

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Řešitel ekonomických úloh

Bakalářská práce

Vedoucí práce:

Ing. Pavel Haluza, Ph.D.

Robin Hynšt

Brno 2015

Poděkování

Chtěl bych poděkovat svojí přítelkyni a své rodině, že mi vytvořili podmínky pro klidné řešení a psaní závěrečné práce, a za jejich podporu a víru ve mně. Také bych chtěl poděkovat svému vedoucímu Ing. Pavlovi Haluzovi, Ph.D. za jeho ochotu, rady a spolupráci.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Řešitel ekonomických úloh** vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 21. května 2015

Abstract

Hynšt, Solver economic tasks. Bachelor thesis. Brno: Mendel university, 2015.
Bachelor thesis describes design, implementation and functionality of desktop application „Řešitel ekonomických úloh“. The completed application is created in the Java programming language and you can run in Windows, Linux and Mac OS. Application contains 25 economic arithmetic tasks, the user can create and add next new arithmetic tasks.

Keywords

Java, economic tasks, arithmetic tasks, Mendelu.

Abstrakt

Hynšt, Řešitel ekonomických úloh. Bakalářská práce. Brno: Mendelova univerzita v Brně, 2015.

Bakalářská práce popisuje návrh, implementaci a funkčnost desktopové aplikace „Řešitel ekonomických úloh“. Výsledná aplikace je zpracována v programovacím jazyce Java a je spustitelná na OS Windows, Linux a Mac OS. Aplikace po spuštění obsahuje 25 ekonomických početních problémů, uživatel má možnost si vytvořit a přidat další početní příklady.

Klíčová slova

Java, ekonomické problémy, početní příklady, Mendelu.

Obsah

1	Úvod a cíl práce	12
1.1	Úvod práce	12
1.2	Cíl práce.....	13
2	Rešerše	14
3	Nástroje a metodika	16
3.1	Visual paradigm for UML.....	16
3.2	Vývojové prostředí	17
3.2.1	Historie NetBeans	17
3.2.2	Výhody NetBeans	17
3.3	Java.....	18
3.3.1	Historie Javy	19
3.3.2	Verze Javy	19
3.4	Komponenty aplikace	20
3.5	Spouštění aplikace	22
3.5.1	Windows.....	22
3.5.2	Linux.....	22
3.5.3	MacOS	22
4	Vlastní práce	23
4.1	Množina ekonomických příkladů.....	23
4.1.1	Příklady makroekonomie	23
4.1.2	Příklady mikroekonomie.....	24
4.1.3	Nezařazené příklady	26
4.2	Návrh aplikace.....	28
4.2.1	Výběr vhodného nástroje.....	28
4.2.2	Popis diagramu tříd.....	28
4.2.3	Obecná funkcionalita aplikace	31
4.2.4	Design a návrh oken aplikace	32

4.3	Struktura aplikace	35
4.3.1	Formát ukládaných dat	35
4.3.2	Formát vstupních a výstupních dat.....	35
4.4	Chod aplikace	37
4.4.1	Chybové hlášky při operacích se soubory	37
4.4.2	Chybové hlášky při počítání příkladů.....	37
5	Diskuze	38
5.1	Ekonomické zhodnocení.....	38
5.2	Technické zhodnocení	38
6	Závěr	39
7	Literatura	41
A	Příložené DVD	45

1 Úvod a cíl práce

1.1 Úvod práce

Všechny ekonomické obory mají jako základní předměty Makroekonomii a Mikroekonomii. Každý kdo tyto předměty studoval, tak ví, že oba jsou dost složité, a náročné na pochopení. Když už se studentovi podaří pochopit smysl a pravidla ekonomických složitostí, čeká na něj množství početních příkladů.

Bez početních příkladů se neobejde žádná složka ekonomie. Při počítání příkladů ve cvičeních studentovi cvičící okamžitě dá zpětnou vazbu, jestli počítá správně, ale když se student připravuje sám na zápočet, či zkoušku nikdo mu neřekne, jestli výsledek je správný. Novým trendem jsou cvičebnice dodávané přesně k učebním textům, kde student nalezne i výsledky k početním příkladům. U zkoušek se ve většině případů objevují příklady pouze podobné, anebo s menší úpravou. Student potřebuje zvládat počítat i jiné příklady a potřebuje být schopen ověřit si správnost výsledků.

Přesně kvůli této myšlence se začala vyvíjet aplikace „Řešitel ekonomických problémů“. Aplikace, která dokáže na vstupní hodnoty vypočítat výsledek, a která dovolí uživateli upravovat si příklady pro alternativní použití.

Tento návrh má podporu pro to, aby cvičící mohli pomoci studentům s kontrolou výsledků. Ochetný cvičící si udělá chvíli času a vytvoří v aplikaci několik příkladů, které by se studentům mohli hodit, a tyto příklady vhodné pro použití v aplikaci dá studentům k dispozici. Takto cvičící ztratí chvíli svého drahocenného času, ale ve výsledku ho ušetří mnohem více, když nebude muset odpovídat jednotlivým studentům na dotazy přes mail, či vysvětlovat na konzultacích.

Aplikace není vyvíjena na operační systémy mobilních telefonů z důvodu možnosti využití k podvodům při zkouškách. Takto aplikace spustitelná pouze na desktopových počítačích bude využívána pouze k ověřování výsledků při přípravě.

Projekt má velký potenciál ve všech oborech, kde se provádí výpočty. Aplikace je na ekonomii omezena pouze svými základními příklady načtenými v aplikaci, jinak je použitelná na jakýkoliv typ příkladů.

1.2 Cíl práce

Cílem této práce je ulehčit studentům a vyučujícím kontrolu správnosti výsledků početních ekonomických příkladů. Práce má část teoretickou a část praktickou.

Do teoretické části této práce patří vytvořit a popsat množinu početních ekonomických problémů, kterou se aplikace bude zabývat. Tyto početní příklady jsou zainteresovány k předmětům Makroekonomie a Mikroekonomie. Dále obsahuje výběr vhodných nástrojů pro návrh a implementaci struktury desktopové aplikace počítající matematické operace. Tyto nástroje jsou popsány včetně jejich stručné historie. Aplikace musí být schopna počítat všechny početní příklady, nejen prvotně vytvořenou množinu. Teoretická část dále obsahuje popis programovacího jazyka Java a návod k aplikaci.

Praktická část obsahuje samotnou implementaci aplikace za použití programovacího jazyka Java. Na závěr bude kritické zhodnocení práce a nastínění možné budoucnosti aplikace.

2 Rešerše

Pro vyhledávání nástrojů, které řeší podobné příklady, jako navržená aplikace „Řešitel ekonomických úloh“ (dále jen navržená aplikace) bylo použito webového vyhledávače google¹ a vyhledávací službu EBSCO Discovery Service². Nalezené aplikace budu porovnávat a hodnotit vůči navržené aplikaci.

Nepodařilo se nalézt žádnou desktopovou aplikaci, která by byla schopná vyrovnat se všem schopnostem a vlastnostem navržené aplikace. Žádný z nástrojů se nevěnoval přímo ekonomickým příkladům. Také se nepodařilo nalézt přímo vytvořenou desktopovou aplikaci, nalezené aplikace byly online, nebo mobilní. Všechny aplikace byly spíše pro jednorázové použití, a ne pro dlouhodobé opakované používání vlastních příkladů.

Většina nástrojů byla vytvořena primárně pro matematické výpočty, což nebrání v nich provádět výpočty ekonomických problémů, to je přeci také matematika. Dva hlavní online nástroje jsou Wolfram Alpha a MAW. Oba dva jsou vytvořeny pro počítání matematických příkladů a funkcí.

Wolfram Alpha je jeden z nejlepších online nástrojů pro matematické výpočty. Celé prostředí je v angličtině, což může být pro některé studenty nevýhoda, ale určitě ne nepřekonatelná. Uživatel vepisuje výrazy, které chce vypočítat, a Wolfram už si s tím nějak poradí. Rozhodně se hodí znát klíčová slova, jako např. když chce uživatel derivovat, použije klíčové slovo „derive“. Wolfram má dobře propracovanou nápovědu s velkou škálou příkladů. Co je překvapující, Wolfram není jen nástroj na počítání příkladů, uživatel mu může klást i jiné otázky, například jaké hlavní město je jaké země.

MAW je původní český nástroj, který vznikl na Mendelu. Umí počítat mnoho různých matematických příkladů, řešit definiční obory funkcí, vykreslit grafy, derivace, průběh funkce, integrály, diferenciální rovnice a běžné rovnice a nerovnice. To vše včetně postupu výpočtu, čímž je hodně napřed před navrženou aplikací. Výsledné řešení se uživateli zobrazí ve vygenerovaném PDF souboru. Celá online aplikace je pěkně přehledná a nabízí výběr z mnoha světových jazyků (MATEMATICKÉ NÁSTROJE, 2011).

Mezi mobilní aplikace, které se alespoň vzdáleně přibližují funkcím navržené aplikace, byly zařazeny PhotoMath, Mathway a Fraction Solver. Hned podle názvu je patrné, že všechny aplikace jsou celé v angličtině. PhotoMath je revolucí, příklad stačí vyfotit, a aplikace už ho dokáže spočítat. Nebudme ale zbytečně překvapení. Text musí být strojový, ne psaný rukou. A výsledné řešení je dosti neohrabané a každý hned pozná, že ho vymyslel stroj. Mathway počítá příklady, které musíte sami zadat, tím se více podobá navržené aplikaci. Poslední aplikací je Fraction Solver, který umí řešit pouze zlomky. Je zaměřený pouze na malou část matematiky, ale řeší to velmi pěkně krok po kroku (POČÍTÁJÍ SAMY, 2014).

¹ <https://www.google.cz/>

² Unikátní vyhledávací služba nové generace. Systém, který umožňuje z jediného místa vyhledávat a získávat obsah z informačních zdrojů Akademické knihovny (EBSCO, 2015).

