberte ten sloupec, ve kterém chcete vypočítat souhrny. Měl by to být sloupec, podle kterého jste v kroku 1 seřadili seznam.
- Voličem **Použití funkce** vyberte funkci, pomocí které chcete souhrny vypočítat.
- V seznamu **Přidat souhrn sloupce** zaškrtněte políčka u sloupců s hodnotami, jejichž souhrny chcete vypočítat.

Tabulka

Vytvoření tabulky dat s jednou proměnnou

Tabulky dat s jednou proměnnou je nutné vytvořit tak, aby vstupní hodnoty byly zapsány buď do sloupce směrem dolů (sloupcová orientace), nebo do řádku (řádková orientace). Vzorce použité v tabulce dat s jednou proměnnou se musí vztahovat ke vstupní buňce.

- Napište seznam hodnot, které chcete nahradit ve vstupní buňce, buď do jednoho sloupce směrem dolů, nebo do jednoho řádku.
- Jestliže jsou vstupní hodnoty zapsány do sloupce směrem dolů, zadejte vzorec do první buňky vpravo od sloupce hodnot v řádku nad první hodnotou. Všechny další vzorce napište vpravo od prvního vzorce.
- Jestliže jsou vstupní hodnoty zapsány do řádku, zadejte vzorec do první buňky pod řádkem hodnot ve sloupci vlevo od první hodnoty. Všechny další vzorce napište pod první vzorec.
- Vyberte oblast buněk obsahující vzorec a hodnoty, které chcete nahradit.
- V nabídce **Data** klepněte na příkaz **Tabulka**.
- Pokud má tabulka dat sloupcovou orientaci, napište odkaz na vstupní buňku do textového pole **Vstupní buňka sloupce**.
- Pokud má tabulka dat řádkovou orientaci, napište odkaz na vstupní buňku do textového pole **Vstupní buňka řádku**.

Tip: K tabulce dat s jednou proměnnou můžete také přidat další vzorce.

Přidání vzorce k již existující tabulce dat s jednou proměnnou

Vzorce použité v tabulce dat s jednou proměnnou se musí vztahovat ke stejné vstupní buňce.

Jestliže jsou vstupní hodnoty zapsány ve sloupci směrem dolů, zadejte nový vzorec do prázdné buňky v horním řádku tabulky vpravo od existujícího vzorce.

Jestliže jsou vstupní hodnoty zapsány v řádku, zadejte nový vzorec do prázdné buňky v prvním sloupci tabulky pod existující vzorec.

Vyberte tabulku dat včetně sloupce nebo řádku, který obsahuje nový vzorec.

V nabídce **Data** klepněte na příkaz **Tabulka**.

Pokud jsou hodnoty zapsány ve sloupci, zadejte odkaz na vstupní buňku do textového pole **Vstupní buňka sloupce**.

Pokud jsou hodnoty zapsány v řádku, zadejte odkaz na vstupní buňku do textového pole **Vstupní buňka řádku**.

Vytvoření tabulky dat se dvěma proměnnými

V tabulkách dat se dvěma proměnnými se používá pouze jeden vzorec se dvěma seznamy vstupních hodnot. Vzorec se musí vztahovat ke dvěma různým vstupním buňkám.

- Vzorec, který se vztahuje ke dvěma vstupním buňkám, zadejte do buňky na listu.
- Jeden seznam vstupních hodnot napište do stejného sloupce pod vzorec. Druhý seznam napište do stejného řádku vpravo od vzorce.
- Vyberte oblast buněk, která obsahuje vzorec a řádek i sloupec hodnot.
- V nabídce **Data** klepněte na příkaz **Tabulka**.
- Do textového pole **Vstupní buňka řádku** zadejte odkaz na vstupní buňku vstupních hodnot v řádku.
- Do textového pole **Vstupní buňka sloupce** zadejte odkaz na vstupní buňku vstupních hodnot ve sloupci.

Iterační výpočet

Hodnotu specifikované buňky na základě výpočetního vztahu obsahujícího odkaz na tuto buňku (iteračním procesem) vyčíslí příkaz **Hledání řešení** z menu **Pomůcky**.

Nastavená buňka – запиšte odkaz na buňku nebo jméno pojmenované buňky, kde je vložen výpočetní vztah, jehož řešení chcete získat. Stiskem pomocného tlačítka v pravé části pole se vrátíte do výchozího dialogu.

Cílová hodnota – запиšte číslo vyjadřující novou nebo cílovou hodnotu, kterou chcete vypočítat.

Měnná buňka – запиšte odkaz na buňku, která se má změnit po ukončení výpočetního procesu. Buňka musí být přímo nebo přímo odkazována ve vztahu zapsaném v poli **Nastavená buňka**. Stiskem pomocného tlačítka v pravé části pole se vrátíte do výchozího dialogu.

Optimalizace

Příkaz otevře dialog s ovladači pro nastavení parametrů pro iterační výpočty.

Do vstupního pole **Nastavit buňku** zadejte odkaz na cílovou buňku nebo její název. Cílová buňka musí obsahovat vzorec.

Má-li cílová buňka obsahovat nejvyšší možnou hodnotu, nastavte přepínač na položku **Maximum**. Pokud má cílová buňka obsahovat nejnižší možnou hodnotu, nastavte přepínač do polohy **Minimum**. Jestliže má cílová buňka obsahovat určitou konkrétní hodnotu, přepínač nastavte na položku **Hodnotu** a do vstupního pole zadejte požadovanou hodnotu.

Do vstupního pole **Změnou buněk** zadejte odkaz na každou měnitelnou buňku nebo napište přímo její název. Nesousedící odkazy oddělte čárkou. Měnitelné buňky musí přímo nebo nepřímo souviset s cílovou buňkou. Pomocí pole můžete zadat až 200 měnitelných buněk.

Do polí v sekci **Volby** zadejte všechny omezující podmínky, které chcete použít – největší dobu v sekundách, maximální počet iterací a požadovanou přesnost (maximální rozdíl hodnot kritéria ve dvou následujících iteracích).

Posledním přepínačem můžete vybrat jednu ze tří optimalizačních metod.

Proces řešení můžete přerušit stisknutím klávesy **Esc**. Obsah celého listu se přepočte s naposledy zjištěnými hodnotami měnitelných buněk.

Náhledy a tisk

Prezentace dat uložených ve vašich sešitech může být poměrně různorodá. Můžete je nejen tisknout na libovolné tiskárně, ale k dispozici jsou vám i další, “modernější” způsoby – tabulky lze přímo z počítače faxovat nebo posílat elektronickou poštou.

Rozměry listu a rozměry tiskové stránky

Velikost tiskové stránky je dána nastavením zařízení, na kterém budete tisknout. Prostředky programu 602Tab můžete určit, jak velké okraje je třeba nechat volné ze všech čtyř stran a jaký prostor má být rezervován pro záhlaví.

Může být problematické, máte-li tabulku na listu velice širokou (vysokou) a její automatické rozčlenění na samostatné stránky při tisku v měřítku 1:1 by bylo nežádoucí.

V dialogu příkazu **Formát stránky** (menu **Soubor**) můžete na kartě **Stránka** v sekci **Velikost** upravit měřítko, ve kterém bude tabulka tištěna. Volbou procentuálního faktoru v poli **Upravit na** můžete list pro tisk přizpůsobit konkrétnímu požadavku. Nastavení můžete průběžně kontrolovat v náhledu – na kartě je k tomu k dispozici tlačítko **Náhled**.

Při nastavení přepínače do polohy **Přizpůsobit na** můžete stanovit, na kolik stránek se má list vejít. Vhodnou kombinací údajů v polích **Počet stránek vodorovně** a **Počet stránek svisle** dosáhnete rozčlenění odpovídajícího vaší představě.

Okraje a záhlaví

Fyzický formát papíru určíte v přednastaveních tiskového zařízení. Prostředky programu 602Tab můžete určit, jak velké okraje je třeba nechat volné ze všech čtyř stran a jaký prostor má být rezervován pro záhlaví.

V menu **Soubor** spusťte příkaz **Formát stránky**. Šest vstupních polí na kartě **Okraje a záhlaví** použijete k nastavení levého, pravého, horního a dolního okraje i prostoru pro obě záhlaví.

Pro posouzení nastavení okrajů je výhodné tlačítkem **Náhled** vyvolat kontrolní zobrazení.

Záhlaví a zápatí se volitelně tiskne na začátku a konci každé stránky. Jeho úkolem je nést informace pro snazší identifikaci tištěných dat.

Prostor pro záhlaví i zápatí se nastaví na kartě **Okraje a záhlaví**. Na kartě jsou voliče pro výběr jednoho z předdefinovaných formátů záhlaví a pro vytvoření záhlaví vlastních. K vytvoření jejich vlastního obsahu je k dispozici dialog **Záhlaví a zápatí**, který otevřete tlačítky **Definice záhlaví** nebo **Definice zápatí**.

Záhlaví i zápatí se mohou skládat z trojice údajů, z nichž jeden je vždy umístěn vlevo, druhý centrovane a třetí vpravo. Jednotlivé údaje se vkládají do trojice sekcí v dialogu a mohou se skládat z pevných textů doplněných proměnnými. Proměnné (číslo stránky, celkový počet stran, datum, čas, název sešitu a jméno listu) lze vkládat pomocí tlačítek. Ani jedna část záhlaví (zápatí) není povinná.

Dialog je doplněn tlačítkem **Písmo**, jehož aktivace vyvolá stejnojmenný dialog pro aktuální nastavení (nebo úpravu) typu písma.

Číslo první stránky tiskové sestavy je implicitně jedna, lze jej však změnit na kartě **Stránka** pomocí vstupního pole **Číslo první strany**. Jméno se přebírá ze záložky příslušného listu. Záhlaví i zápatí mohou být i víceřádková, v tomto případě je však třeba počítat s rezervou při vyhrazování příslušného prostoru.

Vynucené zlomy stránek

Do listu je možné vložit příznak pro vynucený zlom stránky. V jeho místě dojde k odstránkování vždy, bez ohledu na ostatní parametry tisku.

Vynucený zlom se vloží příkazem **Konec stránky** z menu **Vložit**.

Indikátorem zlomu je silnější přerušovaná čára. Pokud je ukazatel v prvním sloupci, vloží se nad ukazatel pouze vodorovný zlom. Je-li ukazatel mimo první sloupec, vloží se jednak vodorovný zlom a jednak svislý zlom vlevo od buňky.

Náhled před tiskem

Před vlastním tiskem je dobré využít možnost zobrazení náhledu, kde uvidíte tabulky na listech tak, jak se budou skutečně tisknout.

Náhled vyžádáte příkazem **Náhled** z menu **Soubor**.

V samostatném okně se zobrazí celostránkové pohledy na jednotlivé listy vašeho sešitu.

Okno se ovládá pomocí tlačítek na liště v horní části. Pro změnu měřítka je také možné použít myš. Nastavte kurzor v podobě lupy nad část, kterou chcete zvětšit a klikněte levým tlačítkem. Výřez se zvětší dvojnásobně. Dalším kliknutím se měřítko dále zvětší. Kliknete-li potřetí, vrátíte se k celostránkovému zobrazení.

Přehled tlačítek:

- **Tisk** – otevře dialog pro spuštění tisku
- **Další strana** – odstránkuje na následující stránku
- **Předchozí** – odstránkuje na předchozí stránku
- **Jedna strana/Dvě strany** – přepínač jednostránkového – dvoustránkového zobrazení
- **Zvětšit** – zdvojnásobí měřítko zobrazení
- **Zmenšit** – sníží měřítko zobrazení na polovinu
- **Zavřít** – přechod k běžnému zobrazení.
- **Formát stránky** – přejde do dialogu příkazu **Formát stránky**.

Rozsah tisku

Pokud budete chtít tisknout jen určitou oblast listu, určete ji v dialogu příkazu **Formát stránky** na kartě **Stránka** zaškrtnutím čtverce **Tisk oblasti** a zápisem rozsahu oblasti do připojeného vstupního pole. Necháte-li pole prázdné, bude se tisknout obsah celého listu.

Součástí tiskového výstupu může být mřížka ohraničující jednotlivé buňky i tlačítka v záhlaví řádků a sloupců. O tisku těchto částí listu rozhodují na kartě **Stránka** dva přepínače v sekci **Tisk**.

Pokud jsou data uložená v listu rozsáhlá a máte jistotu, že se nevejde na jednu tiskovou stránku, může se hodit možnost definice řádku i sloupce, který bude na každé stránce zopakován. Tím se zlepší čitelnost celé tiskové sestavy, protože nevznikne potřeba stále hledat někde na první stránce, co je vlastně v tom kterém sloupci vytištěno. Označení řádků a sloupců napište do polí v sekci **Titulky**.

U obzvláště velkých tabulek může být sporné, jak je tisknout – zda po stránkách vodorovně či svisle. Zde vám pomůže zaškrtačící čtverec **Pořadí stránek zleva doprava**.

Vlastní tisk

Tisková operace proběhne na zařízení nastaveném ve Windows s parametry zvolenými v programu 602Tab.

Tisk spustíte:

- Příkazem **Tisk** z menu **Soubor**.
- Stiskem tlačítka **Tisk** na jedné z karet dialogu příkazu **Vzhled tisku**.
- Tlačítkem **Tisk** z okna náhledu.
- Klávesovým povelům **Ctrl+P**.

Libovolným z uvedených způsobů se otevře standardní dialog Windows pro upřesnění parametrů tisku. Dialog je závislý na ovladači zvoleného zařízení. Bývá zde možné vybrat jiné tiskové zařízení a nastavit jeho parametry, určit počet kopií a způsobu jejich řazení a další.

Zajímavou možnost vám nabízí ovladač pro fax. Máte-li jej instalován a počítač je vybavený faxmodemovou kartou, můžete „tiskem“ na tomto zařízení své tabulky přímo faxovat.

Přehled funkcí

K vyhodnocení obsahu buněk lze kromě základních matematických operací používat také funkce.

Funkce lze formálně rozdělit do několika skupin

- Matematické funkce
- Logické funkce
- Statistické funkce
- Databázové funkce
- Finanční funkce
- Funkce pro práci s datem
- Textové funkce
- Informační funkce
- Vyhledávací funkce

Při zápisu argumentů budou použity tyto konvence

- jako desetinný oddělovač se používá znak 'čárka'
- jako oddělovač prvků seznamu se používá znak 'středník'.

Vzhledem k tomu, že oddělování desetinných míst a položek v seznamu (tedy i argumentů ve funkcích) je závislé na nastavení těchto oddělovačů ve *Windows*, může se stát, že například místo:

MAX(1,2; 3,4)

budete muset psát:

MAX(1.2, 3.4)

Matematické funkce

Pomocí matematických funkcí můžete provádět jednoduché i složité matematické výpočty, jako je například výpočet celkové hodnoty oblasti buněk (nebo celkové hodnoty oblasti buněk), které splňují podmínku v jiné oblasti buněk, nebo zaokrouhlovat čísla.

Abs(Číslo)

Vrátí absolutní hodnotu argumentu *Číslo*.

Příklad: **ABS(-26) = 26**

Acos(Číslo)

Vrátí arcus cosinus argumentu *Číslo* v radiánech. Arcus cosinus je úhel, jehož cosinus je dané číslo. Argument *Číslo* musí být v intervalu od -1 do 1.

602Tab zaokrouhlí výsledek na 14 platných cifer.

Příklad: **ACOS(0) = 1,5707963267949**

Acosh(Číslo)

Vrátí hyperbolický arcus cosinus zadaného *Čísla*. Toto číslo musí být větší nebo rovno 1. Hyperbolický arcus cosinus je hodnota, jejíž hyperbolický cosinus je zadané číslo, tj. **ACOSH(COSH(*Číslo*))** se rovná zadanému číslu.

Číslo je libovolné reálné číslo větší nebo rovno 1.

Příklady: $\text{ACOSH}(1) = 0$
 $\text{ACOSH}(10) = 2,9932228$

Asin(Číslo)

Vrátí arcus sinus argumentu *Číslo* v radiánech. Arcus sinus je úhel, jehož sinus je dané číslo. Argument *Číslo* musí být v intervalu od -1 do 1.

602Tab vrací výsledek s přesností na čtrnáct platných cifer.

Příklad: $\text{ASIN}(1) = 1,5707963267949$

Asinh(Číslo)

Vrátí hyperbolický arcus sinus zadaného čísla. Hyperbolický arcus sinus je hodnota, jejíž hyperbolický sinus je zadané číslo, takže $\text{ASINH}(\text{SINH}(\text{číslo}))$ se rovná zadanému číslu.

Číslo je libovolné reálné číslo

Příklady: $\text{ASINH}(-2,5) = -1,64723$
 $\text{ASINH}(10) = 2,998223$

Atan(Číslo)

Vrátí arcus tangens argumentu *Číslo* v radiánech. Arcus tangens je úhel, jehož tangens je dané číslo.

Příklad: $\text{ATAN}(1) = 0,785398163397448$

Atan2(Číslo1; Číslo2)

Vrací arkustangens úhlu svíraného vodorovnou osou a přímkou procházející počátkem (bodem se souřadnicemi (0, 0)) a bodem se souřadnicemi (x, y). Návrátová hodnota je v radiánech.

Kladný výsledek určuje úhel proti směru hodinových ručiček od osy x, záporný výsledek úhel ve směru hodinových ručiček.

Funkce $\text{ATAN2}(x,y)$ odpovídá $\text{ATAN}(y/x)$, vyjma případu, kdy x je rovno nule.

Pokud potřebujete výsledek ve stupních, vydělte jej hodnotou $180/\text{PI}()$.

Příklady: $\text{ATAN2}(1; 1)$ dává výsledek **0.785398** (to je $\text{PI}/4$ radiánu)
 $\text{ATAN2}(-1; -1)$ dává výsledek **-2.35619** (to je $-3*\text{PI}/4$ radiánu)
 $\text{ATAN2}(-1; -1)*180/\text{PI}()$ dává výsledek -135 stupňů

Atanh(Číslo)

Vrátí hyperbolický arcus tangens zadaného čísla. Toto číslo musí být mezi -1 a 1 (s výjimkou mezních hodnot -1 a 1). Hyperbolický arcus tangens je hodnota, jejíž hyperbolický tangens je zadané číslo, takže $\text{ATANH}(\text{TANH}(\text{číslo}))$ se rovná zadanému číslu.

Číslo je libovolné reálné číslo v rozmezí mezi 1 a -1

Příklady: $\text{ATANH}(0,76159416) = 1$ (přibližně)
 $\text{ATANH}(-0,1) = -0,10034$

Ceiling(Číslo;Přesnost)

Zaokrouhlí číslo se zadanou přesností. Vrací **číslo** zaokrouhlené na nejbližší celistvý násobek argumentu **přesnost**.

- Oba argumenty musí být numerické, jinak funkce vrací chybu.
- Funkce vrací chybovou hodnotu také pokud nemají hodnoty obou argumentů shodná znaménka.

Příklady: $\text{Ceiling}(2,5; 1)$ vrací hodnotu 3
 $\text{Ceiling}(-2,5; -2)$ vrací hodnotu -4
 $\text{Ceiling}(-2,5; 2)$ vrací chybový kód **!!chHod**
 $\text{Ceiling}(1,5; 0,1)$ vrací hodnotu 1,5
 $\text{Ceiling}(0,234; 0,01)$ vrací hodnotu 0,24

Cos(Číslo)

Vrátí cosinus úhlu vyjádřeného argumentem *Číslo* (zadejte v radiánech).

602Tab vrací hodnotu zaokrouhlenou na čtrnáct platných cifer.

Příklad: $\text{COS}(0,5) = 0,877582561890373$

Cosh(Číslo)

Vrátí hyperbolický cosinus úhlu vyjádřeného argumentem *Číslo* (zadejte v radiánech).

Příklad: $\text{COSH}(0,5) = 1,127625965$

Countif(Oblast;Podmínka)

Vrátí počet buněk, které splňují zadanou podmínku.

Oblast je oblast buněk, ve které chcete spočítat buňky

Podmínka definuje, které buňky se budou počítat; udává se jako čísla, výrazy nebo text.
Například 32, "32", ">32", "jablka".

Příklady:

Například buňky A3:A6 obsahují postupně "jablka", "pomeranče", "hrušky", "jablka".

$\text{COUNTIF}(A3:A6;"jablka") = 2$

Pokud buňky B3:B6 obsahují postupně hodnoty 32; 54; 75; 86, potom:

$\text{COUNTIF}(B3:B6;">55") = 2$

Degrees(Číslo)

Konvertuje úhel *Číslo* zadaný v radiánech na stupně.

Příklad: $\text{DEGREES}(\text{PI}()) = 180$

Even (Číslo)

Vrátí argument *Číslo* zaokrouhlený na nejbližší sudé celé číslo. Tuto funkci můžete použít pro zpracování položek, které se vyskytují v párech. Do bedny lze například skládat výrobky do řad po jednom nebo po dvou. Bedna je plná, pokud se počet výrobků, zaokrouhlený nahoru na nejbližší sudé číslo, rovná kapacitě bedny.

Číslo je hodnota, kterou je třeba zaokrouhlit

Poznámky:

Pokud argument *c* není číselného typu, vrátí funkce **Even** chybovou hodnotu **!!chHod**.

Bez ohledu na znaménko se hodnota zaokrouhlí směrem od nuly. Pokud argument *c* je sudé celé číslo, k zaokrouhlení nedojde.

Příklady:

$\text{Even}(1,5) = 2$

$\text{Even}(3) = 4$

$\text{Even}(2) = 2$

$\text{Even}(-1) = -2$

Exp(Číslo)

Vrací základ přirozených logaritmů umocněný na zadaný argument *Číslo*. Konstanta *e* se vkládá s hodnotou 2,71828182845905.

Příklad: $\text{EXP}(1) = 2,71828182845905$

Fact(Číslo)

Fukce vrací faktoriál celočíselného argumentu *Číslo*. Faktoriál čísla **n** se označuje **n!** a je to číslo, které se vypočte jako $1*2*3*...*n$. Pokud místo celočíselného argumentu *n* zadáte argument neceločíselný, zaokrouhlí se na celé číslo.

Příklady: **FACT(1,9)** se transformuje na **FACT(1)** a vrací **1**
 FACT(0) vrací **1**
 FACT(-1) vrací chybovou hodnotu **!!chCis**
 FACT(6) znamená $1*2*3*4*5*6$ a funkce vrací **720**.

Floor(Číslo1;Číslo2)

Zaokrouhlí číslo *Číslo1* dolů, směrem k nule, na nejbližší násobek argumentu *Číslo2*.

Poznámky:

- Pokud některý z argumentů není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Pokud číslo a násobek mají opačná znaménka, vrátí funkce chybovou hodnotu **!!chCis**.
- Bez ohledu na znaménko argumentu násobek se hodnota zaokrouhluje dolů, směrem k nule. Je-li násobek přesným násobkem argumentu zadané hodnoty, k zaokrouhlení nedojde.

Příklady:

Floor(2,5; 1) = 2
Floor(-2,5; -2) = -2
Floor(-2,5; 2) vrací chybový kód **!!chCis**
Floor(1,5; 0,1) = 1,5
Floor(0,234; 0,01) = 0,23

Int(Číslo)

Zaokrouhlí argument *Číslo* na nejbližší menší celé číslo.

