

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Výukový kurz goniometrických funkcí v Javě
Jan Svoboda

Bakalářská práce
2013

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2012/2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan Svoboda**
Osobní číslo: **I10214**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Výukový kurz goniometrických funkcí v Javě**
Zadávající katedra: **Katedra informačních technologií**

Z á s a d y p r o v y p r a c o v á n í :

Anotace:

Cílem práce je vytvoření výukového kurzu goniometrických funkcí pro střední školu v jazyce Java.

Teoretická část:

V teoretické části bakalářské práce bude provedeno zhodnocení technologií využívaných ke zkvalitnění výuky na školách, zejména e-learningu. Bude proveden úvod do problematiky tvorby výukových kurzů a bude provedena analýza řešeného problému a navrženo vhodné řešení. V úvodní části budou popsány Java technologie vhodné pro tvorbu kurzu.

Implementační část:

GUI aplikace bude naprogramována pomocí jazyka Java a bude využívat práci se soubory. Celý kurz se bude skládat z jednotlivých kapitol a závěrečného testu z celého kurzu. Aplikace je určena pro výuku studentů středních škol.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

HEROUT, Pavel. Učebnice jazyka Java. 1. vyd. České Budějovice: Kopp, 2001, 349 s. ISBN 80-723-2115-3.

REKTORYS, Karel. Přehled užité matematiky. 7. vyd. Praha: Prometheus, 2000, xxxii, 874 s. ISBN 80-719-6179-5.

ZOUNEK, Jiří. E-learning - jedna z podob učení v moderní společnosti. Vyd. 1. Brno: Masarykova univerzita, 2009, 161 s. ISBN 978-80-210-5123-2.

Vedoucí bakalářské práce:

Ing. Zdeněk Šilar

Katedra informačních technologií

Datum zadání bakalářské práce:

21. prosince 2012

Termín odevzdání bakalářské práce:

10. května 2013



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 29. března 2013

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 6. 5. 2013

Jan Svoboda

Poděkování

Chtěl bych poděkovat vedoucímu mé bakalářské práce Ing. Zdeňku Šilarovi za veškerou odbornou pomoc a cenné rady při zpracování mé bakalářské práce. Dále bych chtěl poděkovat svým rodičům za psychickou podporu a vytvoření podmínek pro mé studium. Nakonec bych chtěl poděkovat mé přítelkyni za její pomoc při výběru barev aplikace.

Anotace

Cílem práce je vytvoření výukového kurzu goniometrických funkcí pro střední školy v jazyce Java. V první části je provedeno zhodnocení technologií využívaných ke zkvalitnění výuky na školách, zejména e-learningu. Dále je zde popsána základní problematika tvorby výukových kurzů a popis Java technologií vhodných k tvorbě výukových kurzů. Druhá část práce je věnována samotné implementaci výukového kurzu goniometrických funkcí pro střední školy.

Klíčová slova

Java, goniometrické funkce, výukový kurz, GUI

Title

Tutorial of trigonometric functions in Java

Annotation

The goal of this work is to create a tutorial of trigonometric functions intended for high school created by using programming language Java. In the first part is done the evaluation of technologies exploited to improvement education at high school, especially e-learning. There is also written basic dilemma of creating tutorials and description of Java technologies opportune to create tutorials. Second part is dedicated to implementation of tutorial of trigonometric functions intended for high school.

Keywords

Java, trigonometric functions, tutorial, GUI

Obsah

Seznam zkratk	9
Seznam obrázků	10
Úvod	11
1 Technologie využívané ke zkvalitnění výuky	12
1.1 Nástroje podporující spolupráci a sdílení znalostí.....	12
1.1.1 Wiki	12
1.1.2 Weblog (Blog)	12
1.2 Technologie a přístupy umožňující distribuci výukového obsahu	12
1.2.1 Podcast.....	13
1.2.2 Elektronické knihy (učebnice).....	13
1.3 Nástroje zprostředkující komunikaci.....	14
1.3.1 Diskuzní fóra na internetu	14
1.3.2 Nástroje synchronní komunikace	14
1.4 Komplexní nástroje umožňující vyučování a učení	15
1.4.1 Virtuální výuková prostředí.....	15
1.4.2 Google jako nástroj e-learningu	15
1.4.3 Mobilní technologie a učení	16
1.4.4 Sociální sítě na internetu.....	16
2 Tvorba výukových kurzů	17
2.1 Předpoklady k tvorbě výukových kurzů.....	17
2.2 Výběr vhodného software.....	17
2.3 Funkčnost výukového kurzu.....	17
2.4 Obsah výukového kurzu	18
3 Java technologie vhodné k tvorbě výukových kurzů	19
3.1 GUI.....	19
3.1.1 Knihovny AWT a Swing	19
3.1.2 Grafické komponenty	19
3.1.3 Kontejnery – správce rozložení	22
3.2 Kolekce.....	23
3.2.1 Seznamy	24
3.2.2 Mapy.....	24

3.2.3	Množiny.....	24
3.3	Datové streamy.....	24
3.3.1	Práce se soubory.....	24
3.3.2	Serializace a externalizace dat.....	25
4	Analýza a návrh implementace	26
4.1	Rozdělení a vyhodnocení kurzu.....	26
4.2	Správa uživatelů a uchování vyhodnocení.....	27
4.3	Obsah výukového kurzu.....	28
4.4	Komunikace se soubory.....	28
5	Výukový kurz GonioFce	29
5.1	Úvod.....	29
5.1.1	Registrace.....	30
5.1.2	Přihlášení.....	30
5.2	Kurz.....	30
5.2.1	Obslužné menu.....	31
5.2.2	Obsah kurzu.....	31
6	Správa výukového kurzu GonioFce.....	33
6.1	Správa uživatelů.....	33
6.2	Zobrazení testů.....	34
	Závěr.....	35
	Literatura.....	36
	Příloha A – Zjednodušený UML diagram.....	37
	Příloha B – Zdrojový kód naplnění šablony testu.....	38
	Příloha C – Kompletní adresářová struktura.....	39

Seznam zkratek

AWT	Abstract Window Toolkit
DOC	Document
GFC	Goniometrické Funkce Člověk
GFS	Goniometrické Funkce Seznam
GUI	Graphical User Interface
HTML	HyperText Markup Language
ICQ	I Seek You
IP	Internet Protocol
JAR	Java Archive
JPEG	Joint Photographic Experts Group
LMS	Learning Management System
PDB	Portable Digital Book
PDF	Portable Document Format
MD5	Message-Digest Algorithm 5
UML	Unified Modeling Language

Seznam obrázků

Obrázek 1 - GUI Komponenty	21
Obrázek 2 - Border Layout.....	23
Obrázek 3 - Grid Layout.....	23
Obrázek 4 - Dynamický obsah kurzu	27
Obrázek 5 - GonioFce screen přihlášení.....	29
Obrázek 6 - GonioFce screen kurz	30
Obrázek 7 - GonioFce screen testové otázky	32
Obrázek 8 - GonioFce screen vyhodnocení testové otázky.....	32
Obrázek 9 - GonioFce screen správa kurzu.....	33
Obrázek 10 - GonioFce screen vyhodnocení testové otázky 2.....	34

Úvod

Cílem této práce je vytvoření výukového kurzu goniometrických funkcí pro střední školy. Práce se v první kapitole zabývá moderními technologiemi využívanými ve výuce. V této kapitole jsou rozděleny technologie využívané ke zkvalitnění výuky do čtyř hlavních skupin podle zaměření a míry přínosnosti k výuce. Jsou zde popsány nástroje podporující spolupráci a sdílení znalostí, technologie a přístupy umožňující distribuci výukového obsahu, nástroje zprostředkující komunikaci a komplexní nástroje umožňující vyučování a učení.

Druhá kapitola uvádí čtenáře do základů tvorby výukových kurzů. Kapitola je rozdělena na čtyři části a každá část se věnuje jednomu ze základních kroků postupu tvorby výukových kurzů. Postupně je zde popsáno, co je potřeba k tvorbě výukových kurzů, zda-li tvořit celý kurz nebo využít některý z dostupných pomocných nástrojů, jak by měl celý kurz fungovat a co obsahovat, a jak vytvářet studijní materiály.

Třetí kapitola je zaměřena na popis technologií programovacího jazyka Java, které jsou vhodné k tvorbě výukových kurzů. V první části je čtenář seznámen se základy GUI. Dozví se zde, jaké knihovny jsou vhodné pro tvorbu GUI, jaké obsahují tyto knihovny grafické komponenty a co jsou to správci rozložení a jejich základní popis. Další část vysvětluje, co jsou to kolekce, jejich základní rozdělení, použití a vhodný výběr. V poslední části je popsána práce se soubory, obzvláště komunikace mezi programem a soubory.

Ve čtvrté kapitole je provedena analýza řešeného problému a navrženo vhodné řešení. Je zde popsán obsah samotného kurzu, jeho funkčnost, jak pracuje s uživateli, jak je celý kurz vyhodnocen a další otázky vyplývající z druhé kapitoly.

Pátá a šestá kapitola popisují implementace programů GonioFce. Pátá kapitola se věnuje popisu programu Výukový kurz a šestá kapitola popisuje program Správa výukového kurzu, který je určen lektorům výukového kurzu.

1 Technologie využívané ke zkvalitnění výuky

V této kapitole jsou popsány technologie využívané ke zkvalitnění výuky rozdělené do čtyř hlavních skupin podle zaměření a míry přínosnosti k výuce.

1.1 Nástroje podporující spolupráci a sdílení znalostí

Tato část popisuje nástroje, které podporují spolupráci a sdílení znalostí za pomoci volně dostupných internetových aplikací.

1.1.1 Wiki

Jedná se o webové stránky, které umožňují uživatelům přidávat nebo měnit jejich obsah. Uživatelé tak mohou mezi sebou sdílet různé informace (texty, fotky, ...). Většina webů bývá veřejně přístupná bez nutnosti registrace, a tak do nich může přispívat své články téměř kdokoli. Správci těchto stránek však mají možnost blokovat IP adresy uživatelům, kteří nemají zájem o zlepšování informací a sebezdokonalování a místo toho tyto weby pustoší. Na stránkách probíhají pravidelné zálohy, není tedy problém kdykoliv jejich obsah obnovit. Nejznámějším příkladem je celosvětový projekt Wikipedia, což je webová encyklopedie otevřená uživatelům celého světa v několika jazykových verzích. Wiki ale nemusí být jen světu otevřené webové stránky. Mohou to být například aplikace v rámci školní či firemní sítě, kde studenti nebo zaměstnanci sdílí mezi sebou potřebné informace.

Velkou nevýhodou wiki webů je jejich základní vlastnost a to otevřené editování obsahu, což souvisí i s další velkou nevýhodou a tou je nutný dohled nad přidávanými informacemi, který, pokud není dostatečný, může zapříčinit nízkou úroveň zveřejňovaných informací. Ani jedna nevýhoda však není nevýhodou přímo této technologie ale lidí, kteří s ní pracují.

1.1.2 Weblog (Blog)

Weblog je webová stránka, umožňující jeho autorovi zveřejňovat články ve stylu, jako by psal svůj deník. Největší výhodou weblogů je možnost komentářů u jednotlivých článků. Mohou tak zde vznikat velmi zajímavé diskuze o daném tématu. Autorem weblogu může být jeden či více autorů, kteří prezentují své články. Články však na rozdíl od wiki webů může upravovat pouze jejich autor. Založení weblogu a publikování článků jeho prostřednictvím je velmi jednoduché, proto se staly weblogy velmi oblíbeným prostředkem pro sdílení informací. V dnešní době sociálních sítí je však tento způsob sdílení znalostí v úpadku.