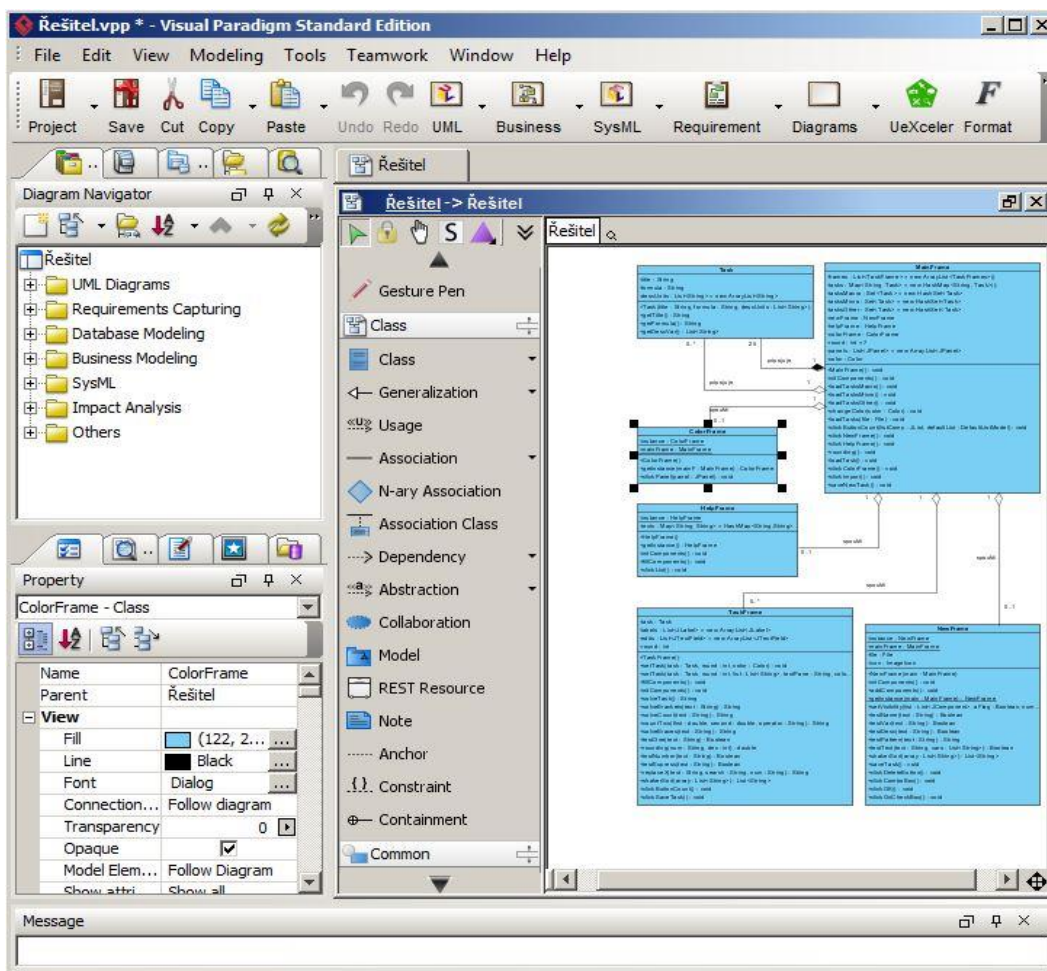
Jako poslední je uvedena grafická kalkulačka, která sice nepočítá žádné příklady z ekonomiky, ale zabývá se výpočtem a grafickým zobrazením funkcí. Podobnost s navrženou aplikací je v tom bodě, že grafická kalkulačka zpracovává vstupní text. Navržená aplikace musí všechny příklady číst ze vstupního textu, a dokázat ho vypočítat (GRAPHICAL CALCULATOR, 2011).

Do zhodnocení tedy určitě patří, že navržená aplikace obsahuje části, které jiné nástroje umí, ale také části, které jiné nezvládají. Rozhodně vyčnívá z řady svojí možností navrhnout své příklady.

3 Nástroje a metodika

3.1 Visual paradigm for UML

Visual paradigm byl stažen přímo ze serveru PEF Mendelu (https://akela.mendelu.cz/~xproch17/visual_paradigm/). Visual paradigm for UML je kompletní nástroj pro tvorbu vizuálních návrhů architektury aplikací. Nabízí propracované IDE pro vizuální návrh. Návrhy lze exportovat do mnoha formátů. Já jsem využil formáty PDF a JPG. Náhled okna Visual Paradigm (Obr. 1).



Obr. 1 Visual Paradigm

3.2 Vývojové prostředí

Aplikaci je naprogramována ve vývojovém prostředí NetBeans IDE³ 8.0. V NetBeans se dají programovat nejen aplikace v programovacím jazyce Java, ale i v jazycích jako jsou C/C++, Groovy, Java Script, PHP, Python, UML, HTML, XML a jiné.

3.2.1 Historie NetBeans

NetBeans začalo jako studentský projekt, který se jmenoval Xelfi, v roce 1996 v České republice. Hlavním cílem bylo napsat vývojové prostředí pro programovací jazyk Java, které bude podobné, jako dřívější vývojové prostředí Delphi pro programovací jazyk Pascal. Xelfi bylo první IDE pro Javu, které bylo napsané v Javě.

Tento projekt přilákal dostatečný zájem, proto se studenti hned po absolvování rozhodli tento projekt komerčně prodávat. Zdroje na webové stránky žádali po svých kamarádech a příbuzných, začali si vytvářet svoji vlastní společnost.

Brzy byli kontaktováni podnikatelem Romanem Staňkem, který již byl zapojen do podobných projektů. Roman hledal dobrý nápad, do kterého by mohl investovat své peníze. Setkal se se zakladateli Xelfi, hned si padli do oka, a skvělý obchod byl hotový. Původní obchodní plán bylo vyvinout síťové JavaBeans⁴ komponenty. S názvem NetBeans přišel jako první Jaroslav Tulach, který navrhl i základní architekturu prvního IDE.

Na jaře roku 1999 spatřil světlo světa NetBeans DeveloperX2, podporující Swing. Pro vylepšení výkonu vytvořili na podzim stejného roku JDK 1.3⁵. Ještě v létě roku 1999 tým velmi tvrdě pracoval na přepracování architektury DeveloperX2 do více modulárních NetBeans, které tvoří základ dnešního programu (NETBEANS, 2015).

3.2.2 Výhody NetBeans

NetBeans má mnoho výhod, které zasahují jak do programovacího jazyku Java, tak i do ostatních jazyků. NetBeans má velmi dobrou podporu syntaxe PHP, včetně napovídání správných typů a metod. Velmi dobré vyhledávání a nahrazování v rámci celých složek a projektů. Možnost přepnout se na deklaraci konkrétních metod, nebo tříd. Výborná podpora SVN⁶, přímo v kódu jsou vidět změny. Dobře zpracovaná lokální historie, umožňuje procházet lokální změny v souborech, vracet se k nim a také obnovovat smazané soubory. Velmi pěkně zpracované porovnávání souborů, které je možné spouštět i mezi různými soubory. Možnost nadefinování šablon kódu, je možné definovat konstrukce, které se často píšou. Podporuje

³ Integrated Development Environment

⁴ JavaBeans jsou třídy v programovacím jazyce Java, které se nějak pojmenují, a k instancím těchto tříd se přistupuje pomocí těchto referencí.

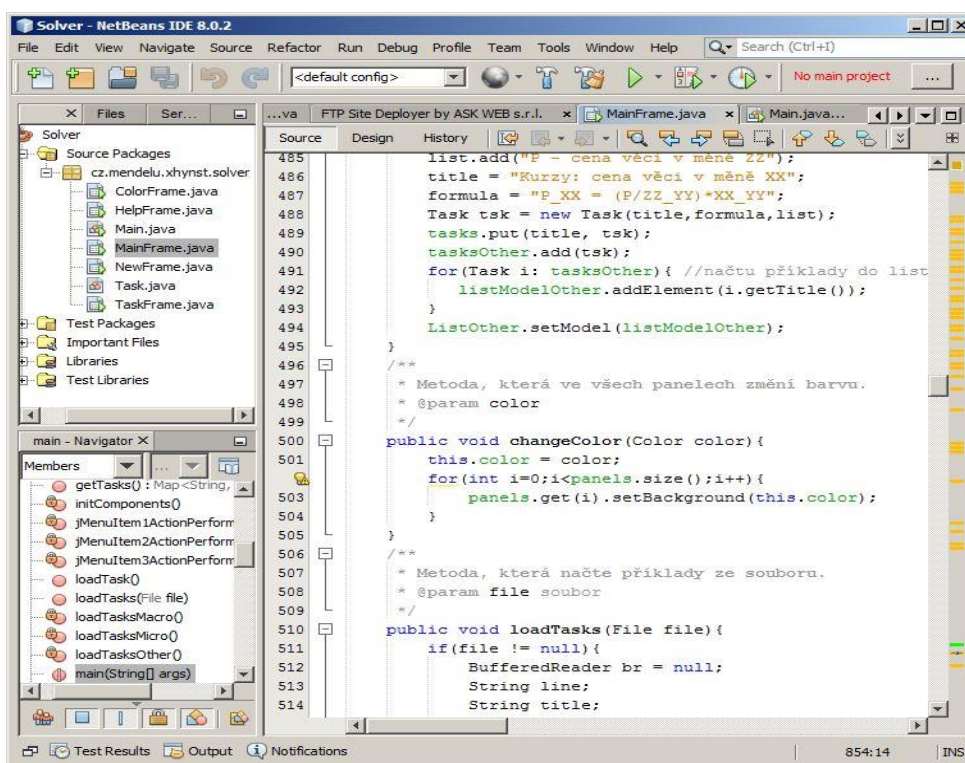
⁵ Java Development Kit

⁶ Subversion - je systém pro správu a verzování zdrojových kódů.

práci s databázemi a integračními servery. Obrovskou výhodou je jednoduchá rozšiřitelnost pomocí pluginů.