Příklad: **INT(2,769) = 2**
INT(-PI) = -4

Ln(Číslo)

Vrátí přirozený logaritmus argumentu *Číslo*, tj. logaritmus o základu **e**. **Ln** je inverzní funkce k funkci **Exp**, takže $\ln(\text{Exp}(\text{číslo})) = \text{číslo}$.

Příklad: **LN(3) = 1,09861228867**

Log(Číslo;Základ)

Vrací jako výsledek logaritmus kladného reálného čísla *Číslo* při zadaném základu *Základ*. Pokud je argument *Základ* vynechán, počítá se dekadický logaritmus.

Příklady:

LOGZ(10) = 1
LOGZ(8;2) = 3
LOGZ(86;2,7182818) = 4,454347

Log10(Číslo)

Vrátí dekadický logaritmus argumentu *Číslo*.

Mod(Číslo;Dělitel)

Počítá zbytek po dělení čísla *Číslo* dělitelem *Dělitel*. Výsledek bude mít takové znaménko, jaké měl *Dělitel*.

Pokud je *Dělitel* roven 0, vrátí funkce MOD chybovou hodnotu **!!ch/0**.

Příklady:

$\text{MOD}(3; 2) = 1$
 $\text{MOD}(-3; 2) = 1$
 $\text{MOD}(3; -2) = -1$
 $\text{MOD}(-3; -2) = -1$

Odd(Číslo)

Funkce zaokrouhlí číslo zadané v argumentu *Číslo* na nejbližší celé liché číslo.

Poznámky:

- Pokud argument *Číslo* není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Zaokrouhlit nahoru znamená, že se zaokrouhluje směrem od nuly. Je-li hodnota argumentu liché číslo, žádné zaokrouhlování se neprovede.

Příklady:

$\text{Odd}(1,5) = 3$
 $\text{Odd}(3) = 3$
 $\text{Odd}(2) = 3$
 $\text{Odd}(-1) = -1$
 $\text{Odd}(-2) = -3$

Pi()

Vrátí hodnotu Ludolfova čísla π zaokrouhlenou na 14 platných cifer.

Příklad: $=\text{PI}()$ vrací hodnotu **3,1415926535898**

Power(Číslo;Mocnina)

Funkce vrací reálné číslo *Číslo* umocněné na exponent daný argumentem *Mocnina*.

Příklady:

$\text{POWER}(5;2) = 25$
 $\text{POWER}(98,6;3,2) = 2401077$
 $\text{POWER}(4;5/4) = 5,656854$

Product(Seznam)

Funkce vynásobí číselné hodnoty uvedené v seznamu argumentů *Seznam*. Pokud je v argumentu uveden odkaz na buňku či oblast, použijí se jen buňky s numerickými hodnotami.

Příklad: $=\text{PRODUCT}(A1:B3;5;F6)$

Pokud je v buňce A1 hodnota 2, v buňce B2 hodnota 7 a v buňce F6 hodnota 10, bude výsledek 700.

Radians(Číslo)

Konvertuje úhel *Číslo* zadaný ve stupních na radiány.

Příklad: $\text{RADIANS}(270) = 4,71238898$

Rand(Číslo)

Funkce generuje náhodné číslo, jehož hodnota je větší nebo rovna nule a menší než jedna. Při opakování téže kalkulace s využitím této funkce se generuje vždy jiné číslo.

Potřebujete-li, aby generované náhodné číslo mělo velikost v rozmezí hodnot **a** a **b**, použijte vztah $\text{RAND}()*(b-a)+a$

Pokud potřebujete jednorázově vygenerovat náhodné číslo, které již nebudete chtít dále měnit, napište do vstupního pole tabulkového kalkulátoru $=\text{RAND}()$ a pak stiskněte klávesu **F9**.

Příklad: $\text{RAND}()*10$

Vygeneruje náhodné číslo v rozsahu od 0 do 10.

Round(Číslo;Míst)

Zaokrouhluje číslo na zadaný počet číslic. Vráti číslo *Číslo* zaokrouhlené na tolik desetinných míst, kolik určuje argument *Míst*.

Příklad: **ROUND(2,34175;2) = 2,34**
ROUND(2896,3;-2) = 2900

Rounddown(Číslo;PočetDesMíst)

Funkce zaokrouhlí číslo *Číslo* směrem dolů (k nule) na počet desetinných míst podle argumentu *PočetDesMíst*.

Poznámka:

- Funkce **Rounddown** se chová jako funkce **Roundup** s tím rozdílem, že vždy zaokrouhluje nahoru.
- Je-li argument *PočetDesMíst* větší než 0, zaokrouhlí se číslo dolů na počet desetinných míst daný touto číslicí.
- Je-li argument *PočetDesMíst* roven 0 nebo je vynechán, zaokrouhlí se číslo na nejbližší nižší celou hodnotu.
- Je-li *PočetDesMíst* menší než 0, zaokrouhlí se číslo dolů doleva od řádové čárky.

Příklady:

ROUNDDOWN(3,2; 0) = 3
ROUNDDOWN(76,9;0) = 76
ROUNDDOWN(3,14159; 3) = 3,141
ROUNDDOWN(-3,14159; 1) = -3,1
ROUNDDOWN(31415,92654; -2) = 31400

Roundup(Číslo;PočetDesMíst)

Funkce zaokrouhlí číslo *Číslo* směrem nahoru (od nuly) na počet desetinných míst daný argumentem *PočetDesMíst*.

Poznámky:

- Funkce **Roundup** se chová jako funkce **Round** s tím rozdílem, že vždy zaokrouhluje nahoru.
- Je-li argument *PočetDesMíst* větší než 0, zaokrouhlí se číslo nahoru na počet desetinných míst daný touto číslicí.
- Je-li argument *PočetDesMíst* roven 0 nebo je vynechán, zaokrouhlí se číslo na nejbližší vyšší celou hodnotu.
- Je-li argument *PočetDesMíst* menší než 0, zaokrouhlí se číslo nahoru doleva od desetinné čárky.

Příklady:

ROUNDUP(3,2;0) = 4
ROUNDUP(76,9;0) = 77
ROUNDUP(3,14159; 3) = 3,142
ROUNDUP(-3,14159; 1) = -3,2
ROUNDUP(31415,92654; -2) = 31500

Sign(Číslo)

Vrací znaménko argumentu. Vráti:

1	jestliže <i>Číslo</i> je kladné
0	jestliže <i>Číslo</i> je nula
-1	jestliže <i>Číslo</i> je záporné.

Příklad: **SIGN(-5) = -1**

Sin(Číslo)

Vráti sinus daného úhlu.

Příklad: $\text{SIN}(\text{PI}()/2) = 1$

Sinh(Číslo)

Vrátí hyperbolický sinus úhlu zadaného v radiánech.

Příklad: $\text{SINH}(\text{PI}()/2) = 2,301298902$

Sqrt(Číslo)

Vrátí druhou odmocninu daného argumentu *Číslo*. Argument *Číslo* musí být větší nebo roven 0.

Příklad: $\text{SQRT}(16) = 4$

$\text{SQRT}(-16) = \#DATA!$

Subtotal(ČísloFunkce; odk1; odk2; ...)

Vrátí souhrn dat v seznamu nebo v databázi. Obecně je snazší vytvořit seznam souhrnů pomocí příkazu **Souhrny** v menu **Data**. Po vytvoření seznamu souhrnů jej lze upravovat prostřednictvím úprav ve vzorci pro funkci **Subtotal**.

ČísloFunkce je číslo od 1 do 11, které určuje typ funkce použité pro výpočet souhrnu v seznamu.

<i>Číslo funkce</i>	<i>Funkce</i>
1	AVERAGE
2	COUNT
3	COUNTA
4	MAX
5	MIN
6	PRODUCT
7	STDEV
8	STDEVP
9	SUM
10	VAR
11	VARP

refn je 1 až 29 oblastí nebo odkazů, u kterých má být proveden souhrn.

Poznámky:

- Případné další souhrny v odkazu (neboli vnořené souhrny) se ignorují, aby se předešlo jejich dvojímu započítání.
- Funkce **Subtotal** ignoruje všechny skryté řádky, které jsou výsledkem filtrování seznamu. Tato vlastnost je důležitá tehdy, když chcete pouze souhrn dat viditelných po filtrování.
- Pokud je některý odkaz ve tvaru 3D, vrátí funkce **Subtotal** chybovou hodnotu **!!chHod**.

Příklad:

SUBTOTAL(9;C3:C5) vypočítá souhrn buněk v oblasti C3:C5 s použitím funkce **Sum**.

Sum(Seznam)

Sečte dané argumenty. Argument *Seznam* je seznamem jedné nebo více hodnot (čísel nebo odkazů), oddělených čárkami.

Příklady: $\text{SUM}(1;5;4;6) = 16$

$\text{SUM}(a1:a10) = 54$

Sumif(TestovacíOblast; Podmínka; OblastHodnot)

Sečte buňky které splňují zadanou podmínku.

TestovacíOblast je vyhodnocovaná oblast buněk

Podmínka udávají podmínku, která definuje přidávané buňky. Vyjadřuje se číslem, výrazem nebo textem. Podmínka může mít například tvar: 32, "32", ">32", "jablka".

OblastHodnot obsahuje přímo sčítané buňky. Tyto buňky se sčítají pouze tehdy, pokud vyhovují podmínce zadané v argumentu *Kritéria*. Pokud je argument *Součet* vynechán, sečtou se buňky v oblasti.

Příklad:

Buňky A1:A4 obsahují například postupně ceny majetku pro čtyři různé byty: 100 000 Kč, 200 000 Kč, 300 000 Kč, 400 000 Kč.

Buňky B1:B4 obsahují postupně poplatky z prodeje pro každou z odpovídajících cen: 7 000 Kč, 14 000 Kč, 21 000 Kč, 28 000 Kč. Pak:

SUMIF(A1:A4;">160000";B1:B4) = 63 000 Kč

Sumproduct(Odkaz1;Odkaz2; ...)

Vynásobí odpovídající položky uvedených polí (matic) a vrátí součet násobků jednotlivých položek.

Odkaz1;Odkaz2;... jsou 2 až 30 polí (matic), jejichž jednotlivé položky chcete násobit a poté sečíst

- Pole, uvedené jako argumenty funkce, musí být stejně velké. Pokud nejsou, vrátí funkce chybovou hodnotu **!!chHod**.
- Položky polí, které nejsou číselného typu, zpracovává funkce jako nuly.

Příklad:

	A	B	C	D	E
1	3	4		2	7
2	8	6		6	7
3	1	9		5	3
4					

Následující vzorec vynásobí všechny položky dvou polí předešlého listu. Výsledek funkce se vypočítá následovně: $3*2 + 4*7 + 8*6 + 6*7 + 1*5 + 9*3$. Pak tedy:

SUMPRODUCT({3;4;8;6;1;9}; {2;7;6;7;5;3}) = 156

Poznámka:

Předcházející příklad vrátí stejný výsledek jako vzorec **SUM(A1:B3*D1:E3)**.

Použití matic poskytuje větší možnosti při provádění operací podobných funkci **SUMPRODUCT**. Můžete například spočítat sumu čtverců položek oblasti A1:B3 tak, že ji zadáte jako matici do vzorce **SUM(A1:B3^2)**.

Sumsq(Seznam)

Vrátí součet čtverců argumentů.

Seznam je až 30 argumentů, jejichž druhé mocniny chcete sečíst. Místo výčtu argumentů oddělených čárkou můžete použít jednorozměrné pole (matici) nebo odkaz.

Příklad:SUMSQ(3; 4) = 25

Sumx2my2(Odkaz1;Odkaz2)

Vrátí součet rozdílů čtverců u odpovídajících prvků dvou polí (matic).

Odkaz1 první matice či oblast dat

Odkaz2 druhá matice či oblast dat

Poznámky:

- Argumenty funkce mohou být čísla nebo názvy, matice nebo odkazy obsahující čísla.
- Jestliže argument označující matici nebo odkaz na matici obsahuje text, logickou hodnotu nebo prázdnou buňku, je hodnota argumentu ignorována. Tyto buňky jsou brány jako buňky obsahující nulovou hodnotu.

- Pokud *Odkaz1* a *Odkaz2* obsahují rozdílný počet hodnot, vrátí funkce chybovou hodnotu **!!chNe**.

Příklad: SUMX2MY2(A2:A5; C2:C5)

Sumx2py2(Odkaz1;Odkaz2)

Vrátí součet součtů čtverců u odpovídajících prvků dvou polí (matic). Součet součtů čtverců je velmi často používán při statistických výpočtech.

Odkaz1 první matice či oblast dat
Odkaz2 druhá matice či oblast dat

Poznámky:

- Argumenty funkce mohou být čísla nebo názvy, matice nebo odkazy obsahující čísla.
- Jestliže argument označující matici nebo odkaz na matici obsahuje text, logickou hodnotu nebo prázdnou buňku, je hodnota argumentu ignorována. Tyto buňky jsou brány jako buňky obsahující nulovou hodnotu.
- Pokud *Odkaz1* a *Odkaz2* obsahují rozdílný počet hodnot, vrátí funkce chybovou hodnotu **!!chNe**.

Příklad: SUMX2PY2(B3:B10;D3:D10)

Sumxmy2(Odkaz1;Odkaz2)

Vrátí součet čtverců rozdílů u odpovídajících prvků dvou polí (matic).

Odkaz1 první matice či oblast dat
Odkaz2 druhé matice či oblast dat

Poznámky:

- Argumenty funkce mohou být čísla nebo názvy, matice nebo odkazy obsahující čísla.
- Jestliže argument označující matici nebo odkaz na matici obsahuje text, logickou hodnotu nebo prázdnou buňku, je hodnota argumentu ignorována. Tyto buňky jsou brány jako buňky obsahující nulovou hodnotu.
- Pokud *Odkaz1* a *Odkaz2* mají rozdílný počet hodnot, vrátí funkce chybovou hodnotu **!!chNe**.

Příklad: SUMXMY2(A1:B3;D1:E3)

Tan(Číslo)

Vrátí tangens argumentu *Číslo* zadaného v radiánech.

Příklad: TAN(PI()/4) = 1

Tanh(Číslo)

Vrátí hyperbolický tangens argumentu *Číslo* zadaného v radiánech.

Příklad: TANH(PI()/4) = 0,655794203

Trunc(Číslo;PočetDesMíst)

Funkce ponechá v čísle *Číslo* pouze počet desetinných míst daný argumentem *PočetDesMíst* a zaokrouhlí ho tak dolů. Implicitní hodnota argumentu přesnost je nula (odříznou se všechna desetinná místa).

Příklady:

Trunc(8,9) = 8
Trunc(-8,9) = -8
Trunc(PI()) = 3

Logické funkce

Logické funkce se používají ke zjištění pravdivosti nebo nepravdivosti podmínky nebo ke kontrole více podmínek. Pomocí funkce **IF** můžete například určit, je-li podmínka pravdivá nebo nepravdivá. Pokud je podmínka pravdivá, bude vrácena určitá hodnota, jestliže je nepravdivá, bude vrácena jiná hodnota.

And(Výraz1;Výraz2; ...)

Funkce vrací hodnotu TRUE, jsou-li všechny argumenty pravdivé – mají hodnotu TRUE.

Příklady: **AND(TRUE; TRUE)** vrací hodnotu **TRUE**
AND(TRUE; TRUE; FALSE) vrací hodnotu **FALSE**
AND(2+2=4; 2+3=5) vrací hodnotu **TRUE**

False()

Vrátí logickou hodnotu nepravda: FALSE.

Podmíněný příkaz IF

Podmíněný příkaz slouží k vyhodnocení obsahu buňky na základě podmínky dané logickým výrazem. Příkaz má syntaxi:

IF(podmínka; if_true; if_false)

kde:

<i>podmínka</i>	logická podmínka
<i>if_true</i>	hodnota, která bude buňce přiřazena v případě splnění podmínky
<i>if_false</i>	hodnota, která bude buňce přiřazena v případě nesplnění podmínky.

Podmínky se vytvářejí z označení buněk, funkcí, konstant a operátorů.

G15 > 0

B2 >= 10000

Lze také vytvářet složené podmínky. Ty vytvoříte ozávkováním jednoduchých podmínek a také s využitím logických funkcí **AND** a **OR**.

Je možné používat tyto logické operátory:

operátor	význam
>	větší než
=	rovno čemu
<	menší než
>=	větší nebo rovno než
<=	menší nebo rovno než
<>	není rovno čemu

Příklady: **=IF(AND (A1 > 0; A2 = 1000); 5000; 50000)**

=IF(TYPE(A1)=5; “ujujujuj, chybička !”; “bez chyby”)

Not(Výraz)

Provede logickou negaci zadané hodnoty.

Or(Výraz1;Výraz2)

Vrací TRUE, je-li alespoň jeden argument z výrazů roven TRUE.

True()

Vrátí pravdivou logickou hodnotu TRUE.

Statistické funkce

Statistické funkce listu slouží ke statistickým analýzám oblastí dat. Poskytuje statistické informace o průměru proložené skupinou hodnot, jako je směrnice a průsečík s osou Y, nebo o skutečných bodech tvořících přímku.

Avedev(Seznam)

Vrátí průměr absolutních odchylek bodů dat od jejich střední hodnoty. Funkce je měřítkem variability množiny dat.

Seznam je 1 až 30 argumentů, jejichž průměr absolutních odchylek chcete zjistit. Místo argumentů oddělených čárkami můžete použít matici nebo odkaz na matici.

Poznámky:

- Argumenty musí být čísla, případně názvy, matice nebo odkazy, obsahující čísla.
- Pokud argument ve tvaru matice nebo odkazu obsahuje text, logické hodnoty nebo prázdné buňky, budou tyto hodnoty ignorovány; buňky s nulovou hodnotou budou však brány v úvahu.

Příklad: **AVEDEV(4; 5; 6; 7; 5; 4; 3) = 1,020408**

Average(Seznam)

Vrací průměrnou hodnotu argumentů *Seznam*.

Poznámky:

- Argumenty musí být čísla nebo odkazy na buňky obsahující číselné hodnoty.
- Nenumerné hodnoty jsou ignorovány, prázdné buňky jsou posuzovány, jako by měly nulovou hodnotu.

Příklad: **Average(5;6)** vrací hodnotu 5,5

AverageA(Seznam)

Vrátí aritmetický průměr argumentů.

Seznam je 1 až 30 číselných argumentů, jejichž průměr chcete zjistit

Poznámky:

- Argumenty musí být čísla, případně názvy, matice či odkazy, obsahující čísla.
- Pokud argument ve tvaru matice nebo odkazu obsahuje text, logické hodnoty nebo prázdné buňky, budou tyto hodnoty ignorovány; buňky s nulovou hodnotou budou však brány v úvahu.

Příklady:

Pokud A1:A5 má název Počty a obsahuje čísla 10, 7, 9, 27 a 2, pak:

AVERAGEA(A1:A5) = 11

AVERAGEA(Počty) = 11

AVERAGEA(A1:A5; 5) = 10

AVERAGEA(A1:A5) = SUMA(A1:A5)/POČET(A1:A5) = 11

Pokud C1:C3 má název JinéPočty a obsahuje čísla 4, 18 a 7, pak:

AVERAGEA(Počty;JinéPočty) = 10,5

Betadist (X;Alfa;Beta;A;B)

Vrátí hodnotu hustoty rozdělení součtové pravděpodobnosti *Beta*. Pomocí hustoty rozdělení součtové pravděpodobnosti beta se obvykle zkoumá změna procentové části určitého jevu pro dané výběry, například část dne, po kterou lidé sledují televizi.

- X* je hodnota mezi hodnotami argumentů A a B, pro kterou chcete zjistit hodnotu funkce.
- Alfa* je parametr rozdělení.
- Beta* je parametr rozdělení.
- A* je volitelná dolní mez intervalu hodnot *X*.

B je volitelná horní mez intervalu hodnot X .

Poznámky:

- Pokud některý z argumentů není číselného typu, vrátí funkce **Betadist** chybovou hodnotu **!!chHod**.
- Jestliže je argument *Alfa* menší nebo rovno 0 nebo *Beta* menší nebo rovno 0, funkce vrátí chybovou hodnotu **!!chCis**.
- Je-li argument $X < A$, $x > B$ nebo $A = B$, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud nezadáte hodnoty A a B , použije funkce standardní součtové rozdělení beta, takže $A = 0$ a $B = 1$.

Příklad:

BETADIST(2;8;10;1;3) = 0,685470581

BetaInv(Pravděpodobnost;Alfa;Beta;A;B)

Vrátí inverzní hodnotu hustoty rozdělení součtové pravděpodobnosti beta. To znamená, že pokud se pravděpodobnost rovná hodnotě funkce **Betadist**, hodnota funkce **BetaInv** se rovná hodnotě x . Součtové rozdělení beta lze použít při plánování projektů pro modelování pravděpodobné doby ukončení, je-li zadána očekávaná doba ukončení a proměnlivost.

Pravděpodobnost je pravděpodobnost rozdělení beta.

Alfa je parametr rozdělení.

Beta je parametr rozdělení.

A je volitelná dolní mez intervalu hodnot x .

B je volitelná horní mez intervalu hodnot x .

Poznámky:

- Pokud některý z argumentů není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Jestliže je argument *alfa* menší nebo roven 0 nebo argument *beta* menší nebo roven 0, vrátí funkce **BETAINV** chybovou hodnotu **!!chCis**.
- Je-li argument *prst* menší nebo roven 0 nebo $prst > 1$, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud nezadáte hodnoty A a B , použije funkce **BETAINV** standardní součtové rozdělení beta, takže $A = 0$ a $B = 1$.
- Funkce **BetaInv** používá k výpočtu iterační metodu. Je-li zadána hodnota pravděpodobnosti, provádí funkce iterace, dokud výsledek nedosáhne přesnosti $\pm 3 \times 10^{-7}$. Pokud funkce **BetaInv** nekonverguje ani po 100 iteracích, vrátí funkce chybovou hodnotu **!!chNe**.

Příklad:

BETAINV(0,685470581;8;10;1;3) = 2

Binomdist(PočetÚspěch;Pokusy;Pravděpodobnost;Par)

Vrátí hodnotu binomického rozdělení pravděpodobnosti jednotlivých veličin. Funkci **Binomdist** lze použít u problémů s pevným počtem testů nebo pokusů, kdy výsledek každého pokusu je pouze úspěch nebo neúspěch, pokusy jsou nezávislé a pravděpodobnost úspěchu během experimentu je konstantní. Pomocí funkce **Binomdist** můžete například vypočítat pravděpodobnost, že dvě z dalších tří narozených dětí budou chlapi.

Syntaxe:

PočetÚspěch je počet úspěšných pokusů.

Pokusy je počet nezávislých pokusů.

Pravděpodobnost je pravděpodobnost každého úspěšného pokusu.

