



Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta
Katedra matematiky

Diplomová práce

Interaktivní aplikace v programu MAPLE pro výuku finanční matematiky

Vypracoval: Bc. Přemysl Rosa
Vedoucí práce: RNDr. Vladimíra Petrášková, PhD.

České Budějovice 2015

Prohlášení

Prohlašuji, že svoji diplomovou na téma Interaktivní aplikace v programu MAPLE pro výuku finanční matematiky jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě, elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích

Anotace

Tato diplomová práce se zabývá možnostmi využití programu Maple při výuce finanční matematiky. Stěžejní je vytvořený maplet, který umožňuje uživatelům využít sílu CAS (systémy počítačové algebry) od divize Maplesoft firmy Waterloo Inc., prostřednictvím jednoduchého grafického uživatelského rozhraní. Maplety poskytují veškeré nástroje pro výpočty, jsou velice názorné a pomáhají studentům lépe pochopit danou problematiku. Práce se také podrobně zabývá maplety od jejich vývoje přes možnosti až po tvorbu.

Abstract

The subject of this diploma thesis is to search for the possibilities of using the Maple program for teaching financial mathematics. The basis is to create the maplet, which allows to users to fully benefit from the the strength of CAS (computer algebra system) created by the division Maplesoft of the company Waterloo Inc., through a simple graphical user interface. Maplets provide all the necessary tools for the calculations, they are very illustrative and help students to better understand certain problem. Detailed evolution of the maplets from its development through possibilities to its creation is also important part of this diploma thesis.

Poděkování

Rád bych poděkoval RNDr. Vladimíře Petráškové, Ph.D a Mgr. Romanovi Haškovi, Ph.D za velmi cenné rady, připomínky a pomoc, kterou mi poskytli při vypracování diplomové práce.

Děkuji

Obsah

Úvod	6
1 Historie programu Maple	7
2 Maplety	11
2.1 Historie mapletů	11
2.2 Rozvržení mapletu	12
2.3 Použitelné prvky	19
2.4 Tvorba mapletů	29
3 Vytvořený maplet	32
3.1 Dluhopisy	33
3.2 Klasické spoření	34
3.3 Penzijní připojištění	35
3.4 RPSN	36
3.5 Spotřebitelský úvěr	37
3.6 Stavební spoření	38
4 Závěr	39
Literatura	40
Seznam příloh	42

Úvod

Tato diplomová práce navazuje na bakalářskou práci Interaktivní aplikace pro výuku finanční matematiky [1], kterou podstatně rozšiřuje. U bakalářské práce byly stěžejní částí práce Smart dokumenty, které byly zaměřeny na grafickou podobu a interaktivnost pracovních listů CAS. Stejně tak v diplomové práci jsou stěžejní maplety, které umožňují uživatelům pomocí grafického uživatelského rozhraní využít CAS k výpočtům a pochopení finanční matematiky, která otevírá dveře rozšíření finanční gramotnosti mezi občany.

Maplety jsou aplikace poskytující vlastní uživatelské rozhraní nad programem Maple, jehož plnou funkcionalitu mohou uživatelé využívat, aniž by měli povědomí o práci se samotným programem.

V první kapitole je v zájmu zachování kontinuity textu použita z bakalářské práce historie firmy Maple a vývoj programu. Maple od divize Maplesoft kanadské firmy Waterloo Maple Inc. je nepochybně jedním z nejznámějších CAS. Cílem této práce je prezentovat jeho možnosti ve formě mapletů, jako podpůrného prostředku při výuce finanční matematiky. Výpočty použité v mapletu vycházejí z matematického aparátu rozvedeného v bakalářské práci. Zároveň tato diplomová práce poskytuje detailní návod potřebný pro tvorbu mapletů. Návod však nenastiňuje pouze způsob tvorby použitý autorem, ale představuje komplexní návod k používání jednotlivých prvků i k možnostem tvorby.

Hlavním vodítkem při počátečním výzkumu a následném zpracování byla nápověda programu Maple a to přímo v programu nebo na webových stránkách. Bohužel neexistuje mnoho relevantních zdrojů, ze kterých by bylo možné čerpat.

Maplety jsou už ze své podstaty na platformě nezávislé, což je jejich nespornou výhodou, neboť je možné je spustit v libovolném operačním systému, pro který existuje distribuce programu Maple.

Výsledný maplet je vytvořen v prostředí programu Maple 13, ale použité prvky garantují zpětnou kompatibilitu minimálně po verzi 9.5.

1 Historie programu Maple

”Za vznikem programu Maple stojí kanadská společnost Waterloo Maple Inc., širokou veřejností známá pod názvem Maplesoft. Společnost byla poprvé registrována jako Waterloo Maple Software v dubnu roku 1988 Keithem Geddesem a Gastonem Gonnetem, kteří byli profesory na katedře počítačové vědy na University of Waterloo v Kanadě. Základem pro tvorbu softwaru se stalo intelektuální vlastnictví vyvinuté z rozsáhlého vědeckého projektu, který proběhl na University of Waterloo začátkem osmdesátých let. V srpnu roku 2009 firma oznámila, že podepsala definitivní dohodu o akvizici společností Cybernet Systems Co., Ltd., která je předním japonským importérem/distributorem CAE (počítačem podporované inženýrství) softwaru. K akvizici došlo v září téhož roku a společnost Maplesoft nadále působí jako samostatná firma, aniž by byla ohrožena péče o zákazníky, či vývoj dalšího softwaru.

Hlavními produkty na kterých společnost staví jsou Maple a MapleSim. Maple je systém počítačové algebry využívající převážně symbolické výpočty, neboli výpočty, při kterých získáváme výsledek úpravou matematických rovnic. Využití Maple nachází ve vzdělávání a využívání matematiky v přírodovědných oborech. Mladší MapleSim, je oproti tomu zaměřený spíše na praktickou činnost. Slouží hlavně k modelování a simulaci problémů z přírodovědných a technických oborů. Oba programy se vyznačují kvalitním zpracováním, vysokým výkonem a všestranností. Výpočetní engine použitý jako základ obou programů je považován za nejvyspělejší symbolický engine na světě. Díky tomu jsou produkty společnosti Maplesoft využívány většinou velkými společnostmi, které kladou velký důraz na kvalitu a přesnost. Mezi tyto firmy se řadí Ford, Toyota, Hewlett Packard, Motorola, Sun Microsystems a mnoho dalších. Zájem je i ze strany univerzit a výzkumných laboratoří, kde za zmínku stojí například MIT, Oxford, Stanford nebo Laboratoř proudového pohonu NASA. 20. dubna 2013 společnost oslavila 25 let od svého založení.

Předchůdci programu Maple, mezi které patří Macsyma, Reduce a další, byly systémy počítačové algebry napsané v programovacím jazyce LISP, který v té době patřil mezi nejrozšířenější. Programy napsané v tomto jazyce byly náročné jak na operační paměť, tak i na výkon procesoru. To se stalo v době jejich největšího rozmachu, v sedmdesátých letech minulého století, omezujícím faktorem, který umožňoval jejich využití pouze v laboratořích majících přístup k sálovým počítačům. Kvůli tomu byla skupina potenciálních uživatelů zredukována na minimum. S přihlédnutím k faktu, aby byl výpočet časově efektivní nesměly tyto počítače sdílet svůj výkon mezi příliš mnoha uživateli, zbývá jen zlomek výzkumníků, kteří mohli potenciál těchto programů naplno využít.

Za otce programu Maple se považují Keith Geddes a Gaston Gonnet, toho času profesori na University of Waterloo. Jejich užší spolupráce začala v červnu roku 1980, poté co zjistili, že jejich výzkumná činnost se zabývá podobnými zájmy. Již v listopadu téhož roku se zrodila myšlenka vytvořit systém počítačové algebry, který by netrpěl nedostatky tehdejších systémů. Jejich cílem bylo navrhnout systém, který by byl přístupný širší vědecké obci, ale i většímu počtu studentů pro jejich vzdělávací potřeby. Hlavní myšlenkou vývoje bylo kladení důrazu na časovou i prostorovou, ve smyslu paměti počítače, nenáročnost výsledného programu. Aby splnili toto kritérium rozhodli se použít programovací jazyk B, který vychází z jazyka BCPL. Jeho zásadními výhodami jsou, oproti jazyku LISP, jednodušší implementace a menší překladač. Záhy se však rozhodli využít jazyk C, který byl podporován více operačními systémy. Díky této změně navíc docílili toho, že vznikající program Maple byl snáze portovatelný na mikropočítače objevující se na tehdejšímu trhu a tím pádem se stal přístupným pro větší počet lidí.

Jelikož měl tým pracující na projektu Maple již připravené návrhy budoucího programu, přešlo se rovnou na jejich implementaci. Už v prosinci roku 1980 byl na světě fungující program Maple, ač s omezenými možnostmi. V lednu následujícího roku byl již začleněn do výuky pro studenty čtvrtých ročníků předmětu Úvod do symbolických výpočtů na University of Waterloo. V roce 1982 začalo program pro svůj výzkum využívat několik matematiků působících na univerzitě. V tom samém roce se program začal dostávat do povědomí i dalším matematikům, kteří měli možnost shlédnout demonstraci programu na různých konferencích. To přispělo k tomu, že roku 1983 našel program využití v různých institucích zabývajících se jak výzkumem, tak i výukou. Koncem roku již existovalo mimo univerzitu padesát instalací.

S narůstajícím počtem instalací, začaly díky zpětné vazbě, narůstat i dotazy na výzkumný tým ohledně portování na další operační systémy, podpory stávajících instalací a dostupnosti programu. Stalo se zřejmým, že problém distribuce programu Maple musí tým přenechat někomu schopnějšímu. To vedlo k tomu, že v roce 1984 výzkumný tým uzavřel dohodu se společností WATCOM Products Inc. týkající se licencování a distribuce programů Maple. Výsledkem dohody byl Maple 3.3, který se 31. července 1984 stal první komerčně dostupnou verzí.

Aby však mohli být další verze distribuovány jako plnohodnotné programy, museli výzkumníci učinit několik významných změn. Program musel být naportován ze svého unixového základu na systémy VM/CMS společnosti IBM a DEC VMS počítačů Alpha, celkové matematické možnosti systému musely být v určitých oblastech zlepšeny a existující referenční příručka musela být celá přepracována, aby mohla být použita jako uživatelská příručka.

Práce na vylepšení programu podle vytyčených požadavků trvala přes rok, čímž se narušila strategie vydávání nové verze jednou ročně. Maple 4.0 byl uveden na trh 15. dubna 1986, a přinesl sebou řadu nových funkcí včetně `solve`, `simplify` a `fsolve`, které dnes neodmyslitelně patří k běžné práci s rovnicemi.

Roku 1987 již existovalo přibližně 300 instalací po celém světě a poptávka po programu Maple dále rostla. S rostoucím počtem majitelů licence však začala být vyžadována i zákaznická podpora na vysoké úrovni, což vedlo Keitha Geddesa a Gastona Gonneta k tomu, že 20. dubna 1988 založili společnost Waterloo Maple Software. Společnost rostla a jednak byla schopna uspokojit požadavky zákazníků a jednak se mohla zabývat dalším vývojem programu.

Verze Maple V z 8. února 1990 přinesla další důležitou změnu. Do té doby byl Maple primárně zaměřen na operační systémy Macintosh a stárnoucí MS-DOS, až verzi Maple V bylo možné regulérně spustit pod operačním systémem Microsoft Windows. Tato verze byla celkově revoluční, protože její další vydání přinesla mnoho nástrojů, které mají uplatnění dodnes. Maple V Release 2 z roku 1992 poskytl první prostředí pro tvorbu pracovních listů, neboli dokumentů, které mohou obsahovat text, výpočty a grafy dohromady. Release 3 v roce 1994 přinesl nástroj pro uživatele, který jim umožňuje tvorbu vlastních balíčků a knihoven. Release 4 z roku 1996 obsahoval debugger, jenž uživatelům programujícím v Maple umožňoval kontrolovat chyby. Release 5 vydaný roku 1998 obsahoval první nástroje Clickable Math, které přinesly revoluci v oládání Maple. Tyto nástroje umožňují nastavení prvků jako jsou grafy v nabídce, jež získáme kliknutím na upravovaný graf. Dále umožňují jednotlivé komponenty, například kostry matematických výrazů, vybírat myší z palet, či je slučovat například přetažením výrazu do grafu. Dalším nově představeným nástrojem je možnost exportovat pracovní listy do HTML nebo naopak HTML dokumenty vkládat do pracovních listů.

Další verze přinášely pro uživatele nepostřehnutelné změny ve výpočetním enginu, vylepšení nástrojů Clickable Math, či propojení s dalšími matematickými platformami jako MATLAB, Microsoft Excel nebo CAD. Některé verze, ale přesto představily novinky, které stojí za zmínku. Ve verzi 8 bylo poprvé možné tvořit maplety, což jsou interaktivní aplikace s vlastním grafickým rozhraním, které si uživatelé mohou vytvořit pro řešení konkrétních případů. Verze 9 představila změnu v grafickém rozhraní určeném pro tvorbu pracovních listů, které bylo nově z větší části napsané v programovacím jazyce Java. Toto rozhraní zůstalo bez větších změn zachováno dodnes. Režim dokumentu představený ve verzi Maple 10 přidává nové rozhraní, které je při spuštění programu nastaveno jako výchozí. Původní režim pracovních listů, představený ve verzi 5, je dodnes spustitelný.

Uživatel má tedy na výběr, jestli mu pro jeho potřeby dostačuje režim pracovních listů, nebo potřebuje k řešení příkladu i vizuální úpravy poskytované v režimu dokumentu. Hlavní výhodou režimu dokumentu je přímý převod matematických výrazů do přirozené sazbové podoby. Tato novinka spolu s paletami programovatelných prvků a úpravami vzhledu textu umožňuje tvorbu smart dokumentů.” (Rosa [1], s. 15-18)

Zatím poslední vydanou verzí je Maple 2015, který byl představen 4. 3. 2015. Společnost Maplesoft neohlásila konec vývoje tohoto softwaru, dají se tedy očekávat jeho další verze.([2])

2 Maplety

Maplety jsou aplikace poskytující vlastní uživatelské rozhraní nad programem Maple, jehož plnou funkcionalitu mohou uživatelé využívat, aniž by měli povědomí o práci se samotným programem. Ovládání vytvořených aplikací je zaměřené na přístup point-and-click. Ve většině případů se maplety vytvářejí pro řešení specifických problémů, které vyžadují možnost dynamické změny vstupních parametrů, a není tedy vhodné použít statický pracovní list. ([3], 455)

Maplety lze standardně spustit třemi způsoby. Prvním způsobem je spuštění přímo z pracovního listu, kde je umístěn zdrojový kód a spouštěcí příkaz. Po provedení příkazu se uživateli otevře nové okno, které již obsahuje jen prostředí naprogramované v mapletu. Druhou možností spuštění je otevření souboru s příponou *.maplet* ze souborového manažeru. Soubor je však nutné napřed vytvořit exportováním pracovního listu, ve kterém je obsažen zdrojový kód mapletu a spouštěcí příkaz, z programu Maple. Vytvořený soubor se otevře v programu Maplet Viewer, který je součástí instalace programu Maple. Poslední nejméně běžnou možností je spuštění z příkazového řádku. Použit lze Command-Line verzi programu Maple, ve které lze maplet vytvořit a spustit, nebo je možné vytvořený pracovní list se zdrojovým kódem uložit s příponou *.mpl*, a tento soubor použít jako argument příkazu *maple* v příkazovém řádku operačního systému.