Největší výhodou, kterou ocení programátoři je automatické generování kódu, které je zpracované na jedničku, většina jiných vývojových prostředí tuto část nepodporuje. NetBeans dokáže sám vygenerovat `gettry`⁷, `settry`⁸, konstruktory podle atributů třídy, ale i překrývané a implementované metody, také testovací třídy, a dokáže i obalit kód try catch blokem⁹. Dále má podporu `phpunit`¹⁰ a vyhodnocení `code-coverage`¹¹ přímo v IDE.

Celkově NetBeans vytváří výborné pracovní podmínky, prostředí je přehledné (Obr. 4) a dobře uspořádané. V jeden okamžik může být spuštěno několik projektů a přepínání mezi nimi je jednoduché (UBUNTU, 2015).



Obr. 2 Ukázka vývojového prostředí NetBeans se zapnutou rozepsanou třídou MainFrame.

3.3 Java

Java je univerzální, moderní, objektově orientovaný programovací jazyk, vyvinutý firmou Sun Microsystems, určený zejména pro vývoj internetových aplikací. Java je

⁷ Metoda `getAtribut()`, pouze vrací hodnotu

⁸ Metoda `setAtribut(hodnota)`, pouze nastaví parametr do atributu

⁹ Slouží pro odchyťávání výjimek.

¹⁰ Testovací framework pro programátory php.

¹¹ Měření kolik řádků, nebo bloků se provedou při spuštění testů.

druhý nejpoužívanější programovací jazyk na světě, hned po jazyku C. Název jazyka je odvozen od slangového označení kávy (Obr. 5), (JAVA, 2015).



Obr. 3 Logo programovacího jazyka Java (<http://answers.ea.com/t5/image/serverpage/image-id/10151i305CAFB28ED1CE16/image-size/small?v=mpbl-1&px=-1>)

3.3.1 Historie Javy

Počátky Javy začali na začátku 90. let, kdy se začala rozšiřovat síla síťových počítačů. V roce 1991 malá skupina inženýrů společnosti Sun pod názvem „Green Team“ věřila, že další vlna v oblasti výpočetní techniky bude spojení digitálních zařízení s počítači. Tým vedl James Gosling, všichni pracovali nepřetržitě a vytvořili programovací jazyk, který byl revolucí. Java byla původně určena pro interaktivní televizi, ale to bylo příliš pokročilé na kabelovou televizi té doby.

Jazyk byl původně pojmenován Oak, to je anglicky dub. Později se začalo tomuto projektu říkat Green, to je zase anglicky zelený. Až nakonec byl projekt přejmenován, podle pojmenování kávy na, Java. Java se v té době nehodila pro interaktivní televizi, ale náramně si padla do oka s Internetem, který měl velký rozlet v té době. V roce 1995 bylo oznámeno, že Java začlenila internetový prohlížeč mezi své hotové projekty, byl to Netscape Navigator.

Dnes Java prospuje nejen do internetu, ale je to neviditelná síla mnoha aplikací a zařízení. Od mobilních telefonů ke kapesním zařízením, hrám a navigačním systémům, Java je prostě všude (JAVA HISTORY, 2015).

3.3.2 Verze Javy

Java má mnoho verzí, prvotně rozdělíme Javu na dva základní sektory, první slouží pro spouštění programů napsaných v Javě a druhý je pro vytváření takových programů.

- Java Runtime Environment (JRE) – prostředí pro běh programů. Má dvě části, Java Virtual Machine¹² a Java Core API¹³.

¹² Abstraktní počítač, virtuální stroj.

¹³ Aplikační programové rozhraní.

- Java Development Kit (JDK) – obsahuje JRE, překladač a další vývojové nástroje.

Verze Javy se číslovají, číslování verzí JRE a JDK je shodné. První verze byla JDK 1.0, další JDK 1.1, následovala JDK 1.2, označovaná jako dvojka. Až od páté verze roku 2004 se začala Java označovat jako 5.0, pětka. Aktuální verze Javy je Java SE 8 (JAVA PŘEDNÁŠKA, 2015).

V navržené aplikaci je využita starší verze JDK 7, hlavně z důvodu, že ne na všech strojích je nainstalována nová verze, a tak by mohl být problém s kompatibilitou.

3.4 Komponenty aplikace

- JFrame – nejdůležitější ze všech kontejnerů, každá aplikace musí mít jako základ svého zobrazování alespoň jeden objekt této třídy. Objekt třídy je orámován, má titulek a všechny běžné ikonky pro zavírání, zvětšení, ikonizaci atd. (HEROUT, str. 115).
- JPanel – nejjednodušší z kontejnerových tříd. Jeho jedinou funkcí je vytvořit místo pro umístění jiných komponent, nebo i dalších panelů (HEROUT, str. 125).
- JTabbedPane – komponenta, která umožňuje uživateli přepínat mezi skupinou záložek pomocí klikání na jednotlivé záložky s názvem, nebo ikonkou (JTabbedPane).
- JMenuBar – abychom mohli vytvořit menu v aplikaci, musíme jako základ použít instanci této třídy. Teprve do instance této třídy můžeme přidávat další komponenty menu. Není možné do jednoho okna vytvořit více menu, pokud se vloží více instancí této třídy, bude se brát poslední přidaný. Do menu lze vkládat položky menu, položky s přepínačem, další podmenu a separátory. Vytvořené menu lze ovládat jak klikáním myši, tak pomocí kurzorových kláves a klávesy Enter. (HEROUT, str. 99).
- JMenu – tento seznam se zobrazí po kliknutí v JMenuBar.
- JMenuItem – položka menu. Tato položka může být nevybratelná. Obsahuje text, jako nápis. (HEROUT, str. 100).
- JCheckBoxMenuItem – položka menu s přidaným zaškrtačacím oknem, které ukazuje, zda je položka vybraná. Máme dva konstruktory, jeden má implicitně vypnutou položku a ve druhém ji můžeme explicitně nastavit (HEROUT, str. 101).
- JButton – jedna z nejpoužívanějších komponent. Po kliknutí vytváří základní událost (HEROUT, str. 76).
- JLabel – tato komponenta je česky zvaná návěští, slouží k popisu, který je umístěný nezávisle na jiných komponentách. Pokud není nastavena barva po-

zadí a není zvolen žádný text, komponenta není na obrazovce vůbec vidět (HEROUT, str. 75).

- JTextField – tato komponenta slouží pro zadávání jednořádkového údaje z klávesnice. Nemá žádný popis, proto se často spojuje s komponentou JLabel (HEROUT, str. 83).
- JTextPane – do této komponenty se dá zapisovat text do více řádků. V aplikaci slouží také pro zobrazování větších bloků textu.
- JComboBox – užitečná komponenta v případě, že chceme uživateli dát na výběr z více možností, ale máme na to na ploše okna málo místa. Položky se dají odlišovat nejen jménem, ale i pořadím, tak může obsahovat více položek stejného jména (HEROUT, str. 80).
- JList – komponenta dost podobná JComboBoxu s hlavním rozdílem, že jsou vidět všechny položky seznamu, které se vlezou do velikosti komponenty. Pokud jsou položky větší než komponenta, po bocích se automaticky objeví scrollbar (HEROUT, str. 91).
- JCheckBox – zaškrtačací pole, dvoustavový přepínač (HEROUT, str. 76).
- JFileChooser – práce se soubory je častou aktivitou aplikace, a tedy volba jejich jmen, nebo výběr souborů z disku počítače. Proto se dá využít této třídy, která zobrazí dialogové okno. Máme dva režimy, můžeme spustit buď OpenFileDialog, nebo SaveDialog, podle toho zda soubor chceme otevírat, nebo ukládat.
- JOptionPane – usnadňuje vyskakování standartních dialogových oken, které vyzývají uživatele k reakci, nebo jen informují. V aplikaci jsou oba dva typy (JOptionPane).

3.5 Spouštění aplikace

Pro spouštění aplikace jsou vytvořeny dva soubory. Jeden vytvořený pomocí Javy, který má příponu jar, a druhý je konvertován z tohoto souboru a je to klasická aplikace s příponou exe. Nejjednodušším řešením by bylo spouštět aplikaci pomocí klasického spouštění přes exe soubor, ale je v tom háček. Tento konvertovaný soubor má problém s nějakými verzemi Javy, přesněji s konkrétními updaty¹⁴. Proto tady je popsáno jak se v různých operačních systémech spouští oba typy souborů, které jsou oba na přiloženém DVD, kde naleznete i instalační soubory pro JRE Javy pro všechny tři typy operačních systémů.

3.5.1 Windows

V operačním systému Windows je spuštění jednoduché, spouští se stejným způsobem oba dva typy souborů. Je k tomu potřeba pouze mít nainstalovanou verzi Javy 7 a vyšší. Stačí JRE, není potřeba vývojářská sada JDK. Instalační soubory přikládám na DVD. Když nemáte Javu nainstalovanou a pokoušíte se spustit aplikaci, vyskočí dialogové okno s textem „Java Runtime Environment not found“.

3.5.2 Linux

V linuxu je také důležité mít nainstalovanou Javu, ovšem spuštění jar souboru je trochu obtížnější. Pro snadné opětovné spouštění souboru jar si klikneme na plochu pravým tlačítkem, zvolíme „vytvořit nový“ a dále „odkaz na aplikaci“. Do názvu si napíšeme, jak chceme, aby se zástupce jmenoval, nejlépe se hodí „Řešitel ekonomických úloh“, nebo „Solver“. Přepneme na záložku „aplikace“ a do pole příkaz napíšeme příkaz „java -jar Solver.jar. Spuštění exe souboru se provádí klasickým otevíráním aplikací.

3.5.3 MacOS

V tomto operačním systému od firmy Apple lze také spouštět java aplikace. Zase prvním krokem je mít nainstalovanou správnou verzi Javy. Spouštění jar aplikace je dost složité a tak se mu vyhneme. Nainstalujeme Javu a aplikaci budeme spouštět pomocí exe souboru.

¹⁴ Update – aktualizace Javy, opravující chyby a nedostatky.