Par je logická hodnota, která určuje tvar funkce. Jestliže se argument *počet* rovná logické hodnotě **TRUE**, funkce **Binomdist** vrátí součtovou distribuční funkci, což je pravděpodobnost, že počet úspěchů bude nejvýše roven hodnotě argumentu *PočetÚspěch*. Pokud se tento argument rovná logické hodnotě **FALSE**, vrátí funkce hromadnou pravděpodobnostní funkci, což je

pravděpodobnost, že počet úspěchů bude roven hodnotě argumentu *PočetÚspěch*.

Poznámky:

- Argumenty *PočetÚspěch* a *pokusy* jsou zkráceny na celá čísla.
- Pokud jeden z argumentů *PočetÚspěch*, *pokusy* nebo *Pravděpodobnost* není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Jestliže je hodnota *PočetÚspěch* < 0 nebo jestliže je hodnota *PočetÚspěch* větší než hodnota *pokusy*, vrátí **Binomdist** chybovou hodnotu **!!chCis**.
- Je-li argument *Pravděpodobnost* < 0 nebo *Pravděpodobnost* > 1, vrátí funkce **Binomdist** chybovou hodnotu **!!chCis**.

Příklad:

Výsledek hodu mincí může být panna nebo orel. Pravděpodobnost, že první hod bude panna, je 0,5. Pravděpodobnost, že z 10 hodů bude přesně šest panna, je:

BINOMDIST(6;10;0,5;FALSE) = 0,205078

Combin(n1;n2)

Funkce vrací počet kombinací daného čísla nebo objektu, který lze převést na objekt numerického typu. Kombinace je sada objektů nebo událostí, pro které nemá význam vnitřní pořadí prvků. Kombinace nesmí být zaměňovány s permutacemi, pro které je naopak vnitřní pořadí významné.

n1 celočíselná hodnota ≥ 0 , udávající počet položek
n2 celočíselná hodnota ≥ 0 , udávající počet kombinací.

Oba argumenty se zaokrouhlují na celočíselné hodnoty. Zadáte-li nenumernický argument, funkce vrací chybovou hodnotu **#NAME?**. Zadáte-li *n1* menší než nula nebo *n2* menší než nula nebo *n1* menší než *n2*, funkce vrací chybovou hodnotu **!!chCis**.

kombinace se počítají podle vztahu:

$$= n! / k! (n - k)!$$

Příklad:

Předpokládejme, že budete chtít vědět, kolika způsoby lze sestavit čtyřčlennou posádku lodi, jestliže je k dispozici 12 kandidátů. Na pořadí, v jakém v lodi budou sedět, nezáleží.

Funkce **=COMBIN(12;4)** dává výsledek **495**

Confidence(Alfa,Odchylka;Vzorek)

Vrátí interval spolehlivosti pro střední hodnotu základního souboru. Interval spolehlivosti je oblast po obou stranách střední hodnoty výběru. Pokud si například objednáte výrobek poštou, můžete s určitým stupněm spolehlivosti určit, kdy nejdříve a kdy nejpozději byste měli zboží dostat.

Alfa je hladina významnosti, pomocí které je vypočítána hladina spolehlivosti. Hladina spolehlivosti se rovná $100 \cdot (1 - \text{alfa})\%$, tzn. je-li argument *alfa* roven 0,05, hladina spolehlivosti je 95%.

Odchylka je směrodatná odchylka základního souboru pro danou oblast dat a pokládá se za známou.

Vzorek je velikost výběru.

Poznámky:

- Pokud některý z argumentů není číselného typu, vrátí funkce **Confidence** chybovou hodnotu **!!chHod**.
- Pokud je argument *Alfa* menší nebo roven 0 nebo *Alfa* větší nebo roven 1, vrátí funkce **Confidence** chybovou hodnotu **!!chCis**.
- Jestliže je argument *Odchylka* menší nebo roven 0, vrátí funkce chybovou hodnotu **!!chCis**.
- Není-li argument *Vzorek* celé číslo, bude zkrácen.
- Pokud je argument *Vzorek* < 1, vrátí funkce **Confidence** chybovou hodnotu **!!chCis**.

Příklad:

Předpokládejme, že ve výběru 50 dojíždějících činí průměrná doba jejich cesty do zaměstnání 30 minut se směrodatnou odchylkou základního souboru 2,5. Můžeme si být na 95 procent jisti, že střední hodnota základního souboru leží podle:

$$\text{CONFIDENCE}(0,05;2,5;50) = 0,692951.$$

Jinými slovy, průměrná délka cesty do zaměstnání je rovna $30 \pm 0,692951$ minut, neboli 29,3 až 30,7 minut.

Correl(Odkaz1;Odkaz2)

Vrátí korelační koeficient oblastí buněk *Odkaz1* a *Odkaz2*. Pomocí korelačního koeficientu je možné určit vztah mezi dvěma vlastnostmi. Můžete například zkoumat vztah mezi teplotou určitého místa a používáním klimatizace.

Odkaz1 je oblast buněk s hodnotami
Odkaz2 je druhá oblast buněk s hodnotami

Poznámky:

- Argumenty musí být čísla, případně názvy, matice nebo odkazy obsahující čísla.
- Pokud je argument matice nebo odkaz obsahující text, logické hodnoty nebo prázdné buňky, budou tyto hodnoty přeskočeny. Buňky s nulovou hodnotou však budou zahrnuty.
- Jestliže argumenty *Odkaz1* a *Odkaz2* mají rozdílný počet datových bodů, vrátí funkce chybovou hodnotu **!!chNe**.
- Je-li jeden z argumentů *Odkaz1* nebo *Odkaz2* prázdný nebo je-li směrodatná odchylka jejich hodnot rovna nule, vrátí funkce chybovou hodnotu **!!ch0**.

Příklad: CORREL({3;2;4;5;6},{9;7;12;15;17}) = 0,997054

Count(Seznam)

Funkce vrací počet buněk obsahujících čísla v oblasti (oblastech) uvedených v *Seznamu* argumentů.

Příklad: COUNT(A1:A10) = 2,

pokud 2 buňky z oblasti **A1:A10** obsahují čísla.

Counta(Seznam)

Funkce vrací počet neprázdných buněk v oblasti (oblastech) uvedených v *Seznamu* argumentů.

Příklad: COUNTA(A1:A10;D5:G10) = 5,

pokud 5 buněk z oblastí **A1:A10** a **D5:G10** obsahuje např. čísla a popisné texty.

Countblank(Odkaz)

Spočítá prázdné buňky v dané oblasti.

Odkaz je oblast, ve které chcete spočítat prázdné buňky

Poznámky:

Buňky se vzorci, které vrací "" (neboli prázdný řetězec), se také započítávají, zatímco buňky s nulovými hodnotami se nezapočítávají.

Příklad:

	A	B	C	D
1				
2		6		
3			27	
4		4	34	
5		4	0	
6				

Například buňka B3 ve výše uvedeném sešitě obsahuje následující vzorec: **IF(C3<30;"";C3)**, který vrací "" (prázdný řetězec). Pak:

$$\text{COUNTBLANK}(B2:C5) = 2$$

Covar(Odkaz1;Odkaz2)

Vrátí kovarianci, průměr součinů odchylek pro každou dvojici bodů dat. Pomocí kovariance určíte souvislost mezi dvěma soubory dat. Můžete například zkoumat, zda vyšší příjmy souvisí s vyšším stupněm vzdělání.

Odkaz1 je první oblast buněk s celými čísly

Odkaz2 je druhá oblast buněk s celými čísly

Poznámky:

- Argumenty musí být čísla nebo názvy, matice a odkazy obsahující čísla.
- Pokud některý argument ve tvaru matice nebo odkazu obsahuje text, logické hodnoty nebo prázdné buňky, budou tyto hodnoty ignorovány. Buňky s nulovou hodnotou však budou zahrnuty.
- Pokud *Odkaz1* a *Odkaz2* mají různý počet bodů dat, vrátí funkce chybovou hodnotu **!!chNe**.
- Pokud je *Odkaz1* nebo *Odkaz2* prázdné, vrátí funkce chybovou hodnotu **!!ch/0**.

Příklad: $\text{COVAR}(\{3; 2; 4; 5; 6\}; \{9; 7; 12; 15; 17\}) = 5,2$

Critbinom(Pokusy;Pravděpodobnost;Alfa)

Vrátí nejmenší hodnotu, pro kterou má součtové binomické rozdělení hodnotu větší nebo rovnu hodnotě kritéria. Tato funkce se používá při kontrole a zajišťování kvality. Pomocí funkce **Critbinom** můžete například určit největší možný počet kazových součástí, které mohou opustit výrobní linku, aniž by bylo třeba odmítnout celou sérii.

Pokusy je počet Bernoulliho pokusů.

Pravděpodobnost je pravděpodobnost každého úspěšného pokusu.

Alfa je hodnota kritéria.

Poznámky:

- Pokud některý z argumentů není číselného typu, vrátí funkce **Critbinom** chybovou hodnotu **!!chHod**.
- Jestliže argument *Pokusy* není celé číslo, bude zkrácen.
- Je-li argument *Pokusy* < 0, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud je argument *Pravděpodobnost* < 0 nebo > 1, vrátí funkce chybovou hodnotu **!!chCis**.
- Jestliže je *Alfa* < 0 nebo *Alfa* > 1, vrátí funkce **CRITBINOM** chybovou hodnotu **!!chCis**.

Příklad:

$$\text{CRITBINOM}(6;0,5;0,75) = 4$$

Devsq(Seznam)

Vrátí součet čtverců odchylek datových bodů od jejich střední hodnoty výběru.

Seznam je 1 až 30 argumentů, pro které chcete vypočítat součet čtverců odchylek. Místo argumentů oddělených středníky lze použít jednu matici nebo odkaz na matici.

Poznámky:

- Argumenty musí být čísla, případně názvy, matice nebo odkazy obsahující čísla.
- Pokud je argument matice nebo odkaz obsahující text, logické hodnoty nebo prázdné buňky, budou tyto hodnoty přeskočeny. Buňky s nulovou hodnotou však budou zahrnuty.

Příklad: $\text{DEVSQ}(4;5;8;7;11;4;3) = 48$

Expondist(x;Lambda;Par)

Vrátí hodnotu exponenciálního rozdělení. Pomocí funkce **Expondist** lze modelovat čas mezi událostmi, jako je například doba, za kterou bankomat vydá peníze. Pomocí funkce **Expondist** můžete například určit pravděpodobnost, že tento proces trvá nanejvýš jednu minutu.

x	je hodnota funkce.
λ	je hodnota parametru.
Par	je logická hodnota, která určuje, který tvar exponenciální funkce chcete vrátit. Pokud je argument Součet logická hodnota TRUE , vrátí funkce Expodist součtovou distribuční funkci. Jestliže je tento argument roven logické hodnotě FALSE , vrátí funkce hustotu rozdělení pravděpodobnosti.

Poznámky:

- Pokud jeden z argumentů x nebo λ není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Jestliže je argument $x < 0$, vrátí funkce chybovou hodnotu **!!chCis**.
- Je-li argument $\lambda = 0$, vrátí funkce chybovou hodnotu **!!chCis**.

Příklady:

EXPONDIST(0,2;10;TRUE) = 0,864665

EXPONDIST(0,2;10;FALSE) = 1,353353

Fdist(x;StVolnosti1;StVolnosti2)

Vrátí hodnotu rozdělení pravděpodobnosti **F**. Pomocí této funkce je možné určit, zda mají dvě množiny dat různé stupně odlišnosti. Můžete například zkoumat výsledky přijímacích zkoušek na střední školu u mužů a žen a rozhodnout, zda se proměnlivost těchto výsledků u mužů a u žen liší.

x je hodnota, pro kterou chcete zjistit rozdělení pravděpodobnosti.

$StVolnosti1$ je počet stupňů volnosti v čitateli.

$StVolnosti2$ je počet stupňů volnosti ve jmenovateli.

Poznámky:

- Pokud některý z argumentů není číselného typu, vrátí funkce **Fdist** chybovou hodnotu **!!chHod**.
- Jestliže je hodnota x záporná, vrátí funkce **Fdist** chybovou hodnotu **!!chCis**.
- Není-li jeden z argumentů $StVolnosti1$ nebo $StVolnosti2$ celé číslo, bude zkrácen.
- Jestliže je argument $volnost1 < 1$ nebo $volnost1 = 10^{10}$, vrátí funkce **Fdist** chybovou hodnotu **!!chCis**.
- Pokud je argument $StVolnosti2 < 1$ nebo $StVolnosti2 = 10^{10}$, vrátí funkce **Fdist** chybovou hodnotu **!!chCis**.
- Funkce **Fdist** je počítána jako $FDIST=P(F < x)$, kde F je náhodná proměnná s rozdělením **F**.

Příklad:

FDIST(15,20675;6;4) = 0,01

Finv(Pravděpodobnost;StVolnosti1;StVolnosti2)

Vrátí hodnotu inverzní funkce k distribuční funkci rozdělení **F**. Jestliže $p = FDIST(x,...)$, $FINV(p,...) = x$.

Pomocí rozdělení **F** je možné porovnat stupeň proměnlivosti dvou množin dat v F-testu. Můžete například analýzou rozdělení příjmů v České republice a na Slovensku rozhodnout, zda je v těchto dvou zemích podobný stupeň odlišnosti.

Pravděpodobnost je pravděpodobnost součtového rozdělení **F**.

$StVolnosti1$ je počet stupňů volnosti v čitateli.

$StVolnosti2$ je počet stupňů volnosti ve jmenovateli.

Poznámky:

- Pokud některý z argumentů není číselného typu, vrátí funkce **Finv** chybovou hodnotu **!!chHod**.
- Jestliže je argument *Pravděpodobnost* < 0 nebo *Pravděpodobnost* > 1 , vrátí funkce chybovou hodnotu **!!chCis**.
- Není-li jeden z argumentů $StVolnosti1$ nebo $StVolnosti2$ celé číslo, bude zkrácen.
- Pokud je argument $StVolnosti1 < 1$ nebo $StVolnosti1$ větší či roven 10^{10} , vrátí funkce chybovou hodnotu **!!chCis**.

- Je-li argument $StVolnosti2 < 1$ nebo $StVolnosti2$ větší či roven 10^{10} , vrátí funkce chybovou hodnotu **!!chCis**.

Pomocí funkce **Finv** můžete vracet kritické hodnoty rozdělení **F**. Výsledek výpočtu ANOVA například často zahrnuje data pro testovací kritérium **F**, pravděpodobnost **F** a kritickou hodnotu **F** na hladině významnosti 0,05. Kritickou hodnotu **F** získáte tak, že hladinu významnosti dosadíte do funkce **Finv** jako argument pravděpodobnosti.

Funkce **Finv** používá k výpočtu iterační metodu. Je-li zadána hodnota pravděpodobnosti, provádí funkce iterace, dokud výsledek nedosáhne přesnosti $\pm 3 \times 10^{-7}$. Pokud funkce **Finv** nekonverguje ani po 100 iteracích, vrátí chybovou hodnotu **!!chNe**.

Příklad:

$$\mathbf{FINV(0,01;6;4) = 15,20675}$$

Fisher(Číslo)

Funkce vrací Fisherovu transformaci numerické hodnoty c . Tato transformace je přibližnou aproximací normálního rozdělení. Používá se k testování korelačních koeficientů.

Jestliže zadáte nenumerickou hodnotu argumentu, funkce vrací chybovou hodnotu **!!chHod**. Pokud je **Číslo** menší nebo rovno -1, případně větší nebo rovno 1, funkce vrací chybovou hodnotu **!!chCis**.

Fisherova funkce se vyčísluje podle vztahu $= 0,5 \ln((1 + c) / (1 - c))$.

Příklad: **FISHER(0,75)** dává výsledek **0,972955**

Fisherinv(Číslo)

Funkce vrací inverzní Fisherovu transformaci numerické hodnoty **Číslo**. Používá se k testování korelací mezi rozsahy dat. Jestliže platí, že $y = \text{FISHER}(x)$, pak $\text{FISHERINV}(y) = x$.

Jestliže zadáte numerickou hodnotu argumentu, funkce vrací chybovou hodnotu **!!chHod**.

Fisherova inverzní funkce se vyčísluje podle vztahu $= (\exp(2*c) - 1) / (\exp(2*c) + 1)$.

Příklad: **FISHER(0,972955)** dává výsledek **0,75**

Forecast(x;ŘadaY;ŘadaX)

Vypočítá nebo odhadne budoucí hodnotu na základě existujících hodnot. Předpovídaná hodnota je hodnota y pro danou hodnotu x . Znamé hodnoty jsou hodnoty x a y . Nová hodnota je předpovídaná na základě lineární regrese. Tuto funkci lze použít k předpovědi objemu budoucího prodeje, požadavků zásob nebo trendů u zákazníků.

x	je datový bod, u kterého chcete předpovědět hodnotu
ŘadaY	je matice nebo oblast závislých dat
ŘadaX	je matice nebo oblast nezávislých dat

Poznámky:

- Pokud x není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Pokud oblasti **ŘadaY** a **ŘadaX** jsou prázdné nebo obsahují odlišný počet bodů dat, vrátí funkce chybovou hodnotu **!!chNe**.
- Pokud se rozptyl argumentu **ŘadaX** rovná nule, vrátí funkce chybovou hodnotu **!!ch/0**.

Příklad: **FORECAST(30;{6;7;9;15;21};{20;28;31;38;40}) = 10,60725**

Ftest(Odkaz1;Odkaz2)

Vrátí výsledek F-testu. F-test vrátí pravděpodobnost, že se rozptyly v argumentech **Odkaz1** a **Odkaz2** významně neliší. Tuto funkci lze použít pro rozhodnutí, zda dva výběry mají odlišný rozptyl. Pokud například porovnáváte výsledky testů ze dvou různých typů škol, můžete zjišťovat, zda mají tyto školy odlišný rozptyl výsledků.

Odkaz1	je první matice nebo oblast dat
Odkaz2	je druhá matice nebo oblast dat.

Poznámky:

- Argumenty musí být čísla, případně názvy, matice či odkazy, které obsahují čísla.
- Pokud argument ve tvaru matice nebo odkazu obsahuje text, logické hodnoty nebo prázdné buňky, tyto hodnoty se ignorují. Buňky s nulovou hodnotou se však započítají.
- Pokud je počet bodů dat v argumentu *Odkaz1* nebo *Odkaz2* menší než 2, nebo pokud je rozptyl argumentu *Odkaz1* nebo *Odkaz2* nulový, vrátí funkce chybovou hodnotu **!!ch/0**.

Příklad: $FTEST(\{6;7;9;15;21\};\{20;28;31;38;40\}) = 0,648318$

Gammadist(x;Alpha;Beta;Par)

Vrátí gama rozdělení. Tuto funkci lze použít pro zkoumání proměnných, které mohou mít zešíkmené rozdělení. Gama rozdělení se obvykle používá při analýze front.

x je hodnota, pro kterou chcete zjistit hodnotu rozdělení.
Alfa je parametr rozdělení.
Beta je parametr rozdělení. Pokud *beta* = 1, funkce vrátí standardní gama rozdělení.
Par je logická hodnota, která určuje tvar funkce. Pokud má součet hodnotu **TRUE**, funkce vrátí součtovou distribuční funkci; má-li hodnotu **FALSE**, vrátí pravděpodobnostní míru.

Poznámky:

- Pokud argument *x*, *Alfa* nebo *Beta* není číselného typu, vrátí funkce **Gammadist** chybovou hodnotu **!!chHod**.
- Pokud $x < 0$, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud je argument *Alfa* menší nebo roven 0 nebo pokud je argument *Beta* menší nebo roven 0, vrátí funkce **Gammadist** chybovou hodnotu **!!chCis**.
- Pokud *Alfa* = 1, **Gammadist** vrátí exponenciální rozdělení
- Pro kladné celé číslo *n*, kde *Alfa* = *n*/2, *Beta* = 2 a *Par* = **TRUE**, funkce vrátí (1 - CHIDIST(*x*)) o *n* stupních volnosti.
- Pokud je argument *Alfa* celé kladné číslo, nazývá se funkce **Gammadist** také Erlangovo rozdělení.

Příklady:

$GAMMADIST(10;9;2;FALSE) = 0,032639$

$GAMMADIST(10;9;2;TRUE) = 0,068094$

Gammainv(Pravděpodobnost;Alfa;Beta)

Vrátí inverzní funkci ke součtovému gama rozdělení. Pokud $p = GAMMADIST(x;...)$, pak $GAMMAINV(p,...) = x$

Pomocí této funkce lze zkoumat proměnné, jejichž rozdělení může být zešíkmené.

Pravděpodobnost je pravděpodobnost spojená s gama rozdělením.
Alfa je parametr rozdělení.
Beta je parametr rozdělení. Pokud *beta* = 1, funkce vrátí standardní gama rozdělení.

Poznámky:

- Pokud některý z argumentů není číselného typu, vrátí funkce **Gammainv** chybovou hodnotu **!!chHod**.
- Pokud $Pravděpodobnost < 0$ nebo $Pravděpodobnost > 1$, funkce vrátí chybovou hodnotu **!!chCis**.
- Pokud je argument *Alfa* menší nebo roven 0 nebo pokud je argument *Beta* menší nebo roven 0, vrátí funkce **Gammainv** chybovou hodnotu **!!chCis**.
- Pokud je argument *Beta* menší nebo roven nule, funkce vrátí chybovou hodnotu **!!chCis**.
- Funkce **Gammainv** počítá hodnoty funkce iterační metodou. Je-li dána hodnota pravděpodobnosti, provádí funkce iterace, dokud výsledek nedosáhne přesnosti $\pm 3 \times 10^{-7}$. Pokud funkce **Gammainv** nekonverguje ani po 100 iteracích, vrátí chybovou hodnotu **!!chNe**.

Příklad:

$GAMMAINV(0,068094;9;2) = 10$.

Gammaln(Číslo)

Vrátí přirozený logaritmus gama funkce, $G(x)$.

Číslo je hodnota, pro kterou chcete vypočítat funkci **Gammaln**.

Poznámky:

- Pokud x není číselného typu, funkce vrátí chybovou hodnotu **!!chHod**.
- Pokud x je menší nebo rovno 0, vrátí funkce chybovou hodnotu **!!chCis**.
- Číslo e umocněné na $Gammaln(i)$, kde i je celé číslo, vrátí stejný výsledek jako $(i - 1)!$.

Příklady:

GAMMALN(4) = 1,791759

EXP(GAMMALN(4)) = 6 nebo **(4 - 1)!**

Geomean(Seznam)

Vrátí geometrický průměr pole nebo oblasti kladných dat. Funkci lze například použít k výpočtu průměrné míry růstu daného složeného úrokování s proměnným úrokem.

Seznam je 1 až 30 argumentů, jejichž průměr chcete vypočítat. Místo argumentů oddělených čárkami lze také použít matici nebo odkaz na matici.

Poznámky:

- Argumenty musí být čísla, případně názvy, matice nebo odkazy na matice, které obsahují čísla.
- Pokud argument ve tvaru matice nebo odkazu obsahuje text, logické hodnoty nebo prázdné buňky, tyto hodnoty se ignorují. Buňky s nulovou hodnotou se však započítají.
- Pokud některý datový bod je menší nebo roven nule 0, vrátí funkce chybovou hodnotu **!!chCis**.