2.1 Historie mapletů

Maplety byly poprvé představeny v prosinci 2001. V té době se ještě nedjednalo o plnohodnotnou součást programu, ale o stáhnutelný doplněk ve formě samostatného balíčku, ze kterých je Maple tvořen. Jednotlivé příkazy balíčku byly napsány v programovacím jazyku Maple, takže na uživatele zvyklé na práci s ním nebyl kladen požadavek znalosti dalšího programovacího jazyka, aby si mohli tvořit vlastní uživatelské rozhraní, jak tomu bylo doposud. Doplněk bylo možné použít jen s tehdy nejnovější verzí programu, kterou byl Maple 7. ([4], [5])

16. května 2002 byla na trh uvedena nová verze programu s pořadovým číslem 8, která už měla původní balíček mapletů zahrnutý v instalaci. Ten ale prošel řadou změn. Největší změnou prošli metody *Get*, *Set* a *SetOption*. Dříve bylo nutné veškeré argumenty těchto metod, včetně booleovských hodnot *true* a *false*, převést na datový typ řetězec, což značně zatěžovalo kód i jeho tvůrce. Od této verze to již není nutné a metody umí zpracovat argumenty v symbolické podobě. Další významnou změnou je přidání nových prvků *CheckBoxMenuItem* a *RadioButtonMenuItem* rozšiřujících možnosti návrhu nabídek v hlavní liště mapletu a prvku *MathMLEditor* pro zápis

matematických výrazů. U dalších prvků také došlo ke změnám, ale nejednalo se o změny zásadně pozměňující jejich funkčnost. S Maple 8 byly navíc představeny dva programy které ovlivnily maplety. Prvním je MapletViewer, díky kterému již není nutné spouštět maplety z pracovního listu a druhým je program MapleNet. Tento program, který není součástí programu Maple, umožňuje publikaci v něm vytvořených materiálů (maplety, smart dokumenty aj.) na serverech společnosti Maple.([6], [7], [8], [9])

Další verze programu Maple již nepřinesly žádné citelné změny, co se funkčnosti jednotlivých prvků balíčku mapletů týče. Za zmínku stojí jen dva nové prvky představené ve verzi Maple 9.5 a to HorizontalGlue a VerticalGlue, které se mohou použít pro zlepšení vizuálního rozvržení prvků v okně mapletu. Ve verzi Maple 10 byl představen asistent Maplet Builder sloužící k tvorbě mapletů způsobem drag-and-drop (přetáhni a pusť) v samostatném okně. Od této verze se balíček už nezměnil a funguje shodně i v posledních verzích programu Maple.([8],[10])

2.2 Rozvržení mapletu

Před začátkem programování mapletů je vhodné seznámit se s možnostmi vizuálního rozvržení mapletu a s prvky použitelnými pro jejich tvorbu.

Z pohledu rozvržení má tvůrce čtyři možnosti, přičemž každou je možné dále zkombinovat s jinou, aby bylo docíleno požadovaného vizuálního efektu. Volba rozvržení nemá vliv na výpočty probíhající na pozadí, ale díky vhodné zvolenému rozvržení může tvůrce docílit větší přehlednosti. Tak příjemní práci uživatelům, kteří budou výsledný maplet používat. V této podkapitole je navíc za každým představeným způsobem vyobrazen maplet včetně zdrojového kódu, aby bylo možné srovnání jednotlivých variant.

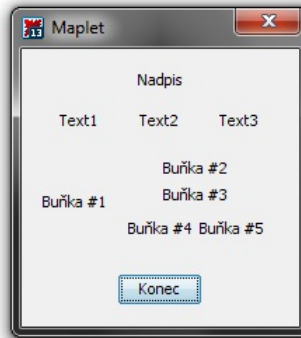
První možností, jak rozvrhnout obsah mapletu, je pomocí vnořených seznamů. Jedná se o velice jednoduchý způsob, při kterém se celý maplet vytvoří jako jeden seznam prvků, přičemž prvky se dají nahradit vnořenými seznamy. Tyto seznamy mohou obsahovat další prvky nebo opět seznamy, přičemž obsah celého mapletu je automaticky zarovnáván na střed. Každé vnoření seznamu mění způsob zarovnání jeho obsahu, a to do řádku nebo do sloupce. Příklad takového rozvržení včetně mapletu z něj vygenerovaného je uveden na obrázku 1.

Jak je patrné, lze tímto způsobem vytvořit kostry, které jsou použitelné pro celou škálu aplikací. Problém může nastat při tvorbě větších mapletů, kde postupné vnořování generuje velké množství závorek a tvůrce tak může ztratit přehled o úrovni vnoření, což může komplikovat jejich samotnou tvorbu. Tento problém je možné odstranit pomocí Layout prvků rozdělených

```

> restart,
> with(Maplets[Elements]) :
> maplet := Maplet([
    ["Nadpis"], #1.řádek
    ["Text1", "Text2", "Text3"], #2.řádek
    ["Buňka #1", #3.řádek 1.sloupec
     ["Buňka #2", "Buňka #3", #3.řádek 2.sloupec
     ["Buňka #4", "Buňka #5"]], #3.řádek 2.sloupec 3. řádek
    [Button("Konec", Shutdown())] #4.řádek
]):
> Maplets[Display](maplet);

```



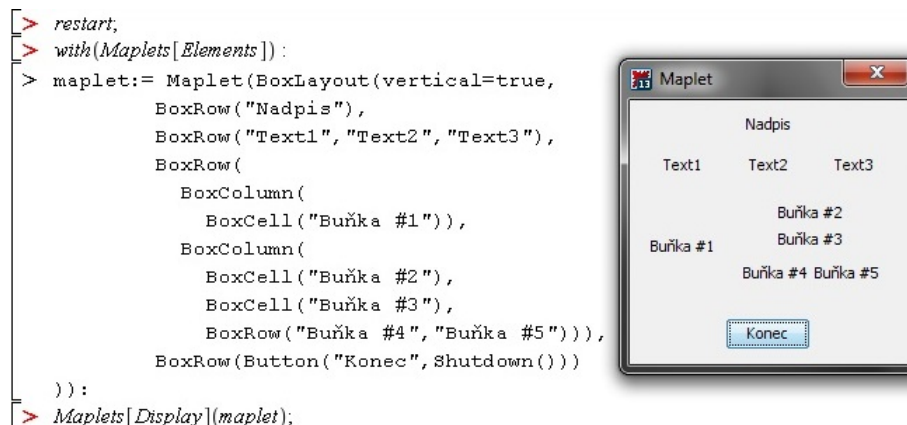
Obrázek 1: Rozvržení mapletu pomocí vnořených seznamů

do tří logických celků vždy s jedním nadřazeným prvkem. Ty poskytují zbylé možnosti rozvržení a lze je zkombinovat jak s vnořenými seznamy, ve kterých nahrazují prvky seznamu, nebo je použít jako výchozí způsob rozvržení mapletu.

Prvním nadřazeným prvkem ze skupiny Layout je `BoxLayout`. Rozvržení pomocí tohoto prvku je obdobou rozvržení pomocí vnořených seznamů, včetně automatického zarovnávání obsahu na střed. Rozdíl mezi těmito způsoby spočívá v tom, že postupné vnořování používané pro docílení zarovnání prvků do sloupce nebo do řádku je nahrazeno jedinečnými prvky `BoxRow`, `BoxColumn` a `BoxCell`. `BoxRow`, jak napovídá název, se používá pro zarovnání prvků v něm obsažených do řádku a je možné ho použít i k zapouzdření výše uvedených prvků. Pouze `BoxRow` může obsahovat i speciální prvek `HorizontalGlue`. Jeho funkce spočívá v tom, že se do něj promítají mezery mezi argumenty vzniklé různou šířkou řádků v rozložení. Díky tomu je možné jednotlivé zobrazované argumenty v řádku odsadit případně sesadit k sobě, podle umístění prvků `HorizontalGlue`. `BoxColumn` oproti tomu zarovnává svůj obsah do sloupce a i tento prvek je možné použít k zapouzdření. Jako ekvivalent k `HorizontalGlue` je možné pouze v `BoxColumn` použít k vertikálnímu odsazení prvky `VerticalGlue`. Poslední prvek `BoxCell` je možné za určitých podmínek použít místo prvku `BoxRow`. Nejčastěji se používá jako argument obou výše uvedených prvků `BoxRow` a `BoxColumn` s tím rozdílem, že ho není možné použít k jejich zapouzdření, a jeho argumentem nemůže

být seznam prvků, protože tento prvek zobrazuje ve výsledném mapletu jen jeden prvek. Výhodou `BoxCell` je však to, že může sloužit k vložení koster rozvržení vytvořených pomocí všech nadřazených prvků ze skupiny `Layout`.

Rozvržení pomocí `BoxLayout` standardně zarovnává vložené `Layout` prvky do řádku. Možným řešením je vložení všech prvků mapletu do seznamu, ale existuje i jednodušší řešení pomocí možnosti vertikálního prvku `BoxLayout`. Při nastavení možnosti `vertical=true` se celý maplet začne automaticky zarovnávat do sloupce. Tato řešení umožňují tvůrci pracovat s rozvržením v logické podobě, na kterou je zvyklý z běžného života. Následné rozšiřování rozvržení probíhá vertikálně směrem dolů pomocí prvků `BoxRow`. Maplet používající `BoxLayout` je včetně jeho zdrojového kódu vyobrazen na obrázku 2.



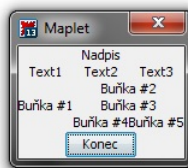
Obrázek 2: Rozvržení mapletu pomocí `BoxLayout`

Dalším nadřazeným prvkem ze skupiny `Layout` je `GridLayout`. Tento prvek jako jediný automaticky zarovnává obsah vlevo a přimyká v sobě obsažené prvky k vnitřní mřížce tabulky. Velikost tabulky je definována počtem řádků a největším počtem sloupců obsažených v jednom řádku. Rozvržení do sloupců a řádků pomocí vnořených seznamů je u `GridLayout` nahrazeno prvky `GridRow` a `GridCell` a je určeno složitějšími pravidly. `GridRow` člení rozvržení na řádky a může být obsažen jen v prvku `GridLayout`. Obsahem `GridRow` mohou být pouze prvky okna nebo prvky `GridCell`. `GridCell` rozděluje prvky řádků do sloupců a může být vložen do rozvržení pouze jako argument `GridRow`. Sám, stejně jako `BoxCell`, může obsahovat jen jeden prvek okna nebo může sloužit k vložení koster, vytvořených na bázi prvků `BoxLayout` nebo `GridLayout`.

Podobně jako měl svá specifika `BoxLayout` i `GridLayout` je možné nastavit tak, aby tvůrce mapletu mohl lépe pracovat s tímto rozložením. Vzhledem k tomu, že tento prvek zarovnává svůj obsah k mřížce a navíc striktně vlevo, mohlo by se zdát, že zarovnání obsahu může být problémem. Existuje složitě

řešení, kdy lze pomocí vkládání prázdných buněk docílit požadovaného odsazení, ale toto řešení je použitelné nanejvýš u malých aplikací, a i v takovém případě nebude programování jednoduché a přehledné. Z tohoto pohledu je možno docílit lepších výsledků užitím možností prvku `GridCell`, jmenovitě `width` a `height`. Pomocí těchto možností lze jednotlivým buňkám `GridCell` změnit jejich velikost. `Width` nastavuje buňkám jejich šířku, která je ve výchozím nastavení rovna jedné. Změnou této velikosti lze docílit toho, že obsah buňky se automaticky zarovná na střed a její šířka se zvětší. Je možné si toto zvětšení představit tak, že došlo ke sloučení teoretických prázdných buněk v řádku s buňkou s obsahem. Oproti tomu `height` nastavuje buňkám jejich výšku. Změnou hodnoty z výchozí 1 se dosáhne vertikálního zarovnání na střed a zvětšení buňky. Opět je možné si toto zvětšení představit jako sloučení s prázdnými buňkami, ale tentokrát ve sloupci. Maplet rozvržený pomocí `GridLayout` včetně jeho zdrojového kódu je prezentován na obrázku 3.

```
> restart;
> with(Maplets[Elements]):
> maplet := Maplet(GridLayout(
  GridRow(
    GridCell('width=3', "Nadpis"),
  GridRow(
    GridCell("Text1"),
    GridCell("Text2"),
    GridCell("Text3"),
  GridRow(
    GridCell('height=3', "Buňka #1"),
    GridCell('width=2', "Buňka #2"),
  GridRow(
    GridCell('width=2', "Buňka #3"),
  GridRow(
    GridCell("Buňka #4"),
    GridCell("Buňka #5"),
  GridRow(
    GridCell('width=3', Button("Konec", Shutdown())))
)):
> Maplets[Display](maplet);
```

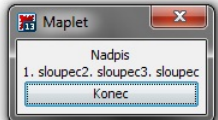


Obrázek 3: Rozvržení mapletu pomocí `GridLayout`

Poslední možností rozvržení je použití prvku `BorderLayout`, který je zároveň posledním nadřazeným prvkem skupiny `Layout`. Rozvržení prvkem `BorderLayout` není typickým rozvržením, protože existuje jeho maximální podoba tvořená pěti prvky, které lze v případě potřeby odebrat, ale žádné se nad rámec přidat nedají. V takové podobě je rozvržení tvořeno dvěma řádky, mezi kterými jsou vloženy tři sloupce. Tyto sloupce se navíc chovají specifickým způsobem, neboť levý a pravý sloupec se těsně přimykají k hraně okna mapletu a obklopujícím řádkům. Vypuštěním sloupce mezi nimi nedojde k jejich přiblížení, ale vznikne mezi nimi mezera. Obsah všech buněk vyplňuje jejich celou velikost, což je nejvíc znatelné, při použití tlačítek. Tvorba rozvržení na bázi `BorderLayout` probíhá pomocí prvků `GridCell2`,

jimž je přidělena jejich unikátní pozice na základě nastavení argumentu `constraint`. Ten může nabývat hodnot `north` (sever) nebo `south` (jih) definujících, že se jedná o horní nebo spodní řádek, dále pak `west` (západ), `east` (východ) nebo `center` (střed), které určují že se jedná o jeden ze tří sloupců. `GridCell2` může obsahovat pouze jeden prvek okna nebo kostru založenou na prvcích `BoxLayout` nebo `GridLayout`. Ukázka maximální kostry i se zdrojovým kódem je na obrázku 4.

```
> restart,
> with(Maplets[Elements]):
> maplet := Maplet(BorderLayout(
    GridCell2(constraint=north, "Nadpis"),
    GridCell2(constraint=south, Button("Konec", Shutdown())),
    GridCell2(constraint=west, "1. sloupec"),
    GridCell2(constraint=center, "2. sloupec"),
    GridCell2(constraint=east, "3. sloupec")
)):
> Maplets[Display](maplet);
```



Obrázek 4: Rozvržení mapletu pomocí `BoxLayout`

Na rozdíl od rozvržení s `GridLayout`, kde je tloušťka vzniklé mřížky pevně dána, je možné od sebe jednotlivé prvky v rozvržení pomocí `BoxLayout` odsadit. Lze toho docílit použitím možností prvku `BoxLayout` `vgap` a `hgap`. Možnost `vgap` určuje vertikální odsazení mezi řádky a sloupci, oproti tomu `hgap` určuje horizontální odsazení mezi jednotlivými sloupci. Hodnotou obou možností může být pouze kladné celé číslo. I přesto je však evidentní, že `Border Layout` je použitelný jen na minimum aplikací. Jeho hlavní přednost však spočívá ve snadném spojení několika typů rozložení v jednom mapletu díky přednastavenému rozdělení na sloupce a řádky.

Obecně je propojování různých způsobů rozvržení při tvorbě mapletů často užívanou metodou, protože každý způsob má svá specifika, a na některé části mapletů je výhodnější užít jiné rozložení než to, které si tvůrce zvolí jako výchozí. Propojení jednotlivých koster probíhá jejich vkládáním do prvků rozvržení, jejichž argumentem může být nadřazený prvek ze skupiny `Layout`. Prvky umožňující toto vložení jsou seznamy nebo elementy typu `Cell`. Na obrázku 5 je předveden maplet i s jeho zdrojovým kódem, který propojuje různá rozvržení.

Výše popsané možnosti rozvržení obsahu vytváří ve výchozím nastavení základní kostru mapletu a definují formátování obsahu včetně jeho grafických úprav. Z hlediska formátování má však tvůrce více možností, neboť většina `Layout` prvků je dále nastavitelná pomocí dalších možností. Ty se



Obrázek 5: Maplet vzniklý propojením rozvržení

prvkům přiřazují jako jeho argumenty ve tvaru rovnice, přičemž je vhodné název možnosti vložit mezi apostrofy např. *název_prvku('možnost' = hodnota, 'možnost' = ...)*. Layout prvky je možné rozdělit do několika logických skupin podle jejich typu na nadřazené Layout prvky (BoxLayout, GridLayout a BorderLayout), Cell prvky (BoxCell, GridCell a GridCell2) a prvky, které lze označit jako Row prvky (BoxRow, BoxColumn a GridRow). Mezi Row prvky je zařazen i BoxColumn, neboť funguje obdobně jako BoxRow s tím rozdílem, že svůj obsah zarovnává v kolmém směru k BoxRow.