4 Vlastní práce

4.1 Množina ekonomických příkladů

Po prostudování ekonomických učebnic od Václava Jurečky (JUREČKA, 2013a, 2013b), které napsal se svým kolektivem, se začala rýsovat množina výpočtu ekonomického oboru. Pro dokončení bylo použito znalostí z dostudovaných předmětů na bakalářském studiu a další online zdroje, které jsou všechny uvedené u jednotlivých příkladů. K makroekonomii a mikroekonomii je přidáno ještě několik příkladů z dalších ekonomických oborů.

4.1.1 Příklady makroekonomie

Pro vytvoření množiny makroekonomických problémů bylo použito převážně materiálů do cvičení předmětu makroekonomie I, který se učí na PEF Mendelu.

Některé příklady jsou triviální, ale hodí se jako mezivýpočty pro jiné příklady. Ostatní příklady jsou základními příklady, které musí znát každý student, který chce absolvovat tento předmět. Celkem je v aplikaci načteno 11 makroekonomických příkladů.

- Hrubý domácí produkt, výdajová metoda – vyčísluje výdaje jednotlivých sektorů na nákup finálních výrobků a služeb. Výdaje domácností na spotřebu (C), soukromé hrubé domácí investice (I), výdaje státu na nákup výrobků a služeb (G), čistý export (CVIČEBNICE MAKROEKONOMIE 2, 2015).

$$HDP = C + I + G + X \quad (1)$$

- Čistý export – slouží k mezivýpočtu, od celkového exportu (E) se odečte celkový import (I), (CVIČEBNICE MAKROEKONOMIE 2, 2015).

$$X = E - I \quad (2)$$

- Hrubý domácí produkt, důchodová metoda – počítá se jako součet národního důchodu, který představuje hrubé mzdy (w), renty (r), hrubé zisky korporací (z), čisté úroky (i), důchody se samozaměstnání (y), amortizace (a), nepřímé daně (n). Tento příklad je převzat z internetového zdroje finance.cz (HDP, 2015).

$$HDP = w + r + z + i + y + a + n \quad (3)$$

- Velikost spotřeby v ekonomice – potřebujeme znát mezní sklon ke spotřebě (mpc), velikost autonomní spotřeby (Ca), výši důchodu v ekonomice (Y). Tento příklad se řeší na prvním cvičení makroekonomie (CVIČEBNICE MAKROEKONOMIE 1, 2015).

$$C = Ca + mpc \cdot Y \quad (4)$$

- Osobní důchod – od čistého národního důchodu (c_{ND}) se postupně odečítají zisky firem (z_f), nepřímé daně (n_d), důchody domácností (d_d), úroky firem (u_f) a nakonec se přičítají transferové platby domácnostem (t_p), (OSOBNÍ DŮCHOD, 2015).

$$PI = c_{ND} - z_f - n_d - d_d - u_f + t_p \quad (5)$$

- Rychlost obratu peněžní jednotky, cenová hladina, reálný produkt, množství peněz v ekonomice – tyto čtyři makroekonomické jednotky jsou obsaženy v tzv. Fisherově transakční rovnici, nebo také známé jako rovnice směny, či kvantitativní rovnice peněz. V aplikaci to musí být uloženo jako 4 různé příklady, pro výpočet jednotlivých proměnných (CVIČEBNICE MAKROEKONOMIE 3, 2015).

$$P \cdot Q = M \cdot V \quad (6)$$

- Celkový přírůstek důchodu/celkový pokles důchodu – tato dvojitá úloha je založena na situaci, kdy v ekonomice došlo ke zvýšení transferových plateb domácnostem, nebo ke zvýšení daňového zatížení domácností (z), při druhé variantě bude číslo záporné. Dále je potřeba znát mezní sklon ke spotřebě (mpc). Výsledný přírůstek/pokles se vypočítá postupným dosazováním mezi výpočtů do dalších rovnic. V aplikaci na to slouží jeden vzorec (CVIČEBNICE MAKROEKONOMIE 4, 2015):

$$prirustek_Y = mpc \cdot z \cdot \left(\frac{1}{1 - mpc} \right) \quad (7)$$

- Únik z výdajového řetězce/pokles úspor – opět se jedná o dvojitou úlohu a je založena na stejných situacích jako předchozí příklad. Zvýšení transferových plateb, nebo zvýšení daňového zatížení (z), a mezní sklon ke spotřebě (mpc). V aplikaci se výsledek vypočítá opět jedním vzorcem (CVIČEBNICE MAKROEKONOMIE 4, 2015):

$$unik = z \cdot \left(\frac{1}{1 - mpc} \right) - (mpc \cdot z) \cdot \left(\frac{1}{1 - mpc} \right) \quad (8)$$

4.1.2 Příklady mikroekonomie

Pro vytvoření množiny mikroekonomických problémů bylo použito učebních materiálů do cvičení předmětu mikroekonomie I, který se učí na PEF Mendelu a webová stránka ekospace.cz. Celkem je do aplikace načteno 9 mikroekonomických problémů.

- Důchodová elasticita poptávky – vyjadřuje citlivost reakce spotřebitele na změnu důchodu. Koeficient udává, o kolik procent se změní poptávané množství, když se důchod změní o procento (CVIČEBNICE MIKROEKONOMIE 1, 2015).

$$E_{ID} = \frac{\frac{Q_2 - Q_1}{(Q_2 + Q_1)/2}}{\frac{I_2 - I_1}{(I_2 + I_1)/2}} \quad (9)$$

- Cenová elasticita poptávky v procentech – podělení procentuálních rozdílů množství (Q) a ceny (P), (CVIČEBNICE MIKROEKONOMIE 1, 2015).

$$E_{PD} = \frac{Q}{P} \quad (10)$$

- Cenová elasticita poptávky – cenová elasticita počítá s výchozí (Q₁) a druhou (Q₂) hladinou množství statku. A dále s výchozí (P₁) a druhou (P₂) hladinou ceny statku (EKOSPACE, 2015).

$$E_{PD} = \frac{Q_2 - Q_1}{(P_2 - P_1) \cdot \left(\frac{P_1}{Q_1}\right)} \quad (11)$$

- Intervalová elasticita poptávky – intervalová elasticita poptávky počítá cenovou elasticitu pouze v určitém intervalu zadaném parametry. Parametry jsou množství jedna (Q₁), množství dva (Q₂), cena jedna (P₁), cena dva (P₂), vlastní kapitál (VK), celkový úročený kapitál (K), (CVIČEBNICE MIKROEKONOMIE 1, 2015).

$$E = \frac{\frac{Q_2 - Q_1}{(Q_2 + Q_1)/2}}{\frac{P_2 - P_1}{(P_2 + P_1)/2}} \quad (12)$$

- Křížová elasticita poptávky – křížová elasticita poptávky vypadá podobně, jako ostatní elasticity, ale rozdíl je v tom, že změna množství (Q_1 , Q_2) jednoho statku se odráží ve změně ceny (P_1 , P_2) statku úplně jiného (EKOSPACE, 2015).

$$E_{DC} = \frac{\left(\frac{Q_2}{Q_1} \cdot 100\right) - 100}{\left(\frac{P_2}{P_1} \cdot 100\right) - 100} \quad (13)$$

- Rovnovážná cena – u výpočtu rovnovážné ceny se můžeme setkat s více typy zadání, proto v aplikaci budou dva příklady na rovnovážnou cenu (EKOSPACE, 2015):

- Dvě rovnice ve tvaru $Q = a + b \cdot P$

$$P = \frac{a_1 - a_2}{b_2 - b_1} \quad (14)$$

- Jedna rovnice ve tvaru $Q = a + b \cdot P$ a druhá ve tvaru $P = a + b \cdot Q$

$$P = \frac{a_2 + b_2 \cdot a_1}{1 - b_2 \cdot b_1} \quad (15)$$

- Rovnovážné množství – u výpočtu rovnovážného množství je stejná situace jako u rovnovážné ceny (EKOSPACE, 2015):

- Dvě rovnice ve tvaru $Q = a + b \cdot P$

$$Q = \frac{b_2 \cdot a_1 - b_1 \cdot a_2}{b_2 - b_1} \quad (16)$$

- Jedna rovnice ve tvaru $Q = a + b \cdot P$ a druhá ve tvaru $P = a + b \cdot Q$

$$Q = \frac{a_1 + b_1 \cdot a_2}{1 - b_1 \cdot b_2} \quad (17)$$

4.1.3 Nezařazené příklady

Nezařazené příklady byli vytvořené ze cvičení předmětů učených na PEF Mendelu. Většina jich je tvořena příklady z Podnikové ekonomiky. Některé jsou složitější a slouží pro uživatele aplikace, aby bylo vidět, jak se zadávají složitější vzorce,

a jiné jsou méně složité, které pouze doplňují ekonomické příklady. Celkem je do aplikace načteno 5 problémů z oblasti ekonomiky podniku a podnikání.