Příklad: GEOMEAN(4;5;8;7;11;4;3) = 5,476987

Harmean(Seznam)

Vrátí harmonický průměr množiny dat. Harmonický průměr je převrácená hodnota aritmetického průměru převrácených hodnot.

Seznam je 1 až 30 argumentů, jejichž průměr chcete vypočítat. Místo argumentů oddělených čárkami můžete také použít jediné pole nebo odkaz na pole.

Poznámky:

- Argumenty musí být čísla, případně názvy, matice nebo odkazy, které obsahují čísla.
- Pokud argument ve tvaru matice nebo odkazu obsahuje text, logické hodnoty nebo prázdné buňky, tyto hodnoty se ignorují. Buňky s nulovou hodnotou se však započítají.
- Pokud některý z datových bodů ≤ 0 , vrátí funkce chybovou hodnotu **!!chCis**.
- Harmonický průměr je vždy menší než geometrický průměr, který je vždy menší než aritmetický průměr.

Příklad: HARMEAN(4;5;8;7;11;4;3) = 5,028376

Hypgeomdist(Úspěch;Celkem;ZákladÚspěch;ZákladCelkem)

Vrátí hypergeometrické rozdělení. Funkce HYPGEOMDIST vrátí pravděpodobnost daného počtu úspěchů ve vzorku, je-li dána velikost vzorku, počet úspěchů v celé populaci a velikost populace. Funkce HYPGEOMDIST se používá u problémů týkajících se konečných populací, kde každé pozorování je buď úspěch nebo neúspěch a kde každá podmnožina dané velikosti má stejnou pravděpodobnost, že bude vybrána.

Úspěch je počet úspěchů ve vzorku.

Celkem je velikost vzorku.

ZákladÚspěch je počet úspěchů v celé populaci.

ZákladCelkem je velikost populace.

Poznámky:

- Všechny argumenty jsou zkráceny na celá čísla.
- Pokud některý z argumentů není číselného typu, vrátí funkce **Hypgeomdist** chybovou hodnotu **!!chHod**.
- Pokud $Úspěch < 0$ nebo úspěch je větší než menší z argumentů *Celkem* a *ZákladÚspěch*, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud *Úspěch* je menší než větší z hodnot 0 a (*Celkem* - *ZákladCelkem* + *ZákladÚspěch*), vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud $Celkem < 0$ nebo $Celkem > ZákladCelkem$ vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud $ZákladÚspěch < 0$ nebo $ZákladÚspěch > ZákladCelkem$, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud $ZákladCelkem < 0$, vrátí funkce chybovou hodnotu **!!chCis**.

Příklad:

Ukázková bonboniéra obsahuje 20 bonbónů. Osm bonbónů je karamelových, zbývajících 12 je s oříšky. Pokud si někdo náhodně vybere 4 bonbóny, vrátí následující funkce pravděpodobnost, že právě jeden z těchto bonbónů je karamelový.

$$\text{HYPGEOMDIST}(1;4;8;20) = 0,363261$$

Chidist(x;StVolnosti)

Vrátí jednostrannou pravděpodobnost rozdělení chí-kvadrát (χ^2). Rozdělení **chí-2** je spojeno s testem **chí-2**. Pomocí testu **chí-2** se porovnávají pozorované a očekávané hodnoty. U genetického experimentu můžete například předpokládat, že následující generace rostlin bude mít květy určitých barev. Porovnáním pozorovaných výsledků s očekávanými výsledky můžete rozhodnout, zda původní předpoklad platí.

x je hodnota, pro kterou chcete zjistit pravděpodobnost rozdělení.
StVolnosti je počet stupňů volnosti.

Poznámky:

- Pokud některý z argumentů není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Jestliže je hodnota *x* záporná, vrátí funkce chybovou hodnotu **!!chCis**.
- Není-li argument *StVolnosti* celé číslo, bude zkrácen.
- Pokud je argument *StVolnosti* < 1 nebo *volnost* je větší či roven 10^{10} , vrátí funkce chybovou hodnotu **!!chCis**.
- Formát sloučení CHIDIST * je vypočítán jako $\text{CHIDIST} = P(X > x)$, kde *X* je náhodná proměnná s rozdělením **chí-2**.

Příklad:

$$\text{CHIDIST}(18,307;10) = 0,050001$$

Chiinv(prst;StVolnosti)

Vrátí hodnotu funkce inverzní k distribuční funkci jednostranné pravděpodobnosti rozdělení chí-kvadrát (χ^2). Pokud je pravděpodobnost rovna hodnotě funkce CHIDIST(*x*,...), hodnota funkce CHIINV(pravděpodobnost,...) je rovna hodnotě *x*. Pomocí této funkce můžete porovnáním pozorovaných výsledků s očekávanými výsledky rozhodnout, zda původní předpoklad platí.

prst je pravděpodobnost rozdělení chí-2.
StVolnosti je počet stupňů volnosti.

Poznámky:

- Pokud některý z argumentů není číselného typu, funkce vrátí chybovou hodnotu **!!chHod**.
- Jestliže je argument *prst* < 0 nebo *prst* > 1 , vrátí funkce chybovou hodnotu **!!chCis**.
- Není-li argument *StVolnosti* celé číslo, bude zkrácen.
- Pokud je argument *StVolnosti* < 1 nebo *StVolnosti* větší či roven 10^{10} , vrátí funkce chybovou hodnotu **!!chCis**.

Funkce **Chiinv** používá pro výpočet funkce iterační metodu. Je-li zadána hodnota pravděpodobnosti, provádí funkce iterace, dokud výsledek nedosáhne přesnosti $\pm 3 \times 10^{-7}$. Pokud funkce **Chiinv** nekonverguje ani po 100 iteracích, vrátí funkce chybovou hodnotu **!!chNe**.

Příklad:

$$\text{CHIINV}(0,05;10) = 18,30703$$

Chitest(Odkaz1;Odkaz2)

Vrátí test nezávislosti. Funkce vrátí hodnotu rozdělení χ^2 pro dané testové kritérium a příslušné stupně volnosti. Pomocí testů χ^2 můžete určit, zda experiment potvrzuje předpokládané výsledky.

Odkaz1 je oblast dat obsahující pozorování, která chcete testovat a srovnávat s předpokládanými hodnotami

Odkaz2 je oblast dat obsahující podíl součinu součtů řádků a sloupců a celkového součtu

Poznámka:

Pokud mají argumenty *Odkaz1* a *Odkaz2* rozdílný počet datových bodů, vrátí funkce chybovou hodnotu **!!chNe**.

Intercept(ŘadaY;ŘadaX)

Vypočte souřadnice bodu, ve kterém čára protne osu y, pomocí existujících hodnot na osách x a y.

Intercept je bod, který je určen proložení nejlepší regresní čáry známými hodnotami na osách x a y.

Intercept se používá v případě, že chcete znát hodnotu závislé proměnné při nezávislé proměnné rovné 0 (nula). Funkci je například možné použít k předpovědi elektrického odporu kovu při teplotě 0°C z měření provedených při pokojové teplotě a vyšších teplotách.

ŘadaY je soubor závislých hodnot

ŘadaX je soubor nezávislých hodnot

Poznámky:

- Jako argumenty mohou být použita čísla, jména, matice nebo odkazy obsahující čísla.
- Pokud matice nebo odkazy použité jako argumenty obsahují text, logické hodnoty nebo prázdné buňky, budou tyto hodnoty ignorovány. Buňky s hodnotou nula budou zahrnuty.
- Pokud *ŘadaY* a *ŘadaX* obsahují různý počet bodů nebo žádné body, vrátí funkce chybovou hodnotu **!!chNe**.

Příklad: $\text{INTERCEPT}(\{2; 3; 9; 1; 8\}; \{6; 5; 11; 7; 5\}) = 0,0483871$

Kurt(Seznam)

Vrátí hodnotu špičatosti množiny dat. Špičatost určuje relativní strmou nebo plochost rozdělení v porovnání s normálním rozdělením. Kladná špičatost znamená, že rozdělení je poměrně strmé. Záporná špičatost znamená, že rozdělení je poměrně ploché.

Seznam je 1 až 30 argumentů, pro které chcete zjistit špičatost. Místo argumentů oddělených čárkami můžete použít jedinou matici nebo odkaz na matici.

Poznámky:

- Argumenty musí být čísla, případně názvy, matice nebo odkazy, které obsahují čísla.
- Pokud argument ve tvaru matice nebo odkazu obsahuje text, logické hodnoty nebo prázdné buňky, tyto hodnoty se ignorují. Buňky s nulovou hodnotou se však započítají.
- Pokud existují méně než čtyři datové body nebo pokud se směrodatná odchylka vzorku rovná nule, vrátí funkce chybovou hodnotu **!!ch/0**.

Příklad: $\text{KURT}(3;4;5;2;3;4;5;6;4;7) = - 0,1518$

Large(Odkaz;Pozice)

Vrátí *k*-tou největší hodnoty ze zadané množiny dat (pro *k* = *Pozice*). Tuto funkci lze použít k výběru hodnoty podle jejího relativního umístění. Funkci můžete použít například ke zjištění nejvyššího skóre, druhého nejvyššího skóre a třetího nejvyššího skóre.

Odkaz je matice nebo oblast dat, pro kterou chcete určit *k*-tou největší hodnotu

Pozice je pozice (od největší) v matici nebo oblasti dat, kterou chcete zjistit

Poznámky:

- Pokud je *Odkaz* prázdný, vrátí funkce chybovou hodnotu **!!chCis**.

- Pokud argument *Pozice* menší i roven 0 nebo pokud argument *Pozice* je větší než počet datových bodů, vrátí funkce chybovou hodnotu **!!chCis**.

Pokud *n* je počet datových bodů v zadané oblasti, pak **LARGE(Odkaz;1)** vrátí největší hodnotu a **LARGE(Odkaz;n)** vrátí nejmenší hodnotu.

Příklady:

LARGE({3;4;5;2;3;4;5;6;4;7};3) = 5

LARGE({3;4;5;2;3;4;5;6;4;7};7) = 4

Loginv(Pravděpodobnost;Střed;Odchylka)

Vrátí inverzní funkci k distribuční funkci logaritmicko-normálního rozdělení hodnot *x*, kde $\ln(x)$ má normální rozdělení s parametry *Střed* a *Odchylka*. Je-li $p = \text{LOGNORMDIST}(x; \dots)$, potom $\text{LOGINV}(p; \dots) = x$.

Distribuční funkci logaritmicko-normálního rozdělení lze použít k analýze logaritmicky transformovaných dat.

Pravděpodobnost je pravděpodobnost odpovídající logaritmicko-normálnímu rozdělení
Střed je střední hodnota $\ln(x)$.
Odchylka je směrodatná odchylka $\ln(x)$.

Poznámky:

- Pokud některý argument není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Je-li argument *Pravděpodobnost* < 0 nebo > 1, vrátí funkce chybovou hodnotu **!!chCis**.
- Je-li *Odchylka* < 0, vrátí funkce chybovou hodnotu **!!chCis**.

Příklad:

LOGINV(0,039084; 3,5; 1,2) = 4,000014

Lognormdist(x;Střed;Odchylka)

Vrátí součtové logaritmicko-normální rozdělení pro *x*, kde $\ln(x)$ má normální rozdělení s parametry průměr a *sm_odchylka*. Tato funkce se používá k analýze dat, která byla transformována pomocí logaritmické funkce.

x je hodnota, pro kterou chcete zjistit hodnotu rozdělení.
Střed je střední hodnota funkce $\ln(x)$.
Odchylka je směrodatná odchylka $\ln(x)$.

Poznámky:

- Pokud některý z argumentů není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Pokud je *x* je menší či rovno 0 nebo pokud *Odchylka* je menší či rovna 0, vrátí funkce chybovou hodnotu **!!chCis**.

Příklad:

LOGNORMDIST(4;3,5;1,2) = 0,039084

Max(Seznam)

Vrátí maximální hodnotu v seznamu argumentů. Argument *Seznam* musí být seznamem jedné nebo více hodnot (čísel nebo odkazů), oddělených čárkami.

Příklady: **MAX(1;2;3;1;8) = 8**

MAX(A1:A10) = 15,

pokud 15 je maximální číslo z oblasti **A1:A10**.

MaxA(Seznam)

Vrátí maximální hodnotu v seznamu argumentů. Text a logické hodnoty TRUE a FALSE jsou srovnávány stejně jako čísla. Funkce je podobná funkci **MINA**.

Seznam je 1 až 30 hodnot, pro které chcete zjistit maximální hodnotu

Poznámky:

- Můžete zadat argumenty, kterými mohou být čísla, prázdné buňky, logické hodnoty nebo text představující čísla. Argumenty, které jsou chybové hodnoty, způsobují chyby. Jestliže výpočet nesmí zahrnovat textové ani logické hodnoty, použijte funkci listu **MAX**.
- Jestliže je argument matice nebo odkaz, budou použity hodnoty pouze z této matice nebo odkazu. Prázdné buňky a textové hodnoty budou v matici nebo odkazu přeskočeny.
- Argumenty obsahující logickou hodnotu TRUE jsou vyhodnoceny jako 1, argumenty obsahující text nebo logickou hodnotu FALSE jsou vyhodnoceny jako 0 (nula).
- Jestliže argumenty neobsahují žádné hodnoty, vrátí funkce hodnotu 0 (nula).

Příklady:

Oblast A1:A5 obsahuje čísla 10, 7, 9, 27 a 2:

MAXA(A1:A5) = 27

MAXA(A1:A5;30) = 30

Oblast A1:A5 obsahuje hodnoty 0; 0,2; 0,5; 0,4 a logickou hodnotu TRUE:

MAXA(A1:A5) = 1

Median(Seznam)

Vrátí medián zadaných čísel. Medián je číslo, které leží uprostřed podle velikosti uspořádaného souboru čísel. Polovina čísel má tedy hodnotu, která je větší nebo rovna mediánu a polovina čísel má hodnotu, která je menší nebo rovna mediánu.

Seznam je 1 až 30 čísel, z nichž má být vypočten medián

- Jednotlivými argumenty musí být čísla nebo názvy, matice nebo odkazy, které obsahují čísla.
- Pokud matice nebo odkaz v argumentu obsahují text, logické hodnoty nebo prázdné buňky, budou tyto hodnoty ignorovány. Buňky s hodnotou nula jsou do výpočtu zahrnuty.

Poznámky:

Pokud je v souboru sudý počet hodnot, vypočítá funkce průměr ze dvou prostředních hodnot (druhý příklad).

Příklady:

MEDIAN(1; 2; 3; 4; 5) = 3

MEDIAN(1; 2; 3; 4; 5; 6) = 3,5, průměr z 3 a 4

Min(Seznam)

Vrátí minimální hodnotu v seznamu argumentů. Jako argument *Seznam* se uvede seznam jedné nebo více hodnot (čísel nebo odkazů), oddělených čárkami.

Příklady:MIN(1;5;3;67;3) = 1

MIN(A1:H10) = 5,

pokud je 5 minimální číslo z oblasti **A1:H10**

MinA(Seznam)

Vrátí minimální hodnotu v seznamu argumentů. Text a logické hodnoty TRUE a FALSE jsou srovnávány stejně jako čísla.

Seznam je 1 až 30 hodnot, pro které chcete zjistit minimální hodnotu

- Můžete zadat argumenty, kterými mohou být čísla, prázdné buňky, logické hodnoty nebo text představující čísla. Argumenty, které jsou chybové hodnoty, způsobují chyby. Jestliže výpočet nesmí zahrnovat textové ani logické hodnoty, použijte funkci listu **MIN**.
- Jestliže je argument matice nebo odkaz, budou použity hodnoty pouze z této matice nebo odkazu. Prázdné buňky a textové hodnoty budou v matici nebo odkazu přeskočeny.
- Argumenty obsahující logickou hodnotu TRUE jsou vyhodnoceny jako 1, argumenty obsahující text nebo logickou hodnotu FALSE jsou vyhodnoceny jako 0 (nula).
- Jestliže argumenty neobsahují žádné hodnoty, vrátí funkce **MINA** hodnotu 0.

Příklady:

Oblast A1:A5 obsahuje čísla 10, 7, 9, 27 a 2:

$$\text{MINA}(A1:A5) = 2$$

$$\text{MINA}(A1:A5;0) = 0$$

Oblast A1:A5 obsahuje hodnoty FALSE; 0,2; 0,5; 0,4 a 0,8:

$$\text{MINA}(A1:A5) = 0$$

Funkce **MINA** je podobná funkci **MAXA**.

Mode(Seznam)

Vrátí **modus** - nejčastěji se vyskytující nebo opakující se hodnotu v poli (matici) nebo oblasti dat. Podobně jako funkce **MEDIAN** je funkce **MODE** zjišťující funkcí.

Seznam je 1 až 30 argumentů, jejichž modus chcete spočítat. Můžete také použít jedinou matici nebo odkaz na matici místo argumentů oddělených středníky.

Poznámky:

- Jako argumenty lze použít čísla, názvy, matice nebo odkazy, které obsahují čísla.
- Pokud matice nebo odkaz obsahuje text, logické hodnoty nebo prázdné buňky, budou tyto hodnoty ignorovány. Buňky obsahující hodnotu 0 budou zahrnuty.
- Pokud soubor dat neobsahuje žádné duplicitní hodnoty, vrátí funkce chybovou hodnotu **!!chNe**.

Modus je nejčastěji se vyskytující hodnota v množině hodnot.

Medián je prostřední hodnota.

Průměr je průměrná hodnota.

Žádné jednotlivé měření nějaké střední hodnoty neposkytuje úplný obraz o datech. Předpokládejme, že polovina dat je soustředěna kolem jedné nízké hodnoty a druhá polovina kolem dvou vysokých hodnot. Funkce **AVERAGE** i **MEDIAN** vrátí asi hodnoty z relativně prázdného středu a funkce **MODE** vrátí dominantní nízkou hodnotu.

Příklad: $\text{MODE}(\{5,6; 4; 4; 3; 2; 4\}) = 4$

Negbinomdist(PočÚspěch;PočNeúspěch;Pravděpodobnost)

Vrátí negativně binomické rozdělení. Funkce **Negbinomdist** vrátí pravděpodobnost, že dojde k *PočÚspěch* neúspěchům, dříve než dojde k úspěchu, jehož pořadí je *PočNeúspěch*, kde konstantní pravděpodobnost úspěchu je *Pravděpodobnost*. Tato funkce se podobá binomickému rozdělení, pouze s tím rozdílem, že počet úspěchů je pevný a počet pokusů proměnný. Jako u binomického rozdělení se jednotlivé pokusy považují za nezávislé.

Potřebujete například najít 10 osob s výjimečnými reflexy, přičemž víte, že pravděpodobnost, že má kandidát tyto vlastnosti, je 0,3. Pomocí funkce **Negbinomdist** lze vypočítat pravděpodobný počet kandidátů, kteří musí být pozváni, aby bylo nalezeno potřebných 10 lidí.

PočÚspěch je počet neúspěšných pokusů.

PočNeúspěch je prahová hodnota počtu úspěchů.

Pravděpodobnost je pravděpodobnost úspěchu.

Poznámky:

- Argumenty *PočÚspěch* a *PočNeúspěch* jsou zkráceny na celá čísla.
- Pokud některý z argumentů není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Pokud *Pravděpodobnost* < 0 nebo pokud *Pravděpodobnost* > 1, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud $(\text{PočÚspěch} + \text{PočNeúspěch} - 1)$ menší nebo rovno 0, vrátí funkce chybovou hodnotu **!!chCis**.

Příklad:

$$\text{NEGBINOMDIST}(10;5;0,25) = 0,055049$$

Normdist(x;Střed;Odchylka;Par)

Vrátí součtové normální rozdělení se zadanou střední hodnotou a směrodatnou odchylkou. Tato funkce má ve statistice velmi široké použití, včetně testování hypotéz.

<i>x</i>	je hodnota, pro kterou chcete zjistit hodnotu rozdělení.
<i>Střed</i>	je aritmetický průměr rozdělení.
<i>Odchylka</i>	je směrodatná odchylka rozdělení.
<i>Par</i>	je logická hodnota, která určuje tvar funkce. Pokud má součet hodnotu TRUE, vrátí funkce NORMDIST součtovou distribuční funkci; má-li hodnotu FALSE, vrátí pravděpodobnostní míru.

Poznámky:

- Pokud argument *Střed* nebo *Odchylka* není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Pokud je parametr *Odchylka* menší nebo roven 0, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud $Střed = 0$ a $Odchylka = 1$, vrátí funkce standardní normální rozdělení.

Příklad:

$$\text{NORMDIST}(42;40;1,5;\text{TRUE}) = 0,908789$$

Norminv(Pravděpodobnost;Střed;Odchylka)

Vrátí inverzní funkci k součtovému normálnímu rozdělení pro zadanou střední hodnotu a směrodatnou odchylku.

<i>Pravděpodobnost</i>	je pravděpodobnost odpovídající normálnímu rozdělení
<i>Střed</i>	je aritmetický průměr rozdělení
<i>Odchylka</i>	je směrodatná odchylka rozdělení

Poznámky:

- Pokud některý z argumentů není číselného typu, funkce vrátí chybovou hodnotu **!!chHod**.
- Pokud $Pravděpodobnost < 0$ nebo pokud $Pravděpodobnost > 1$, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud je *Odchylka* menší nebo rovna 0, vrátí funkce chybovou hodnotu **!!chCis**.

Funkce **Norminv** používá standardní normální rozdělení, pokud $střední = 0$ a $sm_odch = 1$ (viz též nápověda k funkci [Normsinv](#)).

Funkce **Norminv** se počítá pomocí iterací. Při dané pravděpodobnosti provádí funkce iterace, dokud výsledek nedosáhne přesnosti $\pm 3 \times 10^{-7}$. Pokud funkce **Norminv** nekonverguje ani po 100 iteracích, vrátí chybovou hodnotu **!!chNe**.

Příklad:

$$\text{NORMINV}(0,908789;40;1,5) = 42$$

Normsdist(x;Střed;Odchylka;Par)

Vrátí součtovou distribuční funkci standardního normálního rozdělení. Toto rozdělení má střední hodnotu nula a směrodatnou odchylku jedna. Tuto funkci lze použít místo tabulky pro oblasti pod křivkou standardního normálního rozdělení.

<i>x</i>	je hodnota, pro kterou chcete vypočítat rozdělení
<i>Střed</i>	je aritmetický průměr rozdělení
<i>Odchylka</i>	je směrodatná odchylka rozdělení
<i>Par</i>	je logická hodnota, která určuje tvar funkce. Pokud má součet hodnotu TRUE, vrátí funkce součtovou distribuční funkci; má-li hodnotu FALSE, vrátí pravděpodobnostní míru.

Poznámka:

Pokud *x* není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.

Příklad:

$$\text{NORMSDIST}(1,333333) = 0,908789$$

Normsinv(Pravděpodobnost;Střed;Odchylka)

Vrátí inverzní funkci k součtovému standardnímu normálnímu rozdělení. Toto rozdělení má střední hodnotu nula a směrodatnou odchylku jedna.