Nadřazené Layout prvky zapouzdřují celý obsah a někdy je vhodné tento fakt graficky zvýraznit. Existuje několik možností jak toho docílit. Možnost background mění barvu pozadí. Úplné změny pozadí lze docílit jen v Grid Layout, v ostatních rozvrženích barva ohraničuje jednotlivé prvky. Jako hodnotu background je možné použít anglické názvy základních barev ohraničené jako symbol apostrofy nebo hexadecimálním označením uzavřeným v uvozovkách spolu se znakem křížku (#). Další možností je border, která určuje zda bude obsah ohraničen nebo nikoliv a může nabývat pouze booleovských hodnot. S border se pojí další možnosti caption a font, které jsou použitelné jen s *border=true*. Caption přidává do levého horního rohu ohraničení popisek, který může být i víceslovný a jeho formátování se provádí pomocí font. Hodnota font je přiřazována prvkem Font. Ten je definován svými argumenty, jimiž jsou druh písma, daný jeho jménem v uvozovkách, velikost písma, ve formě nezáporného celého čísla, a booleovskými hodnotami možností bold a italic, které určují zda bude popisek napsán tučně a kurzívou. Nastavení možnosti font může vypadat následovně: *font=Font("Arial", bold=true, italic=true, 15)*. Hodnotami left (vlevo), right (vpravo) nebo center (na střed) možnosti align lze změnit horizontální

zarovnání obsahu. Obdobou `halign` je `valign` udávající vertikální zarovnání obsahu. Použitelnými hodnotami `valign` jsou `top` (nahoru), `center` (na střed) a `bottom` (dolů). Prvku `GridLayout` je u `halign` i `valign` možné nastavit hodnotu `full` (plné), která rozvine obsah buňky na celou její šířku u `halign` nebo výšku u `valign`, což je znatelné pokud buňka obsahuje tlačítko. `BorderLayout` nepodporuje nastavení `halign` ani `valign`. Možností `inset` lze zvětšit volné místo obklopující obsah nadřazených `Layout` prvků a v rozvržení prvkem `GridLayout` zároveň zvětšuje i volné místo mezi jednotlivými `Layout` prvky. Hodnotou `inset` může být jen kladné celé číslo. `Visible` je možnost, která udává zda bude celý obsah viditelný uživateli a je řízeno booleovskými hodnotami. Poslední použitelnou možností je `reference`. `Reference` celé rozvržení pojmenovává a její hodnotou může být libolný řetězec znaků umístěný v uvozovkách. Ekvivalentem k `reference` je označení pomocí spodního indexu.

Nastavení možností `Row` prvků probíhá stejně jako u nadřazených `Layout` prvků jen s několika výjimkami. Možnosti `background`, `border`, `caption`, `font`, `inset`, `halign` a `valign` fungují naprosto shodně. `GridRow` je možné nastavit `halign` a `valign` na `full` se stejným efektem jako u `GridLayout`. `GridRow` podporuje jen možnosti `halign` a `valign`. Další shodnou možností je `visible`. Změnu přináší možnosti `spacing`, `hscroll` a `vscroll`. `Spacing` svou nezápornou celočíselnou hodnotou udává velikost odsazení mezi jednotlivými elementy obsaženými v `Row` prvku. Toto odsazení ale probíhá vždy jen v jedné ose. Použití `spacing` u `BoxRow` způsobí odsazení v horizontální ose, které je představitelné jako odsazení teoretických sloupců tvořených obsahem řádku. Podobně použití možnosti `spacing` u `BoxColumn` odsadí obsah ve vertikální ose. `Hscroll` může přidat pod obsah prvku posuvník sloužící k procházení obsahu v horizontálním směru, pokud obsah přesáhne přes levý nebo pravý okraj tohoto prvku. `Vscroll` funguje shodně jako `hscroll`, jen lištu přidává na pravou stranu prvku a slouží k vertikálnímu procházení obsahu přesahujícího horní nebo spodní okraj prvku. Hodnotami určujícími `hscroll` i `vscroll` jsou `never` (nikdy), která je výchozím stavem a nezobrazuje posuvníky, `as_needed` (podle potřeby), jež zobrazí posuvníky jen v případě přesahu obsahu a `always` (vždy), která posuvníky zobrazuje vždy, i když nedojde k přesahu.

Nastavení `Cell` prvků se týká pouze prvků `BoxCell` a `GridCell`, neboť `GridCell2` lze nastavit jen jejich polohu v rámci rozvržení. Tyto prvky umožňují použít možnosti `hscroll` a `vscroll` se skoro shodnou funkcionalitou jako u `Row` prvků. Prvku `GridCell` lze nastavit možnosti `halign` a `valign` i na hodnotu `full`. Pro `BoxCell` tato nastavení fungují v závislosti na tom, jestli je prvek použitý jako argument `BoxRow` nebo `BoxColumn`. V `BoxColumn` má na obsah vliv pouze `valign`, v `BoxRow` pak `halign`. Existují ještě možnosti `hweight` a `vweight`, které jsou podporovány pouze prvky `GridCell`. `Hweight`

udává, jak bude distribuován prostor, vzniklý roztažením okna mapletu v horizontálním směru, mezi jednotlivé buňky. Použití na buňce ovlivní všechny buňky v témže sloupci. Vweight udává distribuci prostoru vzniklého roztažením ve vertikálním směru a jeho použití ovlivňuje všechny buňky ve stejném řádku. Hodnotou hweight i vweight může být jen celé číslo od 0 do 10.

2.3 Použitelné prvky

Označení použitelné prvky zaštiťuje všechny prvky, které nejsou ze skupiny Layout a lze je použít při tvorbě mapletu. Jedná se tedy o prvky, které generující obsah, nastavují nebo propojují prvky nebo slouží k ovládání mapletu. Použitelné prvky je možné rozdělit do pěti skupin podle způsobu jejich použití. Mimo tyto skupiny existují ještě nezařaditelné prvky, z nichž nejdůležitější jsou Maplet, Window, Action a Font, který je představen v podkapitole Rozvržení mapletu. Jednotlivé možnosti se prvkům přiřazují jako jeho argumenty ve tvaru rovnice, přičemž je vhodné název možnosti vložit mezi apostrofy např. *název_prvku ('možnost' = hodnota, 'možnost' = ...)*.

Maplet je prvek, bez kterého by nebylo možné maplety tvořit, protože zapouzdřuje veškerý obsah včetně rozložení a bez něj by program Maple nebyl schopen vytvořenou aplikaci spustit. Prvek je dále nastavitelný pomocí možností abnormalshutdown a onstartup. Hodnotou první možnosti může být pouze řetězec. Tento řetězec je výstupem mapletu do programu Maple pokud dojde k ukončení pomocí prostředků operačního systému (ve Windows symbol X v pravém horním rohu libovolného okna). Onstartup definuje co se stane okamžitě po spuštění mapletu. Hodnotou může být prvek Action nebo některý z příkazových prvků.

Prvek Window definuje detaily okna mapletu a může být použit pouze jako argument prvku Maplet. Jednotlivá specifika je možné nastavit pomocí možností. Možnost defaultbutton, jejíž hodnotou může být pouze prvek Button, určuje, které tlačítko bude po spuštění mapletu reagovat na klávesu enter. Pokud není možnost v prvku zahrnuta, stává se tímto tlačítkem první tlačítko mapletu. Height a width jsou možnosti nastavující celým kladným číslem pevnou výšku a šířku okna. Layout stanovuje rozvržení mapletu. Hodnotou může být vytvořené rozvržení nebo jeho identifikátor. Tuto možnost není nutné používat, protože je možné rozvržení použít přímo jako argument prvku Window, aniž by došlo k narušení funkčnosti. Shodně fungují i možnosti menubar a toolbar. Menubar slouží k připojení ovládací lišty do záhlaví okna mapletu a její hodnotou může být prvek MenuBar nebo jeho identifikátor. Toolbar připojuje do okna mapletu lištu tlačítek. Hodnotou možnosti může být prvek Toolbar nebo jeho identifikátor. Booleovskými hodnotami možnosti resizable lze určit, jestli bude možné měnit velikost okna mapletu.

Řetězec znaků použitý jako hodnota možnosti title udává jméno spuštěného okna mapletu. Možnostmi `xcoord` a `ycoord`, jejichž hodnotami může být jen nezáporné celé číslo, lze nastavit souřadnice levého horního rohu okna mapletu vzhledem k levému hornímu rohu plochy monitoru. Výchozím umístěním okna je střed plochy. Poslední možností je reference, která řetězcem znaků v uvozovkách určuje unikátní označení Window prvku. Prvek je však možné označit i pomocí spodního indexu.

Action v mapletu slouží k definování událostí daných příkazovými prvky nebo dalšími prvky Action. Pokud je argumentem seznam nebo vnořený seznam, vykonávají se akce sekvenčně. To znamená, že akce může být vykonána až po skončení předchozí akce. Nejčastěji je Action používán jako argument prvků, které slouží k interakci uživatele a mapletu. Kromě označení pomocí reference má prvek jen jednu nastavitelnou možnost `run`, která však není určena k nastavení tvůrci. Tato možnost slouží k vyvolání okamžité reakce prostředí mapletu a je nastavována pomocí metody `Set` programu Maple nebo prvku `SetOption`. Použití je vhodné k řešení podmíněného vypnutí mapletu.

První skupinu zbylých prvků tvoří příkazové prvky `CloseWindow`, `Evaluate`, `RunDialog`, `RunWindow`, `SetOption` a `Shutdown`. Tyto prvky slouží k ovládání ostatních prvků a jako jediné se mohou objevit jako argument prvku Action. `RunWindow` a `CloseWindow` se používají výhradně pro manipulaci s oknem mapletu. K manipulaci dochází na úrovni prvků `Window` obsažených v témže prvku `Maplet`. `RunWindow` okno otevírá a `CloseWindow` okno zavírá. Argumentem obou může být pouze odkaz, na existující prvek `Window` pomocí jeho označení a žádné možnosti nejsou k dispozici. `RunDialog` slouží k otevírání oken vytvořených pomocí dialogových prvků. Argumentem může být pouze odkaz na existující dialogový prvek bez dalších možností. Prvek `Evaluate` má jediný účel a to vykonávání procedur předdefinovaných v programu Maple i vytvořených tvůrci mapletu. Procedury se v prvku definují pomocí možnosti `target`, určující prvek, do kterého bude zobrazen výsledek, a `function`, stanovující proceduru. Parametry procedury se vkládají samostatně zapouzdřené v prvku `Argument`. Prvek `Evaluate` ale ve většině případů akceptuje i jednodušší syntax pomocí rovnice, kde levou stranu tvoří odkaz na cílový prvek a pravou jméno procedury s parametry použitými jako argument. Pomocí `SetOption` je možné měnit možnosti prvkům v rámci prvku `Maplet`. Syntax je obdobná jako u `Evaluate`. Pomocí možnosti `target` je určen požadovaný prvek, jehož měněná možnost je v apostrofech použita jako hodnota možnosti `'option'`. Hodnotou možnosti `value` je určena nová hodnota prvku. Zápis je opět možný i pomocí rovnice. Vynecháním možnosti `'option'` lze měnit obsah některých prvků. Posledním prvkem této skupiny je `Shutdown`. Tento prvek slouží k ukončení spuštěného mapletu a používá se většinou bez argumentů. Je ale možné je použít

ve formě odkazů na prvky nebo jako nezávislé řetězce znaků a ty jsou pak mapletem vráceny v podobě seznamu jako výsledek spouštěcího prvku *Maplets[Display]*.

Další skupinu tvoří dialogové prvky *AlertDialog*, *ColorDialog*, *ConfirmDialog*, *FileDialog*, *InputDialog*, *MessageDialog* a *QuestionDialog*, které slouží ke specifikaci dialogových oken. Dialogová okna jsou předdefinovaná okna, u kterých tvůrce může měnit obsah jen minimálně. Každý prvek této skupiny generuje jiné okno s jedinečným rozvržením a funkcionalitou a pojí se s prvkem *RunDialog*. Spojení je uskutečněno použitím označení dialogového prvku jako argumentu prvku *RunDialog*. Nastavení jednotlivých prvků probíhá pomocí možností. U všech prvků je možné určit základní nastavení okna, stejně jako u prvku *Window*, díky možnostem *height*, *width*, *resizable* a *title*. Každý prvek této skupiny lze i označit možností *reference* nebo *spodním indexem*. Další společnou možností je *onapprove*, jejíž hodnotou může být pouze příkazový prvek, prvek *Action* nebo odkaz na něj. *Onapprove* definuje akci vykonanou po zavření dialogového okna potvrzovacím tlačítkem. Další možnosti jsou aplikovatelné pro jeden či více prvků. Prvek *AlertDialog* určuje výstražné okno se dvěma tlačítky *OK* a *Cancel*, sloužícími pro potvrzení a stornování, a zprávou vloženou tvůrcem. Specifické možnosti tohoto prvku jsou *caption*, jejíž hodnota v uvozovkách udává text zobrazované zprávy, a *oncancel*, která definuje událost vzniklou po zavření okna tlačítkem *Cancel* nebo nástroji operačního systému. Hodnota *oncancel* je ve stejném tvaru jako u možnosti *onapprove*. *InputDialog* definuje okno uživatelského vstupu, které je tvořeno zprávou, textovým polem a tlačítky *OK* a *Cancel*. Obsah textového pole je možné dále použít pomocí prvku *Shutdown*, u kterého je označení okna použito jako argument. Zpráva a tlačítka se nastavují možnostmi *caption*, *onapprove* a *oncancel*. Dalšími možnostmi jsou *type* a *value*. *Type* určuje zda je v okně použita přednastavená ikona, případně jakého typu tato ikona je. Hodnotou je jedna z možností *plain* (bez ikony), *'error'* (červená chybová ikona), *information* (ikona s písmenem *i* v modrém poli) nebo *warning* (žlutá výstražná ikona). Hodnota *value* v uvozovkách určuje text zobrazený v textovém poli po otevření okna. Prvek *ColorDialog* utváří předdefinované okno sloužící k výběru barvy. Okno sestává ze tří karet sloužících k výběru barvy různými způsoby a nastavitelných tlačítek *OK* a *Cancel* a tlačítka *Reset*. *Reset* není nastavitelné pomocí možností a pouze nastavuje okno do výchozího stavu. Pomocí možnosti *value* lze nastavit předvybranou barvu místo výchozí bílé. Jako hodnotu *value* lze použít název jedné ze základních barev nebo RGB kód barvy v decimální nebo hexadecimální podobě uzavřené v uvozovkách spolu se znakem křížku (*#*). Dialogový prvek *MessageDialog* generuje okno jednoduché zprávy spolu s tlačítkem *OK*. Specifickými možnostmi jsou *caption* a *type*, které fungují stejně jako u výše zmíněných prvků. *ConfirmDialog* určuje potvrzovací okno, které je složeno z textu, případné ikony a tlačítek

Yes, No a Cancel. Dialogová okna `ConfirmDialog` je vhodné použít jako rozhodovací prvek případného větvení ve výpočtu mapletu. Možnostmi, které nejsou shodné pro všechny prvky skupiny dialogových prvků, jsou `caption`, `oncancel`, `ondecline` a `type`. `Caption` stejně jako u ostatních prvků s touto možností slouží k určení zobrazovaného textu. `Oncancel` propojuje tlačítko `Cancel` s případnou akcí či příkazovým prvkem. `Ondecline` také slouží k propojení s akcí nebo příkazovým prvkem ale u tlačítka `No`. Jako hodnotu možnosti `type`, která určuje případnou ikonu, je možné použít `plain`, `'error'` a `warning`, ale hodnota `information` je nahrazena hodnotou `question` (symbol otazníku v modrém poli). Obdobou prvku `ConfirmDialog` je prvek `QuestionDialog`, který vytváří shodné okno se stejným obsahem jen bez tlačítka `Cancel`. Posledním prvkem skupiny je `FileDialog` sloužící k definici okna pro výběr umístění souboru v počítači. Unikátními možnostmi jsou `approvecaption`, `closeonapprove`, `directory`, `filefilter`, `filename`, `fileselectionmode`, `filterdescription` a `oncancel`, která však funguje stejně ve všech prvcích, jež ji používají. Hodnotou ve formě textu v uvozovkách nastavuje možnost `approvecaption` popis umístění na potvrzovacím tlačítku. `Closeonapprove` booleovskými hodnotami určuje zda bude dialogové okno zavřeno po stisku potvrzovacího tlačítka. `Directory` udává nad obsahem které složky bude okno otevřeno a jako hodnotu lze použít pouze řetězec znaků v uvozovkách odkazující na umístění na disku (např. `C:\\Maple\\`). `Filefilter` svou hodnotou ve tvaru řetězce znaků v uvozovkách určuje, jaké přípony souborů jsou zobrazovány v prohledávaných složkách. K definování přípony se používá standardní zápis s hvězdičkou a řetězec může obsahovat i více těchto přípon oddělených čárkou. `Filename` se používá ke zpřesnění možnosti `directory`, protože udává který soubor má být označen po spuštění dialogového okna. Hodnotou je řetězec znaků v uvozovkách obsahující název souboru včetně přípony. Hodnotami `filesonly` (pouze soubory), `directoriesonly` (pouze složky) a `filesanddirectories` (soubory a složky) možnosti `fileselectionmode` lze definovat, které typy souborů jsou akceptovatelné jako výstup. Možnost `filterdescription` slouží k přidání popisů k příponám zadaným u možnosti `filefilter` a její hodnotou je řetězec znaků v uvozovkách obsahující jednotlivé popisky oddělené čárkou. Možnost `oncancel` má stejná syntaktická pravidla jako u ostatních prvků, které tuto možnost používají. Cestu k hledanému souboru lze získat pomocí prvku `Shutdown`.