1.2 Technologie a přístupy umožňující distribuci výukového obsahu

Následuje popis technologií umožňujících distribuci výukového obsahu pomocí volně dostupných nástrojů.

1.2.1 Podcast

Termín podcasting vznikl spojením dvou slov iPod a broadcasting (vysílání). Podcasting je způsob distribuce zvuku a videa ať už na internetu či v rozhlasu, televizi nebo například z vlastních přehrávačů. Jedná se tak o velkou výhodu moderních zařízení, kdy se studenti mohou vzdělávat kdykoliv a kdekoliv. Nejsou nuceni navštěvovat knihovny nebo neustále sedět u počítače, ale mohou například poslouchat zvukové záznamy nebo sledovat videa z přenosných přehrávačů. Důležitou vlastností podcastingu je, že programy umožňující nahrávání a správu zvuku i videí jsou volně dostupné. Mnoho rozhlasových stanic dává volně ke stažení své pořady formou podcastingu. Této formy využívají i internetové televize, které na internetu zveřejňují své pořady ke zhlédnutí online. Dalšími zástupci video podcastingu jsou streamovací kanály a specializované servery jako např. YouTube.

Velkou výhodou podcastingu je možnost opakovaného spuštění souboru. Student si tak může opakovat potřebné informace do té doby, než si bude jist, že danou problematiku úplně zvládá. Naopak velkou nevýhodou podcastingu je, že je použitelný ke studiu pouze v obecné šíři. Je-li potřeba něco více vysvětlit do detailů, podcasting už přestává být vhodným prostředkem k výuce. Není možnost otevřené diskuze nad aktuálním tématem.

Speciální variantou podcastingu je screencasting. Screencasting využívá snímání obrazovky počítače, ke kterému je poté dopsáno, co se kde na obrazovce děje. Tento způsob podcastingu je zejména vhodný pro tvorbu návodů a postupů. Nejlepší výslednou variantou screencastingu je videonávod.

1.2.2 Elektronické knihy (učebnice)

Nejnámější podobou elektronické učebnice je elektronická verze tištěné knihy, kterou je možné číst na přenosných zařízeních nebo počítačích. Elektronické knihy existují v mnoha formátech a mají různé funkce. Nejčastější formáty elektronických knih jsou DOC, PDF, JPG a PDB. Některé knihy mohou obsahovat zvukové záznamy.

V dnešní době jsou velmi rozšířené čtečky elektronických knih, které se snaží co nejlépe přiblížit zobrazení obsahu knih papírové formě za použití speciálního displeje. Většina čteček podporuje připojení k internetu, odkud je možné si stáhnout knihy, časopisy nebo noviny.

Dalším, zatím ne tolik využívaným typem elektronických učebnic jsou interaktivní učebnice, které jsou určeny pro výuku s využitím interaktivní tabule. Interaktivní učebnice umožňují použití obrázků, zvukových i video nahrávek přímo ve výuce.

Mezi velké výhody elektronických knih patří možnost vyhledávání v textu (na rozdíl od manuálního vyhledávání v tištěných knihách) nebo lepší dostupnost, jedná-li se například o nějaká skriptá (počet „výtisků“ není nijak omezen). Nevýhodou elektronických knih je problematika autorských práv a velmi snadné plagiátorství.

1.3 Nástroje zprostředkující komunikaci

Následující část se věnuje popisu nástrojů zprostředkujících komunikaci mezi lektory a studenty.

1.3.1 Diskuzní fóra na internetu

Jednou z nejvyžívanějších forem komunikace po internetu je email. V emailu je možné poslat formátovaný text s obrázky a do přílohy uložit libovolné soubory (obrázky, zvuky, videa, ...). Lze také vytvořit otevřenou diskuzi prostřednictvím emailu odesláním jednoho emailu více lidem.

Mezi další velmi používané formy komunikace patří online fóra a internetové diskuze. Jsou to v podstatě webové stránky, kam uživatelé přispívají své názory nebo reagují na názory ostatních uživatelů. Fóra mají většinou nějaké zaměření, popřípadě jsou dále separována na podkategorie. Všechny diskuze se odehrávají asynchronně, což znamená, že uživatel nemusí být neustále online, aby si přečetl všechny příspěvky. Každé diskuzní fórum má administrátora, který ho spravuje. Administrátor může mazat nevhodné příspěvky a diskuze, blokovat přístup uživatelům, kteří se nechovají vhodně nebo například nastavovat parametry zveřejňovaných příspěvků.

Diskuzní fóra jsou vhodná jako doplněk k výuce. Studenti zde mohou diskutovat mezi sebou nebo mohou takto komunikovat i s učitelem, který je v roli moderátora fóra. Nevýhody diskuzních fór jsou jejich těžké začlenění do výuky nebo delší doba komunikace.

1.3.2 Nástroje synchronní komunikace

Hlavním znakem všech nástrojů je synchronní komunikace na rozdíl od kapitoly 1.3.1, kde byly popisovány nástroje asynchronní komunikace. V podstatě to znamená, že všichni účastníci spolu komunikují v reálném čase prostřednictvím nějakého komunikačního programu.

První formou synchronní komunikace je tzv. instant messaging, což je internetová služba, která umožňuje mít svůj vlastní seznam přátel, se kterými je možné komunikovat. V seznamu přátel se zobrazují i uživatelé, kteří nejsou zrovna online, a je možné jim nechávat vzkazy. Komunikace probíhá výhradně v textové formě pomocí zpráv doplněných emotikony. Uživatelé mají možnost si posílat soubory různých typů prostřednictvím těchto klientů. Mezi nejznámější zástupce patří ICQ, Windows Live Messenger, Facebook Chat, ...

Další formou synchronní komunikace je internetová telefonie. Nejznámějším představitelem této formy synchronní komunikace je Skype. Tento program umožňuje telefonovat s jinou osobou v reálném čase na libovolnou vzdálenost. Hovor může být pouze zvukový, pokud ale má alespoň jeden účastník hovoru webkameru, může povolit druhému účastníkovi, aby ho v reálném čase i viděl. Skype také, stejně jako programy

instant messagingu, umí posílat textové zprávy s emotikony a soubory. Dalo by se tedy říci, že je jakousi nadstavbou instant messagingu.

Výhodou nástrojů synchronní komunikace může být komunikace se zahraničními studenty nebo virtuální konzultace s vyučujícím. Nevýhodou je nutnost být online a pokud je uživatel na místě se slabým připojením, neprobíhá komunikace v odpovídající kvalitě.

1.4 Komplexní nástroje umožňující vyučování a učení

Poslední část popisuje samotné nástroje s obsáhlými vlastnostmi a funkcemi umožňující vyučování a učení.

1.4.1 Virtuální výuková prostředí

Virtuální výuková prostředí jsou technologie používané k online učení mimo školní budovu. Tato prostředí obsahují nástroje nezbytné pro realizaci výuky včetně její administrace.

Příkladem virtuálních výukových prostředí je LMS neboli systémy určené k řízení studia. Jedná se o propracovaná virtuální výuková prostředí, která umožňují tvorbu a správu jednotlivých kurzů. Jednoduchá tvorba kurzů je umožněna díky integrovaným nástrojům pro tvorbu kurzů. Do kurzů je možné vkládat textové, zvukové i video soubory. Součástí kurzů bývají i komunikační nástroje jako diskuzní fóra a zasílání zpráv jednotlivým účastníkům. K tvorbě kurzů nemusí být autor žádný webový specialista, tyto kurzy lze vytvářet i bez znalosti HTML jazyka. Jednotlivé kurzy mohou být zaheslovány a tak být využívány pouze určitou skupinou studentů, vše záleží pouze na autorovi. Známým příkladem LMS je Moodle. Moodle je rozsáhlý nástroj pro tvorbu virtuálních výukových prostředí umožňující správu činností studentů.

Nevýhodou LMS je, že při velkém počtu kurzů mohou být velké nároky na server, kde jsou tyto kurzy provozovány. LMS tak vyžaduje technické zázemí na odpovídající úrovni.

1.4.2 Google jako nástroj e-learningu

Firma Google nabízí několik nástrojů, které mohou být velmi nápomocné při výuce. Mezi ty nejdůležitější patří:

- Google – vyhledávač umožňující vyhledávat mezi webovými stránkami, obrázky a videi,
- Gmail – emailový klient určený ke komunikaci,
- Google talk – hlasový a textový komunikátor v reálném čase,
- Google dokumenty – nástroj určený ke zpracování textových, tabulkových a prezentačních dokumentů, na kterých se může podílet více uživatelů,
- Google kalendář – pomocník pro organizaci času,

- Google scholar – vyhledávání informací v odborných zdrojích, ...

Díky tomuto systému propojených nástrojů je Google velmi flexibilním komplexním systémem vhodným k výuce. Nevýhodou Google je sdílení svých kolikrát tajných a soukromých dat na serverech Google.

1.4.3 Mobilní technologie a učení

Většina mobilních zařízení se v dnešní době dokáže připojit k internetu nebo se vzájemně propojit s ostatními zařízeními. Vzniká tak samostatný systém komunikace a sdělování prostředků. Na většině mobilních přístrojů, obzvláště na těch, které mají vlastní operační systém, lze instalovat stejné nebo podobné programy jako na stolních počítačích a noteboocích. Je tak možné mít na mobilním zařízení komunikátory instant messagingu, atd. Navíc se zde nachází fotoaparát nebo videokamera. Vyfocené fotky či nahraná videa je možné ihned odeslat komukoliv, kdo vlastní mobilní zařízení. Vzhledem k tomu, že skoro každý student vlastní nějaké mobilní zařízení, je mobilní komunikace vhodný prostředek k výuce.

Velkou výhodou této technologie je, že se student s její pomocí může vzdělávat v podstatě odkudkoliv, kde je signál. Mezi nevýhody těchto zařízení patří jejich malé rozměry a nekvalitní příslušenství. Například nelze přečíst dlouhé texty nebo nahrávat kvalitní videa. Dalším omezením je nutnost častého nabíjení baterie.

1.4.4 Sociální sítě na internetu

Sociální sítě jsou služby dostupné na internetu. Tyto služby spojují skupiny lidí a jednotlivců do jednoho celku. Každý uživatel nebo skupina zde má vytvořený účet a jeho prostřednictvím komunikuje s okolím. Každá sociální síť podporuje jiné způsoby sdílení zveřejňovaných dat, jejich základem však zůstává komunikace pomocí soukromých zpráv. Nejčastější vlastnosti jednotlivých sociálních sítí jsou sdílení textových zpráv, fotografií a videí. Některé sociální sítě umožňují nastavení soukromí. Je tak možné sdílet obsah jen s určitou skupinou lidí. Nejznámější sociální sítě jsou Facebook, MySpace, LinkedIn, YouTube, ...

Velkou a hlavní výhodou sociálních sítí je možnost vytvářet separované komunity s nějakým zájmem. K dalším výhodám patří sdílení materiálů jen pro určitý okruh lidí. Nevýhodou může být například nevyžádaná reklama [3].