Pravděpodobnost je pravděpodobnost odpovídající normálnímu rozdělení
Střed je aritmetický průměr rozdělení
Odchylka je směrodatná odchylka rozdělení

Poznámky:

- Pokud argument *Pravděpodobnost* není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Pokud *Pravděpodobnost* < 0 nebo pokud *Pravděpodobnost* > 1, vrátí funkce chybovou hodnotu **!!chCis**.

Funkce **Normsinv** se počítá pomocí iterací. Při dané pravděpodobnosti provádí funkce iterace, dokud výsledek nedosáhne přesnosti $\pm 3 \times 10^{-7}$. Pokud funkce **Normsinv** nekonverguje ani po 100 iteracích, vrátí chybovou hodnotu **!!chNe**.

Příklad:

NORMSINV(0,908789) = 1,3333

Pearson(Odkaz1;Odkaz2)

Vrátí Pearsonův koeficient korelace r. Hodnota r leží mezi -1.0 pro úplnou zápornou korelaci a +1.0 pro úplnou kladnou korelaci a vyjadřuje lineární vztah mezi dvěma množinami dat.

Odkaz1 je soubor nezávislých hodnot
Odkaz2 je soubor závislých hodnot

Poznámky:

- Argumenty musí být čísla nebo názvy, maticové konstanty nebo odkazy, které obsahují čísla.
- Jestliže matice nebo argument odkazu obsahuje text, logické hodnoty nebo prázdné buňky, jsou tyto hodnoty ignorovány; buňky s nulovou hodnotou jsou však započítávány.
- Pokud jsou *Odkaz1* a *Odkaz2* prázdná nebo mají odlišný počet datových bodů, vrátí funkce chybovou hodnotu **!!chNe**.

Příklad: PEARSON({9;7;5;3;1};{10;6;1;5;3}) = 0,699379

Percentile(Odkaz;k)

Vrátí hodnotu, která odpovídá k-tému percentilu v oblasti hodnot. Tuto funkci můžete použít ke stanovení prahové hodnoty. Můžete například prozkoumat kandidáty, kteří dosáhli alespoň 90tého percentilu.

Odkaz je matice nebo oblast dat, která určuje relativní umístění
k je hodnota percentilu z uzavřeného intervalu 0 až 1

Poznámky:

- Pokud je matice prázdná nebo obsahuje více než 8191 datových bodů, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud argument *k* není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Pokud je $k < 0$ nebo $k > 1$, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud není *k* násobkem $1/(n - 1)$, určuje interpolací hodnotu ke k-tému percentilu.

Příklad: PERCENTILE({1;2;3;4};0,3) = 1,9

Percentrank(Odkaz;x;Desetin)

Vrátí pořadí hodnoty v množině dat vyjádřené procentuální částí množiny dat. Tuto funkci můžete použít k určení relativního umístění pozorování v množině dat. Funkci můžete například použít k určení pořadí výsledků zkoušky způsobilosti v množině všech výsledků testu.

Odkaz je matice nebo oblast dat s numerickými hodnotami, které určují relativní postavení
x je hodnota, u které chcete znát umístění

Desetin je nastavitelný počet desetinných míst, která se budou pro procentuální hodnoty uvažovat. Pokud tuto hodnotu nezadáte, použije funkce tři desetinná místa (0,xxx%).

Poznámky:

- Pokud je pole prázdné, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud je argument *Desetin* < 1, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud argument *x* neodpovídá žádné hodnotě v poli, funkce po interpolaci vrátí správné procentuální pořadí.

Příklad: PERCENTRANK({1;2;3;4;5;6;7;8;9;10};4) = 0,333

Permut(Číslo1;Číslo2)

Funkce vrací počet permutací daného čísla nebo objektu, který lze převést na objekt numerického typu. Permutace je jakákoliv sada objektů nebo událostí, pro které má význam vnitřní pořadí prvků. Permutace nesmí být zaměňovány s kombinacemi, pro které vnitřní pořadí významné není. Funkce může být například využita pro kalkulace pravděpodobností v lotériích.

Číslo1 celočíselná hodnota > 0, udávající počet objektů
Číslo2 celočíselná hodnota >= 0, udávající počet objektů v každé permutaci.

Oba argumenty se zaokrouhlují na celočíselné hodnoty. Zadáte-li nenumernický argument, funkce vrací chybovou hodnotu **#VALUE!**. Zadáte-li *Číslo1* menší či rovno nule nebo *Číslo2* menší než nula nebo *Číslo1* menší než *Číslo2*, funkce vrací chybovou hodnotu **!!chCis**.

Permutace se počítá podle vztahu

$$P_{k,n} = n! / (n - k)!$$

Příklad:

Předpokládejme, že budete chtít vědět, kolika způsoby lze sestavit čísla v loterii o pěti číslech, kde čísla mohou být mezi 0 a 49.

Funkce =PERMUT(50;5) dává výsledek **254 251 200**

Poisson(x;Střed;Par)

Vrátí hodnotu distribuční funkce Poissonova rozdělení. Poissonovo rozdělení se obvykle používá k určení předpokládaného počtu případů za jednotku času, jako je počet automobilů přijíždějících na celnici za jednu minutu.

x je počet případů.
Střed je předpokládaná číselná hodnota.
Par je logická hodnota, která určuje tup distribuční funkce. Pokud nabývá součet hodnotu TRUE, vrátí funkce **Poisson** hodnotu funkce Poissonova rozdělení tak, že počet náhodných případů bude v intervalu nula až *x*; pokud FALSE, vrátí hodnotu funkce tak, že počet náhodných případů bude právě *x*.

Poznámky:

- Pokud argument *x* není celé číslo, je zkrácen.
- Pokud argument *x* nebo střední není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Pokud je *x* menší nebo rovno 0, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud je *Střed* menší nebo rovno 0, vrátí funkce chybovou hodnotu **!!chCis**.

Příklad:

POISSON(2;5;FALSE) = 0,084224
 POISSON(2;5;TRUE) = 0,124652

Prob(Hodnoty;Pravděpodobnosti;DolníLimit;HorníLimit)

Vrátí pravděpodobnost, že hodnoty dané oblasti budou mezi dvěma limity. Pokud není zadán *HorníLimit*, vrátí pravděpodobnost, že hodnoty v oblasti *Hodnoty* se rovnají *DolníLimit*.

Hodnoty je oblast číselných hodnot *x*, s kterými je spojena pravděpodobnost

<i>Pravděpodobnosti</i>	je oblast obsahující seznam pravděpodobností příslušejících k jednotlivým hodnotám v oblasti x
<i>DolníLimit</i>	je volitelná dolní mez rozmezí hodnot, u kterých chcete zjistit pravděpodobnost
<i>HorníLimit</i>	je volitelná horní mez rozmezí hodnot, u kterých chcete zjistit pravděpodobnost

Poznámky:

- Pokud je některá hodnota argumentu *Pravděpodobnosti* ≤ 0 nebo pokud je některá hodnota argumentu *Pravděpodobnosti* > 1 , vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud je součet hodnot argumentu *Pravděpodobnosti* jiný než 1, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud není zadán *HorníLimit*, vrátí funkce pravděpodobnost rovnosti s argumentem *DolníLimit*.
- Pokud *Hodnota a Pravděpodobnosti* obsahují rozdílný počet datových bodů, vrátí funkce chybovou hodnotu **!!chNe**.

Příklady:

$$\text{PROB}(\{0;1;2;3\};\{0,2;0,3;0,1;0,4\};2) = 0,1$$

$$\text{PROB}(\{0;1;2;3\};\{0,2;0,3;0,1;0,4\};1,3) = 0,8$$

Rank(Číslo;Odkaz;Setřídění)

Vrátí pořadí argumentu (podle velikosti) v seznamu čísel. Hodnota pořadí je svou velikostí úměrná jiným hodnotám v seznamu. (Pokud by bylo potřeba seřadit seznam, bude pořadí čísla podle velikosti zároveň jeho pozicí.)

<i>Číslo</i>	je číslo, jehož pořadí hledáte
<i>Odkaz</i>	je matice nebo odkaz na seznam čísel; nečíselné hodnoty jsou ignorovány
<i>Setřídění</i>	je číslo určující, zda se budou hodnoty třídit vzestupně či sestupně

- Pokud je *Setřídění* rovno 0 nebo není zadáno, určuje se pořadí čísla jako v sestupném seznamu.
- Pokud je *Setřídění* jakákoliv nenulová hodnota, určuje se v pořadí čísla jako ve vzestupném seznamu.

Poznámka:

Funkce **RANK** přiděluje stejným číslům stejné pořadí. Přítomnost stejných čísel však ovlivňuje pořadí po sobě následujících čísel. Pokud se v seznamu celých čísel například číslo 10 objeví dvakrát a má pořadí 5, potom číslo 11 má pořadí 7 (žádné číslo nemá pořadí 6).

Příklady:

Pokud buňky A1:A5 obsahují postupně čísla 7; 3,5; 3,5; 1 a 2, pak:

$$\text{RANK}(A2;A1:A5;1) = 3$$

$$\text{RANK}(A1;A1:A5;1) = 5$$

Rsq(ŘadaY;ŘadaX)

Vrátí druhou mocninu Pearsonova korelačního koeficientu pro lineární regresi datových bodů v oblastech *ŘadaY* a *ŘadaX*. Další informace získáte v nápovědě k funkci **PEARSON**. Hodnota r na druhou může být interpretována jako poměr rozptylu v y k rozptylu v x .

<i>ŘadaY</i>	je matice nebo oblast datových bodů
<i>ŘadaX</i>	je matice nebo oblast datových bodů

Poznámky:

- Argumenty musí být čísla nebo názvy, matice nebo odkazy obsahující čísla.
- Jestliže matice nebo odkaz obsahuje text, logické hodnoty nebo prázdné buňky, jsou tyto hodnoty ignorovány; buňky s nulovou hodnotou jsou však započítávány.
- Pokud jsou *ŘadaY* a *ŘadaX* prázdná nebo mají rozdílný počet datových bodů, vrátí funkce **RSQ** chybovou hodnotu **!!chNe**.

Příklad: $\text{RSQ}(\{2;3;9;1;8;7;5\};\{6;5;11;7;5;4;4\}) = 0,05795$

Skew(Seznam)

Vrátí šikmost rozdělení náhodné veličiny. Šikmost označuje stupeň asymetričnosti rozdělení veličiny kolem střední hodnoty. Kladné zešikmení označuje rozdělení s asymetrickou stranou, která se vychyluje směrem k více kladným hodnotám. Záporné zešikmení označuje rozdělení s asymetrickou stranou, která se vychyluje směrem k více záporným hodnotám.

Seznam je 1 až 30 argumentů, pro které chcete spočítat šikmost. Místo argumentů oddělených čárkou můžete také použít jednoduchou matici nebo odkaz na matici.

Poznámky:

- Argumenty musí být čísla nebo názvy, pole nebo odkazy obsahující čísla.
- Jestliže matice nebo odkaz obsahuje text, logické hodnoty nebo prázdné buňky, jsou tyto hodnoty ignorovány; buňky s nulovou hodnotou jsou však započítávány.
- Pokud existují méně než tři datové body nebo pokud je výběrová směrodatná odchylka rovna nule, vrátí funkce chybovou hodnotu **!!ch/0**.

Příklad: **SKEW(3;4;5;2;3;4;5;6;4;7) = 0,359543**

Slope(ŘadaY;ŘadaX)

Vrátí směrnici regresní přímky proložené zadanými body v oblastech *ŘadaY* a *ŘadaX*. Směrnice je vertikální vzdálenost dělená horizontální vzdáleností mezi dvěma body na přímce a vyjadřuje rychlost změny podél regresní přímky.

ŘadaY je matice nebo oblast buněk s číselnými závislými hodnotami y
ŘadaX je množina nezávislých hodnot x

Poznámky:

- Argumenty musí být čísla nebo názvy, matice nebo odkazy na matice obsahující čísla.
- Jestliže matice nebo argument odkazu obsahuje text, logické hodnoty nebo prázdné buňky, jsou tyto hodnoty ignorovány; buňky s nulovou hodnotou jsou však započítávány.
- Pokud jsou *ŘadaY* a *ŘadaX* prázdná nebo mají rozdílný počet datových bodů, vrátí funkce chybovou hodnotu **!!chNe**.

Příklad: **SLOPE({2;3;9;1;8;7;5};{6;5;11;7;5;4;4}) = 0,305556**

Small(Odkaz;Pozice)

Vrátí k-tou nejmenší hodnotu v množině dat. Tato funkce se používá k určení hodnoty, která má v množině dat konkrétní relativní umístění.

Odkaz je matice nebo oblast číselných dat, pro které chcete určit k-tou nejmenší hodnotu
Pozice je pořadí od nejmenšího čísla

Poznámky:

- Pokud je pole prázdné, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud je k menší nebo rovno 0 nebo pokud je k větší než počet datových bodů, vrátí funkce chybovou hodnotu **!!chCis**.

Příklad:

SMALL({3;4;5;2;3;4;5;6;4;7};4) = 4

SMALL({1;4;8;3;7;12;54;8;23};2) = 3

Standardize(x;Střed;Odchylka)

Vrátí normalizovanou hodnotu s normálním rozdělením, které je určeno parametry střední hodnota a *sm_odchylka*.

x je hodnota, kterou chcete normalizovat.
Střed je aritmetická střední hodnota.
Odchylka je směrodatná odchylka distribučního rozdělení.

Poznámka:

Pokud je *Odchylka* menší nebo rovna 0, vrátí funkce **Standardize** chybovou hodnotu **!!chCis**.

Příklad:

STANDARDIZE(42;40;1,5) = 1,333333

Stdev(Seznam)

Funkce počítá standardní odchylku, tj. míru přiléhavosti souboru hodnot ke střední hodnotě vzorku souboru údajů uloženého v oblasti či několika oblastech listu sešitu.

Příklad: STDEV(A1:A5)

Pokud jsou v buňkách **A1** až **A5** hodnoty 1500, 1525, 1490, 1510 a 1499, vrátí funkce hodnotu 13,3304...

StDevA(Seznam)

Vypočte směrodatnou odchylku výběru. Směrodatná odchylka je míra rozptýlení hodnot od průměrné (střední) hodnoty. Do výpočtu je zahrnut i text a logické hodnoty TRUE a FALSE.

Seznam je 1 až 30 hodnot odpovídajících výběru ze základního souboru. Místo argumentů oddělených středníky můžete použít jednu matici nebo odkaz na matici.

Poznámky:

- Při použití funkce se předpokládá, že její argumenty představují výběr ze základního souboru. Jestliže data představují celý základní soubor, musíte směrodatnou odchylku vypočítat pomocí funkce **STDEVPA**.
- Argumenty obsahující logickou hodnotu TRUE jsou vyhodnoceny jako 1, argumenty obsahující text nebo logickou hodnotu FALSE jsou vyhodnoceny jako 0 (nula). Jestliže výpočet nesmí zahrnovat textové ani logické hodnoty, použijte funkci listu **STDEV**.
- Směrodatná odchylka se vypočítá "netendenční" metodou neboli metodou "n-1".

Příklad:

Předpokládejme, že 10 nástrojů vylisovaných stejným strojem během jednoho výrobního procesu je vybráno jako náhodný výběr a je změřena jejich mez pevnosti. Hodnoty výběru (1 345, 1 301, 1 368, 1 322, 1 310, 1 370, 1 318, 1 350, 1 303, 1 299) jsou uloženy v oblasti A2:E3. Funkce **STDEV** odhadne směrodatnou odchylku meze pevnosti všech nástrojů. Pak:

STDEV(A2:E3) = 27,46

Stdevp(Seznam)

Funkce počítá standardní odchylku, tj. míru přiléhavosti souboru hodnot ke střední hodnotě celého souboru údajů uloženého v oblasti či několika oblastech listu sešitu.

Příklad: STDEVP(A1:A5)

Pokud jsou v buňkách **A1** až **A5** hodnoty 1500, 1525, 1490, 1510 a 1499, vrátí funkce hodnotu 11,923...

StDevPA(Seznam)

Vypočte směrodatnou odchylku základního souboru zadaného jako argumenty. Směrodatná odchylka je míra rozptýlení hodnot od průměrné (střední) hodnoty.

Seznam je 1 až 30 hodnot odpovídajících základnímu souboru. Místo argumentů oddělených středníky můžete použít jednu matici nebo odkaz na matici.

Poznámky:

- Při použití funkce se předpokládá, že její argumenty představují celý základní soubor. Jestliže data představují výběr ze základního souboru, musíte směrodatnou odchylku vypočítat pomocí funkce **STDEVA**.

- Argumenty obsahující logickou hodnotu TRUE jsou vyhodnoceny jako 1, argumenty obsahující text nebo logickou hodnotu FALSE jsou vyhodnoceny jako 0 (nula). Jestliže výpočet nesmí zahrnovat textové ani logické hodnoty, použijte funkci listu **STDEVP**.
- U výběrů velké velikosti vrátí funkce **STDEVA** a **STDEVPA** přibližně stejné hodnoty.
- Směrodatná odchylka se vypočítá "tendenční" metodou neboli metodou "n".

Příklad:

Pomocí stejných dat z příkladu funkce **STDEVA** a za předpokladu, že produktem jednoho výrobního procesu bude pouze 10 nástrojů, vypočte funkce **STDEVP** směrodatnou odchylku mezi pevnosti pro všechny nástroje.

$$\text{STDEVP}(A2:E3) = 26,05$$

Steyx(ŘadaY;ŘadaX)

Vrátí standardní chybu při výpočtu lineární regrese. Standardní chyba je určena množstvím chyb při odhadu y pro jednotlivá x.

ŘadaY je matice nebo oblast závislých hodnot

ŘadaX je matice nebo oblast nezávislých hodnot

Poznámky:

- Argumenty musí být čísla nebo názvy, matice nebo odkazy obsahující čísla.
- Jestliže matice nebo odkaz obsahuje text, logické hodnoty nebo prázdné buňky, jsou tyto hodnoty ignorovány; buňky s nulovou hodnotou jsou však započítávány.
- Pokud jsou *ŘadaY* a *ŘadaX* prázdná nebo mají rozdílný počet datových bodů, vrátí funkce chybovou hodnotu **!!chNe**.

Příklad: **STEYX({2;3;9;1;8;7;5};{6;5;11;7;5;4;4}) = 3,305719**

Tdist(x;StVolnosti;Typ)

Vrátí hodnotu distribuční funkce Studentova t-rozdělení. T-rozdělení je používáno při hypotetickém testování malých vzorků dat. Tato funkce zastupuje tabulku kritických hodnot t-rozdělení.

x je číslo, pro které hledáte hodnotu distribuční funkce

StVolnosti je celé číslo, označující počet stupňů volnosti

Typ určuje, zda se jedná o jednostranné či dvoustranné rozdělení. Pokud argument *Typ* = 1, vrátí funkce hodnotu funkce jednostranného rozdělení. Pokud argument *Typ* = 2, vrátí funkce hodnotu funkce dvojstranného rozdělení.

Poznámky:

- Pokud některý argument není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Pokud *StVolnosti* < 1, vrátí funkce chybovou hodnotu **!!chCis**.
- Argumenty *StVolnosti* jsou zkráceny na celá čísla.
- Pokud argument *Typ* nabývá jiných hodnot než 1 nebo 2, vrátí funkce chybovou hodnotu **!!chCis**.
- Funkce se počítá jako **TDIST = p(x < X)**, kde x je náhodná proměnná, která doprovází t-rozdělení.

Příklad:

$$\text{TDIST}(1,96;60;2) = 0,054645$$

Tinv(Pravděpodobnost;StVolnosti)

Vrátí inverzní funkci k distribuční funkci Studentova t-rozdělení pro dané stupně volnosti.

Pravděpodobnost je pravděpodobnost daného dvojstranného t-rozdělení

StVolnosti je počet stupňů volnosti

Poznámky:

- Pokud některý z argumentů není číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Pokud *Pravděpodobnost* < 0 nebo pokud *Pravděpodobnost* > 1, vrátí funkce chybovou hodnotu **!!chCis**.

- Pokud není argument *StVolnosti* celé číslo, je zkrácen.
- Pokud *StVolnosti* < 1, vrátí funkce chybovou hodnotu **!!chCis**.
- Funkce se počítá jako $TINV = p(t < X)$, kde x je náhodná proměnná, která doprovází t-rozdělení.

Funkce **Tinv** používá techniku opakovaného propočítávání funkce. Se zadanou pravděpodobnostní hodnotou opakuje funkce **Tinv** výpočet, dokud není výsledek přesný na $\pm 3 \times 10^{-7}$. Pokud funkce nedosáhne požadovaného výsledku po 100 opakováních, vrátí chybovou hodnotu **!!chNe**.

Příklad:

$$TINV(0,054645;60) = 1,96$$

Trimmean(Odkaz;Procento)

Vrátí průměrnou hodnotu vnitřní části množiny datových hodnot. Funkce určuje střední hodnotu tak, že odřízne daný počet dat z nejvyšších a nejnižších hodnot souboru. Tato funkce se používá v případě, že chcete z analýzy vyloučit vzdálená data.

- Odkaz* je matice nebo oblast hodnot, které chcete oříznout a pro kterou chcete vypočítat průměr
- Procento* je zlomek udávající počet odříznutých dat z nejvyšších a nejnižších hodnot. Jestliže je například *Procento* = 0,2, jsou z množiny 20 hodnot odříznuty 4 hodnoty (20 x 0,2), 2 z nejvyšších a 2 z nejnižších hodnot množiny.

Poznámky:

- Pokud *Procento* < 0 nebo *Procento* > 1, vrátí funkce chybovou hodnotu **!!chCis**.
- Funkce zaokrouhluje počet hodnot, které se mají odříznout, dolů na nejbližší násobek 2. Když *Procento* = 0,1, pak 10 procent z 30 dat odpovídá 3 hodnotám. Funkce však kvůli zachování symetrie odřízne pouze jednu hodnotu z nejvyšších a jednu z nejnižších hodnot množiny.

Příklad: $TRIMMEAN(\{4;5;6;7;2;3;4;5;1;2;3\};0,2) = 3,777778$

Var(Seznam)

Funkce počítá variace ze vzorku souboru hodnot uložených v oblasti nebo oblastech listu sešitu.

Příklad: VAR(A1:A5)

Pokud jsou v buňkách **A1** až **A5** hodnoty 1500, 1525, 1490, 1510 a 1499, vrátí funkce hodnotu 177,7

VarA(Seznam)

Vypočte rozptyl výběru. K číslům je navíc ve výpočtu zahrnut i text a logické hodnoty TRUE a FALSE.

Seznam je 1 až 30 hodnot argumentů odpovídajících výběru ze základního souboru.

Poznámky:

- Při použití funkce se předpokládá, že její argumenty představují výběr ze základního souboru. Jestliže data představují celý základní soubor, musíte rozptyl vypočítat pomocí funkce **VARPA**.
- Argumenty obsahující logickou hodnotu TRUE jsou vyhodnoceny jako 1, argumenty obsahující text nebo logickou hodnotu FALSE jsou vyhodnoceny jako 0 (nula). Jestliže výpočet nesmí zahrnovat textové ani logické hodnoty, použijte funkci listu **VAR**.