Skupinu menu prvků tvoří `CheckBoxMenuItem`, `Menu`, `MenuBar`, `MenuItem`, `MenuSeparator`, `PopupMenu` a `RadioButtonMenuItem`, které se používají pro tvorbu ovládacích lišt. Skupina obsahuje dva nadřazené prvky `MenuBar` a `PopupMenu`, které zapouzdřují libovolné množství zbylých prvků. `MenuBar` lze použít jen jako hodnota možnosti `menubar` prvku `Window` a `PopupMenu` pouze jako hodnota možnosti `popupmenu` prvků `TextField` a `TextBox` ze skupiny prvků okna. Prvky však mohou být zastoupeny i jejich označením. Jedinými možnostmi těchto nadřazených prvků je označení

pomocí možnosti reference nebo použitím spodního indexu a booleovskými hodnotami řízená možnost `visible`, která určuje zda budou vytvořené lišty viditelné uživatelem mapletu. Základními stavebními kameny ovládacích lišt jsou prvky `MenuItem`, `CheckBoxMenuItem` a `RadioButtonMenuItem`, které se v mapletu zobrazují jako jednotlivé položky lišt. Tyto prvky lze použít pouze jako argumenty prvků `Menu` a `PopupMenu` a jejich jediným obsahem jsou jejich nastavitelné možnosti. Těmito možnostmi jsou `caption`, `enabled`, `image`, `onclick`, `visible` a označení pomocí refernce nebo dolního indexu, které jsou společné pro všechny tři prvky. Prvkům `CheckBoxMenuItem` a `RadioButtonMenuItem` je navíc možné nastavit `value` a pro druhý zmíněný ještě `group`. Možnost `caption` řetězcem znaků v uvozovkách použitým jako její hodnota, definuje zobrazovaný název prvku. Booleovské hodnoty `enabled` určují, zda je prvek uživatelem zvolitelný či nikoli. `Image`, jehož hodnotou může být pouze odkaz na prvek `Image`, deklaruje obrázek zobrazovaný vedle prvku v liště. Pomocí možnosti `onclick` se jednotlivé prvky propojují s požadovaným akcemi. Jako hodnoty `onclick` je možné použít příkazový prvek, prvek `Action` nebo odkaz na něj. Tuto možnost je možné vypustit a nahradit ji jen její hodnotou bez ztráty funkčnosti. Možnost `visible` funguje stejně jako u prvků `MenuBar` a `PopupMenu`. Prvek `CheckBoxMenuItem` vytváří položku lišty, která funguje jako zaškrtačací políčko, a proto je rozšířen o možnost `value`, která pomocí booleovských hodnot určuje zda je položka lišty zaškrtnuta. Prvek `RadioButtonMenuItem` funguje jako přepínací políčko, a tak je kromě `value`, která určuje zaškrtnutí, rozšířen i o možnost `group`, která jednotlivé přepínače slučuje do jedné skupiny. Hodnotou `group` je vždy odkaz na prvek `ButtonGroup`, který se deklaruje jen svým označením. Dalším prvkem je `MenuSeparator`, který se používá bez možností a nemůže obsahovat žádné argumenty. Prvek může být obsažen v prvcích `Menu` a `PopupMenu` a slouží k vložení horizontálního grafického oddělovače mezi položky lišty. Posledním prvkem skupiny je prvek `Menu`, který se používá k definování jednotlivých nabídek ovládací lišty prvku `MenuBar` nebo tvorbě víceúrovňových nabídek prvků `MenuBar` a `PopupMenu`. Tento prvek proto může být použit jako argument prvků `MenuBar`, `PopupMenu` nebo `Menu` a sám může mít jako argumenty všechny verze `MenuItem` prvků nebo `Menu` prvky. Možnostmi dalšího nastavení jsou `caption`, `background`, `enabled`, `font`, `foreground`, `image`, `visible` a označení. `Caption` udává název položky ovládací lišty definované prvek `Menu`. Možnost `background` určuje pomocí své hodnoty, kterou může být název jedné ze základních barev nebo RGB kód barvy v decimální nebo hexadecimální podobě uzavřené v uvozovkách spolu se znakem křížku (`#`), jakou barvu bude mít zvýraznění jednotlivých položek prvku `Menu`. Booleovské hodnoty `enabled` určují zda jsou položky prvku uživatelem zvolitelné. Možností `font` se pomocí hodnoty dané prvkem `Font` určuje pro všechny položky prvku použité písmo, jehož barvu lze změnit možnostmi `foreground`, která má stejná syntaktická pravidla jako možnost `background`. Možností `image` je možné vložit vlevo od názvu prvku

obrázek definovaný prvkem Image. Poslední možností `visible` lze pomocí booleovských hodnot nastavit viditelnost prvku pro uživatele.

Prvky `ToolBar`, `ToolBarButton` a `ToolBarSeparator` jsou jedinými prvky skupiny prvků panelu nástrojů. Tyto prvky slouží k vytvoření plovoucí lišty tlačítek v okně mapletu. Lišta je po spuštění mapletu umístěna pod ovládací lištou, ale lze ji libovolně přesouvat v okně mapletu i mimo něj. Prvek `ToolBar` může být použit pouze jako hodnota možnosti `toolbar` prvku `Window` a to přímo nebo odkazem pomocí jména. Obsahem prvku mohou být pouze prvky `ToolBarButton` a `ToolBarSeparator` v libovolném počtu. Možnostmi prvku `ToolBar` jsou `visible` a označení jednou z použitelných možností. Možnost `visible`, která je nastavitelná booleovskými hodnotami, určuje zda bude definovaná lišta viditelná uživatelem mapletu. Prvek `ToolBarButton` primárně nepodporuje prvky jako argumenty, ale je nastavitelný pomocí možností, jejichž hodnotami prvky být mohou. Použitelnými možnostmi jsou `caption`, `enabled`, `image`, `onclick`, `tooltip` a označení. Možnost `Caption` svou hodnotou ve tvaru řetězce v uvozovkách přiřazuje popisec tlačítka určenému tímto prvkem. `Enabled` řízená booleovskými hodnotami určuje zda bude uživatel moci prvek používat. `Image` je možnost, která umožňuje nahradit popisec tlačítka obrázkem a její hodnotou může být pouze prvek `Image`, jehož argumentem je úplná cesta k umístění obrázku uzavřená v uvozovkách, nebo odkaz na něj. Možnost `onclick` stanovuje akci vykonanou po stisku tlačítka uživatelem. Jako hodnota této možnosti může být použit příkazový prvek, prvek `Action` nebo odkaz na něj. Nastavením možnosti `tooltip`, jejíž hodnotou může být pouze řetězec znaků v uvozovkách, se definuje text nápovědy k tlačítku zobrazený po najetí kurzorem na toto tlačítko. K označení prvku je možné použít obou způsobů. Prvek `ToolBarSeparator` určuje grafický rozdělovač mezi tlačítka prvku `ToolBarButton` a nepoužívá argumenty ani možnosti.

Prvky okna jsou prvky, které utvářejí obsah mapletu a zároveň mu dodávají interaktivitu. Všechny tyto prvky lze použít pouze jako argumenty prvků rozvržení. Skupina prvků okna je tvořena prvky `Button`, `CheckBox`, `ComboBox`, `DropDownBox`, `Label`, `ListBox`, `MathMLEditor`, `MathMLViewer`, `PasswordField`, `Plotter`, `RadioButton`, `Slider`, `Table`, `TextBox`, `TextField` a `ToggleButton`. Všechny prvky mají shodnou možnost označení pomocí možnosti `reference` nebo `spodním indexem`. Zbylé možnosti se většinou opakují u více prvků, a proto jsou v následujícím textu uvedeny jejich hodnoty a popisy jen u prvního z prvků, u kterého se možnost vyskytuje, nebo pokud má jinou funkcionalitu. Prvek `Button` slouží k přidání tlačítka, které je dále nastavitelné pomocí možností `background`, `caption`, `enabled`, `font`, `foreground`, `halign`, `height`, `image`, `onclick`, `tooltip`, `valign`, `visible` a `width`. Možnost `background` určuje barvu prvku tlačítka a jako její hodnotu lze použít název jedné ze základních barev nebo RGB kód barvy v decimální

nebo hexadecimální podobě uzavřené v uvozovkách spolu se znakem křížku (#). Hodnotou možnosti `caption`, která je ve formě řetězce znaků v uvozovkách, se stanovuje popisek tlačítka. Booleovskými hodnotami řízená možnost `enabled` určuje zda-li je tlačítko použitelné uživatelem mapletu. Možností `font`, jejíž hodnotou může být pouze prvek `Font` nebo odkaz na něj, se určují vlastnosti písma použitého v popisku tlačítka. `Foreground` je možnost se stejnými syntaktickými pravidly jako `background` a určuje barvu písma. Pomocí možnosti `halign` je možné měnit horizontální zarovnání textu popisku. Hodnotou může být jedna z možností `left` (vlevo), `center` (na střed) nebo `right` (vpravo). Vertikální zarovnání je prováděno hodnotou `top` (nahoru), `center` (na střed) nebo `bottom` (dolů) možnosti `valign`. Možnostmi `height` a `width` je možné změnit výšku a šířku tlačítka, jehož výchozí velikost se přizpůsobuje popisku. Hodnotou těchto možností může být pouze kladné celé číslo. Možností `image` je možné zaměnit popisek tlačítka za obrázek. Hodnotou `image` je prvek `Image` nebo odkaz na něj, přičemž prvek `Image` se používá pouze s argumentem řetězce znaků v uvozovkách, který udává úplnou cestu k požadovanému obrázku. `OnClick` je možnost sloužící k navázání akcí mapletu na stisk tlačítka. Jako hodnota je použit příkazový prvek, prvek `Action` nebo odkaz na něj. Program Maple však akceptuje i použití zápisu s vynechanou možností `onclick` bez ztráty funkčnosti. V takovém případě se prvek vyvolávající akci nebo odkaz na něj použije jako argument prvku `Button`. Další možností je `tooltip`, která definuje text nápovědy k tlačítku zobrazený po najetí kurzorem na toto tlačítko. Hodnotou je řetězec znaků v uvozovkách. Poslední možností prvku `Button` je `visible`, která je řízena booleovskými hodnotami. Tato možnost stanovuje viditelnost prvku pro uživatele. Prvek `CheckBox` přidává do mapletu zaškrťovací políčko. Možnostmi nastavení jsou `background`, `caption`, `enabled`, `font`, `foreground`, `image`, `onchange`, `tooltip`, `value` a `visible`. `Background` mění barvu zvýraznění prvku. Možnost `onchange` propojuje prvek s akcí definovanou hodnotou možnosti. Jako hodnotu lze použít libovolný příkazový prvek, prvek `Action` nebo odkaz na něj. Namísto celého zápisu možnosti i s hodnotou je možné použít pouze hodnotu bez ztráty funkčnosti. Booleovskými hodnotami řízená možnost `value` specifikuje, jestli bude po spuštění mapletu políčko zaškrtnuto. Prvek `ComboBox` vkládá do mapletu rozbalovací seznam s možností užití vlastní odpovědi, pokud se požadovaná odpověď v seznamu nevyskytuje. Argumentem tohoto prvku je seznam všech prvků seznamu, na který je možno aplikovat příkaz `sort` pro setřídění prvků ve spuštěném mapletu. Možnostmi nastavení `ComboBox` jsou `appenditem`, `background`, `editable`, `enabled`, `font`, `foreground`, `halign`, `height`, `itemlist`, `onchange`, `tooltip`, `valign`, `value`, `visible` a `width`. Možnosti `appenditem` a `itemlist` slouží pro práci se seznamem a nemohou být deklarovány v možnostech prvku. `Appenditem` přidává záznam do seznamu a používá se pouze v kombinaci s prvkem `SetOption`. `Itemlist` slouží k práci s celým seznamem v kombinaci s metodami `Set` a `Get` podbalíčku *Maplets/Tools*. Možnost `editable` řízená

booleovskými hodnotami stanovuje, jestli je možné použít vlastní odpověď. Background nastavuje barvu u částí prvku, které neslouží k zadávání obsahu. Nastavení možností `halign` a `valign` se aplikuje na všechny položky seznamu. Možnost `value` určuje, která položka seznamu bude vybrána jako výchozí po spuštění mapletu. Hodnotou může být i řetězec znaků v závorkách, který není umístěn v seznamu. Namísto možnosti `value` je možné použít pouze řetězec bez ztráty funkčnosti. Prvek `DropDownBox` funguje stejně jako `ComboBox` jen neumožňuje vkládání vlastních odpovědí. Možnostmi prvku jsou `appenditem`, `background`, `enabled`, `font`, `foreground`, `halign`, `height`, `itemlist`, `onchange`, `tooltip`, `valign`, `value`, `visible` a `width`. U tohoto prvku je na rozdíl od `ComboBox` nutné u možnosti `value` použít prvek ze seznamu. Dalším prvkem pracujícím se seznamy je prvek `ListBox`. Tento prvek vkládá do mapletu seznam jako velké textové pole, ve kterém jsou jednotlivé položky seznamu vypsány pod sebou a je možné vybrat více než jednu položku. Nastavení prvku probíhá pomocí možností `appenditem`, `background`, `enabled`, `font`, `foreground`, `halign`, `itemlist`, `onchange`, `tooltip`, `valign`, `value`, `visible` a `width`. Prvek `Label` vkládá do mapletu needitovatelný text, který však oproti běžně vloženému textu lze pomocí možností upravovat. Text je vkládán pomocí možnosti `caption`. Možnostmi upravujícími `Label` jsou `background`, `caption`, `enabled`, `font`, `foreground`, `halign`, `height`, `image`, `tooltip`, `valign`, `visible` a `width`. Možnost `enabled` udává zda je možné text označit uživatelem a dále s ním pracovat. `Image` vkládá požadovaný obrázek před text. Další prvkem skupiny prvků okna je `MathMLEditor`, který slouží k zobrazování, úpravě a tvorbě matematických výrazů v režimu 2-D math, jenž se od běžného režimu zobrazení výrazů odlišuje použitím všech běžným matematických symbolů. Do pole editoru zobrazeného v mapletu je možné tyto symboly vkládat z nabídky, která se uživateli zobrazí po stisku pravého tlačítka myši v tomto poli. Označení prvku je možné jen pomocí možnosti `reference` a prvek nemůže obsahovat žádné argumenty. Nastavujícími možnostmi jsou `background`, `breakwidth`, `focus`, `foreground`, `height`, `outputformat`, `palette`, `value`, `visible`, `width` a `wrapped`. Možnost `breakwidth` určuje mezní šířku v pixelech po jejímž překročení se obsah prvku přestane zobrazovat v režimu 2-D math. Hodnota určující tuto možnost je nastavována kladným celým číslem. Pokud je v možnostech prvku nastavena možnost `wrapped` na `true`, je možnost `breakwidth` programem ignorována. Možnost `focus`, jejíž jediná hodnota `true` je nastavitelná pouze prvkem `SetOption`, určuje vlastnost, že po vykonání nějaké akce se na prvek zaměří maplet. Znamená to, že se prvek stane ovladatelný klávesnicí aniž by ho uživatel pro práci vybral. Možností `outputformat` se definuje formát výstupu prvku. Hodnota `presentation` určuje zaměření výstupu na prezentaci (režim 2-D math) oproti tomu hodnota `content` zaměřuje výstup směrem k obsahu (běžný režim). Hodnota `basic` (základní) nebo `extended` (rozšířená) možnosti `palette`, určuje rozsah nabídky zobrazované po stisku pravého tlačítka. Výchozí hodnotou je `basic`. Možnost `wrapped` ovládaná booleovskými hodnotami s výchozí hodnotou