2 Tvorba výukových kurzů

Tato kapitola uvádí čtenáře do základů tvorby výukových kurzů. Kapitola je rozdělena na jednotlivé části a každá část se věnuje jednomu ze čtyř základních kroků postupu tvorby výukových kurzů.

2.1 Předpoklady k tvorbě výukových kurzů

Nejnutnější předpoklady k tvorbě výukových kurzů jsou znalosti dané problematiky, nutné technické vybavení a znalost obsluhy tohoto vybavení. Základním technickým vybavením je osobní počítač, na kterém je možné vytvářet textové dokumenty, zpracovávat fotografie, vytvářet vlastní doprovodné obrázky, nahrávat a upravovat zvuk a zpracovávat nahraná videa.

2.2 Výběr vhodného software

Otázka, kterou musí autor výukového kurzu řešit, je, použije-li ke tvorbě kurzu některý z dostupných specializovaných software zaměřený na tvorbu kurzů (např. LMS), nebo je-li na tak vyspělé úrovni obsluhy počítače, že zvládne takový kurz vytvořit celý sám. Půjde-li tou snazší cestou a zvolí specializovaný software, ušetří si spoustu práce a dále již řeší jen nastavení funkčnosti a obsahovou stránku kurzu. Naopak, zvolí-li tu těžší cestu a chce si vytvořit svůj vlastní kurz, ať už mu nevyhovuje žádný specializovaný software svou funkčností nebo vzhledem, má autor možnost ovlivnit celý průběh kurzu a není omezen ničím, kromě možností programovacího jazyka.

2.3 Funkčnost výukového kurzu

Dalším krokem je promyšlení funkčnosti celého kurzu. Autor si musí promyslet, co všechno má kurz umět. Základní otázkou každého kurzu je, bude-li kurz spravovat uživatele a pamatovat si jejich studijní výsledky, nebo jestli bude vytvářet pouze kurz na procvičení bez jakékoliv správy. Co se týká specializovaných software na tvorbu výukových kurzů, tak záleží na zvolené aplikaci. Většinou jsou zde však uchovávány údaje o uživateli. Pokud autor tvoří celý kurz sám, je toto rozhodnutí zcela na něm. Může zvolit lehčí cestu a neukládat žádné údaje o uživateli. Výsledná aplikace pak bude mít menší paměťové nároky, není však příliš vhodná jako náhrada výuky. Jako náhradu výuky lze však zařadit aplikaci, která si pamatuje jednotlivé uživatele a nabízí jejich správu a prohlížení studijních výsledků.

Mezi další otázky, které si musí položit autor kurzu, patří, jak vhodně kurz rozdělit na jednotlivé části a co bude každá kapitola obsahovat. Každá kapitola by měla určitě obsahovat nějaký studijní materiál. Je jedno, jestli to bude text s obrázky nebo video. Jde o to, aby studentům byly nějak předány nezbytné informace, kvůli čemuž vlastně vzniká celý kurz. Dále je doporučeno, pokud je to možné, aby byly nabitě informace nějak otestovány, ať už formou procvičení nebo testu, který se započítá do celkového hodnocení. Každá kapitola může být doplněna o další libovolné pomůcky, které uzná autor za vhodné.

Další základní otázkou je jak celý kurz vyhodnotit. Je zcela na autorovi jak se s tímto vyrovná. Speciální software pro tvorbu výukových kurzů nabízejí většinou několik možností vyhodnocení díky zabudovaným modulům, které se do jednotlivých částí vkládají. U každého modulu lze nastavit určitý způsob hodnocení a počet opakování. Mezi časté způsoby hodnocení patří výběr nejlepšího dosaženého výsledku, poslední absolvovaný pokus nebo pouze jeden povolený pokus. Výsledek jednotlivého pokusu lze také nastavit na různé možnosti – sečtení pouze správných odpovědí, odečítání špatných odpovědí, ... Lze také vkládat otázky s jednou správnou odpovědí, s více správnými odpověďmi nebo také například slovní doplnění. V případě programování vlastní aplikace je toto opět zcela v režii autora. Autor může ponechat na správcích kurzů, aby vybrali vhodnou možnost k jednotlivým kapitolám, nebo může vytvořit funkčně slabší výukový kurz s pouze jednou možností hodnocení a stanoveným počtem opakování.

Zbývá zde ještě jedna základní otázka týkající se funkčnosti aplikace a tou je komunikace mezi uživateli. Speciální software pro tvorbu výukových kurzů tuto možnost podporují formou zpráv mezi účastníky nebo pomocí diskuzních fór. V případě programování vlastního kurzu záleží na typu aplikace. Jedná-li se o webovou aplikaci, není problém zde vložit nějakou diskuzi, návštěvní knihu nebo vzkazníček. Tyto miniaplikace nemusí sám autor ani programovat, dají se dohledat na internetu volně ke stažení. Co se týká desktopových aplikací, musel by autor sám naprogramovat nějaký chat nebo komunikovat se studenty pomocí nějakého komunikátoru nebo emailu.

2.4 Obsah výukového kurzu

Tvorba obsahu kurzu čeká autora výukového kurzu, ať už zvolí cestu vývoje pomocí specializovaného software nebo vlastní aplikaci. K tvorbě obsahu potřebuje autor další podpůrný software. Zejména bude autor kurzu potřebovat nějaký textový editor, kam bude psát samotný text popřípadě vkládat doplňující obrázky. Dále jsou vhodné software pro upravování obrázků, pokud bude kurz obsahovat i zvuk tak software pro nahrávání a úpravu zvuku a pokud zde budou vkládána i videa, tak software pro zpracování videí.

Co se samotného obsahu kurzu týká, tak každý vysvětlující studijní materiál by měl být smysluplný, srozumitelný a úplný. Určitě by neměl být nepřehledný a měl by předat všechny potřebné informace studentům. V procvičovací části, je-li tedy v kurzu obsažená, by měly být jasně dány podmínky, jak je tato část vyhodnocena a kolikrát jí může student opakovat. Všechny úlohy v procvičovací části musejí mít smysluplné zadání, aby studenti pochopili, na co mají jak odpovídat.

3 Java technologie vhodné k tvorbě výukových kurzů

Tato kapitola je zaměřena na popis technologií programovacího jazyka Java, které jsou vhodné k tvorbě výukových kurzů.

3.1 GUI

V této části jsou popsány základní vlastnosti a funkčnost grafického uživatelského rozhraní.

3.1.1 Knihovny AWT a Swing

Knihovna `java.awt` obsahuje grafické komponenty určené k tvorbě GUI aplikací. AWT je tzv. *heavyweight* knihovna, což znamená, že se její komponenty na různých operačních systémech nevykreslují totožně.

Knihovna `javax.swing` obsahuje také grafické komponenty určené k tvorbě GUI aplikací, na rozdíl od AWT je ale *lightweight* knihovnou, tudíž jsou její komponenty na různých systémech vykresleny stejně. Je to způsobeno tím, že knihovna Swing provádí téměř veškeré vykreslování sama na rozdíl od AWT.

Obě knihovny umožňují práci s vlákny, ale ani jedna není vláknově bezpečná. Budou-li přistupovat dvě vlákna k jedné komponentě najednou, může aplikace skončit s chybou [4].

3.1.2 Grafické komponenty

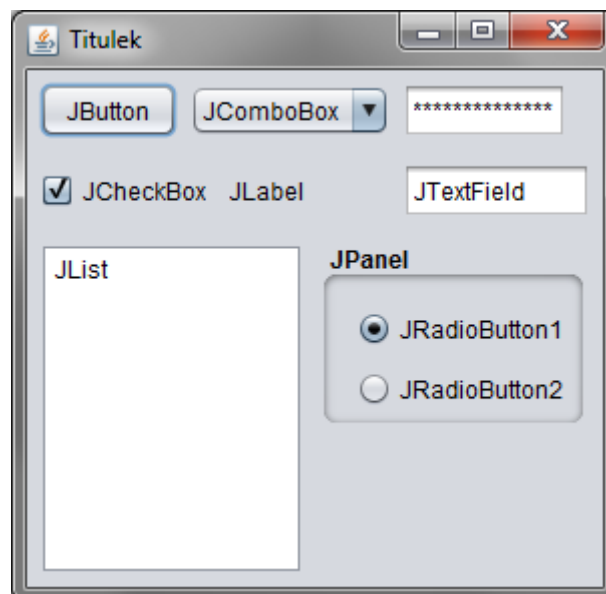
Všechny komponenty mají shodné vlastnosti, které ovlivňují její vzhled. Jsou to například výška, šířka, barva písma, atd. Každá komponenta má však navíc vlastnosti, které jsou specifické přímo pro ni. Dále mají všechny komponenty možnost reagovat na události, jako stisknutí tlačítka myši, stisknutí klávesy, změna velikosti, atd. Při vkládání komponent z knihovny Swing na formulář vytváříme jednotlivé instance pomocí třídy `Jnavevkomponenty`. Mezi nejpoužívanější komponenty patří (viz. Obrázek 1) [13]:

- `Button` – stiskací tlačítko, kterému se dá nastavit popisek a obrázek. Tlačítko se nejčastěji používá v kombinaci s událostí `mouseClicked()`, neboli při stisku tlačítka myši se vykoná nějaká akce.
- `ButtonGroup` – skupina tlačítek, kde pouze jedno může být vybráno jako aktuální (vlastnost `selected`). Může uchovávat libovolná tlačítka, která jsou děděna od `AbstractButton`. Nejčastěji udržuje instance `JRadioButton`, `JRadioButtonMenuItem` nebo `JToggleButton`. Nemá smysl zde udržovat instance `JButton` nebo `JMenuItem`, protože tyto komponenty nemají vlastnost `selected`.
- `ComboBox` – rozbalovací list s možnostmi, kde může být vybrána pouze jedna možnost jako aktuální. Je-li u `ComboBoxu` nastavena vlastnost `editable`, objeví

se zde editovací pole, kam lze psát vybranou hodnotu. Pomocí vlastnosti `model` lze nastavit implicitní hodnoty, které budou zobrazeny mezi možnostmi `ComboBoxu`. Implicitní nastavení vybrané hodnoty lze nastavit pomocí vlastností `selectedIndex` nebo `selectedItem`.

- `FileChooser` – dialog pro výběr souboru. Základní vlastností dialogu je `dialogType`, kde můžeme nastavit, bude-li se soubor ukládat nebo otevírat. Další vlastnosti jsou `fileFilter`, který nastavuje, s jakými soubory bude `FileChooser` pracovat a `fileSelectionMode`, který určuje, jestli se budou zobrazovat jen složky, jen soubory nebo oboje. `FileChooser` lze zobrazit jako dialogové okno nebo přímo ve formuláři obsažený v kontejneru.
- `Frame` – formulářové okno, na které se vkládají jednotlivé komponenty. U formulářového okna se dá nastavit pomocí vlastnosti `title` titulek okna a pomocí vlastnosti `defaultCloseOperation` se nastavuje akce, která nastane při stisku zavíracího křížku. Jsou zde možnosti `hide` (schování okna), `exit` (ukončení celé aplikace), `do_nothing` (nic se nestane) a `dispose` (odstranění okna z aplikace).
- `CheckBox` – zaškrťovací položka, která může být vybrána nebo nevybrána nastavením vlastnosti `selected`. Pokud je více komponent `CheckBox` obsaženo v `JButtonGroup`, může být libovolný počet komponent `CheckBox` označen.
- `Label` – komponenta pro uchování textu, obrázku nebo kombinace obojího. `Label` není fokusovatelný a nereaguje na vstupní události. To znamená, že se na něj nelze přepnout tabulátorem a je používán pouze pro zobrazování textů a obrázků.
- `List` – výpisový list objektů, kde může být vybrán pouze jeden nebo více objektů jako aktuální vzhledem k nastavení vlastnosti `selectionMode`. Ta nabývá hodnot `single` (výběr jednoho objektu), `single_interval` (výběr více objektů v rámci jednoho intervalu) a `multiple_interval` (výběr více objektů v rámci více intervalů). Pomocí vlastnosti `model` lze nastavit implicitní hodnoty, které budou zobrazeny mezi možnostmi `Listu`.
- `OptionPane` – zobrazení vstupního nebo informačního dialogu. Jsou zde k dispozici 4 typy dialogů - `confirmDialog` (dialog s dotazem a možností ano/ne, ...), `inputDialog` (dialog s dotazem na zadání nějaké vstupní hodnoty), `messageDialog` (dialog, který zobrazí informaci o proběhlé události) a `optionDialog` (kombinace předchozích tří). `OptionPane` lze zobrazit jako dialogové okno nebo přímo ve formuláři obsažený v kontejneru.