Příklad:

Předpokládejme, že 10 nástrojů vylisovaných stejným strojem během jednoho výrobního procesu je vybráno jako náhodný výběr a je změřena jejich mez pevnosti. Hodnoty výběru (1 345, 1 301, 1 368, 1 322, 1 310, 1 370, 1 318, 1 350, 1 303, 1 299) jsou uloženy v oblasti A2:E3. **VARA** odhadne rozptyl meze pevnosti nástrojů. Pak:

$$VARA(A2:E3) = 754,3$$

Varp(Seznam)

Funkce počítá variace z celého souboru hodnot uložených v oblasti nebo oblastech listu sešitu.

Příklad: VARP(A1:A5)

Pokud jsou v buňkách **A1** až **A5** hodnoty 1500, 1525, 1490, 1510 a 1499, vrátí funkce hodnotu 142,16.

VarPA(Seznam)

Vypočte rozptyl základního souboru. K číslům je navíc ve výpočtu zahrnut i text a logické hodnoty TRUE a FALSE.

Seznam je 1 až 30 hodnot argumentů odpovídajících základnímu souboru

Poznámky:

- Při použití funkce se předpokládá, že její argumenty představují celý základní soubor. Jestliže data představují výběr ze základního souboru, musíte rozptyl vypočítat pomocí funkce **VARA**.
- Argumenty obsahující logickou hodnotu TRUE jsou vyhodnoceny jako 1, argumenty obsahující text nebo logickou hodnotu FALSE jsou vyhodnoceny jako 0 (nula). Jestliže výpočet nesmí zahrnovat textové ani logické hodnoty, použijte funkci listu **VAR**.

Příklad:

Pomocí dat z příkladu funkce **VARA** a za předpokladu, že produktem jednoho výrobního procesu je pouze 10 nástrojů, vypočte funkce **VARPA** rozptyl mezi pevností všech nástrojů.

$$\text{VARPA}(A2:E3) = 678,8$$

Weibull(x;Alpha;Beta;Par)

Vrátí hodnotu distribuční funkce Weibullova rozdělení. Rozdělení se používá při analýze spolehlivosti, například k určení střední doby mezi poruchami.

<i>x</i>	je hodnota, pro kterou chcete zjistit hodnotu rozdělení
<i>Alfa</i>	je parametr rozdělení
<i>Beta</i>	je parametr rozdělení
<i>Par</i>	určuje druh použité funkce

Poznámky:

- Pokud argumenty *x*, *Alfa* nebo *Beta* nejsou číselného typu, vrátí funkce chybovou hodnotu **!!chHod**.
- Pokud $x < 0$, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud je argument *Alfa* menší nebo roven 0 nebo pokud argument *Beta* je menší nebo roven 0, vrátí funkce chybovou hodnotu **!!chCis**.
- Pokud $Alfa = 1$, vrátí funkce exponenciální rozdělení.

Příklady:

$$\text{WEIBULL}(105;20;100;TRUE) = 0,929581$$

$$\text{WEIBULL}(105;20;100;FALSE) = 0,035589$$

Ztest(Odkaz;x;Odchylka)

Vrátí dvoustrannou hodnotu P, která je výsledkem z-testu. Při z-testu se generují hodnoty pro argument *x* s ohledem na množinu dat a matici (pole) a je vrácena pravděpodobnost u normálního rozdělení. **Ztest** se používá pro určení pravděpodobnosti toho, že náhodný výběr pochází z jediného základního souboru.

Odkaz je matice nebo oblast dat, proti které je testována hodnota *x* je testovaná hodnota

Odchylka je směrodatná odchylka základního souboru či výběru. Pokud není zadána, je použita směrodatná odchylka vzorku.

Poznámka:

Pokud je pole prázdné, vrátí funkce **ZTEST** chybovou hodnotu **!!chNe**.

Příklad: **ZTEST**({3;6;7;8;6;5;4;2;1;9};4) = 0,090574

Databázové funkce

Databázové funkce umožňují analyzovat, zda hodnoty seznamu vyhovují určitým podmínkám nebo kritériím. Funkce databáze a správy seznamu listu začínající písmenem “D” mají tři argumenty:

- Argument *Databáze* je oblast, která obsahuje daný seznam. Do této oblasti musíte zahrnout řádek, který obsahuje popisky sloupce.
- Argument *Pole* je popis sloupce, který chcete sumarizovat.
- Argument *Kritéria* je oblast obsahující určenou podmínku.

Daverage(Databáze;Pole;Kritéria)

Vrátí průměr hodnot ve sloupci seznamu nebo databáze, které splňují zadaná kritéria.

- Databáze* je oblast buněk, které tvoří seznam nebo databázi. Databáze je seznam souvisejících dat, ve které řádky souvisejících informací představují záznamy a sloupce dat jsou pole. První řádek seznamu obsahuje popisky sloupců.
- Pole* určuje, který sloupec je ve funkci používán. Argument *Pole* může být zadán jako text s popisem sloupce v uvozovkách, například “Stáří” nebo “Výnos”, nebo jako číslo představující umístění sloupce v seznamu: hodnota 1 představuje první sloupec, hodnota 2 druhý sloupec atd.
- Kritéria* je oblast buněk, která obsahuje zadaná kritéria. Pro argument *Kritéria* můžete použít libovolnou oblast, která zahrnuje nejméně jeden popis sloupce a nejméně jednu buňku pod popisem sloupce určující podmínku pro sloupec.

Dcount(Databáze;Pole;Kritéria)

Vrátí počet buněk obsahujících čísla ve sloupci seznamu nebo databáze, které splňují zadaná kritéria.

Argument *Pole* je volitelný. Je-li vynechán, funkce vrátí počet všech záznamů databáze, které splňují daná kritéria.

- Databáze* je oblast buněk, která tvoří seznam nebo databázi. Databáze je seznam souvisejících dat, ve kterých řádky souvisejících informací představují záznamy a sloupce dat jsou pole. První řádek seznamu obsahuje popisky sloupců.
- Pole* určuje, který sloupec je ve funkci používán. Argument *Pole* může být zadán jako text s popisem sloupce v uvozovkách, například “Stáří” nebo “Výnos”, nebo jako číslo představující umístění sloupce v seznamu: hodnota 1 představuje první sloupec, hodnota 2 druhý sloupec atd.
- Kritéria* je oblast buněk, která obsahuje zadaná kritéria. Pro argument *Kritéria* můžete použít libovolnou oblast, která zahrnuje nejméně jeden popis sloupce a nejméně jednu buňku pod popisem sloupce určující podmínku pro sloupec.

Dcounta(Databáze;Pole;Kritéria)

Vrátí počet všech neprázdných buněk ve sloupci seznamu nebo databáze, které splňují zadaná kritéria.

- Databáze* je oblast buněk, která tvoří seznam nebo databázi. Databáze je seznam souvisejících dat, ve kterých řádky souvisejících informací představují záznamy a sloupce dat jsou pole. První řádek seznamu obsahuje popisky sloupců.
- Pole* určuje, který sloupec je ve funkci používán. Argument *Pole* může být zadán jako text s popisem sloupce v uvozovkách, například “Stáří” nebo “Výnos”, nebo jako číslo představující umístění sloupce v seznamu: hodnota 1 představuje první sloupec, hodnota 2 druhý sloupec atd.
- Kritéria* je oblast buněk, která obsahuje zadaná kritéria. Pro argument *Kritéria* můžete použít libovolnou oblast, která zahrnuje nejméně jeden popis sloupce a nejméně jednu buňku pod popisem sloupce určující podmínku pro sloupec.

Dget(Databáze;Pole;Kritéria)

Extrahuje ze sloupce seznamu nebo databáze jednu hodnotu, která splňuje zadaná kritéria.

<i>Databáze</i>	je oblast buněk, která tvoří seznam nebo databázi. Databáze je seznam souvisejících dat, ve kterých řádky souvisejících informací představují záznamy a sloupce dat jsou pole. První řádek seznamu obsahuje popisky sloupců.
<i>Pole</i>	určuje, který sloupec je ve funkci používán. Argument <i>Pole</i> může být zadán jako text s popisem sloupce v uvozovkách, například “Stáří” nebo “Výnos”, nebo jako číslo představující umístění sloupce v seznamu: hodnota 1 představuje první sloupec, hodnota 2 druhý sloupec atd.
<i>Kritéria</i>	je oblast buněk, která obsahuje zadaná kritéria. Pro argument <i>Kritéria</i> můžete použít libovolnou oblast, která zahrnuje nejméně jeden popisek sloupce a nejméně jednu buňku pod popisem sloupce určující podmínku sloupce.

Poznámky:

- Pokud se s kritérii neshoduje žádný záznam, vrátí funkce chybovou hodnotu **!!chHod**.
- Jestliže se s kritérii shoduje více než jeden záznam, vrátí funkce chybovou hodnotu **!!chCis**.

Dmax(Databáze;Pole;Kritéria)

Vrátí maximální hodnotu ve sloupci seznamu nebo databáze, která splňuje zadaná kritéria.

<i>Databáze</i>	je oblast buněk, která tvoří seznam nebo databázi. Databáze je seznam souvisejících dat, ve kterých řádky souvisejících informací představují záznamy a sloupce dat jsou pole. První řádek seznamu obsahuje popisky sloupců.
<i>Pole</i>	určuje, který sloupec je ve funkci používán. Argument <i>Pole</i> může být zadán jako text s popisem sloupce v uvozovkách, například “Stáří” nebo “Výnos”, nebo jako číslo představující umístění sloupce v seznamu: hodnota 1 představuje první sloupec, hodnota 2 druhý sloupec atd.
<i>Kritéria</i>	je oblast buněk, která obsahuje zadaná kritéria. Pro argument <i>Kritéria</i> můžete použít libovolnou oblast, která zahrnuje nejméně jeden popisek sloupce a nejméně jednu buňku pod popisem sloupce určující podmínku sloupce.

Dmin(Databáze;Pole;Kritéria)

Vrátí minimální hodnotu ve sloupci seznamu nebo databáze, která splňuje zadaná kritéria.

<i>Databáze</i>	je oblast buněk, která tvoří seznam nebo databázi. Databáze je seznam souvisejících dat, ve kterých řádky souvisejících informací představují záznamy a sloupce dat jsou pole. První řádek seznamu obsahuje popisky sloupců.
<i>Pole</i>	určuje, který sloupec je ve funkci používán. Argument <i>Pole</i> může být zadán jako text s popisem sloupce v uvozovkách, například “Stáří” nebo “Výnos”, nebo jako číslo představující umístění sloupce v seznamu: hodnota 1 představuje první sloupec, hodnota 2 druhý sloupec atd.
<i>Kritéria</i>	je oblast buněk, která obsahuje zadaná kritéria. Pro argument <i>Kritéria</i> můžete použít libovolnou oblast, která zahrnuje nejméně jeden popisek sloupce a nejméně jednu buňku pod popisem sloupce určující podmínku sloupce.

Dproduct(Databáze;Pole;Kritéria)

Vynásobí hodnoty ve sloupci seznamu nebo databáze, které splňují zadaná kritéria.

<i>Databáze</i>	je oblast buněk, která tvoří seznam nebo databázi. Databáze je seznam souvisejících dat, ve kterých řádky souvisejících informací představují záznamy a sloupce dat jsou pole. První řádek seznamu obsahuje popisky sloupců.
<i>Pole</i>	určuje, který sloupec je ve funkci používán. Argument <i>Pole</i> může být zadán jako text s popisem sloupce v uvozovkách, například “Stáří” nebo “Výnos”, nebo jako číslo představující umístění sloupce v seznamu: hodnota 1 představuje první sloupec, hodnota 2 druhý sloupec atd.
<i>Kritéria</i>	je oblast buněk, která obsahuje zadaná kritéria. Pro argument <i>Kritéria</i> můžete použít libovolnou oblast, která zahrnuje nejméně jeden popisek sloupce a nejméně jednu buňku pod popisem sloupce určující podmínku sloupce.

Dstdev(Databáze;Pole;Kritéria)

Vrátí směrodatnou odchylku výběru pomocí čísel ve sloupci seznamu nebo databáze, která splňují zadaná kritéria.

- Databáze* je oblast buněk, která tvoří seznam nebo databázi. Databáze je seznam souvisejících dat, ve kterých řádky souvisejících informací představují záznamy a sloupce dat jsou pole. První řádek seznamu obsahuje popisky sloupců.
- Pole* určuje, který sloupec je ve funkci používán. Argument *Pole* může být zadán jako text s popisem sloupce v uvozovkách, například “Stáří” nebo “Výnos”, nebo jako číslo představující umístění sloupce v seznamu: hodnota 1 představuje první sloupec, hodnota 2 druhý sloupec atd.
- Kritéria* je oblast buněk, která obsahuje zadaná kritéria. Pro argument *Kritéria* můžete použít libovolnou oblast, která zahrnuje nejméně jeden popisek sloupce a nejméně jednu buňku pod popisem sloupce určující podmínku sloupce.

Dstdevp(Databáze;Pole;Kritéria)

Vrátí směrodatnou odchylku základního souboru pomocí čísel ve sloupci seznamu nebo v databázi, která splňují zadaná kritéria.

- Databáze* je oblast buněk, která tvoří seznam nebo databázi. Databáze je seznam souvisejících dat, ve kterých řádky souvisejících informací představují záznamy a sloupce dat jsou pole. První řádek seznamu obsahuje popisky sloupců.
- Pole* určuje, který sloupec je ve funkci používán. Argument *Pole* může být zadán jako text s popisem sloupce v uvozovkách, například “Stáří” nebo “Výnos”, nebo jako číslo představující umístění sloupce v seznamu: hodnota 1 představuje první sloupec, hodnota 2 druhý sloupec atd.
- Kritéria* je oblast buněk, která obsahuje zadaná kritéria. Pro argument *Kritéria* můžete použít libovolnou oblast, která zahrnuje nejméně jeden popisek sloupce a nejméně jednu buňku pod popisem sloupce určující podmínku sloupce.

Dsum(Databáze;Pole;Kritéria)

Sečte čísla ve sloupci seznamu nebo databáze, která splňují zadaná kritéria.

- Databáze* je oblast buněk, která tvoří seznam nebo databázi. Databáze je seznam souvisejících dat, ve kterých řádky souvisejících informací představují záznamy a sloupce dat jsou pole. První řádek seznamu obsahuje popisky sloupců.
- Pole* určuje, který sloupec je ve funkci používán. Argument *Pole* může být zadán jako text s popisem sloupce v uvozovkách, například “Stáří” nebo “Výnos”, nebo jako číslo představující umístění sloupce v seznamu: hodnota 1 představuje první sloupec, hodnota 2 druhý sloupec atd.
- Kritéria* je oblast buněk, která obsahuje zadaná kritéria. Pro argument *Kritéria* můžete použít libovolnou oblast, která zahrnuje nejméně jeden popisek sloupce a nejméně jednu buňku pod popisem sloupce určující podmínku sloupce.

Dvar(Databáze;Pole;Kritéria)

Vrátí rozptyl výběru základního souboru pomocí čísel ve sloupci seznamu nebo databáze, která splňují zadaná kritéria.

- Databáze* je oblast buněk, která tvoří seznam nebo databázi. Databáze je seznam souvisejících dat, ve kterých řádky souvisejících informací představují záznamy a sloupce dat jsou pole. První řádek seznamu obsahuje popisky sloupců.
- Pole* určuje, který sloupec je ve funkci používán. Argument *Pole* může být zadán jako text s popisem sloupce v uvozovkách, například “Stáří” nebo “Výnos”, nebo jako číslo představující umístění sloupce v seznamu: hodnota 1 představuje první sloupec, hodnota 2 druhý sloupec atd.
- Kritéria* je oblast buněk, která obsahuje zadaná kritéria. Pro argument *Kritéria* můžete použít libovolnou oblast, která zahrnuje nejméně jeden popisek sloupce a nejméně jednu buňku pod popisem sloupce určující podmínku sloupce.

Dvarp(Databáze;Pole;Kritéria)

Vrátí rozptyl základního souboru pomocí čísel ve sloupci seznamu nebo databáze, která splňují zadaná kritéria.

Databáze je oblast buněk, která tvoří seznam nebo databázi. Databáze je seznam souvisejících dat, ve kterých řádky souvisejících informací představují záznamy a sloupce dat jsou pole. První řádek seznamu obsahuje popisky sloupců.

Pole určuje, který sloupec je ve funkci používán. Argument *Pole* může být zadán jako text s popisem sloupce v uvozovkách, například “Stáří” nebo “Výnos”, nebo jako číslo představující umístění sloupce v seznamu: hodnota 1 představuje první sloupec, hodnota 2 druhý sloupec atd.

Kritéria je oblast buněk, která obsahuje zadaná kritéria. Pro argument *Kritéria* můžete použít libovolnou oblast, která zahrnuje nejméně jeden popisek sloupce a nejméně jednu buňku pod popisem sloupce určující podmínku sloupce.

Finanční funkce

V obchodních funkcích se často setkáváte s úrokovou mírou. Abyste se vyhnuli zbytečným komplikacím, musíte si uvědomit, že:

- úroková míra je číslo jako každé jiné, tzn. že chcete-li zadat úrok 15 %, musíte napsat 0,15
- úrok se vždy myslí za jedno splátkové období.

Ddb(Cena;ZůstHodnota;Životnost;Období;Faktor)

Vrátí odpis aktiva za určité období pomocí dvojité degresivní metody odpisu nebo jiné metody, kterou zadáte.

<i>Cena</i>	je pořizovací cena aktiva
<i>ZůstHodnota</i>	je zůstatková cena na konci období odpisu (někdy se nazývá zůstatková hodnota aktiva)
<i>Životnost</i>	je počet období, po které je aktivum odepisováno (někdy se nazývá životnost aktiva)
<i>Období</i>	je období, za které chcete vypočítat odpis. Argument <i>Období</i> musí být ve stejných jednotkách jako argument <i>Životnost</i>
<i>Faktor</i>	je míra poklesu zůstatku. Jestliže argument <i>Faktor</i> vynecháte, bude jeho hodnota 2 (dvojitá degresivní metoda odpisu)

Všech pět argumentů musí být kladná čísla.

Poznámky:

Dvojitá degresivní metoda odpisu počítá odpis při zvýšené sazbě. Odpis je nejvyšší v prvním období a v následujících obdobích klesá. Funkce používá k výpočtu odpisu za určité období následující vzorec:

$$\text{náklady} - \text{zůstatek}(\text{celkový odpis za předchozí období}) * \text{faktor} / \text{životnost}$$

Nechcete-li použít dvojitou degresivní metodu odpisu, změňte argument *Faktor*.

Příklady:

Předpokládejme, že chcete pro podnik koupit nový stroj. Stroj stojí 2 400 Kč a má životnost 10 let. Zůstatková cena stroje je 300 Kč. Následující příklady ilustrují odpis stroje během několika období. Výsledky jsou zaokrouhleny na dvě desetinná místa.

Ddb(2400;300;3650;1) vrátí hodnotu 1,32 Kč, odpis za první den. Automaticky se předpokládá, že argument *Faktor* je roven 2.

Ddb(2400;300;120;1;2) vrátí hodnotu 40,00 Kč, odpis za první měsíc.

Ddb(2400;300;10;1;2) vrátí hodnotu 480,00 Kč, odpis za první rok.

Ddb(2400;300;10;2;1,5) vrátí hodnotu 306,00 Kč, odpis za druhý rok, hodnota argumentu *Faktor* je 1,5 místo dvojité degresivní metody odpisu.

Ddb(2400;300;10;10) vrátí hodnotu 22,12 Kč, odpis za desátý rok. Automaticky se předpokládá, že argument *Faktor* je roven 2.

Fv(Úrok;PočetObdobí;Splátka;SoučasnáHodnota;Typ)

Vrátí budoucí hodnotu investice. Argumenty *současná_hodnota* a *typ* jsou nepovinné. Jestliže je vynecháte, 602Tab je považuje za nulové.

<i>Úrok</i>	číselná hodnota ≥ 0 , představující úrokovou míru
<i>PočetObdobí</i>	celé číslo ≥ 2
<i>Splátka</i>	číselná hodnota, představující výši pravidelné splátky
<i>SoučasnáHodnota</i>	současný stav účtu (nepovinný argument, pokud není uveden nahradí se 0).
<i>Typ</i>	hodnota je 0 - jestliže jsou splátky splatné na konci období hodnota je 1 - jestliže jsou splátky splatné na začátku období. Je to nepovinný argument; pokud není uveden, nahradí se 0.

Příklady: Vklad 1000 Kč, úrok 14%. Kolik bude na účtu za 10 let?

$$FV(14\%; 10; -1000) = 19\,337,30$$

Vklad 100 Kč každý měsíc, úrok 10%. Kolik bude na účtu za 10 let?

$$FV(10\%/12; 10*12; -100; 0) = 20\,484,50$$

Ipmt(Úrok;SplátkovéObdobí;PočetObdobí; SoučasnáHodnota;BudoucíHodnota;Typ)

Vrátí výšku úroku v určitém úrokovém období.

Platí, že $PMT = IPMT + PPMT$, kde funkce **PMT** počítá celkové splátky, **IPMT** úrokovou část splátky a **PPMT** jistinnou část splátky. **IPMT** se počítá výpočtem jednoduchého úročení zbývajících jistiny z minulého měsíce.

<i>Úrok</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující pevnou periodickou úrokovou míru (číslo větší než -1)
<i>SplátkovéObdobí</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, určující splátkové období pro které chcete spočítat IPMT
<i>PočetObdobí</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující počet splátkových období (číslo větší než 0)
<i>SoučasnáHodnota</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující vypůjčenou částku (jistinu)
<i>BudoucíHodnota</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující budoucí hodnotu investice (tento argument je nepovinný, pokud není uveden nahradí se 0).
<i>Typ</i>	hodnota je 0- jestliže jsou splátky splatné na konci období hodnota je 1- jestliže jsou splátky splatné na začátku období. Je to nepovinný argument; pokud není uveden, nahradí se 0.

Příklad:

Předpokládejte, že jste si půjčili 1000000 Kč na 10 let s 12% úrokem. Potom úroky z půjčky po 12 měsících dávají v současném měsíci:

$$IPMT(12\%/12; 12; 10*12; 1000000) = -9497,18 \text{ Kč}$$

Výsledek vyšel záporně, protože se jedná o výdaje a ne o příjem.

Irr(Hodnoty;Odhad)

Vrátí míru výnosnosti pro sérii peněžních toků představovaných číselnými hodnotami. Tyto peněžní toky si nemusí být rovny jako v případě annuity. K peněžním tokům však musí docházet v pravidelných intervalech, například jednou za měsíc nebo za rok. Míra výnosnosti je úroková míra investice zahrnující výdaje (záporné hodnoty) a příjmy (kladné hodnoty), které jsou rozloženy v pravidelných intervalech.