false stanovuje, jestli bude výraz přesahující okraj prvku zalomen na další řádek. Možnost value nastavuje výchozí výraz zobrazený po spuštění mapletu a její hodnota může být zapsána pouze ve tvaru algebraického výrazu. Dalším prvkem je MathMLViewer, který na rozdíl od MathMLEditoru pouze zobrazuje matematické výrazy. Označení prvku je možné jen pomocí možnosti reference a prvek nemůže obsahovat žádné argumenty. Možnostmi, které definují tento prvek jsou background, breakwidth, fontsize, foreground, height, value, visible, width a wrapped. Možnost fontsize definuje velikost zobrazovaného textu a může nabývat pouze kladných celočíselných hodnot. Možnost value stejně jako u předchozího prvku nastavuje výchozí výraz zobrazený po spuštění mapletu a její hodnota může být zapsána pouze ve tvaru algebraického výrazu. Prvek Plotter přidává do okna mapletu pole sloužící k zobrazování grafů, které mohou být dvourozměrné i třírozměrné. Obsahem pole může být i animace grafu. Plotter nepoužívá argumenty a k jeho nastavení se při zobrazování grafů používají možnosti background, height, tooltip, value, visible a width. Pokud je obsahem animace je možné ji nastavit přidáním možností continuous, delay, frame_backwards, frame_forward, pause, play, 'stop', to_end a to_start. Jako hodnotu možnosti value lze použít pouze konstrukce založené na bázi funkcí plot a plot3D nebo funkcí z balíčku plots. Continuous určuje, zda animace bude probíhat nepřetržitě a její hodnotu je jedna z booleovských hodnot. Kladným celým číslem řízená možnost delay stanovuje prodlevu mezi jednotlivými snímky animace. Možnost play, jejíž hodnotou je pouze true, určuje, je-li animace spuštěna. Nezařazení možnosti znamená její implicitní nastavení na false. Zbylé možnosti prvku Plotter lze nastavit pouze pomocí prvku SetOption a jejich hodnotou může být jen true. Možnosti frame_backwards a frame_forward každým nastavením hodnoty animaci nastavují na předchozí/následující snímek. Možnost pause pozastavuje animaci. Po použití možnosti play animace pokračuje od bodu, kde byla zastavena. Stop animaci také zastavuje, ale zároveň ji nastavuje na výchozí snímek. Pomocí možnosti to_end a to_start lze nastavit jako zobrazovaný snímek animace poslední nebo první snímek animace. Prvek RadioButton vkládá do mapletu přepínač jehož vzhled a funkce jsou nastavovány pomocí možností background, caption, enabled, font, foreground, group, image, tooltip, value a visible. Povinná možnost group přepínač zařazuje do skupiny přepínačů, ve které je vždy pouze jeden přepínač zaškrtnutý. Hodnotou group je vždy odkaz na prvek ButtonGroup, který se deklaruje jen svým označením. Možnost value řízená booleovskými hodnotami určuje zda je prvek označen zaškrtnutím. Prvek Slider umožňuje umístit do mapletu číselnou osu s jezdcem použitelnou pro výběr čísla uživatelem. Slider je definován možnostmi background, enabled, filled, foreground, lower, majorticks, minorticks, onchange, orientation, showlabels, showticks, snapticks, tooltip, upper, value a visible. Background nastavuje pouze barvu zvýraznění daného označením prvku. Možnost filled stanovuje zobrazení souvislé osy pomocí booleovských hodnot. Celočíslnými hodnotami možností lower a upper

se definují krajní body s nejnižší a nejvyšší hodnotou. Majorticks a minorticks udávají frekvenci výskytu značek označujících jednotlivé hodnoty, přičemž značky možnosti majorticks zobrazují pod značkou hodnotu. Hodnotou možností je kladné celé číslo, které udává velikost mezery mezi jednotlivými značkami. Orientation určuje orientaci osy a může nabývat hodnot vertical nebo horizontal pro svislou nebo vodorovnou orientaci. Možnost showlabels deklaruje použití hodnot pod značkami vytvořenými možností majorticks. Použitelnými hodnotami jsou booleovské hodnoty. Těmi jsou řízeny i možnosti showticks, která určuje jestli budou u osy zobrazené značky reprezentující hodnoty osy a snapticks, která stanovuje přichycení jezdeck k hodnotám značek. Celočíslnou hodnotou se nastavuje možnost value určující hodnotu, na které je jezdec umístěn. Tabulku je možné do mapletu vložit prvkem Table. Použitelnými argumenty jsou prvky TableHeader, TableRow a TableItem, ale je vhodnější je nahradit dvěma seznamy. První seznam definuje záhlaví tabulky a druhý seznam, který může být při použití více řádků tabulky vnořený, samotný obsah. Prvek Table je nastavitelný možnostmi background, font, foreground, height, tooltip, visible a width. Prvkem TextBox je možné v mapletu specifikovat ohraničené víceřádkové textové pole. Pole je nastavováno možnostmi append, appendline, background, cursor, editable, enabled, focus, font, foreground, height, onchange, popupmenu, tooltip, value, visible, width a wrapped. Možnosti append a appendline vkládají text, za stávající obsah prvku, přičemž appendline umísťuje za vkládaný text symbol pro nový řádek. Obě možnosti je možné nastavit jen pomocí prvku SetOption a jejich hodnotou může být je řetězec znaků v uvozovkách. Cursor řízený nezáporným celým číslem určuje počet znaků, za které má být umístěn kurzor. Možnost popupmenu, jejíž hodnotou je prvek PopupMenu nebo odkaz na něj, definuje ovládací lištu zobrazovanou po stisku pravého tlačítka myši na prvku. Možnost value, která je nahraditelná zápisem obsahujícím pouze její hodnotu, určuje obsah textového pole. Prvky TextField a PasswordField jsou ekvivalentem prvku TextBox. Tyto prvky také vkládají do mapletu textové pole, ale pouze s jedním řádkem, a oba jsou nastavovány možnostmi append, background, cursor, editable, enabled, focus, font, foreground, halign, onchange, popupmenu, tooltip, value, visible a width. Rozdíl mezi TextField a PasswordField spočívá v prezentaci obsahu textového pole, neboť PasswordField zobrazuje vkládaný text jako hvězdičky, což je vhodné pro práci s hesly. Posledním ve skupině prvků okna je ToggleButton, který představuje ekvivalent k prvku CheckBox. ToggleButton vkládá do mapletu tlačítko fungující jako spínač. Při použití možnosti group se stejnou hodnotou pro více prvků lze docílit funkcionality prvku RadioButton. ToggleButton je možné nastavit možnostmi background, caption, enabled, font, foreground, image, group, onchange, tooltip, value a visible. Background nastavuje barvu tlačítka ve stisknutém stavu.

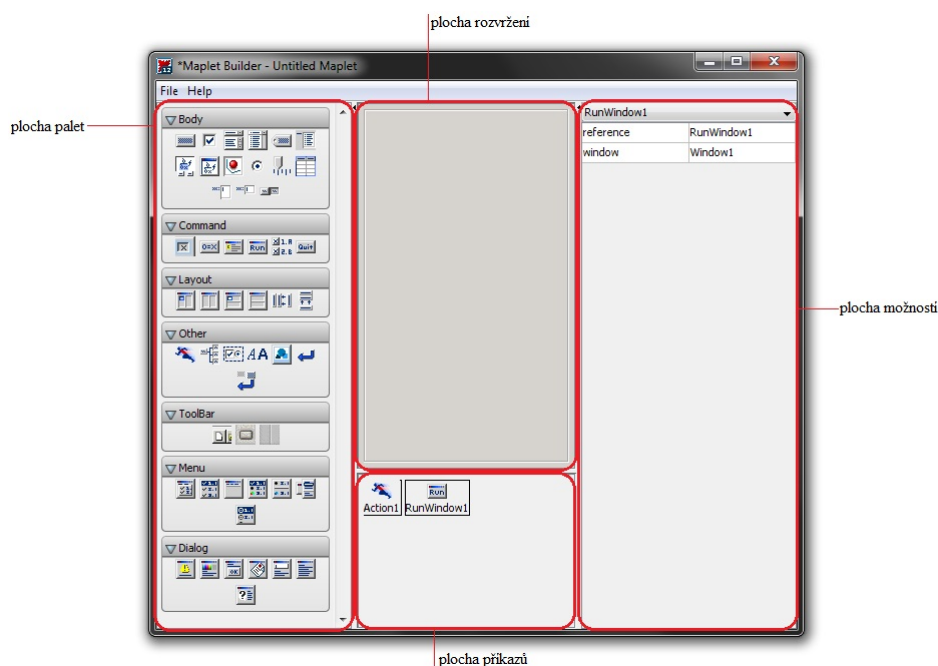
2.4 Tvorba mapletů

Před tvorbou mapletu je důležité utvořit si sumarizaci požadavků na finální maplet. Prvním atributem, na který je nutné se zaměřit, je odhad toho, co bude maplet řešit. To znamená, že tvůrce by měl mít představu, jestli budou použity uživatelské vstupy, jaké budou výstupy, jak budou tyto výstupy vyřešeny a jaké budou možnosti ovládní mapletu uživatelem. Dalším faktem ovlivňujícím tvorbu je rozvržení prvků okna a jejich funkce. Například pokud budou jako vstupní prvky použita pouze tlačítka, je vhodné tato tlačítka umístit pomocí prvku `GridLayout`. Tvůrce sám podle svého estetického hlediska rozhoduje, kam umístí blok vstupních prvků. Jde pouze o vizuální podobu, která však definuje tvorbu mapletu. Posledním aspektem tvorby je použitý způsob vytváření mapletu. V tomto směru má tvůrce dvě možnosti a to použití pracovního listu nebo asistenta `Maplet Builder`.

Asistenta `Maplet Builder` lze spustit pouze z programu `Maple`. Nachází se na kartě `Tools`, která je umístěná na ovládací liště programu, pod možností `Assistants`. Asistent se spouští v novém okně a s běžícím prostředím `Maple` není propojen. Okno je rozdělené na několik částí. V horní části se nachází ovládací lišta, obsahující základní možnosti pro práci s mapletem, jako je uložení, otevření dříve vytvořeného a spuštění nového. Prostor pod lištou vyplňují čtyři plochy. Celou levou část zabírá plocha palet. Jednotlivé palety obsahují použitelné prvky včetně prvků ze skupiny `Layout`. Střed prostředí je věnován ploše rozvržení a pod ní umístěné ploše příkazů. V pravé části je umístěna plocha možností, ve které je možné jednotlivým prvkům nastavit jejich možnosti. Obrázek 6 demonstruje spuštěný asistent.

Práce s prostředím je jednoduchá. Po spuštění může tvůrce použít výchozí rozvržení založené na `BoxLayout`. Existuje i možnost, vybrat si rozvržení v ovládacím panelu možností `New with Layout` (Nový s rozvržením), kde lze nadefinovat i počty `Row` prvků. Další práce spočívá v přetahování prvků z plochy palet do plochy rozvržení, a jejich nastavení v ploše možností. Některé prvky však nelze přidat. Jedná se o prvky ze skupin příkazová, menu, dialogová a nastavující prvky z palety `Other` (Ostatní), které se přetahují do plochy příkazů. Možnosti i těchto prvků se však nastavují v ploše možností.

Tento způsob vytváření mapletů je vhodný pro začínající tvůrce, neboť syntax jednotlivých prvků je v režii asistenta. Další výhodou je, že asistent poskytuje dostatečný aparát programu `Maple`, takže je možné tvořit větší aplikace se sofistikovanou funkcionalitou. Tvoření mapletů pomocí `Maplet Builder` má i své nedostatky. Vytvoření aplikace, která je složena z několika oken mapletů, ale zároveň je k jejímu spuštění potřeba jen jednoho souboru, není možné. Problém může nastat i v případě, že tvůrce potřebuje použít k výpočtu jím vytvořené procedury, protože prostředí asistenta neumožňuje



Obrázek 6: Vzhled asistenta Maplet Builder

vkládání takových procedur. Funkcionalita menu prvků je značně omezená. Zmíněným nedostatkům lze předejít tvorbou mapletů v pracovním listu.

Použití pracovního listu může pro mnohé tvůrce představovat výzvu, jelikož neposkytuje žádné silné pomocné prostředky. Jedinou pomocí poskytovanou programem Maple je kontrola parity závorek podtržením výrazu, přičemž lepší kontroly je možné docílit nastavením stylu odstavce s prvkem Maplet na Text místo výchozího Math. Na druhou stranu má tvůrce v rukách plnou funkcionalitu programu Maple, včetně všech prvků z balíčku *Maplets[Elements]*. Tvorbu mapletů touto cestou lze označit za programování, neboť musí být dodržována syntaktická pravidla programovacího jazyka Maple a samotný proces tvorby zahrnuje všechny činnosti typické pro programování jako jsou návrh řešení, zápis, testování a ladění vytvořeného kódu.

Při programování mapletu je nutné dodržet logickou strukturu kódu. Úvod je většinou věnován vytvořeným procedurám a příkazům načítajícím balíčky potřebné pro funkci mapletu. Další část definuje kód zapouzdřený v prvku Maplet. Při tvorbě mapletu, který je tvořen několika nezávislými okny, lze použít dva způsoby propojení těchto oken. První variantou je definovat každé okno zvlášť, a poté ho připojit pomocí příkazového prvku RunWindow. Argumentem jednotlivých prvků je unikátní

pojmenování každého okna. Druhou variantou je zapouzdřit každé okno do vlastního prvku Maplet. Tyto prvky se připojují příkazovým prvkem Evaluate ve formě *Evaluate('function' = 'Maplets[Display](identifikátor_mapletu)')*. Obě možnosti je možné kombinovat. Kód je zakončen spouštěcím prvkem Maplets[Display], jehož argumentem je unikátní název mapletu, který má být otevřen jako první po spuštění aplikace.

3 Vytvořený maplet

Celý maplet je vytvořený na rozvržení pomocí prvku `BoxLayout`. Je tvořen jedenácti dílčími maplety, z nichž jeden funguje jako ovládací. Ovládací maplet obsahuje tlačítkové menu sloužící k navigaci mezi jednotlivými maplety. Ty jsou rozděleny do šesti skupin na základě finančních výpočtů a každý z nich obsahuje, kromě vstupů a výstupů, tři tlačítka sloužící k ovládnutí mapletu. Tlačítko "Vypočítat" slouží k vygenerování výstupu, "Vymazat" anuluje vstupní pole parametrů a "Návrat do Menu" zavírá spuštěný maplet a navrácí uživatele do hlavní nabídky.

Kompletní zdrojový kód je umístěn v přílohách.



Obrázek 7: Maplet menu

3.1 Dluhopisy

První skupinou jsou Dluhopisy. Obsahuje pouze jeden maplet, který se zabývá výpočtem teoretické hodnoty dluhopisu. Výpočet ovlivňují čtyři vstupní pole: tržní úroková míra, kupónová úroková míra, doba splatnosti dluhopisu a nominální hodnota dluhopisu. Jejich vyplnění a stisk tlačítka výpočtu vygeneruje teoretickou hodnotu dluhopisu.