- `Panel` – komponentový kontejner. Slouží k zřehlednění formulářového okna. O komponentě `Panel` více v další kapitole.
- `PasswordField` – komponenta pro uchování textu, která zobrazuje jednotlivé znaky jako znak `*`. Díky této vlastnosti je tato komponenta vhodná pro zadávání hesel nebo tajných textů.
- `RadioButton` – přepínací položka, která může být vybrána nebo nevybrána nastavením vlastnosti `selected`. Pokud je více komponent `RadioButton` obsaženo v `JButtonGroup`, může být označena pouze jedna komponenta `RadioButton`.
- `ScrollPane` – panel, který může obsahovat komponentu větších rozměrů, než je on sám. V takovém případě se na větších rozměrech mohou zobrazit posuvníky. Zobrazení posuvníků se nastavuje vlastnostmi `horizontalScrollBarPolicy` a `verticalScrollBarPolicy`. Tyto vlastnosti mohou nabývat možností `as_needed` (zobrazí se jen v případě, je-li obsah vnitřní komponenty větší), `always` (zobrazí se vždy) a `never` (nezobrazí se nikdy).
- `Table` – klasická tabulka s několika sloupci a řádky. Pomocí vlastnosti `model` lze nastavit implicitní hodnoty, které budou zobrazeny v komponentě `Table`.
- `TextField` – komponenta pro uchování textu. Na rozdíl od `Labelu` je tato komponenta fokusovatelná a reaguje na vstupní události, pokud je vlastnost `editable` nastavena na `true`.

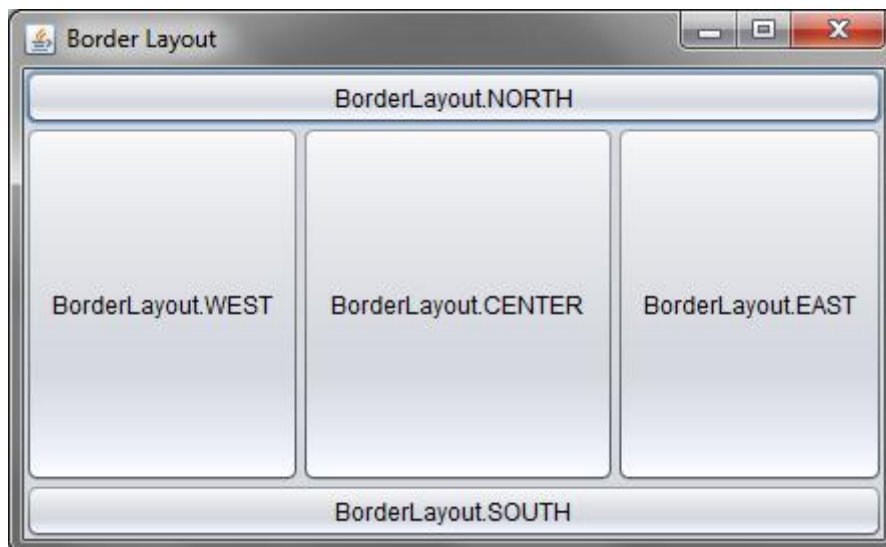


Obrázek 1 - GUI Komponenty

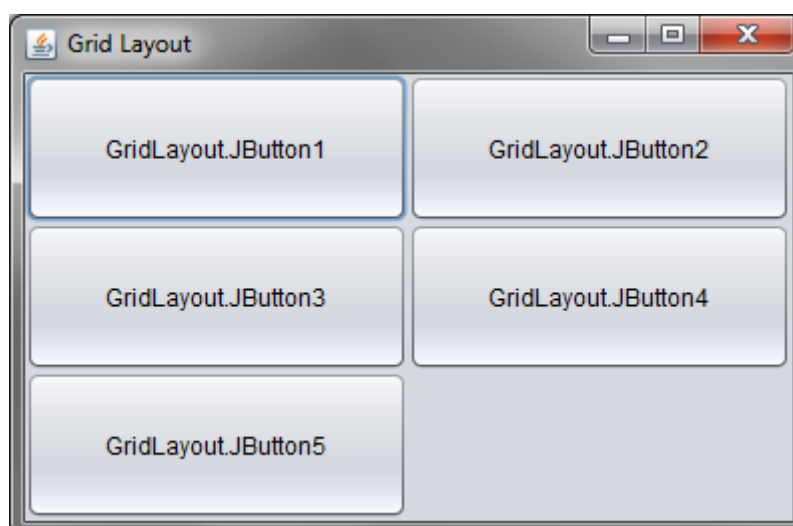
3.1.3 Kontejnery – správce rozložení

Všichni potomci třídy `Container` mají nastavený vlastní `Layout Manager` neboli správce rozložení. Ten zajišťuje rozložení jednotlivých komponent na formuláři. Nejpoužívanější komponentou v tomto ohledu je `JPanel`, který se využívá ke skládání složených komponent. Jednotlivé typy správců rozložení jsou [6]:

- `Absolute Layout` – komponenty jsou zarovnány na své absolutní pozice a nijak se nemění. Při zmenšení okna na menší velikost, než je pozice některé z komponent, zůstává komponenta stále na svém místě. Není tedy viditelná při zmenšení.
- `Border Layout` – zarovnává jednotlivé komponenty do všech světových stran a na střed (viz. Obrázek 2). Jedná se o nejpoužívanější rozložení u aplikací, které jsou dělené na obslužnou a hlavní část. Stačí na střed umístit komponentu `JPanel`, která bude vykreslovat hlavní obsah a na libovolnou stranu umístit jinou komponentu `JPanel`, která bude aplikaci ovládat.
- `Box Layout` – zarovnání všech komponent do jedné horizontální nebo vertikální linie. Do jaké linie se komponenty zarovnají, záleží na nastavení vlastnosti `axis`. Všechny komponenty jsou velikostně přizpůsobeny, aby vyplnily celý prostor kontejneru a vešly se do jedné linie. Není možné, aby některá komponenta přetekla na nový řádek.
- `Card Layout` – správce, který uchovává několik komponent. V jeden moment smí být však zobrazena pouze jedna. Je vhodný v kombinaci s `Border Layoutem` jako přepínač hlavního obsahu aplikace. Ovládací panel tak určuje, která komponenta se vykreslí v hlavním obsahu aplikace.
- `Flow Layout` – jednotlivé komponenty jsou skládány za sebou v pořadí, jak byly přidány. Nevejdou-li se vedle sebe, další komponenta se přidá na nový řádek. Pomocí vlastností `alignOnBaseline` a `Alignment` lze nastavit zarovnání jednotlivých komponent. Vlastnosti `horizontalGap` a `verticalGap` pak nastavují vnitřní okraje kontejneru.
- `Grid Layout` – mřížkové rozmístění. Vytvoří se mřížka, jejíž všechny komponenty mají stejnou velikost (viz. Obrázek 3). Počet sloupců a řádků mřížky je ovlivněn nastavením vlastností `rows` a `columns`.
- `Grid Bag Layout` – mřížkové rozmístění na pozadí. Vytvoří se mřížka, jejíž políčka mají stejnou velikost. Každá komponenta zabírá několik políček, mohou tak na rozdíl od `Grid Layoutu` mít komponenty různé velikosti.
- `Null Layout` – žádná správa umístění. Kam se komponenta vloží, tam také zůstane.



Obrázek 2 - Border Layout



Obrázek 3 - Grid Layout

3.2 Kolekce

Kolekce se používají k práci s předem neznámým počtem objektů. Většinu kolekcí najdeme v balíčku `java.util`. Jsou součástí `Collections Frameworku` a realizují všechny důležité datové struktury. Důležitou součástí kolekcí jsou iterátory, což jsou prostředky zajišťující sekvenční přístup k datům. Používají se v případech, kdy je potřeba projít celou kolekci prvek po prvku a pracovat s daty jednotlivých prvků.

Volba vhodné kolekce závisí na několika faktorech. Přístup k datům může být náhodný nebo sekvenční. Kolekce s náhodným přístupem se dají procházet i sekvenčně, což je ale v případě, kdy je potřeba přistoupit jen k jednomu hledanému prvku zbytečné. Dalším důležitým faktorem je manipulace s daty. Rozhodující je zde četnost čtení, přidávání, odebrání a přesunů prvků. Následuje výčet nejčastěji používaných rozhraní a jednotlivých kolekcí [10].

3.2.1 Seznamy

- `ArrayList` – seznam nad polem. Umožňuje přímý přístup k prvkům. Potřeba relokace paměti při změně velikosti.
- `LinkedList` – obousměrně zřetěžený spojový seznam. Využívá sekvenční přístup k prvkům. Nepotřebuje relokovat paměť při změně velikosti.
- `Vector` – stejná funkčnost jako `ArrayList`, navíc přístup z více vláken, ale je zastaralý. Dnes se používá metoda `synchronizedList()` třídy `Collections`.

3.2.2 Mapy

- `HashMap` – mapa, která vyhledává prvky s využitím hashovací funkce. Negarantuje žádné řazení prvků.
- `TreeMap` – mapa, jejíž klíče jsou seřazeny podle předaného `Comparatoru`. Je implementována jako výškově vyvážený binární strom.

3.2.3 Množiny

Kolekce prvků, která neumožňuje jejich duplicitu. Vlastnosti z hlediska řazení dat a jejich procházení jsou u `HashSet` a `TreeSet` podobné jako u `HashMap` a `TreeMap` [5].

3.3 Datové streamy

V Javě jsou základní třídy pro práci s datovými streamy obsaženy v balíčku `java.io`. Základní typy streamů jsou dva – znakový a binární. Znakové streamy pracují se skupinami bajtů podle nastavení prostředí, zatímco binární pracují přímo s jednotlivými bajty. Znakové streamy jsou odvozeny od tříd `Reader` a `Writer`. Binární jsou odvozeny od tříd `InputStream` a `OutputStream`. Při používání datových streamů se nesmí zapomínat na výjimky. Výjimky ošetřují chyby spojené s komunikací mezi zařízeními [1] [8].