<i>Hodnoty</i>	je matice hodnot nebo odkaz na buňky s hodnotami, pro které chcete vypočítat míru výnosnosti. <ul style="list-style-type: none"> Aby bylo možné vypočítat míru výnosnosti, musí být mezi čísla určenými argumentem hodnoty alespoň jedno kladné a jedno záporné číslo. Funkce řadí peněžní toky podle pořadí hodnot v argumentu. Proto je potřeba při zadávání výdajů a příjmů dbát na správné seřazení. Pokud matice nebo argument odkazu obsahuje text, logické hodnoty nebo prázdné buňky, jsou tyto hodnoty ignorovány.
<i>Odhad</i>	je číslo, které se podle vašeho odhadu blíží výsledku funkce. <ul style="list-style-type: none"> V aplikaci 602Tab se míra výnosnosti počítá pomocí iterací. Začne se od zadaného odhadu a pokračuje se, dokud se následující výsledky neliší o méně než 0,00001 procenta. Jestliže se nepodaří najít výsledek po 20 iteracích, vrátí funkce chybovou hodnotu !!chCis. Ve většině případů není nutné odhad zadávat. Nezádáte-li ho, bude použita implicitní hodnota 0,1 (10 procent).

- Vráti-li funkce chybovou hodnotu **!!chCis** nebo je-li výsledek příliš vzdálený očekávané hodnotě, zkuste použít jiný odhad.

Příklady:

Chcete například otevřít restauraci. Odhadujete, že náklady spojené se zahájením podnikání by činily 700 000 Kč a čistý zisk v prvních pěti letech by byl postupně: 120 000 Kč, 150 000 Kč, 180 000 Kč, 210 000 Kč a 260 000 Kč. Oblast B1:B6 bude obsahovat následující hodnoty: -700 000 Kč, 120 000 Kč, 150 000 Kč, 180 000 Kč, 210 000 Kč a 260 000 Kč.

Míra výnosnosti investice po čtyřech letech se vypočítá následujícím způsobem:

Irr(B1:B5) rovná se -2,12 procenta

Míra výnosnosti po pěti letech se vypočítá následujícím způsobem:

Irr(B1:B6) rovná se 8,66 procenta

Pro výpočet míry výnosnosti po dvou letech je již potřeba zadat odhad:

Irr(B1:B3;-10%) rovná se -44,35 procenta

Mirr(Odkaz;Úrok;ÚrokReinvest)

Vráti modifikovanou vnitřní míru výnosnosti pro sérii pravidelných peněžních toků, uvažuje náklady na investici a úrok získaný z reinvestování hotovosti.

Odkaz matice nebo odkaz na buňky, které obsahují čísla, pro která se má počítat vnitřní míra výnosnosti. Tato čísla reprezentují sérii plateb (záporné hodnoty) a příjmů (kladné hodnoty), prováděných periodicky.

- Hodnoty musí obsahovat nejméně jednu kladnou a jednu zápornou hodnotu, aby bylo možno vypočítat modifikovanou vnitřní míru výnosnosti. V opačném případě funkce vrátí chybovou hodnotu **!!ch/0**.
- Obsahuje-li matice nebo odkaz text, logické hodnoty nebo prázdné buňky, jsou tyto hodnoty ignorovány. Buňky, které obsahují nulovou hodnotu, jsou do výpočtu zahrnuty.

Úrok úroková sazba, která je placena za využití peněžní toky

ÚrokReinvest je sazba získaná při reinvestici peněžních toků

Poznámka:

Funkce **MIRR** používá pořadí hodnot k interpretaci pořadí peněžních toků. Ujistěte se, že platby a příjmy jsou ve správném pořadí a se správnými znaménky (kladné hodnoty pro příjmy a záporné pro platby).

Příklady:

Před 5 lety jste například založili obchod s rybami. Vypůjčili jste si tehdy 120 000 Kč na 10% úrok na koupi lodě. Za úlovky jste obdrželi 39 000, 30 000, 21 000, 37 000 a 46 000 Kč. Tyto obnosy jste reinvestovali při 12 procentním úroku. V tabulce je půjčka 120 000 Kč uložena v buňce B1 a příjmy za každé z pěti let v buňkách B2:B6.

K výpočtu modifikované vnitřní míry výnosnosti této investice po 5 letech použijte vzorec:

MIRR(B1:B6; 10%; 12%) = 12,61 procenta

K výpočtu modifikované vnitřní míry výnosnosti této investice po 3 letech použijte vzorec:

MIRR(B1:B4; 10%; 12%) = - 4,80 procenta

K výpočtu modifikované vnitřní míry výnosnosti této investice po 5 letech, je-li reinvestiční úroková míra 14 procent, použijte vzorec:

MIRR(B1:B6; 10%; 14%) = 13,48 procenta

Nper(Úrok;Splátka;SoučasnáHodnota;BudoucíHodnota;Typ)

Vráti počet úročených období pro investici.

Úrok číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující pevnou periodickou úrokovou míru (číslo větší než -1)

<i>Splátka</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující pravidelnou splátku <i>SoučasnáHodnota</i> číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující současnou hodnotu investice
<i>SoučasnáHodnota</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující vypůjčenou částku (jistinu)
<i>BudoucíHodnota</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující budoucí hodnotu investice (tento argument je nepovinný, pokud není uveden, nahradí se 0)
<i>Typ</i>	hodnota je 0- jestliže jsou splátky splatné na konci období; hodnota je 1- jestliže jsou splátky splatné na začátku období. Je to nepovinný argument; pokud není uveden, nahradí se 0.

Příklad:

Za jak dlouho našetřím 100000 Kč, budu-li měsíčně vkládat 3000 Kč na 10% úrok?

$$\text{NPER}(10\%/12; -3000; 0; 100000) = 29,53 \text{ měsíců}$$

Npv(x;Hodnota1;Hodnota2; ...)

Vypočítává čistou současnou hodnotu investice na základě diskontní sazby, hodnot budoucích plateb (záporné hodnoty) a příjmů (kladné hodnoty).

x je diskontní sazba vztažená k úročenému období.

Hodnota1, Hodnota2,... je 1 až 29 hodnot představujících peněžní toky.

- Argumenty *hodnota1, Hodnota2,...* musí být rovnoměrně rozloženy v čase a realizovány na konci období.
- Funkce užívá pořadí argumentů *Hodnota1, Hodnota2...* k interpretaci pořadí peněžních toků. Zadávejte proto platby a příjmy ve správném pořadí.
- Jako argumenty mohou být použita čísla, prázdné buňky, logické hodnoty a texty reprezentující čísla. Argumenty obsahující chybové hodnoty nebo texty, které nelze převést na číslo, jsou ignorovány.
- Je-li argumentem matice nebo odkaz, jsou pro výpočet uvažována pouze čísla. Prázdné buňky, logické hodnoty, text nebo chybové hodnoty jsou ignorovány.

Investice je realizována jedno období před datem prvního peněžního toku daného argumentem *Hodnota1* a končí posledním peněžním tokem v seznamu. Výpočet funkce je založen na budoucích peněžních tocích. Je-li první peněžní tok realizován na počátku prvního období, musí být tato hodnota k získanému výsledku přičtena a nesmí se udávat v seznamu hodnot. Další informace získáte v níže uvedeném příkladu.

Příklady:

Předpokládáte, že zaplatíte za rok za investici 10 000 Kč a roční příjmy ve třech následujících letech jsou 3 000 Kč, 4 200 Kč a 6 800 Kč. Roční diskontní sazba je 10%. Potom čistá současná hodnota investice je:

$$\text{Npv}(10\%; -10\,000; 3\,000; 4\,200; 6\,800) \text{ rovná se } 1188,44 \text{ Kč}$$

V tomto příkladu jsou náklady ve výši 10 000 Kč jednou z hodnot, protože platba proběhne na konci 1. období.

Předpokládáte investici, která začíná platbou 40 000 Kč na začátku 1. období. v dalších pěti letech očekáváte příjmy z investice 8 000 Kč, 9 200 Kč, 10 000 Kč, 12 000 Kč a 14 500 Kč. Roční diskontní sazba je 8%, což může reprezentovat míru inflace nebo úrokovou míru z alternativní investice.

Pokud platba a příjmy jsou v odpovídajícím pořadí uloženy v buňkách B1 až B6, potom čistá současná hodnota investice je:

$$\text{Npv}(8\%; B2:B6)+B1 \text{ rovná se } 1922,06 \text{ Kč}$$

V tomto příkladu nelze zahrnout náklady 40 000 Kč do seznamu hodnot, protože platba proběhla na počátku 1. období.

Předpokládáte, že investice bude mít v 6. roce ztrátu 9 000 Kč. Čistá současná hodnota po 6 letech bude:

Npv(8%; B2:B6; -9000)+B1 rovná se -3 749,47 Kč

Pmt(Sazba;Úrok;PočetObdobí;SoučasnáHodnota;BudoucíHodnota;Typ)

Vrátí periodickou konstantní platbu pro anuitu.

sazba = úroková sazba

Platí, že **PMT = IPMT + PPMT**, kde funkce **PMT** počítá celkové splátky, **IPMT** úrokovou část splátky a **PPMT** jistinnou část splátky. **PMT** umožňuje počítat předběžnou splatnost a bere v úvahu budoucí hodnotu jistiny.

<i>Sazba</i>	úroková sazba
<i>Úrok</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující pevnou periodickou úrokovou míru (číslo větší než -1)
<i>PočetObdobí</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující počet splátkových období (číslo větší než 0)
<i>SoučasnáHodnota</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující vypůjčenou částku (jistinu)
<i>BudoucíHodnota</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující budoucí hodnotu investice (tento argument je nepovinný, pokud není uveden nahradí se 0)
<i>Typ</i>	hodnota je 0- jestliže jsou splátky splatné na konci období hodnota je 1- jestliže jsou splátky splatné na začátku období. Je to nepovinný argument; pokud není uveden, nahradí se 0.

Příklad:

Jaké budou měsíční splátky při půjčce 50000 na 3 roky s úrokem 15%?

PMT(15%/12; 3*12; 50000) = -1733,30 Kč

Ppmt(Sazba;SplátkovéObdobí;PočetObdobí;SoučasnáHodnota;BudoucíHodnota;Typ)

Vrátí rozdíl základní jistiny v roční splátce za dané období při konstantní úrokové sazbě. Platí, že **PMT = IPMT + PPMT**, kde funkce **PMT** počítá celkové splátky, **IPMT** úrokovou část splátky a **PPMT** jistinnou část splátky.

<i>Sazba</i>	úroková sazba
<i>Úrok</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující pevnou periodickou úrokovou míru (číslo větší než -1)
<i>SplátkovéObdobí</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, určující splátkové období
<i>PočetObdobí</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující počet splátkových období (číslo větší než 0)
<i>SoučasnáHodnota</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující vypůjčenou částku (jistinu)
<i>BudoucíHodnota</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující budoucí hodnotu investice (tento argument je nepovinný, pokud není uveden nahradí se 0)
<i>Typ</i>	hodnota je 0-jestliže jsou splátky splatné na konci období; hodnota je 1- jestliže jsou splátky splatné na začátku období. Je to nepovinný argument; pokud není uveden, dosadí se 0.

Příklad:

Chcete si vzít půjčku 1500000 Kč na 20 let s roční úrokovou mírou 14%. Jak velká bude jistinná část měsíční částky po dvanácti měsících?

PPMT(14%/12; 12; 12*20; 1500000) = -1309,70

Všimněte si, že půjčku (příjem) jsme zapsali kladně a že výsledek (výdaj) vyšel záporně.

PV(Úrok; PočetObdobí; Splátka; BudoucíHodnota; Typ)

Vrátí aktuální hodnotu investice.

<i>Úrok</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující pevnou periodickou úrokovou míru (číslo větší než -1)
<i>PočetObdobí</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující počet splátkových období (číslo větší než 0)
<i>Splátka</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující pravidelnou splátku
<i>BudoucíHodnota</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující budoucí hodnotu investice (tento argument je nepovinný, pokud není uveden nahradí se 0);
<i>Typ</i>	hodnota je 0 -jestliže jsou splátky splatné na konci období; hodnota je 1 -jestliže jsou splátky splatné na začátku období. Je to nepovinný argument; pokud není uveden, dosadí se 0.

Příklad:

Kolik musím uložit na 10% úrok, abych za 10 let měl 1000 Kč?

PV(10%; 10; 0; 10000) = -3855,40

Budu si chtít v období 10 let vybírat měsíčně 2000 Kč. Kolik si musím uložit při 10% úroku?

PV(10%/12; 10*12; 2000) = -151 342,30

Rate(PočetObdobí; Splátka; SoučasnáHodnota; BudoucíHodnota; Typ)

Vrátí úrokovou sazbu vztaženou k úročenému období anuity.

<i>PočetObdobí</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující počet splátkových období (číslo větší než 0)
<i>Splátka</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující pravidelnou splátku
<i>SoučasnáHodnota</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující současnou hodnotu investice
<i>BudoucíHodnota</i>	číslo nebo odkaz na buňku obsahující číselnou hodnotu, představující budoucí hodnotu investice (tento argument je nepovinný, pokud není uveden nahradí se 0);
<i>Typ</i>	hodnota je 0 - jestliže jsou splátky splatné na konci období; hodnota je 1 -jestliže jsou splátky splatné na začátku období. Je to nepovinný argument; pokud není uveden, dosadí se 0.

Příklad:

Mohu ukládat 1000 Kč měsíčně na konto se současným stavem 100 Kč. Jaký musí být úrok, abych po 5 letech mohl vybrat 100000 Kč?

RATE(5*12; -1000; -100; 100000; 0) = 2 %

Měsíční úrok musí být 2%.

Sln(Cena; ZůstHodnota; Životnost)

Vrátí lineární odpisy aktiva pro jednoduché období.

<i>Cena</i>	je pořizovací cena aktiva
<i>ZůstHodnota</i>	je zůstatková cena na konci doby odepisování
<i>Životnost</i>	je počet let, ve kterých se aktivum odepisuje

Příklad:

Koupili jste například nákladní automobil za 30 000 Kč, který má životnost 10 let a zůstatková cena je 7 500 Kč. Povolená odpisovaná částka pro každý rok je:

Sln(30000; 7500; 10) rovná se 2 250 Kč

Syd(Cena;ZůstHodnota;Životnost;Období)

Vrátí degresivní odpisy aktiva pro určené období.

<i>Cena</i>	je pořizovací cena aktiva
<i>ZůstHodnota</i>	je zůstatková cena na konci doby odepisování
<i>Životnost</i>	je počet let, ve kterých se aktivum odepisuje
<i>Období</i>	je období, pro které se počítá výše odpisů (ve stejných jednotkách jako životnost)

Příklady:

Pokud jste například koupili zboží za 30 000 Kč, které má životnost 10 let a zbytkovou hodnotu 7500 Kč, je roční odepsaná částka:

SYD(30000;7500;10;1) rovná se 4090,91 Kč

Roční odepsaná částka v 10-tém roce je:

SYD(30000;7500;10;10) rovná se 409,09 Kč

Funkce pracující s datem

Pomocí datových a časových funkcí můžete analyzovat hodnoty data i času a pracovat ve vzorcích.

Date(Rok;Měsíc;Den)

Funkce **DATE** pro zadaný letopočet, číslo měsíce a dne vrací kalendářní datum ve zvoleném formátu.

<i>Rok</i>	letopočet zapsaný dvojčíslicím nebo čtyřčíslicím
<i>Měsíc</i>	číslo měsíce (1 až 12)
<i>Den</i>	číslo dne (1 až 31)

Příklad:DATE(97;11;10) = 11.10.1997

Datevalue(Text)

Vrátí pořadové číslo data vyjádřeného argumentem *text*. Pomocí funkce lze převádět datum z textového tvaru na pořadové číslo.

Text text, který udává datum ve formátu pro datum aplikace 602Tab

- Při použití původního nastavení formátu data v aplikaci 602Tab musí argument datum vyjadřovat datum od 1. ledna 1900 do 31. prosince 9999.
- Vynecháte-li v argumentu *Text* část udávající rok, funkce použije právě platný rok z vestavěných hodin počítače. Informace o čase v argumentu *Text* se ignorují.

Poznámka:

Většina funkcí automaticky převede hodnoty kalendářních dat na pořadová čísla.

Příklady:

Následující příklady pracují se systémem dat 1900:

DATEVALUE("8/22/55") = 20323

DATEVALUE ("22-Srp-55") = 20323

Pokud jsou vestavěné hodiny počítače nastaveny na 1993 a používáte systém dat 1900:

DATEVALUE ("5-Čec") = 34155

Day(Datum)

Funkce **DAY** převádí hodnotu *datum* na číslo dne v měsíci (1 - 31). Parametr *datum* musí být typu datum nebo číslo nebo odkaz na buňku obsahující tyto typy.

Příklad:DAY(10.11.97) = 10

je-li obsah buňky **A1** 1.1.98, pak **DAY(A1) = 1**

Hour(Čas)

Převede argument *Čas* na pořadové číslo odpovídající naposledy uběhlé celé hodině.

Příklady:HOURL(10:21) = 10

HOURL(1.12.97 15:41) = 15

Minute(Čas)

Převede argument *Čas* na pořadové číslo odpovídající poslední celé uběhlé minutě. Argument musí být hodnota typu datum a čas nebo odkaz na buňku tohoto typu.

Příklady:MINUTE(21:45) = 45

MINUTE(24.12.1997 12:28) = 28

Month(Datum)

Funkce převádí hodnotu argumentu na pořadové číslo měsíce. Argument *Datum* musí být hodnota typu datum nebo číslo či odkaz na buňku těchto typů.

Příklady: MONTH(31.12.1997) vrací hodnotu 12

MONTH(24.6.1998) vrací hodnotu 6

Now()

Vrátí „pořadové číslo“ současného data a času. Jeho celočíselná část označuje počet dní uplynulých od 1.1.1900. Desetinná část pak vyjadřuje zlomek času uplynulého z „tohoto“ dne. Nasadíte-li na výsledek funkce formát datum a čas, obdržíte okamžité kalendářní datum a okamžitý časový údaj.

Příklad:

Funkce vrátí číslo 36545,4655555556. Nasadíte-li na příslušnou buňku formát **d.m.yy h:mm**, obdržíte 20.1.00 11:10.

Second(Čas)

Převádí argument *Čas* na pořadové číslo poslední uběhlé sekundy.

Příklad:

Vrátí-li funkce =Second(Now()) číslo 35, znamená to, že z aktuální minuty právě uběhla pětaticátá sekunda.

Time(Hodina;Minuta;Vteřina)

Na základě časového údaje, zadaného v hodinách, minutách a vteřinách, vrátí desetinné číslo mezi nulou a jedničkou vyjadřující uplynulou část dne. Vynásobíte-li výsledek stovkou, získáte údaj o tom, kolik procent ze dne již uplynulo.

Příklady:

TIME(12;00;00) = 0,5

TIME (06;00;00) = 0,25

Timevalue(Text)

Vrátí pořadové číslo času zadaného jako textový řetězec. Pořadové číslo je desetinné číslo v rozmezí od 0 (nula) do 0,99999999, reprezentující čas od 0:00:00 (12:00:00 dop.) do 23:59:59 (11:59:59 odp.). Funkce se používá k převodu času v textové podobě na pořadové číslo.

Text textový řetězec představující čas v libovolném formátu používaném v aplikaci 602Tab. Doplnující informace u času se ignorují.

Příklady:

TIMEVALUE("2:24 dop.") = 0,1

TIMEVALUE ("22-Srp-55 6:35 odp.") = 0,274305556

Today()

Funkce vrátí pořadové číslo dne aktuálního data – vztaženo k 1.1.1900 (aktuální datum se přebírá podle nastavení hodin počítače). Pořadové číslo je kód data, který lze používat při výpočtech. Nasadíte-li na buňku některý z formátů typu datum a čas, bude zobrazeno přímo kalendářní datum.

Weekday(Datum;Typ)

Funkce převádí *datum* na pořadové číslo dne v týdnu (1-neděle, 2-pondělí, 3-úterý...). Argument *datum* musí být typu datum.

Příklady: WEEKDAY(19.10.94) = 3

WEEKDAY(1.1.2000) = 6

WEEKDAY(Now()) = 3

Year(Datum)

Funkce separuje z hodnoty *Datum* letopočet (resp. počet roků od roku 1900). Argument *Datum* musí být hodnota typu datum nebo číslo, případně odkaz na buňku těchto typů.

Příklady: **YEAR(24.12.1997) = 97**

pokud je **D25 = 25.6.1999**, **YEAR(D25) = 99**

Textové funkce

Pomocí textových funkcí je možné upravit textové řetězce ve vzorcích. Můžete například změnit velikost písma nebo určit délku textového řetězce. Lze také například spojit (neboli zřetěžit) datum do textového řetězce.

Char(Číslo)

Funkce převádí argument *Číslo* na znak (WinEECS). Argumentem *n* musí být číslo z intervalu 1 až 255; výsledek je textového typu.

Příklady: CHAR(43) vrací znak +
 CHAR(65) vrací znak A
 CHAR(250) vrací znak ú

Clean(Text)

Odstraní z textu všechny netisknutelné znaky. Funkci lze použít u textů importovaných z jiných aplikací, obsahujících znaky, které se ve vámi používaném prostředí nevytisknou. Pomocí funkce můžete například odstranit některé kódy, které se často vyskytují na počátku a na konci datových souborů a které nelze vytisknout.

Text je libovolná informace z tabulky, ze které chcete odstranit netisknutelné znaky

Příklad:

Jelikož ZNAK(7) vrátí netisknutelný znak:

CLEAN(ZNAK(7)&"text"&ZNAK(7)) = "text"

Code(Text)

Funkce vrací kódovou hodnotu (WinEECS) prvního znaku *Text*. Opačnou funkcí je funkce **Char**. Parametr *Text* musí být textový výraz. Výsledek je numerického typu.

Příklady: CODE("A") = 65
 CODE("?") = 63
 CODE("Zlato") = 90

Concatenate(Text1;Text2)

Sloučí několik textových řetězců do jednoho.

Text1;Text2; je 1 až 30 textových položek, které mají být sloučeny do jediné. Tyto položky mohou obsahovat textové řetězce, čísla nebo odkazy na jednotlivé buňky.

Poznámky:

Místo klíčového slova **CONCATENATE** lze pro slučování textu použít operátor "&".

Příklady:

CONCATENATE("Hodnota "; "celkem") rovná se "Hodnota celkem".

Je to totéž, jako když napíšete: "Hodnota"&" "&"celkem" .

V sešitu s přehledem o živočišstvu v řece je například sloupec C2 obsahující "druh", sloupec C5 obsahující "říční pstruh" a sloupec C8 obsahující celkový počet 32.

CONCATENATE("Populace ";C5;"u ";C2;"je ";C8;"kilometr") rovná se "Populace druhu říční pstruh je 32/kilometr".

Exact(Text1;Text2)

Porovná dva textové řetězce, zda jsou shodné; v kladném případě vrací hodnotu TRUE, jinak FALSE.

Příklady: `Exact("wintext","wintext")` vrací hodnotu TRUE
`Exact("WinText","wintext")` vrací hodnotu FALSE
`Exact("WinText","Win Text")` vrací hodnotu FALSE

Find(HledanýText;VTextu;Od)

Vyhledá jeden textový řetězec (*HledanýText*) uvnitř jiného (*VTextu*) a vrátí číslo pozice prvního znaku nalezeného podřetězce (*HledanýText*) vzhledem k prvnímu znaku zleva v řetězci *VTextu*.