Teoretická hodnota dluhopisů

Výpočet teoretické hodnoty dluhopisu

Úroková míra i: % Kupónová úroková míra C: %

Doba splatnosti n: roků Nominální Hodnota NH: Kč

Výpočet provedeme pomocí vzorce:

$$P = \frac{C(1+i)^n - C + NH}{i(1+i)^n}$$

Vypočítat teoretickou hodnotu

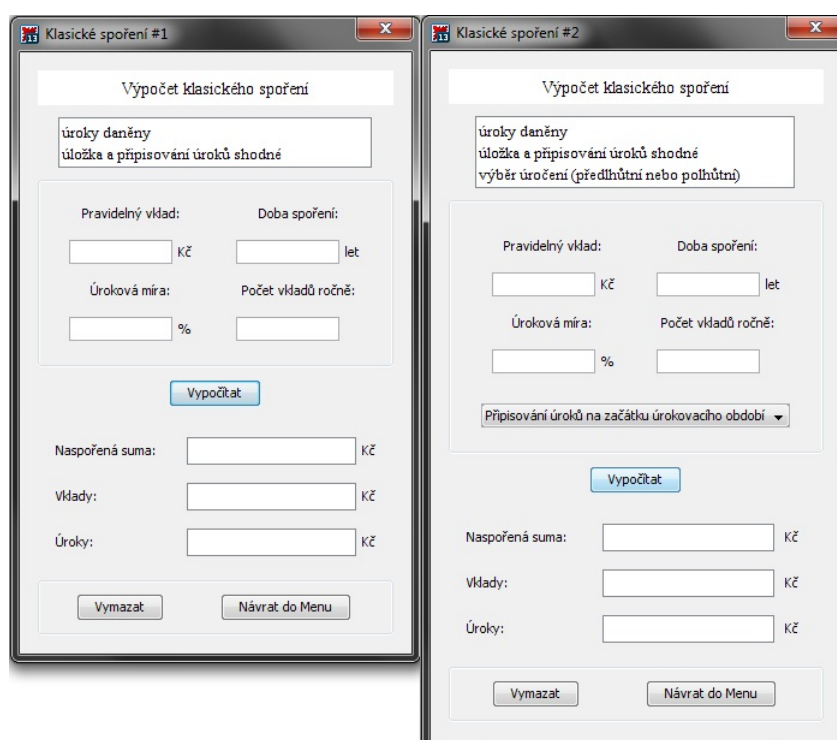
Teoretická hodnota dluhopisu: Kč

Vymazat Návrat do Menu

Obrázek 8: Maplet dluhopisů

3.2 Klasické spoření

Skupina Klasické spoření obsahuje dva maplety, které se od sebe odlišují způsobem úročení. První úročí předlůhůtně a jeho parametry jsou: výše pravidelné úložky, doba spoření v letech, roční úroková míra, počet úložek za rok. Kliknutím na tlačítko výpočtu obdržíme částku, kterou jsme za zadaných podmínek uspořili. Tato částka je následně rozdělena na naše vklady a úroky z těchto vkladů, které jsou daně 15%. U druhého je možnost volby úročení. Zbylé vstupy a výstupy jsou stejné jako u prvního mapletu.



Obrázek 9: Maplety klasického spoření

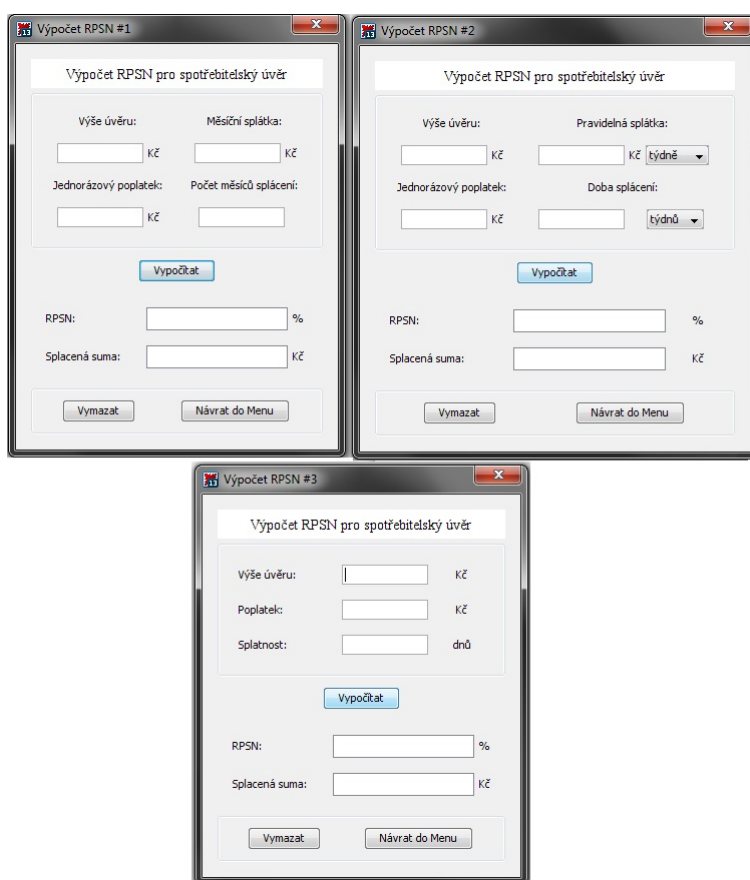
3.3 Penzijní připojištění

Skupina Penzijní připojištění obsahuje dva maplety. Vstupní parametry obou mapletů tvoří: měsíční příspěvek účastníka, předpokládané zhodnocení, měsíční příspěvek zaměstnavatele a počet roků placeného pojištění. Výstupem je jednorázové vyrovnání před a po zdanění doplněné informacemi o celkových příspěvcích účastníka, zaměstnavatele a státu, včetně výnosů z nich. První maplet vypočítává státní příspěvky na základě změny zákona z roku 2013, kdy došlo ke změně jejich velikostí. Před změnou bylo možné získat státní příspěvek již při měsíčním příspěvku účastníka 100 korun. Tato hranice se změnou posunula na výši 300 korun. Maximální výše státního příspěvku se po změně zákona zvyšuje ze 150 na 230 korun. Zároveň se však zvýšila hranice pro jeho získání. Dříve bylo možné získat maximální příspěvek od státu již při úložce 500 korun, ale od roku 2013 je k tomu zapotřebí dvojnásobná úložka. Vstupní parametr rok uzavření smlouvy druhého mapletu umožňuje simulovat situaci, ke které došlo při přechodu mezi změnami zákona. V této situaci je část státních příspěvků vypočítávána podle původních pravidel a zbytek podle nových. Vhodným nastavením tohoto parametru je možné docílit i situace, kdy je výše všech státních příspěvků určena pravidly platnými před rokem 2013.

Obrázek 10: Maplety penzijního připojištění

3.4 RPSN

Skupinu RPSN tvoří tři maplety. Vstupní parametry prvních dvou mapletů jsou: výše úvěru, pravidelná splátka, která je u prvního měsíční, ale u druhého lze přepínat mezi měsíční a týdenní, jednorázový poplatek a doba splácení, jenž je u prvního možné zadat počtem měsíců ale u druhého lze přepínat mezi měsíci a týdny. Třetí maplet je řízen vstupy: výše úvěru, poplatek a splatnost ve dnech. V tomto případě je výpočet RPSN vhodný u tzv. rychlých půjček, kdy splácení půjčky je jednorázové a doba splatnosti je zpravidla do jednoho měsíce. Výstupem všech je vypočítané RPSN v procentech a celková zaplacená částka.



Obrázek 11: Maplety RPSN

3.5 Spotřebitelský úvěr

Skupina Spotřebitelský úvěr je zastoupena pouze jedním mapletem jehož vstupy jsou: výše úvěru, roční úroková míra, doba splácení v rocích, frekvence splátek, kterou je možné zvolit jako měsíční, čtvrtletní nebo roční, a výběr mezi předlůtním a polhůtním úročením. Výstupy jsou vypočtená anuita a umořovací plán. Ten obsahuje údaje: období, splátka, úrok a úmor, které se rozpočítávají podle nastavených parametrů a zůstatek.

Spotřebitelský úvěr

Spotřebitelský úvěr

Výše úvěru: Kč

Roční úroková míra: %

Doba splácení: roků

Platba splátek:

Připisování úroků na začátku úrokovacího období

Vypočítat

Výše splátky: Kč

Umořovací plán

Vymazat

Návrat do Menu

Obrázek 12: Maplet spotřebitelského úvěru

3.6 Stavební spoření

Do skupiny Stavební spoření je zahrnut jen jeden maplet. Vstupy tohoto mapletu tvoří: pravidelný měsíční vklad, roční úroková míra, vstupní poplatek, který je získán jako relativní část cílové částky, cílová částka, počet roků spoření a celkový roční poplatek. Po provedení výpočtu obdržíme naspořenou částku a dílčí informace o: poplatcích zaplacených za celou dobu spoření, příspěvcích účastníka, výnosech z těchto příspěvků, státních příspěvcích a výnosech ze státních příspěvků. Výpočet je upraven podle změny zákona z roku 2011. Ta mění výši státních příspěvků, jejichž velikost je snížena na 10 % ročního vkladu. Maximální velikost příspěvku je však 2000 korun. Změnou prošly i připisované výnosy. Ty se od roku 2011 daní 15% a to i výnosy ze státních příspěvků. Změnila se také vázací doba na současných 6 let.

Stavební spoření

Výpočet stavebního spoření

Pravidelný měsíční vklad: Kč

Úroková míra: %

Vstupní poplatek (z cílové částky): %

Cílová částka: Kč

Počet roků spoření: let

Roční poplatek (vedení účtu a výpisy): Kč

Vypočítat

Poplatky: Kč

Příspěvky účastníka: Kč

Výnosy z příspěvků účastníka: Kč

Státní příspěvky: Kč

Výnosy ze státních příspěvků: Kč

Naspořená částka: Kč

Vymazat Návrat do Menu

Obrázek 13: Maplet stavebního spoření

4 Závěr

Cílem této práce bylo vytvořit sadu mapletů na základě matematického aparátu pro finanční matematiku. Další devizou této práce je vytvořený podrobný návod na tvorbu a práci s programem Maple.

Práce je strukturována do několika kapitol. První shrnuje historii programu Maple a vývoj programu. Na tuto kapitolu navazuje seznámení s maplety, jejich historií a zmapováním jejich aktualizací v čase. Zároveň je zde popsána a definována grafická podoba mapletu. Možnosti jak rozvrhnout obsah a vzhled. Práce poukazuje na důležitost vytvoření koster, které jsou použitelné pro celou škálu aplikací. U grafického vzhledu koster jsou prezentovány možné výstupy a specifika. Důležitou podkapitolou jsou jednotlivé prvky, které určují charakter mapletu. Prvky jsou rozděleny do skupin podle jejich funkcionality. Některé slouží pouze k ovládní mapletu, další skupiny jsou určeny k propojování jednotlivých prvků. Významný prostor je věnován prvkům sloužícím k interakci uživatele s výpočtem. Tyto prvky mohou být použity k zadávání informací ovlivňující výpočet uživatelem, nebo k interpretaci získaných výstupů.

V podkapitole tvorba mapletů jsou shrnuty možnosti tvorby s použitím prvků představených v předchozích podkapitolách. Práce je postavena tak, aby mohla být použita jako manuál pro tvorbu a práci s maplety uživateli s minimálními znalostmi programu Maple.

Samotná sada mapletů je sloučena v jeden maplet ovládaný jednoduchou grafickou nabídkou, která umožňuje přepínat mezi jednotlivými maplety. Ty obsahují výpočty spojené s nejčtenějšími finančními produkty na trhu. Sada disponuje maplety se základními jednoduchými výpočty například spotřebitelské úvěry, které jsou snadno dohledatelné v internetových kalkulačkách. Osahuje ale také sofistikovanější výpočty typu RPSN spojené s problematikou spotřebitelských úvěrů, které nejsou běžně dostupné. Zajímavá je také změna ve výpočtech penzijního připojištění před a po zákonné změně. Nesporně přínosná je i možnost ve vybraných mapletech přepínat mezi frekvencemi splácení.

Ač je možnost tvorby mapletů stará přes čtrnáct let, představuje silný nástroj který je minimálně využíván. Krása programu Maple, respektive mapletů, spočívá jednak v jejich jednoduchosti, ale také v širokém uplatnění v jednotlivých exaktních vědách.

Literatura

- [1] ROSA, Přemysl. *Interaktivní aplikace pro výuku finanční matematiky*, České Budějovice: Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta, [2013].
- [2] *Maple 2015 includes powerful tools for access, analysis, and visualization of data* [online]. Datum vytvoření www stránek nedostupné, [cit. 2015-4-17]. Dostupné z WWW: <<http://www.maplesoft.com/company/publications/articles/view.aspx?SID=153764>>
- [3] BERNARDIN, L., CHIN, P., DEMARCO, P., et. al. *Maple Programming Guide* [online]. 4. 3. 2014, [cit. 2015-4-17]. Dostupné z WWW: <http://www.maplesoft.com/documentation_center/maple18/ProgrammingGuide.pdf>
- [4] CURRIER, Madalaine. *Waterloo Maple Introduces MapletsTM* [online]. 6. 12. 2001, [cit. 2015-4-17]. Dostupné z WWW: <<http://www.maplesoft.com/view.aspx?SF=41/2001-12-06.pdf>>
- [5] GAMBLE, Greg, KEADY, Grant. *MapleNet and Maplets under Maple 8* [online]. 18. 2. 2003, [cit. 2015-4-17]. Dostupné z WWW: <<http://www.maplesoft.com/view.aspx?SF=138/maplenetmaplets.pdf>>
- [6] CURRIER, Madalaine. *Maple 8 Now Shipping* [online]. 15. 5. 2002, [cit. 2015-4-17]. Dostupné z WWW: <<http://www.maplesoft.com/view.aspx?SF=33/2002-05-16.pdf>>
- [7] MEADE, Douglas B. *Maple 8 and Maplets: A New Use of Computer Algebra for Teaching Mathematics and Science* [online]. 14. 2. 2003, [cit. 2015-4-17]. Dostupné z WWW: <<http://people.math.sc.edu/meade/bmcc-maplets/proceedings/BMCCmeadePROC.pdf>>
- [8] *Maplets Package Updates* [online]. Datum vytvoření www stránek nedostupné, [cit. 2015-4-17]. Dostupné z WWW: <<http://www.maplesoft.com/support/help/Maple/view.aspx?path=Maplets/updates>>
- [9] CURRIER, Madalaine. *Waterloo Maple Launches MapleNetTM* [online]. 22. 7. 2002, [cit. 2015-4-17]. Dostupné z WWW: <<http://www.maplesoft.com/view.aspx?SF=32/2002-07-23.pdf>>
- [10] *Major New Release of Maplesoft Flagship Product Radically Improves Ease of Use* [online]. 16. 5. 2005, [cit. 2015-4-17]. Dostupné

z WWW: <<http://www.maplesoft.com/company/news/html/2005-05-16-maple10launch.aspx>>

Seznam příloh

Příloha A - Zdrojový kód	I
--------------------------------	---

Příloha A - Zdrojový kód

```

> restart;
> with(Maplets[Elements]) :
> an := proc (Dl, i, n, m) options operator, arrow;
local T:
T := convert(Maplets:-Tools:-Get('DB2'), string) :
if T = DBtext1 then  $\left( \frac{Dl * i * 0.01}{m} \right) / \left( \left( 1 - \left( 1 + i * \frac{0.01}{m} \right)^{-n \cdot m} \right) \left( 1 + i * \frac{0.01}{m} \right) \right)$ 
else  $\left( \frac{Dl * i * 0.01}{m} \right) / \left( 1 - \left( 1 + i * \frac{0.01}{m} \right)^{-n \cdot m} \right)$  end if:
end proc:
> an2 := proc (Dl, i, n) options operator, arrow; (Dl * i * 0.01) / (1 - (1 + i * 0.01)^(
-n)) end proc:
> ani := proc (i, n, m) options operator, arrow; (1 - v(i, m)^n) /  $\left( i * \frac{0.1e-1}{m} \right)$  end proc:
> KN := proc(aa, nn, ii);
local k, M, a, b, c;
M := 0;
b :=  $\frac{0.85 \cdot ii}{1200}$ ;
for k from 1 to nn do c := (nn - k) * 12; M := M + (aa * (1 + b)^c) end do;
M
end proc:
> MM := proc (m) options operator, arrow;
local TT;
TT := convert(Maplets:-Tools:-Get('DB1'), string);
if TT = "měsíčně" then 12 else if TT = "čtvrtletně" then 4 else 1 end if end if:
end proc:
> PH := proc (C, n, i, NH) options operator, arrow;
-1 * (((C/100) * NH) * (1 / (1 + i/100))^n - ((C/100) * NH) - NH * (1 + i/100)^(
-n) * (i/100)) / (i/100) end proc:
> POP := proc (C, n, vp, p) options operator, arrow;
 $\left( C \cdot \left( \frac{vp}{100} \right) + n \cdot p \right)$ 
end proc:
> PREVOD := proc(aa, r) options operator, arrow;
if aa ≤ 0 then 0 else if aa ≥ r then r else aa end if end if:
end proc:
> RPSN := proc (K0, a, n, P) options operator, arrow;
local pom, r;
pom := K0 = sum(a / (1 + r)^(k/12), k = 1 .. n) + P;
fsolve(pom, r) * 100
end proc:
> RPSN2 := proc (K0, a, n, P, spl, poc) options operator, arrow;
local pom, r, nn, poc1, ff;
if poc = "týdnu" then poc1 := n * 7 else poc1 :=  $n \cdot \frac{365}{12}$  end if;
if spl = "měsíčně" then nn := ceil $\left( \frac{poc1}{\left( \frac{365}{12} \right)} \right)$ ; ff :=  $\frac{1}{12}$ ; else nn := ceil $\left( \frac{poc1}{7} \right)$ ; ff