3.3.1 Práce se soubory

Nejčastějším způsobem komunikace mezi programem a vnějším prostředím je práce se soubory. Balíček `java.io` obsahuje třídu `File`, která je určena k práci se soubory. Objekt třídy `File` je jakousi abstraktní cestou k souboru a nabízí nám řadu operací. Mezi ty nejčastěji využívané patří [9]:

- `getPath()` – vrátí cestu k souboru zadanou při vytváření objektu (absolutní, relativní, atd.),
- `getAbsolutePath()` – vrátí absolutní cestu k souboru,
- `getName()` – vrátí pouze jméno souboru,

- `exists()` – vrátí `true`, jestli soubor existuje, jinak `false`,
- `length()` – vrátí velikost souboru,
- `createNewFile()` – vytvoří nový soubor.

3.3.2 Serializace a externalizace dat

Serializace je ukládání objektů do takového formátu, aby mohl být uložen nebo poslán po síti. Pro práci se serializací lze využít streamy `ObjectInputStream` nebo `ObjectOutputStream` a rozhraní `Serializable`. Použití serializace je podmíněno třemi podmínkami:

- třída ukládaného objektu musí implementovat rozhraní `Serializable`,
- každý předek třídy, který neimplementuje `Serializable` musí mít veřejný bezparametrický konstruktor,
- všechny položky objektu musí být primitivního typu nebo musejí být sami serializovatelné.

Vlastní serializační protokol lze nastavit pomocí implementace rozhraní `Externalizable` namísto `Serializable` u třídy objektu, s nímž se pracuje. Použití externalizace má stejné podmínky jako serializace a práce s objekty implementujícími `Externalizable` je také stejná. Rozhraní `Externalizable` implementuje dvě metody:

- `readExternal(ObjectInput in)` – volá metody třídy `DataInput` pro čtení primitivních typů a `readObject()` pro čtení objektů, řetězců a polí.
- `writeExternal(ObjectOutput out)` – volá metody třídy `DataOutput` pro zápis primitivních typů a `writeObject()` pro zápis objektů, řetězců a polí [7].

Pomocí správně implementovaného serializačního protokolu lze urychlit komunikaci mezi programem a souborem. Rozdíl mezi `Serializable` a `Externalizable` je v tom, že `ObjectOutputStream` nepoužívá buffer, zatímco `OutputStream` ano [11].

Další velkou výhodou implementace rozhraní `Externalizable` jsou právě jeho dvě metody pro zápis a čtení dat, pomocí kterých programátor přebírá plnou kontrolu nad komunikací mezi třídou a komunikačním zařízením. Pokud třída `Class` implementuje rozhraní `Externalizable` a zavolá se metoda `writeObject(Class object)` při práci s komunikačním zařízením, je volána metoda `writeExternal(ObjectOutput out)` z třídy `Class` a instance objektu je uložena tak, jak je nadefinována v této metodě [12].

4 Analýza a návrh implementace

V této kapitole je provedena analýza řešeného problému a navrženo vhodné řešení. Nejnutnější předpoklady k tvorbě výukových kurzů jsou splněny, je tedy možné začít s analýzou tvorby výukového kurzu. Úkolem této práce je naprogramovat GUI aplikaci v jazyce Java, která bude využívat práci se soubory.

4.1 Rozdělení a vyhodnocení kurzu

Kurz by měl být přehledně rozdělen do několika kapitol dle obsáhlosti látky goniometrické funkce pro střední školy. Ze začátku by se měl student seznámit s jednotlivými goniometrickými funkcemi a dále s jejich aplikací v příkladech. Ze znalosti probírané látky byl kurz rozdělen do celkem šesti kapitol:

- Funkce sinus,
- Funkce cosinus,
- Funkce tangens a cotangens,
- Goniometrické vzorce,
- Goniometrické rovnice a nerovnice,
- Obecný trojúhelník.

První tři kapitoly seznamují studenty s jednotlivými goniometrickými funkcemi, čtvrtá kapitola se věnuje úpravám a vzájemné závislosti goniometrických funkcí mezi sebou, v páté kapitole se studenti učí, jak řešit rovnice a nerovnice, ve kterých se neznámá hodnota nachází v argumentu funkce a poslední kapitola vysvětluje řešení sinové a cosinové věty v rámci obecného trojúhelníka.

Každá kapitola by měla obsahovat nějaký studijní materiál. Nejvýhodnějším řešením bylo vytvořit studijní materiál textový doplněný vysvětlujícími obrázky. Dále každá kapitola obsahuje test na procvičení látky z dané kapitoly a celý kurz je zakončen závěrečným testem.

Při tvorbě testů je vhodné si rozmyslet, jaké budou jejich vlastnosti. Bude-li test možné absolvovat jednou či vícekrát, budou-li testy obsahovat otázky s více správnými odpověďmi, budou se strhávat body za špatné odpovědi, jaká bude váha testů v celkovém hodnocení, atd. V případě kurzu na téma výuka goniometrických funkcí není možné vytvořit test s více správnými odpověďmi, tudíž je vždy pouze jedna správná odpověď a je nutné odpovědět na všechny otázky (test nelze jinak vyhodnotit), takže se pouze přičítají kladné body za správně zodpovězené otázky. Vzhledem k tomu, že výukový kurz obsahuje šest kapitol a závěrečný test, bylo zvoleno, že každý test z jednotlivých kapitol bude mít váhu 10% celkového hodnocení kurzu a závěrečný test bude mít váhu 40%. Co se týká počtu pokusů, tak u testů z kapitol je možné každý test absolvovat celkem třikrát a do

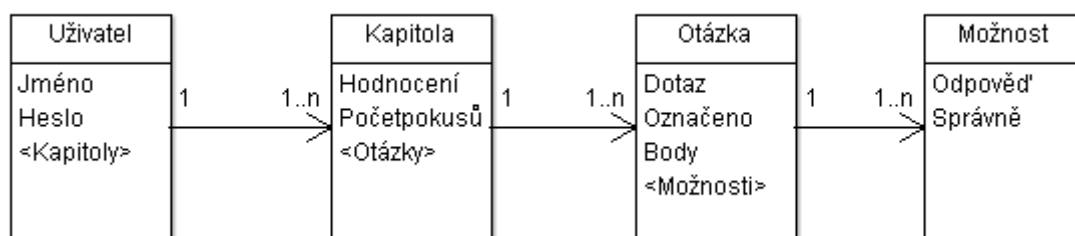
celkového hodnocení se započítává nejlepší dosažený výsledek. Závěrečný test lze absolvovat pouze jednou.

4.2 Správa uživatelů a uchování vyhodnocení

Vzhledem k tomu, že aplikace by měla být určená pro výuku středních škol, je určitě vhodné, aby si pamatovala jednotlivé uživatele i jejich údaje a studijní výsledky. Představa aplikace je taková, že kurz poběží na počítačích, kde se budou střídát studenti, je tedy vhodné vytvořit takový kurz, který si bude pamatovat více uživatelů, a ti budou mít přístup ke svému kurzu za pomoci hesla. Je také dobré, aby byl obsah kurzu, co se týká testů dynamický (každý uživatel měl jiné otázky a jinak seřazené možnosti) je tedy vhodné ukládat nastavení pro každého studenta zvlášť do jedné entity. Teoretická stránka kurzu je pro všechny uživatele stejná, není tedy nutné ji udržovat vícekrát a neměla by být součástí ukládaných údajů o každém studentovi.

Aby mohla být část kurzu dynamická, je potřeba právě komunikace se souborem. Komunikace by měla být šifrovaná, aby se ze souboru nedaly vyčíst žádné údaje o kurzu a hlavně ne hesla jednotlivých studentů. Nejvhodnější k ukládání hesel je použití jednosměrné hashovací funkce, která zašifruje heslo a nelze ho odšifrovat zpět. K porovnání hesel při přihlášení se zadané heslo také zahashuje a porovnají se dva hashe mezi sebou. V kurzu bylo použito kódování BASE64 a jednosměrná hashovací funkce MD5.

K uchování samotného obsahu kurzu jednotlivých účastníků je potřeba vytvořit strukturu uživatelů (viz. Obrázek 4). U každého uživatele by se mělo ukládat jméno, zašifrované heslo a odkazy na jednotlivé kapitoly. Každá kapitola by měla obsahovat hodnocení testu, počet absolvovaných pokusů a odkazy na jednotlivé otázky. U otázky by se měly uchovávat dotaz otázky, která možnost byla označená, kolik získal student za danou otázku bodů a odkazy na jednotlivé možnosti. Nejnižší uchovávanou vrstvou jsou možnosti, kde by se měly uchovávat odpověď a je-li možnost správně nebo špatně. Takto organizovaná struktura využívá na odkazy k objektům nižší úrovně objekty třídy `ArrayList` z důvodu lepšího přístupu k jednotlivým objektům.



Obrázek 4 - Dynamický obsah kurzu

Do aplikace výukový kurz byla přidána funkce exportovat kurz, která je schopná uložit do souboru, jehož umístění si student vybere, aktuální stav výukového kurzu jednoho studenta. Tento student pak předá lektorovi svůj soubor a lektor si jej otevře

v doplňkové aplikaci správa kurzu, kde vidí jednotlivé testy studenta a navíc i správné řešení otázek, v případě že student zvolil špatnou odpověď, což student nevidí.

4.3 Obsah výukového kurzu

Jak už bylo řečeno, co se týká teorie, tak jsou v kurzu použity studijní materiály textové doplněné obrázky. Vysvětlovaná látka ve výukovém kurzu je v souladu s knihou Přehled užití matematiky [2]. Samotné texty jsou napsány v programu Microsoft Office Word a obrázky byly vytvořeny v programu Ipe 6.0 a Matlabu. Případné úpravy popisků obrázků pak byly provedeny v Malování. Po vytvoření textových dokumentů byl proveden export do PDF, odkud se následně vyexportovaly snímky jednotlivých stránek. Snímky stránek byly poskládány pod sebe a vznikl tak pro každou kapitolu jeden dlouhý obrázek jako studijní materiál, který se dá lehce importovat do GUI.

Vzhledem k tomu, že programovací jazyk Java nemá žádný prostředek pro zápis rozsáhlých matematických rovnic, musely být do testů vkládány všechny materiály jako obrázky. Pro vytvoření textového obsahu byl použit nástroj pro tvorbu rovnic v programu Microsoft Office Word, odkud se poté exportovaly obrázky pro zadání otázek, jednotlivé možnosti otázek, pro označené možnosti a pro vyhodnocení možnosti. Varianta tří různých obrázků možnosti byla vybrána proto, že k výběru odpovědi byly vybrány komponenty `JRadioButton`, které mohou obsahovat obrázek a text, a je-li vložen obrázek, je jím nahrazen klasický přepínač a je potřeba nějak zdůraznit vybrání a následné vyhodnocení možnosti.

4.4 Komunikace se soubory

Je důležité si uvědomit, že nesmí dojít ke kolizím, tudíž by měl se souborem komunikovat pouze jeden jediný objekt. Objekt by také měl udržovat stav uživatelů, aby se předcházelo častým čtením a zápisům do souboru. Tudíž by bylo nejlepší, kdyby se při startu aplikace načetli všichni uživatelé do tohoto stavu a při ukončení aplikace, se do souboru uložil celý upravený stav nebo pouze upravily změny stavu.

Obě aplikace `GonioFce` obsahují jeden statický objekt třídy `Stream`. Tento objekt má jeden atribut obsahující seznam odkazů na jednotlivé uživatele typu `ArrayList`. Dále se zde nacházejí metody pro práci se souborem `users.gfs` a přístupové metody, které spolupracují se samotným kurzem.