HledanýText je hledaný řetězec

- Pokud argument *HledanýText* je "" (prázdný řetězec), potom podle funkce odpovídá hledaný řetězec prvnímu porovnávanému znaku druhého řetězce (znak, jehož číslo se rovná hodnotě argumentu *Od* nebo hodnotě 1).
- Argument *HledanýText* nemůže obsahovat žádné zástupné znaky.

VTextu je řetězec, který bude prohledán.

Od je pozice znaku v řetězci *VTextu* od kterého se má začít prohledávat. První znak v řetězci *VTextu* je na pozici 1. Pokud je argument *Od* vynechán, začne se prohledávat od pozice 1.

Poznámky:

- Pokud se řetězec *co* v řetězci kde nevyskytuje, vrátí funkce **Find** chybovou hodnotu **!!chHod**.
- Pokud hodnota argumentu *start* není větší než nula, vrátí funkce **Find** chybovou hodnotu **!!chHod**.
- Je-li hodnota argumentu *start* větší než délka řetězce kde, vrátí funkce **Find** chybovou hodnotu **!!chHod**.

Příklady:

`Find("M";"Martina Malá") = 1`
`Find("m";"Martina Malá") = 6`
`Find("M";"Martina Malá";3) = 8`

Fixed(Číslo;PočetDesMíst;NeOddělovač1000)

Zaokrouhlí číslo na určený počet desetinných míst, zformátuje toto číslo v desetinném formátu s tečkou a čárkami a výsledek vrátí v podobě textu.

Číslo je číslo, které chcete zaokrouhlit a převést na text

PočetDesMíst je počet číslic vpravo od desetinné čárky

NeOddělovač1000 je logická hodnota, která, pokud je TRUE, zakazuje funkci v navráceném textu použít čárky. Pokud má argument *NeOddělovač1000* hodnotu FALSE nebo není uveden, budou v navráceném textu uvedeny čárky jako obvykle.

- Čísla v aplikaci 602Tab nesmějí mít nikdy více než 15 platných číslic, ale argument *PočetDesMíst* může být až 127.
- Pokud je argument *PočetDesMíst* záporné číslo, zadané číslo se zaokrouhlí po levé straně desetinné čárky.
- Pokud argument *PočetDesMíst* vynecháte, předpokládá se, že jeho hodnota je 2.

Poznámka:

Hlavní rozdíl mezi formátováním buňky obsahující číslo pomocí příkazu **Buňky** (nabídka **Formát**) a formátováním tohoto čísla přímo funkcí **FIXED** je ten, že funkce **FIXED** převede výsledek na textový tvar. Číslo zformátované příkazem **Buňka** je stále číslo.

Příklady:

`FIXED(1234,567; 1) = "1234,6"`
`FIXED(1234,567; -1) = "1230"`
`FIXED(-1234,567; -1) = "-1230"`
`FIXED(44,332) = "44,33"`

Left(Text;N)

Funkce vrátí prvních *N* znaků odleva z řetězce *Text*. První znak v řetězci má číslo 1. Výsledek je typu řetězec.

Text textový výraz
N celé číslo větší než nula.

Příklady:LEFT("ahoj";2) = "ah"

LEFT("Budlák";1) = "B"

Len(Text)

Funkce vrátí délku textového řetězce *Text*. Parametr *Text* musí být textový výraz. Výsledek je typu číslo.

Příklady:LENGTH(B6) = 14 (například)

LENGTH("WinTab602") = 9

Lower(Text)

Funkce vrátí textový řetězec *Text* převedený na malá písmena. Argument *Text* musí být textový výraz. Výsledek je typu řetězec.

Příklady:LOWER("AHoj") = "ahoj"

LOWER("VÝPOČET") = "výpočet"

Mid(Text;PrvníZnak;PočetZnaků)

Funkce vrací podřetězec vzniklý z celkem *n2* znaků textového řetězce *řetězec*, počínajíc od znaku *n1* zleva. Výsledek je typu řetězec.

Text textový výraz
PrvníZnak Celé číslo větší než nula udávající, od které pozice výrazu *řetězec* bude podřetězec začínat.

- Je-li *PrvníZnak* větší než délka *řetězce*, funkce vrací prázdný řetězec.
- Je-li *PrvníZnak* menší než délka *řetězce*, ale součet *PrvníZnak* + *PočetZnaků* délku řetězce přesahuje, funkce vrací znaky od pozice *PrvníZnak* do konce řetězce.
- Pokud je *PrvníZnak* menší než nula, funkce vrací chybovou hodnotu **!!chHod**.

PočetZnaků Celé číslo větší než nula udávající délku požadovaného podřetězce. Pokud je *PočetZnaků* menší než nula, funkce vrací chybovou hodnotu **!!chHod**.

Příklady:MID("Software602";1;8) = "Software"

MID("Software602";9;3) = "602"

MID("Software602";18;8) = ""

Proper(Text)

Funkce převádí první písmeno každého slova v řetězci na velké písmeno a zbytek znaků na malá písmena. Za oddělovač slov je považována mezera, interpunkční znaménko, číslice atd. Parametr *řetězec* musí být textový výraz. Výsledek je typu řetězec.

Příklad:PROPER("SOFTWARE602") = "Software602"

Replace(Text;Pozice;N;NovýText)

Nahradí znaky v textu. Pozice prvního písmene v řetězci je 1.

Text textový výraz
Pozice číslo větší než nula
N číslo větší než nula
NovýText textový výraz

Příklad:

REPLACE("pes kočka"; 5; 5; "lev") vrátí řetězec "pes lev"

Rept(Text;nKrát)

Několikrát zopakuje zadaný *text*. Funkce **Rept** se používá k vložení několikrát se opakujícího textového řetězce do buňky.

Text je text, který chcete zopakovat

nKrát je kladné číslo určující počet opakování. Pokud je *nKrát* roven 0 (nula), vrátí funkce **REPT ""** (prázdný řetězec). Jestliže argument *nKrát* není celé číslo, bude zkrácen. Výsledek funkce nemůže být delší než 255 znaků.

Tip: Tuto funkci můžete použít k vytvoření jednoduchého histogramu na vašem listu.

Příklady:

Rept("*-"; 3) rovná se "*-*-*"

Pokud buňka A3 obsahuje "Prodej", pak:

Rept(\$A\$3; 2,9) rovná se "ProdejProdej"

Right(Text;N)

Funkce vrátí *n* posledních znaků vyskytujících se napravo v řetězci *řetězec*. Výsledek je typu řetězec.

Text textový výraz

N číslo větší než nula.

Příklady:RIGHT("ahoj";2) = "oj"

RIGHT("Petr";1) = "r"

Substitute(Text;StarýText;NovýText;N)

Nahradí v textu zadaný řetězec jiným. Chcete-li nahradit určitý text v textovém řetězci, použijte funkci [Replace](#). Funkce **Substitute** se používá k nahrazení libovolného textu, který se nachází na určitém místě v textovém řetězci.

Text je text nebo odkaz na buňku s upravovaným textem

StarýText je textový řetězec, který chcete zaměnit

NovýText je nový textový řetězec

N určuje, který výskyt starého řetězce chcete zaměnit. Jestliže zadáte argument *N* bude nahrazen pouze zadaný výskyt starého řetězce. Jinak budou všechny výskyty starého řetězce nahrazeny řetězcem novým.

Příklady:

Substitute("Datum prodeje"; "prodeje"; "nákupu") rovná se "Datum nákupu"

Substitute("1.čtvrtletí; 1991"; "1"; "2"; 1) rovná se "2.čtvrtletí, 1991"

Substitute("1.čtvrtletí; 1991"; "1"; "2"; 3) rovná se "1.čtvrtletí, 1992"

T(Hodnota)

Vrátí text, na který se odkazuje argument *Hodnota*.

Hodnota převáděná hodnota. Pokud je hodnota text nebo odkaz na něj, vrátí funkce **T** hodnotu. Pokud není hodnota odkaz na text, vrátí funkce **" "** (prázdný textový řetězec).

Příklady:

Jestliže buňka B1 obsahuje text "Vodní srážky":

T(B1) = "Vodní srážky"

Pokud buňka B2 obsahuje číslo 19:

T(B2) = ""

T("TRUE") = "TRUE"

T(TRUE) = ""

Trim(Text)

Odstraní nadbytečné mezery v textu tak, aby byla slova oddělena pouze jednou mezerou. Funkce se používá u textů importovaných z jiných aplikací, které mohou obsahovat velký počet nadbytečných mezer.

Text text, ze kterého chcete odstranit nadbytečné mezery

Příklad:

TRIM(" Výdělek za první čtvrtletí ") = "Výdělek za první čtvrtletí"

Upper(Text)

Funkce převede řetězec znaků na velká písmena. Argument *Text* musí být textový výraz. Výsledek je typu řetězec.

Příklady:UPPER("ahoj") = "AHOJ"

UPPER("používat") = "POUŽÍVAT"

Value(Text)

Funkce konvertuje znakový *Text* reprezentující číslo na číselnou hodnotu.

Informační funkce

Informační funkce listu se používají k určení typu dat uložených v buňce. Informační funkce zahrnují skupinu funkcí listu, které vrátí hodnotu TRUE, pokud buňka splňuje určitou podmínku.

IsBlank(Hodnota)

Funkce vrací TRUE, pokud je argument *výraz* odkazem na prázdnou buňku.

Příklad: Je-li A1 = 10, pak **ISBLANK(A1) = FALSE**

IsErr(Hodnota)

Funkce vrací hodnotu TRUE, pokud je argument *výraz* chybou.

Příklady: A1 = sqrt(10), **iserr(a1) = FALSE**

C5 = sqrt(-10), **iserr(c5) = TRUE**

IsError(Hodnota)

Vrací hodnotu TRUE, je-li argumentem jakýkoli příznak chyby.

IsLogical(Hodnota)

Funkce vrací TRUE, pokud je argumentem logická hodnota.

Příklady: A1 = True, **ISLOGICAL(A1) = TRUE**

ISLOGICAL(False) = TRUE

ISLOGICAL("True") = FALSE

IsNa(Hodnota)

Funkce vrací TRUE je-li argument příznak chyby: **!!chNE**.

IsNontext(Hodnota)

Funkce vrací hodnotu TRUE není-li argumentem text.

IsNumber(Hodnota)

Funkce vrací hodnotu TRUE je-li argumentem číslo.

IsRef(Hodnota)

Funkce vrací hodnotu TRUE jedná-li se o odkaz.

IsText(Hodnota)

Funkce vrací hodnotu TRUE jedná-li se o text.

N(Hodnota)

Vrátí hodnotu převedenou na číslo.

Hodnota převáděná hodnota. Funkce převádí hodnoty podle následující tabulky:

Hodnota	Výsledek
Číslo	Stejně číslo
Datum v některém z předdefinovaných formátů	Pořadové číslo tohoto data

TRUE	1
Cokoli jiného	0

Příklady:

Pokud A1 obsahuje "7", A2 obsahuje "sudá" a A3 obsahuje "TRUE", potom:

N(A1) = 7

N(A2) = 0, protože A2 obsahuje text

N(A3) = 1, protože A3 obsahuje TRUE

N("7") = 0, protože "7" je text

N("4/17/91") = 0, protože "4/17/91" je text

Na()

Vrátí chybovou hodnotu: **!!chNe**

Type(Hodnota)

Vrátí typ hodnoty. Tato funkce se používá, když chování některé funkce závisí na typu hodnoty v určité buňce.

Hodnota může být libovolná hodnota používaná v aplikaci (např. číslo, text, logická hodnota apod.).

<u>Typ hodnoty</u>	<u>Výsledek</u>
Číslo	1
Text	2
Logická hodnota	4
Vzorec	8
Chybová hodnota	16

Poznámky:

Funkce se používá hlavně ve spojení s funkcemi, které mohou mít argumenty různého typu. Pomocí funkce **TYPE** můžete zjistit, jaký typ hodnoty daná funkce vrátí.

Příklady:

Obsahuje-li buňka A1 text "Novák", pak:

TYPE(A1) = TYP("Novák"), čili **2**

TYPE("pan "&A1) = 2

TYPE(2+A1) = TYP (!!chHod) = 16

TYPE({1;2|3;4}) = 64

Vyhledávací funkce

Tyto funkce slouží k výběru hodnot. Upozornění: funkce vrací vždy číselnou hodnotu!

Address (ČísloŘádky;ČísloSloupce;RefTyp;A1;JménoListu)

Vytvoří adresu buňky ve tvaru textu, vrátí čísla určeného řádku a sloupce.

<i>ČísloŘádky</i>	číslo řádku, které je použito v odkazu na buňku
<i>ČísloSloupce</i>	číslo sloupce, které je použito v odkazu na buňku
<i>Typ</i>	určuje typ odkazu, který bude vrácen:

Argument Typ Typ vráceného odkazu

1 nebo vynechán	Absolutní
2	Absolutní řádek; relativní sloupec
3	Relativní řádek; absolutní sloupec
4	Relativní
<i>A1</i>	je logická hodnota, která určuje typ odkazu - A1 nebo R1C1. Má-li argument <i>a1</i> hodnotu TRUE nebo je-li vynechán, vrátí funkce odkaz ve tvaru A1; má-li argument <i>a1</i> hodnotu FALSE, vrátí funkce ADDRESS odkaz ve tvaru R1C1.
<i>JménoListu</i>	je textový řetězec určující název listu, který má být použit jako externí odkaz. Tento argument může být vynechán.

Příklady:

ADDRESS(2;3) = "\$C\$2"
ADDRESS(2;3;2) = "C\$2"
ADDRESS(2;3;2;FALSE) = "R2C[3]"
ADDRESS(2;3;1;FALSE;"[Sesit1]List1") = "[Sesit1]List1!R2C3"

Column(Odkaz)

Vrací číslo sloupce daného odkazu. Pokud je argumentem odkaz na oblast, vrací číslo prvního sloupce oblasti. Není-li uveden žádný argument, funkce vrací sloupcovou pozici ukazatele.

Příklad: **Column(A3)** vrací hodnotu 1

Jestliže je ukazatel nastaven na buňku **C5**, pak zápis **Column()** je ekvivalentní **Column(C5)** a vrací hodnotu 3.

Columns(Odkaz)

Vrátí počet sloupců v matici nebo odkazované oblasti.

Odkaz je matice, maticový vzorec nebo oblast buněk, u kterých chcete zjistit počet sloupců.

Příklady:

SLOUPCE(A1:C4) = 3
SLOUPCE({1;2;3|4;5;6}) = 3

Hlookup(X;rozsah;Y)

Tato funkce slouží k vyhledávání hodnoty ve vodorovném směru. Funkce hledá obsah buňky na pozici vztážené k pozici jiné určené buňky. Při použití funkce je třeba se řídit těmito pokyny:

- Je třeba určit oblast hodnot, které mají být vyhledány.
- Horní řádek oblasti musí být uspořádán zleva doprava, od nejnižší hodnoty k nejvyšší. Žádná hodnota se nesmí opakovat.

- Horní řádek je prohledáván zleva doprava. Jakmile je nalezena hodnota větší než X , je vybrán sloupec umístěný vlevo od buňky s touto hodnotou. Je-li nalezena hodnota X , je vybrán sloupec obsahující tuto hodnotu.
- Funkce vrací hodnotu umístěnou Y řádků pod vybranou buňkou. Y musí adresovat buňku ve zvolené oblasti.

Příklad:

V buňkách **A1** až **D4** jsou uložena čísla:

	A	B	C	D
1	1	2	3	4
2	0	1	0	1
3	2	0	2	0
4	0	3	0	3

V této situaci:

HLOOKUP(3; A1:D4;3) = 0

HLOOKUP(2,3; A1:D4;3) = 3

Lookup(Hodnota;Vektor1;Vektor2)

Nalezne hodnotu zadanou parametrem *Hodnota* v řádku nebo sloupci (*Vektor1*) a vrátí příslušnou hodnotu z řádku nebo sloupce *Vektor2*.

<i>Hodnota</i>	je hodnota, kterou má funkce Lookup nalézt v prvním vektoru <i>Vektor1</i> . Argument co může být číslo, text, logická hodnota, název nebo odkaz na hodnotu.
<i>Vektor1</i>	je oblast obsahující jeden řádek nebo sloupec. Hodnoty tohoto argumentu mohou obsahovat text, čísla nebo logické hodnoty. Upozornění: Hodnoty argumentu hledat musí být seřazeny vzestupně: ..., -2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE; jinak by funkce mohla vrátit nesprávnou hodnotu. Při vyhledávání se nebere ohled na malá a velká písmena.
<i>Vektor2</i>	je oblast obsahující jeden řádek nebo sloupec. Měla by mít stejný rozměr jako prohledávaný vektor (<i>Vektor1</i>).

Nenajde-li funkce *Hodnotu*, použije největší hodnotu z prohledávaného vektoru (*Vektor1*), která je menší nebo rovna hledané hodnotě.

Je-li hodnota argumentu *Hodnota* menší než nejmenší hodnota z prohledávaného vektoru (*Vektor1*), funkce vrátí chybovou hodnotu.

Match(Hodnota;Odkaz;Typ)

Funkce vrací relativní pozici hodnoty *Hodnota* v zadané oblasti *Odkaz*. Používá místo funkce **Lookup** v případech, kdy není potřeba získat hledaný prvek, ale jeho pozici.

<i>Hodnota</i>	je hodnota, pomocí které hledáte v tabulce požadovanou hodnotu. Je to hodnota, kterou chcete nalézt v oblasti <i>Odkaz</i> . Je to podobné, jako když hledáte v telefonním seznamu číslo určitého člověka. Nehledáte požadované číslo, ale jméno. Argument <i>Hodnota</i> může být hodnota (číslo, text nebo logická hodnota) nebo odkaz na buňku s číslem, textem nebo logickou hodnotou.
<i>Odkaz</i>	je odkaz na souvislou oblast buněk, v níž chcete nalézt hledané hodnoty.
<i>Typ</i>	je číslo -1, 0 nebo 1. Určuje, jakým způsobem má 602Tab porovnávat hledanou hodnotu s hodnotami v prohledávané matici.

- Pokud *Typ* = 1, najde funkce největší hodnotu, která je menší nebo rovna hledané hodnotě (*Hodnota*). Hodnoty argumentu prohledat musí být přitom seřazeny vzestupně: ..., -2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE.
- Pokud *Typ* = 0, najde funkce první hodnotu, která se přesně shoduje s hledanou hodnotou (*Hodnota*). Hodnoty argumentu prohledat přitom nemusí být nijak seřazeny.

- Pokud *Typ* = -1, najde funkce nejmenší hodnotu, která je větší nebo rovna hledané hodnotě (*Hodnota*). Hodnoty argumentu prohledat přitom musí být seřazeny sestupně: TRUE, FALSE, Z-A,...2, 1, 0, -1, -2,... atd.
- Pokud argument *Typ* nezadáte, implicitně se použije *Typ* = 1.

Poznámky:

- Funkce vrátí pozici nalezené hodnoty v prohledávané oblasti, nikoliv hodnotu samotnou.
- Funkce nerozlišuje při porovnávání textových hodnot malá a velká písmena.
- Nenajde-li funkce žádnou vyhovující hodnotu, vrátí chybovou hodnotu.
- Je-li *Typ* = 0 a argument *Hodnota* je text, můžete v argumentu *Hodnota* použít zástupné znaky hvězdička (*) a otazník (?). Hvězdička zastupuje libovolnou posloupnost znaků, otazník zastupuje jeden libovolný znak.

Offset(Odkaz;Řádky;Sloupce;Výška;Šířka)

Vrátí odkaz na oblast, která obsahuje určený počet řádků a sloupců, od určité buňky nebo oblasti buněk. Vracený odkaz může být jedna buňka nebo oblast buněk. Počet řádků a sloupců, které se mají vrátit, můžete určit.

<i>Odkaz</i>	je původní odkaz, vůči kterému provádíte posun. Pokud odkaz není tvořen odkazem na buňku nebo oblast sousedících buněk, vrátí funkce chybovou hodnotu !!chHod .
<i>Řádky</i>	je počet řádků, o které se má posunout levá horní buňka nového odkazu (nahoru nebo dolů). Zadáte-li například číslo 5, levá horní buňka odkazu bude pět řádků pod levou horní buňkou původního odkazu. Můžete použít kladnou (dolů od původního odkazu) nebo zápornou hodnotu (nahoru od původního odkazu).
<i>Sloupce</i>	je počet sloupců vlevo nebo vpravo, o které se má posunout levá horní buňka výsledného odkazu vzhledem k původnímu odkazu. Zadáte-li například číslo 5, bude levá horní buňka odkazu o pět sloupců vpravo od levé horní buňky původního odkazu. Můžete použít kladnou (posun doprava od původního odkazu) i zápornou hodnotu (posun doleva od původního odkazu). Přesáhne-li takto posunutý odkaz okraje listu, vrátí funkce Offset chybovou zprávu.
<i>Výška</i>	je požadovaná výška (počet řádků) výsledného odkazu. Výška je vždy kladné číslo.
<i>Šířka</i>	je požadovaná šířka (počet sloupců) výsledného odkazu. Šířka je vždy kladné číslo.

Vynecháte-li argument *Výška* nebo *Šířka*, implicitně se dosadí výška nebo šířka původního odkazu.

Poznámky:

Funkce ve skutečnosti žádné buňky nepřesunuje ani nemění označenou oblast; pouze vrátí hodnotu typu odkaz. Funkci lze použít ve spojení s libovolnou funkcí, která očekává argument typu odkaz.

Například pomocí vzorce **SUM(POSUN(C2;1;2;3;1))** se vypočítá celková hodnota oblasti o třech řádcích a jednom sloupci, která je umístěna 1 řádek pod a 2 sloupce vpravo vzhledem k buňce C2.

Příklady:

OFFSET(C3;2;3;1;1) = F5.

Zadáte-li tento vzorec na listu, zobrazí aplikace hodnotu uloženou v buňce F5.

OFFSET(C3:E5;-1;0;3;3) = C2:E4

OFFSET(C3:E5;0;-3;3;3) = #REF!

Row(Odkaz)

Vrací číslo řádku uvedeného v argumentu *Odkaz*. Argument *Odkaz* musí být odkaz na buňku. Argument může být vynechán.

Příklady:ROW = 1

ROW(A1) = 1

ROW(B4) = 4

Rows(Odkaz)

Vrátí počet řádků daného odkazu nebo matice.

Odkaz je matice, maticový vzorec nebo odkaz na oblast, jejíž počet řádků chcete zjistit

Příklady:

ROWS(A1:C4) = 4

ROWS({1;2;3|4;5;6}) = 2

Vlookup(X;rozsah;Y)

Funkce pro vyhledávání ve svislém směru se podobá funkci **Hlookup**; první sloupec bloku je uspořádán shora dolů, od nejnižší do nejvyšší hodnoty. Sloupec je prohledáván shora dolů. Jakmile je nalezena hodnota větší než *X*, je vybrán řádek umístěný nad buňkou s touto hodnotou. Je-li nalezena hodnota *X*, je vybrán řádek, který ji obsahuje. Funkce vrací hodnotu umístěnou *Y* sloupců vpravo od vybrané buňky. *Y* musí adresovat buňku ve zvolené oblasti.