- Nákladová funkce – výpočet pro celkové náklady. Fixní náklady (F), průměrné variabilní náklady na jeden kus výroby (V) a objem výroby (Q). Tento příklad mám převzat z přednášky (PŘEDNÁŠKA PODNIKOVÁ EKONOMIKA 2, 2015).

$$N = F + V \cdot Q \quad (18)$$

- Optimální kapitálová výroba – považujeme takový dlouhodobý kapitál, při kterém dosahujeme minimálních průměrných nákladů kapitálu. Potřebujeme znát průměrnou úrokovou míru z cizího úročeného kapitálu (i_{ck}), sazbu daně z příjmů (t), cizí úročený kapitál (CK), požadovaná výkonost vlastního kapitálu (i_{vk}), vlastní kapitál (VK), celkový úročený kapitál (K), (PŘEDNÁŠKA PODNIKOVÁ EKONOMIKA 2, 2015).

$$WACC = i_{ck} \cdot (1-t) \cdot \frac{CK}{K} + i_{vk} \cdot \frac{VK}{K} \quad (19)$$

- Harrisův-Wilsonův vzorec – počítá optimální velikost jedné dodávky. Potřebujeme znát potřebu dodávek (D), náklady na jednu dodávku (Nd), náklady na jeden den a jednu zásobu (Ns), délku období (T).

$$D_0 = \sqrt{\frac{2 \cdot D \cdot Nd}{Ns \cdot T}} \quad (20)$$

- Potřebný počet dělníků – potřebujeme znát počet strojů (Z), koeficient směnnosti (s), formu individuální obsluhy (o), (Přednáška Podniková ekonomika 3, 2015), (POČET DĚLNÍKŮ, 2015).

$$D_0 = \frac{Z \cdot s}{o} \quad (21)$$

- Kurzy, cena věci v měně XX – potřebujeme znát kurz první měny ku druhé (XX_YY), kurz třetí měny ku druhé (ZZ_YY) a cenu věci v třetí měně (P).

$$P_{XX} = \frac{P}{ZZ_{YY}} \cdot XX_{YY} \quad (22)$$

4.2 Návrh aplikace

4.2.1 Výběr vhodného nástroje

Při návrhu struktury aplikace byl prvotně vytvořen UML¹⁵ diagram tříd. Pro vytvoření tohoto diagramu musel být vybrán vhodný nástroj. Muselo být zahrnuto mnoho kritérií. Hlavním byla cena, užití tohoto nástroje nebylo naplánováno ke komerčním účelům, tak většina nástrojů obsahuje trochu ořezanou verzi zdarma ke stažení, nebo jsou levnější verze, které mají většinu nástrojů, které byly potřeba pro diagram tříd. Dalším a hlavním kritériem bylo, jaká bude práce s kterým nástrojem, a jak ho již znám z minulosti. Zajistil jsem si 4 nástroje pro návrh UML diagramů a porovnával je. Porovnával jsem tyto: Office Visio, Dia, AgroUML a Visual Paradigm for UML.

Nástroj Office Visio od firmy Microsoft jsem již jednou používal, když nám byl doporučen jako jeden z mnoha při výuce v předmětu Databázové systémy na PEF Mendelu. Celkově se mi nezdál úplně vhodný, práce v něm mi nevyhovovala.

Dalším nástrojem byl Dia, open source¹⁶ software, který jsme zase používali v předmětu Počítačové sítě na PEF Mendelu. Sice pěkný přehledný nástroj s mnoha možnostmi, ale nehodil se pro diagram tříd aplikace v Javě.

Dalším z řady open source softwarů je AgroUML, tento nástroj byl vynikající, ale když jsem ho zkoušel poprvé, docela jsem nevěděl na co kde kliknout. Jazyková podpora v angličtině mi nestačila, na start bych nejspíše potřeboval český překlad.

Jako nejlepší mi vyšel Visual Paradigm. Tento nástroj se dal spustit v češtině, takže jsem všemu plně rozuměl, také jsem již s tímto nástrojem pracoval při studiu předmětů Základy objektově orientovaného návrhu a C++ na PEF Mendelu. Tento nástroj má dohodu s naší univerzitou, a tak jsem mohl využít akademickou licenci. Tato licence se kontroluje při každém spuštění programu, takže vždy je nutné připojení k internetu. To mi při práci nevadilo. Zvolil jsem tento nástroj a navrhl jsem v něm diagram tříd.

4.2.2 Popis diagramu tříd

V této podkapitole podrobně popíšu navržený diagram tříd (Obr. 2). Na obrázku UML diagramu je všech 6 tříd, které jsou navrženy v aplikaci. Třídy, které mají v názvu slovo „Frame“, jsou třídy, které se zobrazují jako okna v aplikaci. Jediná nezobrazovaná třída, je třída Task. Třída Task slouží pro ukládání příkladů do aplikace.

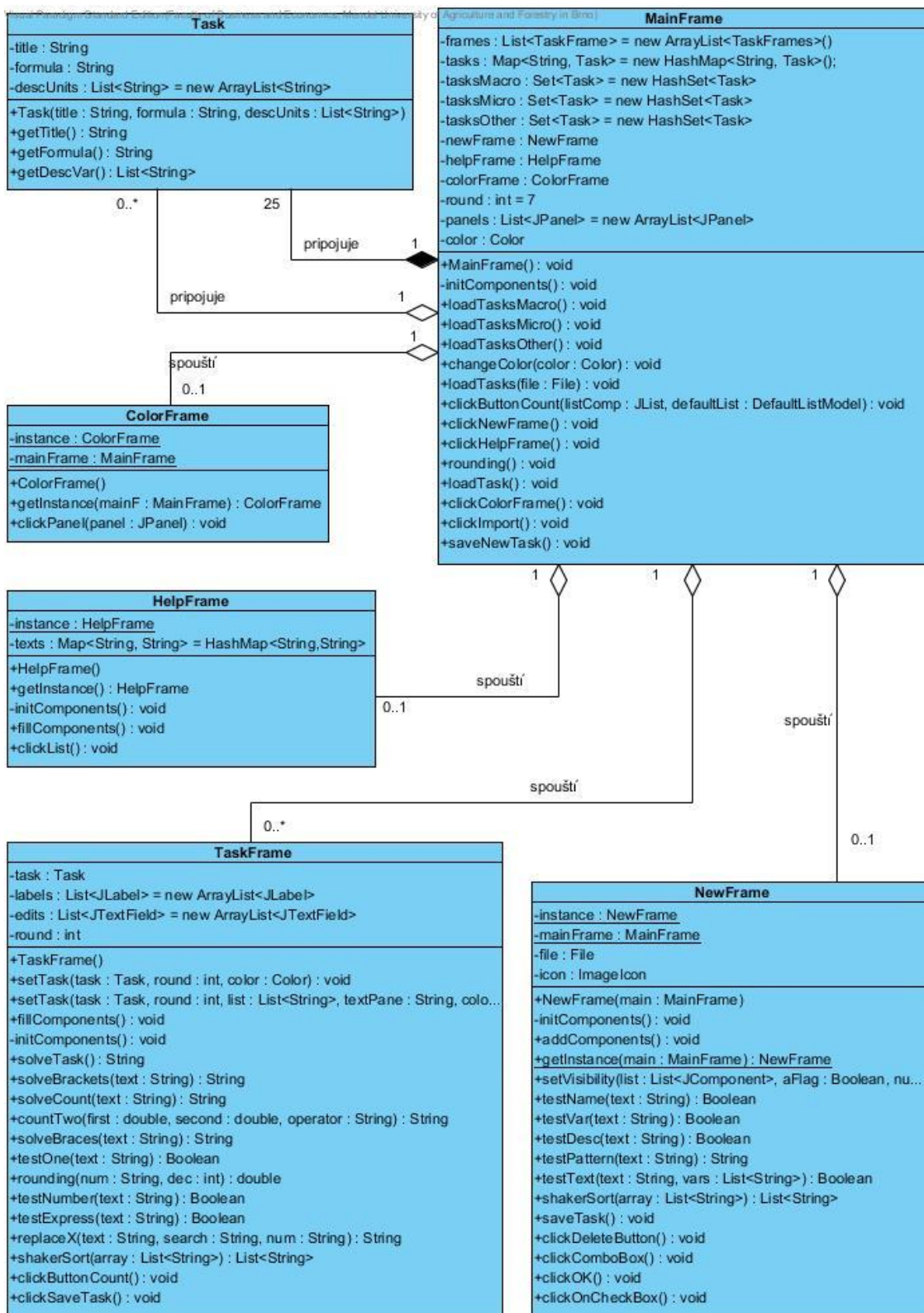
V pravém horním rohu obrázku je MainFrame, hlavní okno aplikace, všechny ostatní třídy mají vazby s touto třídou. Třída Task má vazby dvě, kompozicí je při spuštění vytvořeno 25 příkladů, a agregací se mohou připojovat a vytvářet další příklady. S třídou NewFrame, která slouží pro vytváření nových příkladů je hlavní okno spojeno agregací, třída je vytvořena jako jedináček. Nelze vytvořit více než

¹⁵ Unified Modeling Language

¹⁶ Software s otevřeným zdrojovým kódem

jednu instanci. Při prvním pokusu o spuštění se vytvoří okno, při dalších spuštěních se už jen vrací první otevřené okno, které bylo skryto. Stejně fungují i okna HelpFrame pro zobrazení nápověd a ColorFrame pro výběr barvy pozadí.

Poslední třída je TaskFrame, která je s hlavní třídou spojená agregací, těchto oken může být otevřeno neomezeně. Každý příklad se dá počítat víckrát, i více stejných příkladů najedou.



Obr. 4 UML diagram tříd

4.2.3 Obecná funkcionalita aplikace

V následujících odstavcích je popsána navržená obecná funkcionalita aplikace. Řešitel ekonomických úloh je GUI¹⁷ aplikace vytvořená v programovacím jazyce Java. Aplikace se ovládá pomocí myši, číselné hodnoty se zadávají z klávesnice, nebo se dají kopírovat do textových polí ze schránky.

Po spuštění se zapne pouze hlavní okno. Z hlavního okna lze spouštět všechny ostatní okna. Zvláště je dobré si hned po spuštění zapnout okno nápovědy, kde se uživatel dozví jak správně s programem pracovat. Pro třídy nápovědy, nového příkladu a výběru barvy pozadí lze otevřít jen jedno okno. Pro třídu počítání příkladu, lze otevírat oken neomezeně a počítat více příkladů zároveň. Již otevřená okna s příklady nemění barvu pozadí s ostatními okny. Toho se dá využívat, nastavíme barvu pozadí na modrou, začneme počítat první příklad, pozadí tohoto okna bude modré. Změníme barvu pozadí na červenou a spustíme nové okno s počítáním příkladu. Tím docílíme toho, že budeme počítat dva příklady, a každý bude mít jinou barvu pozadí, i když v okně počítání příkladu nelze měnit barvu pozadí přímo. Okno nápovědy nemění barvu pozadí nikdy, je stále v defaultní šedé barvě.