Programy `GonioFce` obsahují také vlastní serializační protokol pro zápis a čtení jednotlivých dat uživatelů. Ten je zde využit zejména proto, že bylo potřeba nadefinovat šifrovaný zápis jednotlivých prvků zapisovaných objektů a jejich následné odšifrování při čtení. Poté stačí ve třídě `Stream` zavolat zápis nebo čtení celého seznamu uživatelů a všechny podobjekty jsou uloženy nebo načteny přesně tak, jak je definováno v našem serializačním protokolu. Dalším důvodem aplikace rozhraní `Externalizable` je menší doba komunikace mezi programem a souborem.

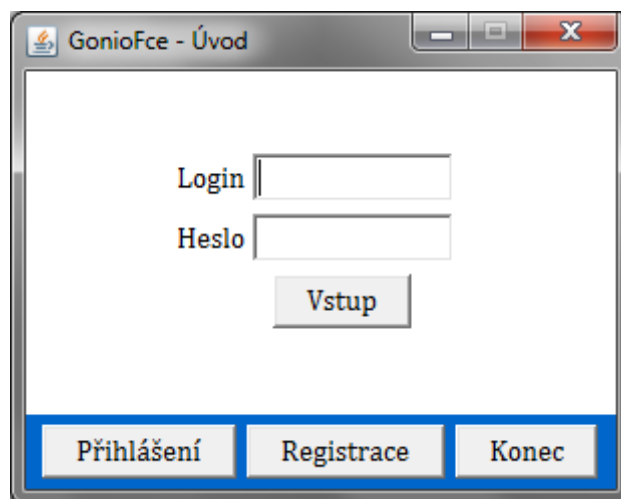
5 Výukový kurz GonioFce

Tato kapitola popisuje implementaci výukového kurzu GonioFce. Výukový kurz je tvořen následující adresářovou strukturou (kompletní adresářová struktura viz. Příloha C – Kompletní adresářová struktura):

/	kořenový adresář
... aplikace/	balíček obsahující zdrojové kódy aplikace výukový kurz
... .. Kurz/	obsahuje GUI výukového kurzu a pomocné šablony
... .. Main/	zde je spouštěcí třída programu, instance streamu a GUI oken
... .. Uvod/	obsahuje GUI úvodního formuláře
... entity/	zde jsou uchovány jednotlivé entity dynamické části kurzu
... obrazky/	obrázky vkládané do obsahu kurzu (studijní materiály a testy)
... rozhrani/	zde jsou pomocná rozhraní k entitám
... stream/	obsahuje třídu <code>Stream</code>
... streamEntity/	obsahuje potomky tříd ze složky entity určené ke komunikaci se soubory

5.1 Úvod

Následuje popis úvodního formuláře aplikace GonioFce (viz. Obrázek 5). Tento formulář se načítá při spuštění aplikace. Nachází se zde stránka pro přihlášení uživatele a stránka pro registraci uživatele. Na pozadí je při spuštění zahájena první komunikace `Streamu` se souborem `users.gfs` a tou je načtení existujících uživatelů do seznamu uživatelů ve `Streamu`. Pokud žádný uživatel ještě neexistuje, je zde vytvořen prázdný soubor `users.gfs` a seznam uživatelů ve `Streamu` se inicializuje jako nový `ArrayList`.



Obrázek 5 - GonioFce screen přihlášení

5.1.1 Registrace

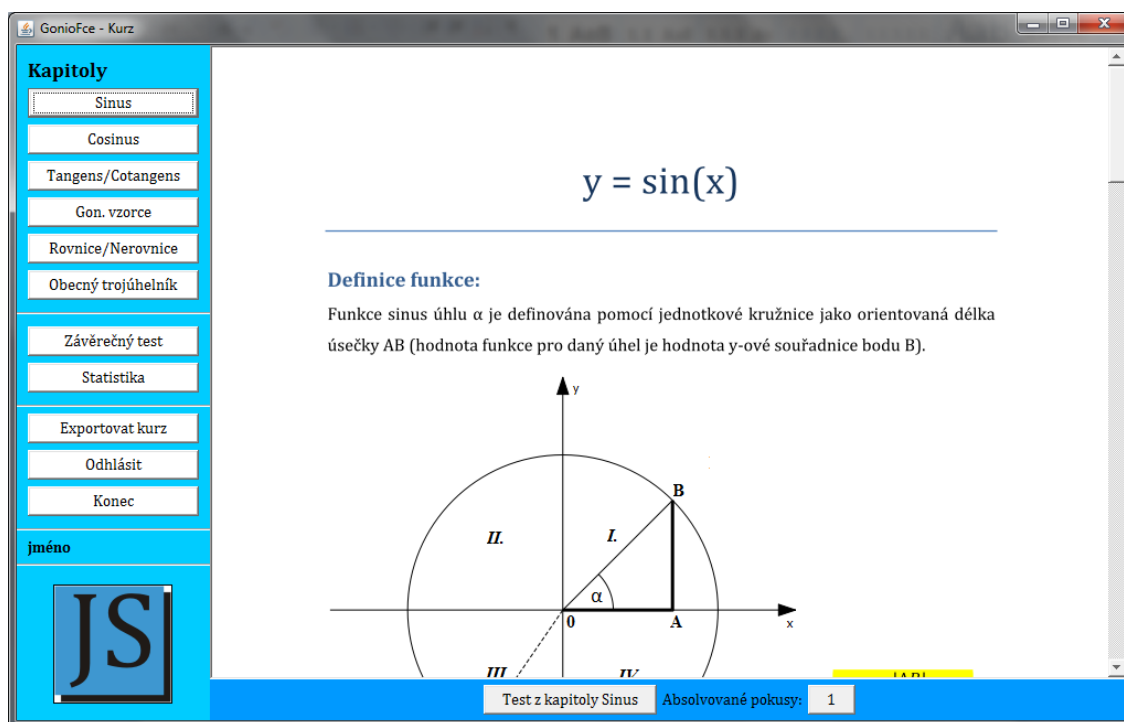
Pro registraci uživatele je vyžadováno zadání uživatelského jména, hesla a potvrzení hesla. Aplikace ošetřuje chyby zadání různých hesel, nevyplnění některého z požadovaných polí, existenci jména uživatele a zadání jména delšího než 20 znaků. Po vyplnění požadovaných vstupních údajů lze kliknout myší na tlačítko Registrace nebo stisknout klávesu enter. Aplikace uživatele buď upozorní na jednu z chyb, nebo nahlásí jeho úspěšnou registraci. Zatímco uživatel vidí hlášku o úspěšné registraci, na pozadí právě došlo jeho přidání do seznamu uživatelů ve Streamu a byl mu vytvořen obsah všech testů.

5.1.2 Přihlášení

Pro přihlášení uživatele musí mít uživatel vytvořený účet a musí zadat své přihlašovací jméno a heslo. Aplikace ošetřuje chyby zadání špatného hesla, nevyplnění některého z požadovaných polí a neexistenci uživatele. Po vyplnění jména a hesla lze kliknout myší na tlačítko Vstup nebo stisknout klávesu enter. Zobrazí se hláška s chybou nebo úspěšným přihlášením. Pokud se zobrazí úspěch, uživatel odklikne hlášení, zavře se úvodní okno a otevře se mu okno nové, kde se nachází samotný obsah kurzu.

5.2 Kurz

Tato část obsahuje popis formulářového okna výukového kurzu (viz. Obrázek 6). Okno je tvořeno dvěma panely – první je obslužné menu a druhé je obsah kurzu. U obsahu kurzu je nastaven Card Layout, který obsahuje celkem tři šablony panelů a je obsluhován pomocí obslužného menu. Obsah kurzu obsahuje šablony pro studijní materiál, testy a statistiku výsledků studenta v daném kurzu.



Obrázek 6 - GonioFce screen kurz

5.2.1 Obslužné menu

V obslužném menu se nacházejí tlačítka pro přepínání mezi jednotlivými kapitolami. V závislosti na přepnutí do některé z kapitol se mění obsah kurzu. Dále se dole na tmavší části obslužného panelu mění popisek tlačítka, které spouští test z dané kapitoly, a je zde zobrazen různý počet tlačítek absolvovaných pokusů dle počtu absolvovaných pokusů. Jakmile jsou absolvovány všechny tři pokusy z dané kapitoly, tlačítka na spuštění testu zmizí a jsou zde zobrazena pouze tlačítka na zobrazení absolvovaných pokusů.

Při stisknutí tlačítka myši nad tlačítkem, které spouští test se uživateli zobrazí upozornění, jestli chce doopravdy spustit test a kolik mu zbývá pokusů. Pokud potvrdí, je spuštěn test. Když zvolí, že nechce absolvovat test, je vše vráceno do původního stavu před zobrazením upozornění. Stejnou funkčnost má i další tlačítka v obslužném menu, které spouští závěrečný test. Ten je možné absolvovat pouze jednou, takže po jeho absolvování je zde zobrazeno vyhodnocení testu. Další funkcí obslužného menu je zobrazení statistiky uživatele. Po kliknutí na toto tlačítka se zobrazí uživateli průběh jeho studijních výsledků a průběžné hodnocení.

V předposlední části se nachází funkce související s uživatelským účtem. První funkcí je možnost exportovat kurz. Po stisknutí tohoto tlačítka je uživateli zobrazena hláška s návodem, ať zadá název souboru a koncovka bude doplněna automaticky. Po potvrzení této informace se uživateli otevře `JFileChooser`, pomocí kterého vybere uživatel umístění pro generovaný soubor a název souboru. Po potvrzení umístění a správně vyplněném názvu je vygenerován soubor `jmeno.gfc`, který obsahuje údaje o přihlášeném uživateli a tento soubor je nutné poslat lektorovi kurzu, který s tímto souborem pracuje v části správa kurzu.

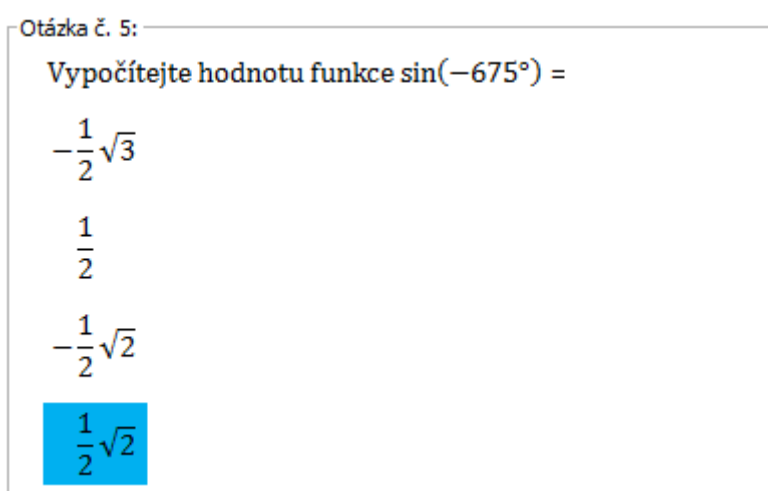
Další funkcí je odhlášení z kurzu pomocí tlačítka Odhlásit. Po stisku tohoto tlačítka, je zobrazen dotaz, chce-li se uživatel doopravdy odhlásit, a pokud ano, je uzavřeno okno kurzu a je obnoveno úvodní okno s možností registrace a přihlášení. Poslední funkcí je ukončení aplikace pomocí tlačítka Konec. Zde je funkčnost stejná jako u tlačítka odhlásit, akorát se po potvrzení neotevře úvodní okno a celá aplikace je ukončena. Při ukončení aplikace běží na pozadí uložení aktuálního seznamu uživatelů do souboru `users.gfs`. Poslední částí obslužného menu je zobrazení přezdívky právě přihlášeného uživatele a logo autora kurzu s plovoucím popiskem.