```

```

:=  $\frac{7}{365}$  end if;
pom := K0 = sum(a / (1 + r)^(k*ff), k = 1 ..nn) + P;
fsolve(pom, r) * 100
end proc:
> RPSN3 := proc (K0, a, n) options operator, arrow;
local pom, r;
pom := K0 = (a + K0) / (1 + r)^(n/365);
fsolve(pom, r) * 100
end proc:
> S := proc (a, n, m, i) options operator, arrow;
(a * (1 + 0.85 * i / (m * 100)) * ((1 + 0.85 * (i / (m * 100)))^(n * m) - 1)) / (0.85 * (i / (m * 100)))
end proc:
> S2 := proc (a, n, m, i) options operator, arrow;
local T;
T := convert(Maplets:-Tools:-Get('DBI'), string);
if T = DBtext1 then (a * (1 + 0.85 * i / (m * 100)) * ((1 + 0.85 * (i / (m * 100)))^(n * m) - 1)) / (0.85 * (i / (m * 100)));
else (a * ((1 + 0.85 * (i / (m * 100)))^(n * m) - 1)) / (0.85 * (i / (m * 100))); end if;
end proc:
> S3 := proc (a, n, i) options operator, arrow;
(a * ((1 + (i / 1200))^(n * 12) - 1)) / (i / 1200)
end proc:
> S4 := proc (a, n, i) options operator, arrow;
a *  $\left(1 + \frac{i}{1200}\right)^{(12 \cdot n)}$ 
end proc:
> S5 := proc (a, n, i) options operator, arrow;
(a * ((1 + 0.85 * (i / 1200))^(n * 12) - 1)) / (0.85 * (i / 1200))
end proc:
> SPNov := proc (b) options operator, arrow;
if b < 300 then 0 elif b < 1000 then (90 + 0.2 * (b - 300)) elif b ≥ 1000 then 230 end if;
end proc:
> SPSta := proc (b) options operator, arrow;
if b < 100 then 0 elif b < 200 then (50 + 0.4 * (b - 100)) elif b < 300 then (90 + 0.3 * (b - 200)) elif b < 400 then (120 + 0.2 * (b - 300)) elif b < 500 then (140 + 0.1 * (b - 400)) elif b ≥ 500 then 150 end if;
end proc:
> STAT := proc(a) options operator, arrow;
local sp;
sp := round(a * 1.2);
if sp > 2000 then 2000 else sp end if;
end proc:
> SUMA := proc(a, n, P, spl, poc) options operator, arrow;
local poc1, nn, K;
if poc = 'týdnů' then poc1 :=  $\frac{365}{7}$  else poc1 := 12 end if;
if spl = 'měsíčně' then nn := ceil(n * 12 / poc1) else if spl = 'ročně' then nn := ceil(n * 1 / poc1) else nn := ceil(n *  $\frac{365}{7}$  / poc1) end if end if;
nn * a + P

```

```

end proc:
> Umor := proc (Dl, i, n, m) options operator, arrow;
local a, n1 :
a := an(Dl, i, n, m) :
n1 := n·m :
matrix([[období, splátka, úrok, úmor, zůstatek], [0, ``; ``; ``; Dl], seq([j, Z2D(a), Z2D(a
* (1-v(i, m)(n1 - j + 1))), Z2D(a * v(i, m)(n1 - j + 1)), Z2D(a * ani(i, n1 - j,
m)), j=1 .. n1 - 1], [n1, Z2D(a), Z2D(a * (1-v(i, m))), Z2D(a * v(i, m)), ``],
[``; ``; ``; ``; ``], [``; Z2D(n1 * a), Z2D(n1 * a - Dl), Dl, ``]])
end proc:
> v := proc (i, m) options operator, arrow; 1 / ( 1 + 0.1e-1 *  $\frac{i}{m}$  ) end proc:
> Z2D := proc (c) options operator, arrow; evalf(0.1e-1 * round(100 * c)) end proc:
> DBtext1 := "Připisování úroků na začátku úrokovacího období" : DBtext2
:= "Připisování úroků na konci úrokovacího období" :
>
> mapletMENU := Maplet(Window['WMenu']('title'="Hlavní
nabídka",
BoxLayout('vertical'='true', 'inset'=10, 'background'='
white',
BoxRow('border'='true', [BoxRow("Dluhopisy"),
BoxRow(
BoxCell(Button("Maplet #1", 'tooltip'="Maplet na
výpočet teoretické hodnoty dluhopisu", Evaluate('function'='
Maplets[Display](mapletDLH)')))),
BoxRow('border'='true', [BoxRow("Klasické spoření"),
BoxRow(
HorizontalGlue(),
BoxCell(Button("Maplet #1", 'tooltip'="Maplet na
výpočet klasického spoření", Evaluate('function'='Maplets
[Display](mapletKS1)'))),
BoxCell(Button("Maplet #2", 'tooltip'="Maplet na
výpočet klasického spoření s možností výběru
úročení", Evaluate('function'='Maplets[Display](mapletKS2)'))
)),
HorizontalGlue()))),
BoxRow('border'='true', [BoxRow("Penzijní
připojištění"),
BoxRow(
HorizontalGlue(),
BoxCell(Button("Maplet #1", 'tooltip'="Maplet na
výpočet penzijního připojištění založeného po 1. lednu
2013", Evaluate('function'='Maplets[Display](mapletPP1)'))),
BoxCell(Button("Maplet #2", 'tooltip'="Maplet na
výpočet penzijního připojištění s možností zadání
roku založení (změna státních příspěvků od 1.1.2013)",
Evaluate('function'='Maplets[Display](mapletPP2)')))),

```



```

        BoxRow(HorizontalGlue(),TextField['TFC'](10),"%",
HorizontalGlue()),
        BoxRow(),
        BoxRow("Nominální Hodnota NH: "),
        BoxRow(HorizontalGlue(),TextField['TFNH'](10),
"Kč",HorizontalGlue()))),

        BoxRow('border'='false','background'='white',Label
("Výpočet provedeme pomocí vzorce:",'font'=Font("Times",14))
),
        BoxRow([MathMLViewer('value' = MathML[Export](P=(C*
(1+i)^n-C+NH*i)/(i*(1+i)^n),'width'='250','height'='100',
'fontsize'=12)]),
        BoxRow(Button("Vypočítat teoretickou hodnotu",
Action ( Evaluate ('TB1' = 'evalf(PH(TFC, TFn, TFi, TFNH))' )
)),
        BoxRow('border'='true','background'='white',Label
("Teoretická hodnota dluhopisu:",'font'=Font("Times",14))),
        BoxRow(HorizontalGlue(),TextBox['TB1']('editable' =
'false','font'=Font("Times",12), 1..20),"Kč",HorizontalGlue()
),

        BoxRow('border'='true',
        BoxCell(Button("Vymazat",Action(SetOption('TFC' =
"0"),SetOption('TFi' = "0"),SetOption('TFn' = "0"),SetOption
('TFNH' = "0"),SetOption('TB1' = ""))),),
        BoxCell(Button("Návrat do Menu",Shutdown()))))
)):
> mapletKS1:= Maplet(Window['WKS1']('title'="Klasické spoření
#1",
        BoxLayout('vertical'='true','inset'=15,
        BoxRow('halign'='center','border'='false',
'background'='white',Label("Výpočet klasického spoření",
'font'=Font("Times",14))),
        BoxRow('halign'='center','valign'='none','border'='
'false',TextBox['TBUP']("úroky daněny
úložka a připisování úroků shodné",'font'=Font("Times",13),
'editable'='false',2..27)),

        BoxRow('border'='true',
        BoxColumn(
        BoxRow("Pravidelný vklad:"),
        BoxRow(HorizontalGlue(),TextField['TFvklad'](10),
"Kč",HorizontalGlue()),
        BoxRow("Úroková míra:"),

```

```

        BoxRow(HorizontalGlue(),TextField['TFi'](10),"%",
HorizontalGlue()),
        BoxColumn(
        BoxRow("Doba spoření:"),
        BoxRow(HorizontalGlue(),TextField['TFn'](10),
"let",HorizontalGlue()),
        BoxRow("Počet vkladů ročně:"),
        BoxRow(HorizontalGlue(),TextField['TFm'](10),"
",HorizontalGlue()))),

        BoxRow(Button("Vypočítat",Action(Evaluate('TB1'='S
(TFvklad,TFn,TFm,TFi)'),
        Action(Evaluate('TB2' = 'TFn'*'TFm'*'TFvklad'),
Action(Evaluate ('TB3' = 'TB1'-'TB2'))))),),
        BoxRow(
        BoxColumn('halign'='left',
        BoxRow("Naspořená suma:"),
        BoxRow("Vklady:"),
        BoxRow("Úroky:")),
        BoxColumn(
        BoxRow(TextBox['TB1']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
        BoxRow(TextBox['TB2']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
        BoxRow(TextBox['TB3']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"))),

        BoxRow('border'='true',
        BoxCell(Button("Vymazat",Action(SetOption
('TFvklad' = 0),SetOption('TFn' = 0),SetOption('TFi' = 0),
SetOption('TFm' = 0),SetOption('TB1' = ""),SetOption('TB2' =
""),SetOption('TB3' = ""))),),
        BoxCell(Button("Návrat do Menu",Action(Shutdown(
['TFvklad', 'TFn', 'TFi', 'TFm', 'TB1', 'TB2', 'TB3'])))
))):
> mapletKS2:= Maplet(Window['WKS2']('title'="Klasické spoření
#2",
        BoxLayout('vertical'='true', 'inset'=15,
        BoxRow('halign'='center', 'border'='false',
'background'='white',Label("Výpočet klasického spoření",
'font'=Font("Times",14))),
        BoxRow('halign'='center', 'valign'='none', 'border'=
'false',TextBox['TBUP']("úroky daněny
úložka a připisování úroků shodné
výběr úročení (předlůhnutí nebo polhůtní)"),'font'=Font

```



```

("Times",13),'editable'='false',3..27)),

    BoxRow('border'='true',[BoxRow(
        BoxColumn(
            BoxRow("Pravidelný vklad:"),
            BoxRow(HorizontalGlue(),TextField['TFvklad'](10),
"Kč",HorizontalGlue()),
            BoxRow("Úroková míra:"),
            BoxRow(HorizontalGlue(),TextField['TFi'](10),"%",
HorizontalGlue()))),

        BoxColumn(
            BoxRow("Doba spoření:"),
            BoxRow(HorizontalGlue(),TextField['TFn'](10),
"let",HorizontalGlue()),
            BoxRow("Počet vkladů ročně:"),
            BoxRow(HorizontalGlue(),TextField['TFm'](10),"
",HorizontalGlue()))),

        BoxRow(DropDownBox['DB1'](DBtext1,[DBtext1,DBtext2]
)))]),

    BoxRow(Button("Vypočítat",Action(Evaluate('TB1' =
'S2(TFvklad,TFn,TFm,TFi)'),Action(Evaluate('TB2' = 'TFn'*
'TFm'*'TFvklad'),Action(Evaluate ('TB3' = 'TB1'-'TB2'))))),
    BoxRow(
        BoxColumn('halign'='left',
            BoxRow("Naspořená suma:"),
            BoxRow("Vklady:"),
            BoxRow("Úroky:")),

        BoxColumn(
            BoxRow(TextBox['TB1']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
            BoxRow(TextBox['TB2']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
            BoxRow(TextBox['TB3']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"))),

    BoxRow('border'='true',
        BoxCell(Button("Vymazat",Action(SetOption
('TFvklad' = 0),SetOption('TFn' = 0),SetOption('TFi' = 0),
SetOption('TFm' = 0),SetOption('TB1' = ""),SetOption('TB2' =
""),SetOption('TB3' = ""))),
        BoxCell(Button("Návrat do Menu",Action(Shutdown(

```

```

    ['TFvklad', 'TFn', 'TFi', 'TFm', 'TB1', 'TB2', 'TB3', 'DB1']]))))
  ))):
> mapletPP1:= Maplet(Window['WPP1']('title'="Penzijní
připojištění #1",
    BoxLayout(vertical=true,inset=15,
      BoxRow('halign'='center','border'='false',
'background'='white',Label("Výpočet penzijního připojištění",
'font'=Font("Times",14))),

      BoxRow('border'='true',
        BoxColumn(
          BoxRow("Měsíční příspěvek účastníka:"),
          BoxRow(HorizontalGlue(),TextField['TFucas'](10),
"Kč",HorizontalGlue()),
          BoxRow("Předpokládané zhodnocení:"),
          BoxRow(HorizontalGlue(),TextField['TFi'](10),"%",
HorizontalGlue()))),

        BoxColumn(
          BoxRow("Měsíční příspěvek zaměstnavatele:"),
          BoxRow(HorizontalGlue(),TextField['TFzam'](10),
"Kč",HorizontalGlue()),
          BoxRow("Počet roků placeného pojištění:"),
          BoxRow(HorizontalGlue(),TextField['TFn'](10),
"let",HorizontalGlue()))),

      BoxRow(Button("Vypočítat",
Action(Evaluate('TB1' = 'evalf(TFucas*TFn*12)'),
Action(Evaluate('TB2' = 'trunc(evalf(S3(TFucas,TFn,
TFi)-'TB1')))'),
Action(Evaluate('TB3' = 'evalf(TFzam*TFn*12)'),
Action(Evaluate('TB4' = 'trunc(evalf(S3(TFzam,TFn,
TFi)-'TB3')))'),
Action(Evaluate('TB5' = 'evalf(SPNov(TFucas)*TFn*
12)'),
Action(Evaluate('TB6' = 'trunc(evalf(S3(SPNov
(TFucas),TFn,TFi)-'TB5')))'),
Action(Evaluate('TB7' = ''TB1'+''TB2'+''TB3'+''TB4'+
'TB5'+''TB6'''),
Action(Evaluate('TB8' = ''TB7'-(0.15*( 'TB2'+''TB3'+
'TB4'+''TB6'''))')
)))))))))
    BoxRow(
    BoxColumn('halign'='left',

```

```

BoxRow("Celkové příspěvky účastníka:"),
BoxRow("Výnosy z příspěvků účastníka:"),
BoxRow("Celkové příspěvky zaměstnavatele:"),
BoxRow("Výnosy z příspěvků zaměstnavatele:"),
BoxRow("Celkové státní příspěvky:"),
BoxRow("Výnosy ze státních příspěvků:"),
BoxRow("Jednorázové vyrovnání před zdaněním:"),
BoxRow("Jednorázové vyrovnání po zdanění:")),

BoxColumn(
    BoxRow(TextBox['TB1']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
    BoxRow(TextBox['TB2']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
    BoxRow(TextBox['TB3']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
    BoxRow(TextBox['TB4']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
    BoxRow(TextBox['TB5']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
    BoxRow(TextBox['TB6']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
    BoxRow(TextBox['TB7']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
    BoxRow(TextBox['TB8']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"))),

BoxRow('border'='true',
    BoxCell(Button("Vymazat",Action(SetOption('TFucas'
= 0),SetOption('TFi' = ""),SetOption('TFn' = ""),SetOption
('TFzam' = 0),SetOption('TB1' = ""),SetOption('TB2' = ""),
SetOption('TB3' = ""),SetOption('TB4' = ""),SetOption('TB5' =
""),SetOption('TB6' = ""),SetOption('TB7' = ""),SetOption
('TB8' = ""))),
    BoxCell(Button("Návrat do Menu",Action(Shutdown(
['TFucas', 'TFn', 'TFi', 'TFzam', 'TB1', 'TB2', 'TB3', 'TB4',
'TB5', 'TB6', 'TB7', 'TB8']))))))
)):
> mapletPP2:= Maplet(Window['WPP2']('title'="Penzijní
připojištění #2",
    BoxLayout(vertical=true,inset=15,
        BoxRow('halign'='center','border'='false',
'background'='white',Label("Výpočet penzijního připojištění",
'font'=Font("Times",14))),