Aplikace dovoluje vytvářet nové příklady, to je jedna z hlavních výhod. V prvotním návrhu bylo obsaženo, že nové příklady se budou ukládat do přednastaveného souboru, a automaticky bude tento soubor načítán při opětovném spuštění aplikace, stejně tak při ukládání se provede pokus o načtení celého souboru. Tento návrh nebyl zcela ideální, a proto z něj bylo upuštěno a bylo zavedeno ukládání do souboru, který si uživatel sám zvolí. Po restartu aplikace si musí uživatel manuálně soubor nainportovat. Příklady se ukládají do souboru za sebe na jednotlivé řádky. Tak i v případě když bude vytvořeno velké množství nových příkladů, dají se nainportovat do aplikace pomocí tří kliknutí a výběru jednoho souboru. Při ukládání nového příkladu je možnost příklad neukládat do souboru, v takovémto případě se nově vytvořený příklad automaticky po dokončení nahraje do aplikace, ale na žádné jiné místo se neuloží. Jako s každým příkladem v aplikaci je možné s ním provádět všechny operace, jako ho spouštět, ukládat a nahrávat. Bohužel, bez uložení se příklad po restartu aplikace nadobro ztratí, a nebude možné ani nahrávat již uložené rozpočítané instance tohoto příkladu. Možnost neukládat příklad je v návrhu pouze pro zkušební příklady, nebo pro dočasné mezivýpočty. Je doporučeno všechny příklady ukládat.

Okamžitě od prvních návrhů bylo rozhodnuto, že aplikace nebude uživateli vysvětlovat, co znamená výsledná hodnota, ani nebude oznamovat, jaké výsledné číslo by se dalo v jakých typech příkladů očekávat. Aplikace je řešitel, zadané hodnoty dokáže podle vzorce matematicky vyřešit.

¹⁷ Graphical User Interface

4.2.4 Design a návrh oken aplikace

Všechna okna aplikace jsou navržena v podobném duchu. Všechna mají neměnnou pevně danou velikost. Neměnná velikost oken je zvolena, protože všechny okna aplikace nejsou příliš velká, a tak není důležité si je zmenšovat, nebo si je roztahovat ne celou obrazovku. Zároveň také není důležité vidět mnoho oken aplikace zároveň, bohatě si uživatel vystačí se třemi okny. Pro konkrétní rozmístění komponent v okně by se musel využít jeden z množiny Layout Managerů¹⁸ a složitě ho aplikovat na různá oddělení okna, protože nelze využít jednoduchých Managerů, které komponenty řadí pouze pod sebe, nebo vedle sebe. Komponenty musí být seřazeny do sloupců, jinde zase za sebou, v nějaké části okna je pouze jedna komponenta. Rozhodl jsem se nevyužít z žádných základních Managerů, jako jsou FlowLayout, GridLayout, BorderLayout, CardLayout, nebo DridBagLayout. Dříve v Javě pro variantu absolutně zadávaných pozic nebyl žádný Manager, tato varianta je doporučovaná až v krajní nouzi. Nyní již je vytvořen, a je využit v aplikaci, jmenuje se Absolute Layout Manager, který umísťuje komponenty přímo na určené pozice. Při tomto rozmísťování se komponenty neroztahují při změně velikosti celého okna, další důvod navíc nechat okna v neměnné velikosti.

Panel menu je pouze v hlavním okně, ostatní okna nevyžadují tolik akcí, aby se nedali vyřešit pomocí tlačítek. Je to pro uživatele rychlejší volba, než stále otevírat menu. Všechna okna jsou jako JFrame¹⁹, takže mají svoje záhlaví, ve kterém mají tlačítka pro minimalizování okna, maximalizování okna a zavření okna. Tlačítko pro maximalizování je zašedlé a z důvodu neměnnosti velikosti okna nelze použít. V některých oknech se objevují dva otazníky zapsané za sebe. Tato část slouží jako nápověda, po ponechání kurzoru na těchto otaznících se zobrazí nápověda, co se jakou částí okna dá dělat.

Pro styl zobrazení má Java více různých stylů. Je zde možnost i nenastavovat vůbec žádný styl. Při takovém nastavení zůstanou komponenty zobrazovány jako komponenty Windows. V aplikaci je zvolen styl Nimbus, který vývojové prostředí nastavuje automaticky po vytvoření projektu. Bylo zvoleno, aby aplikace byla odlišena od ostatních starých aplikací. Tento styl má hladší hrany a rohy komponent. Celkově je zobrazení příjemnější. Barvy nejsou tak tmavě šedivé, ale přechází do modravé. Velký rozdíl je hlavně vidět u tlačítek, která jsou krásně zakulacena. Porovnání dvou stylů (Obr. 3).

V hlavním okně při horním okraji je panel menu a v něm položky panelu menu. Panel menu má celkem 3 kategorie, hlavní nabídka, nastavení a nápovědu. V hlavní nabídce se nabízejí možnosti přidání nového příkladu, načtení příkladu z textového souboru, import nových příkladů z textového souboru a ukončení aplikace. V kategorii nastavení se nastavuje barva pozadí a zaokrouhlování výsledků. Poslední kategorie, nemá žádné podpoložky, pouze zapíná okno s nápovědou. Ve spodní části hlavního okna je dlouhé tlačítko, kterým se zapíná počítání příkladu.

¹⁸ Objekt třídy `java.awt.LayoutManager`, který je zodpovědný za rozmísťování a zobrazování komponent ukládaných do objektů typu kontejner. (HEROUT, 129 str.)

¹⁹ Základní okno, instance třídy `Javax.swing.JFrame`

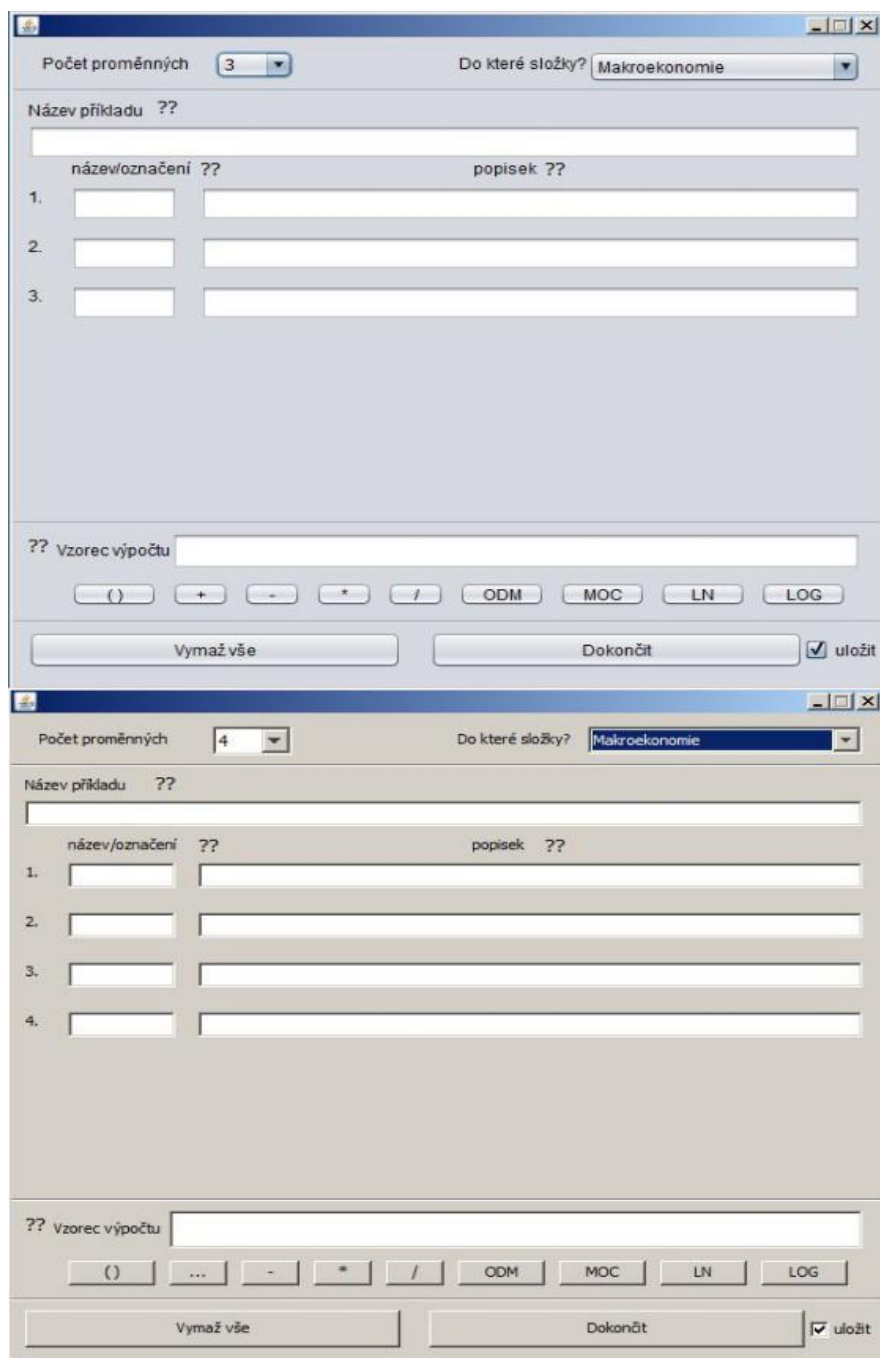
Takřka přes celý zbytek okna je záložkový panel. V něm jsou 3 záložky pro každou z kategorií příkladů. Po kliknutí na příklad se příklad vybere, spustit se musí potvrzením na tlačítko ve spodní části okna.

Okno nápovědy je nejjednodušší ze všech oken, na levé straně je seznam témat, a na pravé straně velké textové pole. Po kliknutí na položku v seznamu se zobrazí podrobnější text nápovědy do textového pole.

Naopak nejsložitějším je bezesporu okno vytvoření nového příkladu. Okno je vodorovně rozděleno do čtyř částí, oddělených separátory²⁰. Při prvním spuštění se zobrazí pouze první část, ve které musí uživatel zvolit, kolik proměnných bude v novém příkladu obsaženo. V této části se také vybírá do které ze tří složek se má příklad uložit, složky zastupují záložky v hlavním okně. Po provedení změny v části počtu proměnných se odkryjí ostatní části okna. V pořadí druhá část od horního okraje okna slouží pro vyplnění názvu příkladu, názvu proměnných a popisu proměnných. Tady je první pole roztaženo přes celou šíři okna, ale další už jsou ve dvou sloupcích. Třetí částí je část nejdůležitější, vyplnění vzorce. V této části je textové okno pro zapsání vzorce a 9 tlačítek, které vkládají operátory do vzorce. Poslední část obsahuje pouze zaškrtačací pole, které určuje zda se příklad bude ukládat do souboru, a tlačítka pro vymazání obsahu okna a dokončení příkladu.