5.2.2 Obsah kurzu

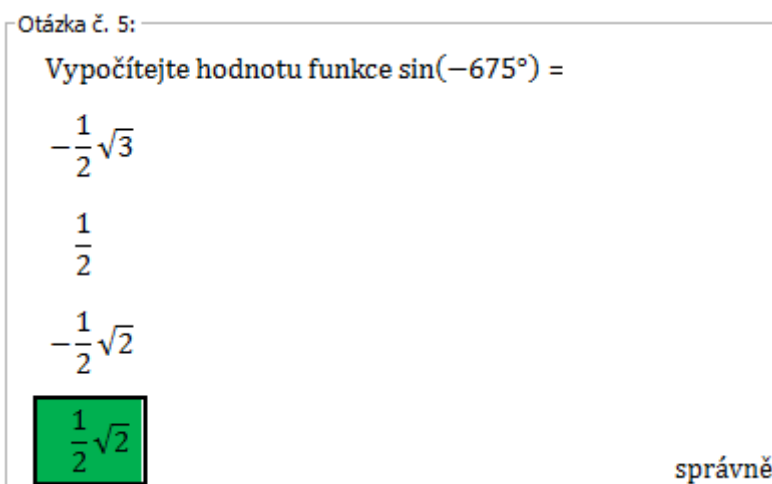
Obsah kurzu je šablona obsahující panel, který obsahuje instanci `JLabel`, přes celou svoji plochu. V závislosti na zvolení kapitoly z obslužného menu se zde mění obrázky s obsahem studijních materiálů.

Šablona statistika obsahuje dva objekty typu `JTable`. V první tabulce jsou vypsány výsledky všech tří pokusů testů z jednotlivých kapitol v procentuální úspěšnosti. Ve druhé tabulce je zobrazen výsledek závěrečného testu a je zde počítáno průběžné celkové hodnocení v procentuální úspěšnosti.

Poslední použitou šablonou je šablona na test. Aplikace využívá tyto šablony celkem dvě. První obsahuje pět otázek a používá se k testům v kapitolách. Druhá obsahuje otázek 12 a je použita pro závěrečný test. Šablona je v obou případech naplněna cyklem for-each přes jednotlivé komponenty. Šablona také obsahuje vyhodnocovací tlačítko Vyhodnotit, které je možné stisknout až v případě, že jsou zodpovězeny všechny otázky, jinak se zobrazí chybová hláška. Při vybrání možnosti se změní barva jejího pozadí z bílé na modrou (viz. Obrázek 7). Po úspěšném vyhodnocení testu se uživateli zobrazí hodnocení, které je ihned uloženo do studentových uživatelských dat a zapsáno do statistiky. Student může ihned také nahlédnout na vyhodnocení testu, kde se mu zobrazí jak písemně tak i barevně správnost vybraných odpovědí (viz. Obrázek 8, zelená barva je správně a červená barva je špatně).



Obrázek 7 - GonioFce screen testové otázky



Obrázek 8 - GonioFce screen vyhodnocení testové otázky

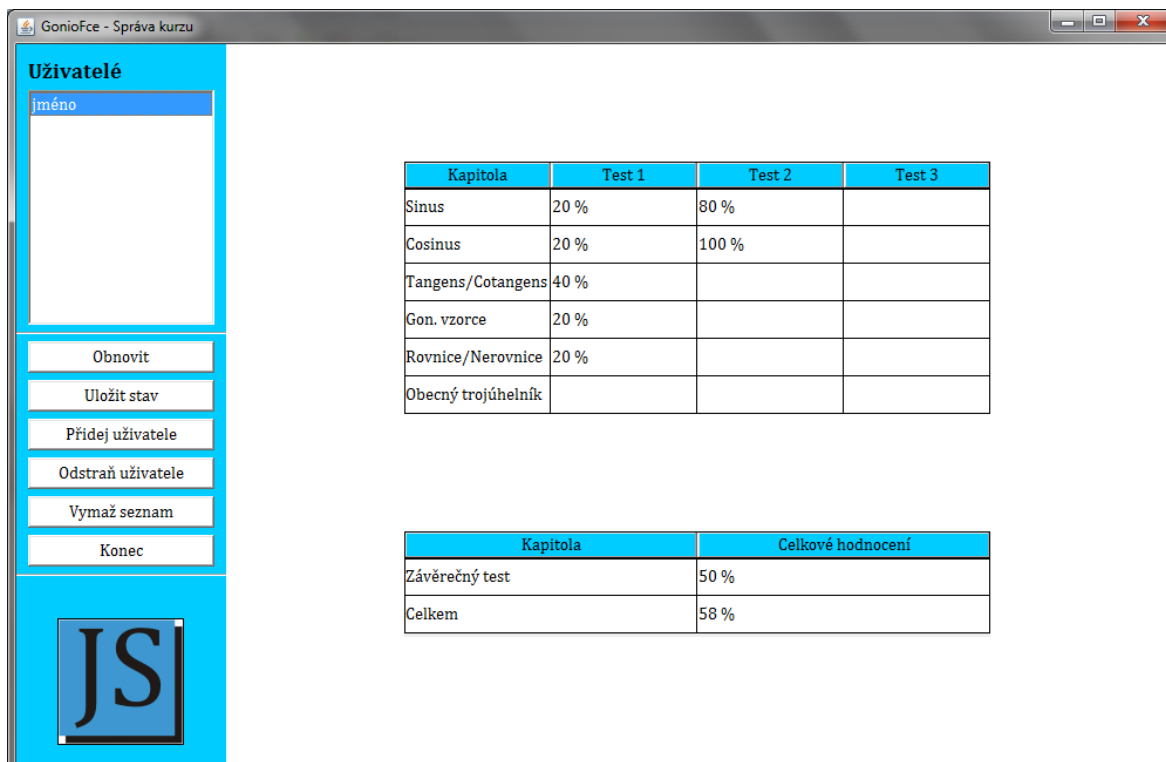
6 Správa výukového kurzu GonioFce

V této kapitole je popsána implementace správy výukového kurzu GonioFce. Správa výukového kurzu je tvořena následující adresářovou strukturou (kompletní adresářová struktura viz. Příloha C – Kompletní adresářová struktura):

/	kořenový adresář
... aplikace/	balíček obsahující zdrojové kódy aplikace výukový kurz
... ... Main/	zde je spouštěcí třída programu, instance streamu a GUI okna
... entity/	zde jsou uchovány jednotlivé entity dynamické části kurzu
... obrázky/	obrázky vkládané do obsahu kurzu
... rozhraní/	zde jsou pomocná rozhraní k entitám
... stream/	obsahuje třídu Stream
... streamEntity/	obsahuje potomky tříd ze složky entity určené ke komunikaci se soubory

6.1 Správa uživatelů

Tato část obsahuje popis formulářového okna správy výukového kurzu (viz. Obrázek 9).



Obrázek 9 - GonioFce screen správa kurzu

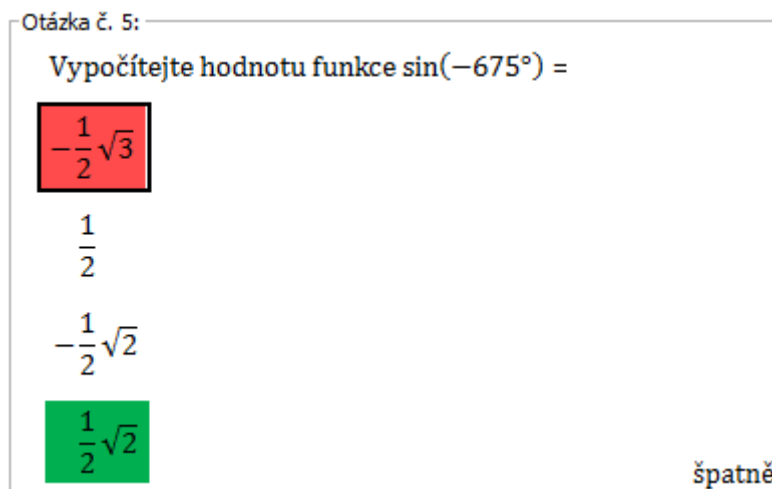
Správa uživatelů obsahuje v levé části aplikace uživatelskou část. Jako první seshora se zde nachází seznam uživatelů, které momentálně lze spravovat. Seznam uživatelů je realizovaný pomocí instance objektu `JList`. Po vybrání uživatele se v pravé části vyplní tabulky testů a celkového hodnocení stejně jako v aplikaci výukový kurz. Seznam reaguje jak na klikání myši, tak i na pohyb klávesových šipek nahoru a dolů.

Pod seznamem uživatelů se nachází obslužné menu pro práci se seznamem uživatelů. Po stisknutí tlačítka obnovit bude uživatel dotázán, chce-li doopravdy obnovit stav aplikace ze souboru `users.gfs`, pokud ano, je přemazán aktuální stav aplikace, do posledního uloženého stavu. S tím souvisí hned další tlačítko uložit stav, které se zeptá, chce-li uložit uživatel aktuální stav, protože tím přemaže poslední uložený stav.

Další volbou je tlačítko přidej uživatele. Po stisknutí tohoto tlačítka se otevře `JFileChooser`, pomocí kterého uživatel vybere jednoho či více uživatelů, které chce přidat do seznamu uživatelů. Všechny soubory se studenty musejí mít povinně koncovku `gfc`, jinak nebude žádný z vybraných uživatelů přidán. Tlačítko odstraň uživatele, po potvrzení od uživatele, vymaže vybrané studenty. Pokud není žádný student vybrán, je uživatel upozorněn, že nevybral žádného studenta. Tlačítko vymaž seznam smaže celý seznam studentů a uvede aplikaci do stavu po zapnutí. Tlačítkem konec se ukončuje aplikace, stejně jako ve výukovém kurzu je zde uživatel dotázán, chce-li ukončit aplikaci a po potvrzení je program ukončen.

6.2 Zobrazení testů

Aby lektor věděl více, než jen procentuální výsledek studentů, je zde integrována ještě jedna pomocná funkce. A tou je klikání do tabulky. Pokud lektor klikne do tabulky do políčka, které obsahuje hodnotu nějakého testu, tak se mu ten test zobrazí v novém okně přesně tak, jak ho vidí vyhodnocený sám student s tím rozdílem, že lektor navíc vidí i zobrazení správné možnosti (viz. Obrázek 10).



Obrázek 10 - GonioFce screen vyhodnocení testové otázky 2

Závěr

Program GonioFce splňuje všechny požadavky podle zadání. Program je plně funkční a může být ihned nasazen do provozu. Studenti jsou nejdříve seznámeni s probíranou látkou, kde jsou jim představeny teoretické základy dané látky doplněné několika příklady s názorným postupem a popisem jednotlivých kroků. Poté mají možnost si natřikrát vyzkoušet, jak dané problematice porozuměli. Po absolvování všech výukových kategorií absolvují závěrečný test a mohou odeslat výsledky absolvovaného kurzu svému lektorovi.