```

```

BoxRow('border'='true',[BoxRow(
  BoxColumn(
    BoxRow("Měsíční příspěvek účastníka:"),
    BoxRow(HorizontalGlue(),TextField['TFucas'](10),
    "Kč",HorizontalGlue()),
    BoxRow("Předpokládané zhodnocení:"),
    BoxRow(HorizontalGlue(),TextField['TFi'](10),"%",
HorizontalGlue()))),

  BoxColumn(
    BoxRow("Měsíční příspěvek zaměstnavatele:"),
    BoxRow(HorizontalGlue(),TextField['TFzam'](10),
    "Kč",HorizontalGlue()),
    BoxRow("Počet roků placeného pojištění:"),
    BoxRow(HorizontalGlue(),TextField['TFn'](10),
    "let",HorizontalGlue()))),

  BoxRow(HorizontalGlue(),"Rok uzavření smlouvy:
",TextField['TFrok'](10),HorizontalGlue()))),

  BoxRow(Button("Vypočítat",
  Action(Evaluate('TB1' = 'evalf(TFucas*TFn*12)'),
  Action(Evaluate('TB2' = 'trunc(evalf(S3(TFucas,TFn,
TFi)-'TB1'))'),
  Action(Evaluate('TB3' = 'evalf(TFzam*TFn*12)'),
  Action(Evaluate('TB4' = 'trunc(evalf(S3(TFzam,TFn,
TFi)-'TB3'))'),
  Action(Evaluate('TB5' = 'evalf(SPSta(TFucas)*PREVOD
(2013-TFrok,TFn)*12+SPNov(TFucas)*(TFn-PREVOD(2013-TFrok,TFn)
)*12)'),
  Action(Evaluate('TB6' = 'trunc(evalf(S4(S3(SPSta
(TFucas),PREVOD(2013-TFrok,TFn),TFi),(TFn-PREVOD(2013-TFrok,
TFn)),TFi)+S3(SPNov(TFucas),(TFn-PREVOD(2013-TFrok,TFn)),TFi)
-'TB5'))'),
  Action(Evaluate('TB7' = ''TB1'+''TB2'+''TB3'+''TB4'+
'TB5'+''TB6'')),
  Action(Evaluate('TB8' = ''TB7'-(0.15*( 'TB2'+''TB3'+
'TB4'+''TB6''))'))
))))))))) ,
BoxRow(
BoxColumn('halign'='left',
  BoxRow("Celkové příspěvky účastníka:"),
  BoxRow("Výnosy z příspěvků účastníka:"),
  BoxRow("Celkové příspěvky zaměstnavatele:"),
  BoxRow("Výnosy z příspěvků zaměstnavatele:"),

```

```

        BoxRow("Celkové státní příspěvky:"),
        BoxRow("Výnosy ze státních příspěvků:"),
        BoxRow("Jednorázové vyrovnání před zdaněním:"),
        BoxRow("Jednorázové vyrovnání po zdanění:")),

        BoxColumn(
            BoxRow(TextBox['TB1']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
            BoxRow(TextBox['TB2']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
            BoxRow(TextBox['TB3']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
            BoxRow(TextBox['TB4']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
            BoxRow(TextBox['TB5']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
            BoxRow(TextBox['TB6']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
            BoxRow(TextBox['TB7']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
            BoxRow(TextBox['TB8']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"))),

        BoxRow('border'='true',
            BoxCell(Button("Vymazat",Action(SetOption('TFucas'
= 0),SetOption('TFi' = 0),SetOption('TFrok' = "2013"),
SetOption('TFn' = 0),SetOption('TFzam' = 0),SetOption('TB1' =
""),SetOption('TB2' = ""),SetOption('TB3' = ""),SetOption
('TB4' = ""),SetOption('TB5' = ""),SetOption('TB6' = ""),
SetOption('TB7' = ""),SetOption('TB8' = ""))),
            BoxCell(Button("Návrat do Menu",Action(Shutdown(
['TFucas', 'TFn', 'TFi', 'TFzam', 'TFrok', 'TB1', 'TB2', 'TB3',
'TB4', 'TB5', 'TB6', 'TB7', 'TB8']))))))
    )):
> mapletRPSN1:= Maplet(Window['WRPSN1']('title'="Výpočet RPSN
#1",
        BoxLayout(vertical=true,inset=15,
            BoxRow('halign'='center','border'='false',
'background'='white',Label("Výpočet RPSN pro spotřebitelský
úvěr", 'font'=Font("Times",14))),

            BoxRow('border'='true',
                BoxColumn(
                    BoxRow("Výše úvěru:"),
                    BoxRow(HorizontalGlue(),TextField['TFuv'](10),

```

```

"Kč",HorizontalGlue()),
    BoxRow("Jednorázový poplatek:"),
    BoxRow(HorizontalGlue(),TextField['TFpop'](10),
"Kč",HorizontalGlue()))),

    BoxColumn(
    BoxRow("Měsíční splátka:"),
    BoxRow(HorizontalGlue(),TextField['TFsp'](10),
"Kč",HorizontalGlue()),
    BoxRow("Počet měsíců splácení:"),
    BoxRow(HorizontalGlue(),TextField['TFn'](10),
HorizontalGlue()))),

    BoxRow(Button("Vypočítat",Action ( Evaluate ('TB1'
= 'RPSN(TFuv,TFsp,TFn,TFpop)'),
    Action ( Evaluate ('TB2' = 'TFsp'*'TFn'+TFpop'))))),

    BoxRow(
    BoxColumn('halign'='left',
    BoxRow("RPSN:"),
    BoxRow("Splacená suma:")),

    BoxColumn(
    BoxRow(TextBox['TB1']('editable' = 'false',
'font'=Font("Times",12), 1..15),"%"),
    BoxRow(TextBox['TB2']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč")),

    BoxRow('border'='true',
    BoxCell(Button("Vymazat",Action(SetOption('TFuv' =
0),SetOption('TFsp' = 0),SetOption('TFn' = 0),SetOption
('TFpop' = 0),SetOption('TB1' = ""),SetOption('TB2' = ""))),
    BoxCell(Button("Návrat do Menu",Action(Shutdown(
['TFuv', 'TFsp', 'TFn', 'TFpop', 'TB1', 'TB2']))))
    ))):
> mapletRPSN2:= Maplet(Window['WRPSN2']('title'="Výpočet RPSN
#2",
    BoxLayout(vertical=true,inset=15,
    BoxRow('halign'='center','border'='false',
'background'='white',Label("Výpočet RPSN pro spotřebitelský
úvěr",'font'=Font("Times",14))),

    BoxRow('border'='true',
    BoxColumn(
    BoxRow("Výše úvěru:"),

```

```

        BoxRow(HorizontalGlue(),TextField['TFuv'](10),
        "Kč",HorizontalGlue()),
        BoxRow("Jednorázový poplatek:"),
        BoxRow(HorizontalGlue(),TextField['TFpop'](10),
        "Kč",HorizontalGlue()))),

        BoxColumn(
        BoxRow("Pravidelná splátka:"),
        BoxRow(HorizontalGlue(),TextField['TFsp'](10),
        "Kč",DropDownBox['DB1']("týdně",["týdně","měsíčně"]
        ),HorizontalGlue()),
        BoxRow("Doba splácení:"),
        BoxRow(HorizontalGlue(),TextField['TFn'](10),"
        ",DropDownBox['DB2']("týdnů",["týdnů","měsíců"]
        ),HorizontalGlue()))),

        BoxRow(Button("Vypočítat",
        Action(Evaluate('TB1' = 'RPSN2(TFuv,TFsp,TFn,TFpop,
        DB1,DB2)')),
        Action(Evaluate('TB2' = 'SUMA(TFsp,TFn,TFpop,DB1,
        DB2)'))
        ))),

        BoxRow(
        BoxColumn('halign'='left',
        BoxRow("RPSN:"),
        BoxRow("Splacená suma:")),

        BoxColumn(
        BoxRow(TextBox['TB1']('editable' = 'false',
        'font'=Font("Times",12), 1..15),"%"),
        BoxRow(TextBox['TB2']('editable' = 'false',
        'font'=Font("Times",12), 1..15),"Kč")),

        BoxRow('border'='true',
        BoxCell(Button("Vymazat",Action(SetOption('TFuv' =
        0),SetOption('TFsp' = 0),SetOption('TFn' = 0),SetOption
        ('TFpop' = 0),SetOption('TB1' = ""),SetOption('TB2' = ""))),
        BoxCell(Button("Návrat do Menu",Action(Shutdown(
        ['TFuv', 'TFsp', 'TFn', 'TFpop', 'TB1', 'TB2'])))
        ))):
> mapletRPSN3:= Maplet(Window['WRPSN3']('title'="Výpočet RPSN
#3",
        BoxLayout(vertical=true,inset=15,
        BoxRow('halign'='center','border'='false',

```

```

'background'='white',Label("Výpočet RPSN pro spotřebitelský
úvěr", 'font'=Font("Times",14))),

    BoxRow('border'='true',
    BoxColumn('halign'='left',
    BoxRow("Výše úvěru:"),
    BoxRow("Poplatek:"),
    BoxRow("Splatnost:")),

    BoxColumn(
    BoxRow(TextField['TFuv'](10)," KČ "),
    BoxRow(TextField['TFpop'](10)," KČ "),
    BoxRow(TextField['TFn'](10),"dnů")),

    BoxRow(Button("Vypočítat",
    Action(Evaluate('TB1' = 'RPSN3(TFuv,TFpop,TFn)'),
    Action(Evaluate('TB2' = 'TFuv'+'TFpop'))))),

    BoxRow(
    BoxColumn('halign'='left',
    BoxRow('halign'='left',"RPSN:"),
    BoxRow('halign'='left',"Splacená suma:")),

    BoxColumn(
    BoxRow(TextBox['TB1']('editable' = 'false',
'font'=Font("Times",12), 1..15),"%"),
    BoxRow(TextBox['TB2']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč")),

    BoxRow('border'='true',
    BoxCell(Button("Vymazat",Action(SetOption('TFuv' =
0),SetOption('TFn' = 0),SetOption('TFpop' = 0),SetOption
('TB1' = ""),SetOption('TB2' = ""))),
    BoxCell(Button("Návrat do Menu",Action(Shutdown(
['TFuv', 'TFn', 'TFpop','TB1','TB2']))))))
)):
> mapletSU:= Maplet(Window['WSU']('title'="Spotřebitelský
úvěr",
    BoxLayout('vertical'='true', 'inset'=15,
    BoxRow('halign'='center', 'border'='false',
'background'='white',Label("Spotřebitelský úvěr", 'font'=Font
("Times",14))),

    BoxRow('border'='true',[
    BoxRow(

```



```

        BoxColumn(
            BoxRow("Výše úvěru:"),
            BoxRow(HorizontalGlue(), TextField['TFuv'](10),
                "Kč", HorizontalGlue()),
            BoxRow("Doba splácení:"),
            BoxRow(HorizontalGlue(), TextField['TFn'](10),
                "roků", HorizontalGlue()))),

        BoxColumn(
            BoxRow("Roční úroková míra:"),
            BoxRow(HorizontalGlue(), TextField['TFi'](10), "%",
                HorizontalGlue()),
            BoxRow("Platba splátek:"),
            BoxRow(HorizontalGlue(), DropDownBox['DB1']
                ("měsíčně", ["měsíčně", "čtvrtletně", "ročně"]), HorizontalGlue()
            )),

            BoxRow(DropDownBox['DB2'](DBtext1, [DBtext1,
                DBtext2]))),

            BoxRow(Button("Vypočítat",
                Action(Evaluate('TB1' = 'Z2D(evalf(an(TFuv, TFi,
                    TFn, MM(DB1))))')),
                Action(Evaluate('MMLV1' = 'MathML[Export](evalm
                    (Umor(TFuv, TFi, TFn, MM(DB1))))')))),

            BoxRow(
                BoxColumn(BoxRow(HorizontalGlue(), "Výše
                    splátky:")),
                BoxColumn(BoxRow(TextBox['TB1']('editable' =
                    'false', 'font'=Font("Times", 12), 1..15), "Kč", HorizontalGlue()
                ))),

            BoxRow("Umožňovací plán"),
            BoxRow((MathMLViewer['MMLV1'](value="", 'wrapped'=
                'true', 'width'='450', 'height'='300'))),

            BoxRow('border'='true',
                BoxCell(Button("Vymazat", Action(SetOption('TFuv' =
                    0), SetOption('TFi' = 0), SetOption('TFn' = 0), SetOption
                    ('MMLV1' = ""), SetOption('TB1' = "")))),
                BoxCell(Button("Návrat do Menu", Action(Shutdown(
                    ['TFuv', 'TFi', 'TFn', 'TB1'])))
                ))):
> mapletSS:= Maplet(Window['WSS']('title'="Stavební spoření",

```

```

BoxLayout('vertical'='true', 'inset'=15,
    BoxRow('halign'='center', 'border'='false',
'background'='white', Label("Výpočet stavebního spoření",
'font'=Font("Times",14))),

    BoxRow('halign'='center', 'border'='true',
    BoxColumn(
        BoxRow("Pravidelný měsíční vklad:"),
        BoxRow(HorizontalGlue(), TextField['TFvklad'](10),
"Kč", HorizontalGlue()),
        BoxRow("Úroková míra:"),
        BoxRow(HorizontalGlue(), TextField['TFi'](10), "%",
HorizontalGlue()),
        BoxRow("Vstupní poplatek (z cílové částky):"),
        BoxRow(HorizontalGlue(), TextField['TFvpop'](10),
"%", HorizontalGlue()))),

    BoxColumn(
        BoxRow("Cílová částka:"),
        BoxRow(HorizontalGlue(), TextField['TFcil'](10),
"Kč", HorizontalGlue()),
        BoxRow("Počet roků spoření:"),
        BoxRow(HorizontalGlue(), TextField['TFn'](10),
"let", HorizontalGlue()),
        BoxRow("Roční poplatek (vedení účtu a výpisy):"),
        BoxRow(HorizontalGlue(), TextField['TFpop'](10),
"Kč", HorizontalGlue()))),

    BoxRow(Button("Vypočítat",
Action(Evaluate('TB1' = 'POP(TFcil,TFn,TFvpop,TFpop)'),
Action(Evaluate('TB2' = 'TFn'*12*'TFvklad'),
Action(Evaluate('TB3' = 'S5(TFvklad,TFn,TFi)'-'TB2'),
Action(Evaluate('TB4' = 'STAT(TFvklad)'*'TFn'),
Action(Evaluate('TB5' = 'KN(STAT(TFvklad),TFn,TFi)'-'TB4'),
Action(Evaluate('TB6' = ('TB2'+ 'TB3'+ 'TB4'+ 'TB5')-'TB1'
)))))))),

    BoxRow(
    BoxColumn('halign'='left',
        BoxRow("Poplatky:"),
        BoxRow("Příspěvky účastníka:"),
        BoxRow("Výnosy z příspěvků účastníka:"),
        BoxRow("Státní příspěvky:"),
        BoxRow("Výnosy ze státních příspěvků:"),
        BoxRow("Naspořená částka:")),

```

```

        BoxColumn(
            BoxRow(TextBox['TB1']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
            BoxRow(TextBox['TB2']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
            BoxRow(TextBox['TB3']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
            BoxRow(TextBox['TB4']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
            BoxRow(TextBox['TB5']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"),
            BoxRow(TextBox['TB6']('editable' = 'false',
'font'=Font("Times",12), 1..15),"Kč"))),

        BoxRow('border'='true',
            BoxCell(Button("Vymazat",Action(SetOption
('TFvklad' = 0),SetOption('TFcil' = 0),SetOption('TFi' = 0),
SetOption('TFn' = 0),SetOption('TFvpop' = 0),SetOption
('TFpop' = 0),SetOption('TB1' = ""),SetOption('TB2' = ""),
SetOption('TB3' = ""),SetOption('TB4' = ""),SetOption('TB5' =
""),SetOption('TB6' = ""))))),
            BoxCell(Button("Návrat do Menu",Action(Shutdown(
['TFvklad', 'TFn', 'TFi', 'TFcil', 'TFvpop', 'TFpop', 'TB1',
'TB2', 'TB3', 'TB4', 'TB5', 'TB6']))))))
    )):
    > Maplets[Display](mapletMENU);

```