Okno počítání příkladu se nastavuje podle počítaného příkladu. Vždy se zobrazí název příkladu. Níže se zobrazí určený počet proměnných, které se musí zadat, aby se mohl provést výpočet. Pod zadáním hodnot se nachází velké textové pole, ve kterém je zapsán vzorec výpočtu, a po vypočítání příkladu se tam zapíše zadané hodnoty a výsledek. Příklad můžeme počítat několikrát po sobě i s jinými hodnotami v jednom okně, a výsledky se budou zapisovat pod sebe. Ve spodní části jsou ještě dvě tlačítka, první pro uložení rozpočítaného příkladu do souboru a druhé slouží pro samotné počítání.

²⁰ Tenká čára rozdělující části oken.



Obr. 5 Porovnání oken aplikace, Nahoře Nimbus styl, dole Windows styl.

4.3 Struktura aplikace

4.3.1 Formát ukládaných dat

V aplikaci se ukládají informace spojené s příklady, dále údaje o zaokrouhlování, barvě pozadí oken a informace ohledně nápovědy.

Pro ukládání příkladů je vytvořena samostatná třída. Tato třída má jako atributy název příkladu, který je ukládán jako datový typ `String`²¹, stejným typem je uložen i vzorec příkladu. Označení proměnných je uloženo v seznamu tvořeném opět z datových typů `String`.

Údaj o zaokrouhlování je uložen v hlavní třídě, která tvoří hlavní okno aplikace. Je uložen jako celočíselný 32 bitový datový typ `Integer`²². Barva pozadí oken je uložena také v hlavní třídě aplikace, jako instance třídy `Color`²³.

Informace v nápovědě jsou uloženy jako instance třídy `String` v rozhraní `Map`²⁴. Kde klíčem je název napovídáného textu a hodnota je samotný text nápovědy. Toto rozhraní je zvoleno tak, protože nápověda je zobrazována pomocí témat, a až po kliknutí se zobrazí celý text.

4.3.2 Formát vstupních a výstupních dat

Všechna data ukládaná do textových souborů z aplikace mohou být dříve, či později z aplikace čtena. Proto formát musí být jednotný. Jak rozpočítané příklady, tak příklady pro import mohou být v textových souborech vytvořena manuálně, bez použití aplikace. Tento postup osobně rozhodně nedoporučuji. Snadno se dá udělat chyba, po které aplikace nedovolí soubor načíst celý, někdy ho nemusí načíst vůbec. Při importu příkladů ze souboru, kdy je několik příkladů napsaných pod sebou v textovém souboru, dokáže aplikace pomocí metod načíst pouze tolik řádků, kolik se jich dokáže načíst bez chyby, řádek s chybou a všechny řádky po něm se vůbec nenačtou. Pokud do souboru ukládáte příklady pouze z aplikace, chyba by se neměla vyskytnout. Pokud uživatel sám bude manipulovat se souborem, je dobré manuálně psané příklady dávat až na konec souboru, tímto se dá vyhnout případnému nenačtení většího počtu příkladů.

Při uložení nového příkladu se ukládají tyto údaje: název příkladu, počet proměnných, složka, označení a názvy proměnných, a na závěr matematický vzorec. Všechny tyto informace jsou zapsány na jeden řádek. Jako oddělovače slouží svislí-

²¹ Třída `String` představuje řetězce znaků. Všechny řetězcové literály v Java programech, jsou implementovány jako instance této třídy (`STRING`).

²² Třída `Integer` zabalí hodnotu primitivního typu `int` jako objekt. Objekt typu `Integer` obsahuje jediné jednoduché pole, jehož typ je `int`. Navíc, tato třída poskytuje několik metod pro konverzi `int` na řetězec a řetězec na `int`, stejně jako jiné konstanty a metody užitečné při jednání s `int` (`INTEGER`).

²³ Třída `Color` se používá k zapouzdření barvy z výchozího barevného RGB prostoru (`COLOR`).

²⁴ Objekt, který mapuje klíče k hodnotám. Mapa nemůže obsahovat duplicitní klíče, každý klíč může mapovat na maximálně jednu hodnotu (`MAP`).

ce známá jako nebo „|“ a znak mínus „-“. Celý řádek je potom v tomto obecném formátu:

```
Název příkladu|počet proměnných-složka|označení první pro-
měnné-popis první proměnné|...|označení poslední proměnné-
popis poslední proměnné|označení výsledku=vzorec
```

Toto je formát řádku konkrétního vzorového příkladu:

```
Vzorový příklad|2-Makroekonomie|A-toto je áčko|B-toto je
béčko|V=A+B
```

Při ukládání rozpočítaného příkladu se ukládají tyto údaje: název příkladu, hodnoty proměnných a text, který byl již zapsán do textového pole komponenty *JTextPane* v okně výpočtu příkladu. Tentokrát je příklad zapisován na více řádků. Každá položka je zapsaná na vlastní řádek. Na prvním řádku je zapsán informativní text, že tento soubor obsahuje uložený příklad aplikace Řešitel ekonomických úloh, tento řádek není nutné opisovat, ale je důležité, aby na prvním řádku byl nějaký text. Tato skutečnost je zde, protože na prvním řádku se po úpravě souborů mohou objevovat netisknutelné znaky, které ovšem aplikace načítá. Kdyby v takovém případě na prvním řádku byl název příkladu, nemusel by odpovídat již známému příkladu a celý příklad by nemohl být načten, jen z důvodu, že před načtený název by se připojil netisknutelný znak začátku souboru. Název příkladu proto musí být až na druhém řádku. Pro rozhodnutí, zda bude zbytek souboru načten, nebo ne rozhoduje hlavně druhý řádek s názvem. Pokud název příkladu nebude napsán přesně ve formátu, v jakém je uložen v aplikaci, nebude vůbec načítán zbytek souboru. Pokud bude název souboru napsán správně, ale bude chybět nějaký další řádek s hodnotami, aplikace nebude upozorňovat na chybu, ale načte textové řádky do špatných polí. Taková chyba je vidět na první pohled, ale rozhodně se neřadí do chyb závažných pro chod aplikace. Aplikace neošetřuje toto špatné čtení, protože když se do textových polí pro hodnoty proměnných dostane nesprávný text, nebude možné příklad vypočítat a nehrozí pád aplikace. I pokud další řádky budou úplně chybět, aplikace je bude číst jako prázdné a do okna pro výpočet příkladu vyplní prázdná pole. Celý soubor je v tomto obecném formátu:

```
Uložený příklad aplikace Řešitel ekonomických úloh.
Název příkladu
Hodnota první proměnné
Hodnota x proměnné
Hodnota poslední proměnné
Text v komponentě JTextPane
```

Toto je formát souboru pro uložení rozpočítaného vzorového příkladu:

```
Vzorový příklad
20
50
Vzorec: V=A+B
```

4.4 Chod aplikace

Při chodu aplikace se může stát, že něco nefunguje tak jak má. Většinou je to způsobeno chybnou akcí ze strany uživatele. Pro zamezení vstupu chybných dat do aplikace jsou vytvořeny chybové hlášky, které upozorní uživatele na chybu a zamezí mu v nesprávném postupu. Všechny chybové hlášky se zobrazují na vyskakovacích dialogových oknech. Vysvětlení co znamenají texty chybových hlášek je zapsáno v nápovědě aplikace.

4.4.1 Chybové hlášky při operacích se soubory

- Nepodařilo se načíst všechny příklady – nějaké příklady mají shodné jména s již načtenými příklady v aplikaci.
- Chyba při čtení příkladu na x. řádku souboru xxx – příklad je zadán ve špatném formátu, dialogové okno informuje i o čísle řádku na jakém se vyskytla chyba.
- Soubor se nepodařilo načíst – poškozený soubor, soubor nelze nalézt.
- Aplikace nezná příklad – při čtení rozpočítaného příkladu se pokoušíte načíst příklad, který aplikace nezná. Buď jej musíte naimportovat, nebo opravit název příkladu.
- Soubor není kompatibilní ke čtení – chyba v zápisu, soubor poškozen, nebo nelze nalézt.
- Nepředvídatelná chyba při ukládání – soubor, který vybíráte je poškozen, nebo do něj nelze zapisovat, nebo nelze nalézt.

4.4.2 Chybové hlášky při počítání příkladů

- Závažná chyba ve výpočtu, možné dělení nulou, nebo jiná závažná chyba – tato chyba se vyvolá, když ve zlomku se pokoušíte dělit nulou, nebo se do výpočtu dostala nečitelná hodnota. Při nečitelné hodnotě se musela poškodit aplikace a je vhodné ji opětovně instalovat²⁵.
- Tento příklad teď není možné vypočítat – opět se může jednat o dělení nulou. Tato chyba se vyskytuje převážně u nově vytvořených příkladů, kde se špatně načte vzorec příkladu. Musíte opravit příklad.

²⁵ Instalací se zde rozumí znova ji nakopírovat do počítače.

5 Diskuze

5.1 Ekonomické zhodnocení

Při návrhu a vývoji každé aplikace jsou na začátku důležité finance. Já jsem se na toto dívat nemusel, protože jsem práci vypracovával sám. Aplikace je tedy úplně zadarmo, nejsou žádné náklady. Všechny nástroje, které byly využity pro návrh a implementaci aplikace, jsou zdarma. Kdyby takový podobný projekt byl objednan na zakázku, práce, kterou jsem nad návrhem a implementací strávil, by stála desetitisíce korun českých. Při větším rozsahu projektu, se cena projektů obvykle pohybuje i ve statisících.