Nad rámec zadání byla vypracována druhá aplikace GonioFce a to správa výukového kurzu. Tato aplikace je určena pro lektory, kteří si zde zobrazí hodnocení studentů a jejich absolvované testy. Oba programy obsahují spustitelné JAR soubory.

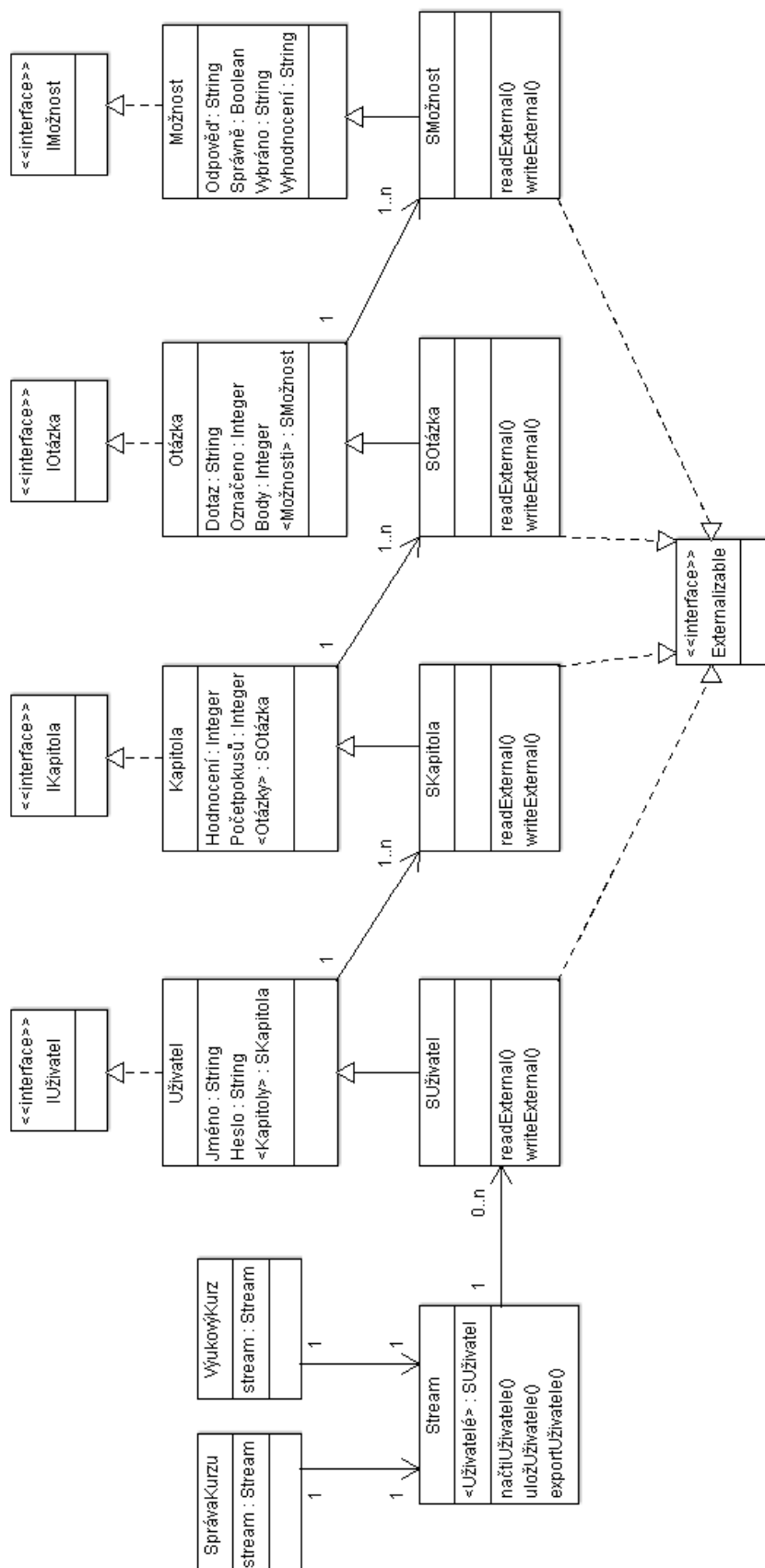
Díky této práci jsem se seznámil s novou variantou komunikace mezi programem a ostatními zařízeními. Tou variantou je externalizace. Její implementace je velmi jednoduchá a lehce pochopitelná. Navíc urychluje proces komunikace a programátor má možnost plné správy průběhu komunikace. Při programování složitějších struktur bych už nikdy nedal přednost serializaci před externalizací.

Do budoucna by bylo dobré rozšířit databázi příkladů a rozvinout tím různorodost obsahu kurzů mezi jednotlivými studenty. Osobně bych doporučoval program GonioFce jako doprovodnou pomůcku k výuce goniometrických funkcí na středních školách.

Literatura

- [1] HEROUT, Pavel. Učebnice jazyka Java. 1. vyd. České Budějovice: Kopp, 2001, 349 s. ISBN 80-723-2115-3.
- [2] REKTORYS, Karel. Přehled užití matematiky. 7. vyd. Praha: Prometheus, 2000, xxxii, 874 s. ISBN 80-719-6179-5.
- [3] ZOUNEK, Jiří. E-learning - jedna z podob učení v moderní společnosti. Vyd. 1. Brno: Masarykova univerzita, 2009, 161 s. ISBN 978-80-210-5123-2.
- [4] Java pro začátečníky: Grafické rozhraní. MIČKA, Pavel. *Algoritmy.net: příručka pro vývojáře* [online]. 2011 [cit. 2013-04-28]. Dostupné z: <http://www.algoritmy.net/article/39898/Graficke-rozhrani-23>
- [5] Java pro začátečníky: Kolekce. MIČKA, Pavel. *Algoritmy.net: příručka pro vývojáře* [online]. 2011 [cit. 2013-04-28]. Dostupné z: <http://www.algoritmy.net/article/34009/Kolekce-21>
- [6] Java pro začátečníky: Správci rozmístění. MIČKA, Pavel. *Algoritmy.net: příručka pro vývojáře* [online]. 2011 [cit. 2013-04-28]. Dostupné z: <http://www.algoritmy.net/article/41225/Spravci-rozmisteni-24>
- [7] Java pro začátečníky: Proudí serializace. MIČKA, Pavel. *Algoritmy.net: příručka pro vývojáře* [online]. 2011 [cit. 2013-04-28]. Dostupné z: <http://www.algoritmy.net/article/30183/Proudy-serializace-18>
- [8] Java: I/O operace I. JELÍNEK, Lukáš. *Linuxsoft.cz* [online]. 2005 [cit. 2013-04-28]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=836
- [9] Java: práce se soubory. JELÍNEK, Lukáš. *Linuxsoft.cz* [online]. 2005 [cit. 2013-04-28]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=866
- [10] Java: kontejnery I. JELÍNEK, Lukáš. *Linuxsoft.cz* [online]. 2005 [cit. 2013-04-28]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=735
- [11] Externalizable vs DataOutputStream/DataInputStream. *OTN Discussion Forums* [online]. 2010 [cit. 2013-04-28]. Dostupné z: <https://forums.oracle.com/forums/thread.jspa?threadID=1150222>
- [12] Difference between Serializable and Externalizable in Java Serialization. *Javarevisited* [online]. 2012 [cit. 2013-04-28]. Dostupné z: <http://javarevisited.blogspot.cz/2012/01/serializable-externalizable-in-java.html>
- [13] Javax.swing. *Java™ Platform, Standard Edition 7 API Specification* [online]. © 1993, 2013 [cit. 2013-04-28]. Dostupné z: <http://docs.oracle.com/javase/7/docs/api/javax/swing/package-frame.html>

Příloha A – Zjednodušený UML diagram



Příloha B – Zdrojový kód naplnění šablony testu

```
String dotaz;
Iterator<SOtazka> itOtazka = dotazy.iterator();
Iterator<SMoznost> itMoznost = moznosti.iterator();
SMoznost moznost;
String obsah, vybrano;
for (Component pan : jPanelOtazky.getComponents()) {
    if (pan instanceof JPanel) {
        for (Component lbl : ((JPanel) pan).getComponents()) {
            if (lbl instanceof JLabel && ((JLabel) lbl).getText().startsWith("Dotaz") && itOtazka.hasNext()) {
                dotaz = itOtazka.next().getDotaz();
                ((JLabel) lbl).setIcon(new ImageIcon(getClass().getResource(dotaz)));
                ((JLabel) lbl).setText("");
            }
            if (lbl instanceof JRadioButton && itMoznost.hasNext()) {
                moznost = itMoznost.next();
                obsah = moznost.getObsah();
                ((JRadioButton) lbl).setIcon(new ImageIcon(getClass().getResource(obsah)));
                vybrano = moznost.getVybrano();
                ((JRadioButton) lbl).setSelectedIcon(new ImageIcon(getClass().getResource(vybrano)));
            }
        }
    }
}
```

Příloha C – Kompletní adresářová struktura

/	kořenový adresář
... aplikace/	balíček obsahující zdrojové kódy aplikace
... .. Kurz/	obsahuje GUI výukového kurzu a pomocné šablony
... <i>JPanelStatistika.java</i>	GUI šablona statistiky
... <i>JPanelTeorie.java</i>	GUI šablona teorie
... <i>JPanelTest.java</i>	GUI šablona testu s 5 otázkami
... <i>JPanelTest12.java</i>	GUI šablona testu s 12 otázkami
... <i>Kurz.java</i>	GUI okno výukového kurzu
... .. Main/	zde je spouštěcí třída programu, instance streamu a GUI oken
... <i>Main.java</i>	spouštěcí a ovládací třída výukového kurzu
... .. Uvod/	obsahuje GUI úvodního formuláře
... <i>JPanelPrihlaseni.java</i>	GUI panel pro přihlášení do kurzu
... <i>JPanelRegistrace.java</i>	GUI panel pro registraci do kurzu
... <i>Uvod.java</i>	GUI okno pro přihlášení a registraci uživatelů
... entity/	zde jsou uchovány jednotlivé entity dynamické části kurzu
... .. <i>Kapitola.java</i>	entita kapitola
... .. <i>Moznost.java</i>	entita možnost
... .. <i>Otazka.java</i>	entita otázka
... .. <i>Uzivatel.java</i>	entita uživatel
... obrazky/	obrázky vkládané do obsahu kurzu
... .. teorie/	obrázky studijních materiálů
... .. testy/	zadání a možnosti testů
... rozhrani/	zde jsou pomocná rozhraní k entitám
... .. <i>IKapitola.java</i>	rozhraní entity kapitola
... .. <i>IMoznost.java</i>	rozhraní entity možnost
... .. <i>IOtazka.java</i>	rozhraní entity otázka
... .. <i>IUzivatel.java</i>	rozhraní entity uživatel
... stream/	obsahuje třídu <code>Stream</code>
... .. <i>Stream.java</i>	třída pro komunikaci mezi programem a souborem
... streamEntity/	obsahuje potomky tříd ze složky entity určené ke komunikaci se soubory
... .. <i>SKapitola.java</i>	streamovací entita kapitola
... .. <i>SMoznost.java</i>	streamovací entita možnost
... .. <i>SOtazka.java</i>	streamovací entita otázka
... .. <i>SUzivatel.java</i>	streamovací entita uživatel