5.2 Technické zhodnocení

Aplikace je dle mého na dobré technické úrovni. Je zajištěn bezproblémový chod, chybové hlášky, i nápověda. Aplikace zvládá také složitější matematické operace. Rozhodně není dokonalá, má své výhody i nevýhody. Jako hlavní výhodou je možnost přidání vlastních příkladů, které je opravdu výbornou pomůckou. Nevýhody jsou zadávání vzorců jako jednořádkový text, kde zadání zlomků je náročnější, a je důležitá přesnost. Zadávání zlomků pod sebe odděleně zlomkovou čarou by bylo velice složité a časově náročné na implementaci, na tomto by se mohlo zapracovat do budoucna. Aplikace není zatím dostupná na webu, toto by se mohlo změnit, aplikace by se mohla dát spouštět přímo z prohlížeče. Aplikace neumí derivovat, ani integrovat, tyto funkce jsou občas v ekonomickém oboru potřebné, toto je další podnět pro zlepšení. Aplikace má velké výhledy do budoucna, dá se pokračovat v rozvíjení a vylepšování.

6 Závěr

Hned při výběru tohoto tématu jsem si udělal představu, jak bude výsledná aplikace vypadat. Tato představa se hodně změnila při prvních pokusech. Prvotní návrh vypadal slibně, ale začátky implementace byli složité. Od začátku bylo jasné, že nejsem schopen vytvořit návrh, který bych nemusel už ani jednou měnit při jeho implementaci. Toto bylo způsobeno mou neznalostí všech nástrojů jazyka Java. Nějaký čas jsem strávil nad studiem knih o navrhování aplikací v jazyce Java, nad samotným programováním jsem ale strávil času daleko více. Stále se objevovali nějaké chyby, které jsem musel odstraňovat. Někdy bylo zapotřebí měnit částečně strukturu celé aplikace, jindy zase stačila malinká úprava. Abych tyto chyby mohl objevit, musel jsem dalších hodně hodin strávit nad testováním aplikace. Nešlo jen o testování funkčnosti aplikace samotné, ale zda půjde spustit na všech desktopových počítačích.

Po dokončení funkční aplikace, která by byla schopná počítat matematicky správně příklady ze zadaného textu, jsem musel vytvořit základní množinu ekonomických příkladů, které budou součástí aplikace. Při zvolení tohoto tématu jako bakalářské práce jsem nečekal, že to bude problém. Většina příkladů v ekonomickém oboru ke svému řešení potřebuje graf, a to navržená aplikace nedokázala podporovat. Nakonec po prostudování mnoha zdrojů jsem do aplikace nahrál 25 příkladů.

Nakonec se mi podařilo vytvořit aplikaci, která dokáže vypočítat všechny matematické příklady, které obsahují základní matematické operátory, logaritmy, mocniny a odmocniny.

7 Literatura

- COLOR. *Class Color* [online]. [Cit 11. 5. 2015]. Dostupný na WWW:
<https://docs.oracle.com/javase/7/docs/api/java/awt/Color.html>
- CVIČEBNICE MAKROEKONOMIE 1. *Dokument ve složce MA1 Makroekonomie I* [online]. [Cit 13. 5. 2015]. Dostupný na WWW:
https://is.mendelu.cz/auth/dok_server/dokumenty_cteni.pl?id=63000;on=0;dok=99629
- CVIČEBNICE MAKROEKONOMIE 2. *Dokument ve složce MA1 Makroekonomie I* [online]. [Cit 13. 5. 2015]. Dostupný na WWW:
https://is.mendelu.cz/auth/dok_server/dokumenty_cteni.pl?id=63000;on=0;dok=102503
- CVIČEBNICE MAKROEKONOMIE 3. *Dokument ve složce MA1 Makroekonomie I* [online]. [Cit 13. 5. 2015]. Dostupný na WWW:
https://is.mendelu.cz/auth/dok_server/dokumenty_cteni.pl?id=63000;on=0;dok=105077
- CVIČEBNICE MAKROEKONOMIE 4. *Dokument ve složce MA1 Makroekonomie I* [online]. [Cit 13. 5. 2015]. Dostupný na WWW:
https://is.mendelu.cz/auth/dok_server/dokumenty_cteni.pl?id=63000;on=0;dok=106195
- CVIČEBNICE MIKROEKONOMIE 1. *Dokument ve složce Cvičení* [online]. [Cit 13. 5. 2015]. Dostupný na WWW:
https://is.mendelu.cz/auth/dok_server/dokumenty_cteni.pl?id=60428;on=0;dok=92227
- EBSCO. *Akademická knihovna spustila EBSCO Discovery Service* [online]. [Cit 19. 4. 2015]. Dostupný na WWW: <https://www.jcu.cz/o-univerzite/aktuality/akademicka-knihovna-spustila-ebSCO-discovery-service>
- EKOSPACE. *Mikroekonomie 1* [online]. [Cit. 9. 5. 2015]. Dostupný na WWW: <http://www.ekospace.cz/1-mikroekonomie-1>
- GRAPHICAL CALCULATOR. *Lukáš Rusek 2011* [online]. [Cit. 19. 4. 2015]. Dostupný na WWW: <http://dspace.vsb.cz/handle/10084/87476>
- HDP. *Jak se počítá HDP* [online]. [Cit 13. 5. 2015]. Dostupný na WWW:
<http://www.finance.cz/makrodata-eu/hdp/vypocet/>
- HEROUT, P. *Java: grafické uživatelské prostředí a čeština*. 2. vyd. České Budějovice: Kopp, 2009. 316 s. ISBN 978-80-7232-328-9.
- INTEGER. *Class Integer* [online]. [Cit 11. 5. 2015]. Dostupný na WWW:
<https://docs.oracle.com/javase/7/docs/api/java/lang/Integer.html>
- JAVA. *Dokument ve složce PJJ Programovací jazyk Java* [online]. [Cit 9. 5. 2015]. Dostupný na WWW: https://is.mendelu.cz/auth/dok_server/dokumenty_cteni.pl?id=80371;on=1;dok=143036

- JAVA HISTORY. *The History of Java Technology* [online]. [Cit. 9. 5. 2015]. Dostupný na WWW:
<http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html>
- JAVA PŘEDNÁŠKA. *Přednášková prezentace č. 1A* [online]. [Cit. 12. 5. 2015]. Dostupný na WWW:
https://is.mendelu.cz/auth/dok_server/slozka.pl?id=80371;zobraz=0;typ=mod;on=1
- JOPTIONPANE. *Class JOptionPane* [online]. [Cit 11. 5. 2015]. Dostupný na WWW:
<http://docs.oracle.com/javase/7/docs/api/javax/swing/JOptionPane.html>
- JTABBEDPANE. *Class JTabbedPane* [online]. [Cit 11. 5. 2015]. Dostupný na WWW:
<http://docs.oracle.com/javase/7/docs/api/javax/swing/JTabbedPane.html>
- JUREČKA, V. a kol. *Makroekonomie*. 2. vyd. Praha: Grada, 2013a. 342 s. ISBN 978-80-247-4386-8.
- JUREČKA, V. a kol. *Mikroekonomie*. 2. vyd. Praha: Grada, 2013b. 366 s. ISBN 978-80-247-4385-1.
- MAP. *Class Map* [online]. [Cit 11. 5. 2015]. Dostupný na WWW:
<https://docs.oracle.com/javase/7/docs/api/java/util/Map.html>
- MATEMATICKÉ NÁSTROJE. *Matematické nástroje 2011* [online]. [Cit 19. 4. 2015]. Dostupný na WWW: <http://www.matematika.cz/nastroje>
- NETBEANS. *A Brief History of NetBeans* [online]. [Cit. 7. 5. 2015]. Dostupný na WWW: <https://netbeans.org/about/history.html>
- OSOBNÍ DŮCHOD. *Osobní důchod, Ekonomie otázky* [online]. [Cit 13. 5. 2015]. Dostupný na WWW: <http://ekonomie-otazky.studentske.cz/2008/07/osobndchod.html>
- POČET DĚLNÍKŮ. *Proces řízení zásob ve firmách* [online]. [Cit 13. 5. 2015]. Dostupný na WWW: <http://portal.pohoda.cz/pro-podnikatele/uz-podnikam/proces-rizeni-zasob-ve-firmach/>
- POČÍTAJÍ SAMY. *Jitka Rambousková 2014* [online]. [Cit 19. 4. 2015]. Dostupný na WWW:
[http://www.cojsemvyzkousela.cz/?p=1033#14320287063924&{"type":"auth","argument":"","result":null}](http://www.cojsemvyzkousela.cz/?p=1033#14320287063924&{)
- PŘEDNÁŠKA PODNIKOVÁ EKONOMIKA 1. *Složka ZPE Podniková ekonomika I* [online]. [Cit 13. 5. 2015]. Dostupný na WWW:
https://is.mendelu.cz/auth/dok_server/slozka.pl?download=83744;id=55113;edit_priloha=1;dokument=83744;hlavni=1;on=3
- PŘEDNÁŠKA PODNIKOVÁ EKONOMIKA 2. *Složka ZPE Podniková ekonomika I* [online]. [Cit 13. 5. 2015]. Dostupný na WWW:
https://is.mendelu.cz/auth/dok_server/slozka.pl?download=84937;id=55113;edit_priloha=1;dokument=84937;hlavni=1;on=3

- PŘEDNÁŠKA PODNIKOVÁ EKONOMIKA 3. *Složka ZPE Podniková ekonomika I* [online]. [Cit 13. 5. 2015]. Dostupný na WWW:
https://is.mendelu.cz/auth/dok_server/slozka.pl?download=87534;id=55113;edit_priloha=1;dokument=87534;hlavni=1;on=1
- STRING. *Class String* [online]. [Cit 11. 5. 2015]. Dostupný na WWW:
<http://docs.oracle.com/javase/7/docs/api/java/lang/String.html>
- UBUNTU. *NetBeans* [online]. [Cit 7. 5. 2015] Dostupný na WWW: <http://wiki.ubuntu.cz/programování/netbeans>

Přílohy

A Příložené DVD

Příložené DVD obsahuje:

- Soubory na spuštění aplikace
- Zdrojové kódy aplikace
- Uživatelskou příručku
- Soubory pro instalaci Javy