

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

Počítačová podpora výuky geometrie,
konstrukce trojúhelníků



2010

Peter Garláthy

Anotace

Cílem práce je vytvoření didaktické aplikace pro demonstraci a procvičování algoritmů konstrukcí trojúhelníků. Měla by být použitelná k výukovým účelům na základních školách v České republice a jejím hlavním přínosem možnost výuky geometrie netradiční formou.

Děkuji vedoucímu práce, Mgr. Tomáši Kührovi za jeho vedení, odborné rady a drahocenný čas strávený na konzultacích. Rovněž děkuji své rodině za podporu při studiu a kvalitní zázemí.

Obsah

1. Úvod	8
1.1. Zaměření aplikace	8
1.2. Návrh řešení	8
1.3. Použité technologie	8
2. Uživatelská dokumentace	9
2.1. Systémové požadavky	9
2.2. Ovládání	9
2.3. Výuková část	10
2.3.1. Krokování konstrukce	10
2.3.2. Pohyb obrázku	10
2.3.3. Změna měřítka	10
2.4. Procvičovací část	10
2.4.1. Zásady pro rýsování objektů	11
2.4.2. Rýsování úsečky	11
2.4.3. Rýsování kružnice	13
2.4.4. Rýsování bodu	14
2.4.5. Rýsování přímky	14
2.4.6. Pohyb obrázku	15
2.4.7. Změna měřítka	15
2.4.8. Mazání narýsovaných objektů	15
2.4.9. Kontrola narýsovaného trojúhelníku	15
2.4.10. Náповěda	16
2.5. Seznam klávesový zkratky	16
3. Programátorská část	17
3.1. Teoretická část	17
3.1.1. Změna měřítka	17
3.1.2. Translace	17
3.1.3. Analytický popis objektů	17
3.2. Vývojové prostředí	18
3.3. Grafické uživatelské rozhraní	19
3.4. Návrh tříd	19
3.4.1. Rozdělení tříd	19
3.4.2. Grafické třídy	19
3.4.3. Datové třídy	20
3.4.4. Formulářové třídy	20
3.4.5. Nezařazené třídy	21
Závěr	22

Reference	23
4. Obsah přiloženého CD	24

Seznam obrázků

1.	Dialogové okno pro vložení hodnot zvoleného algoritmu.	9
2.	Kreslení úsečky.	13
3.	Strom dědičnosti grafických objektů.	20

Seznam tabulek

1.	Možnosti nastavení úsečky.	12
2.	Možnosti nastavení kružnice.	14
3.	Možnosti nastavení bodu.	14
4.	Možnosti nastavení přímky.	15
5.	Přehled soustav použitých při hledání průsečíku.	18

1. Úvod

V době, kdy informační technologie pronikají do všech odvětví lidského zájmu, není ani školní prostředí výjimkou.

Znalost konstrukce trojúhelníků nalezneme v osnovách již prvního stupně základní školy. Na druhém stupni se už žáci učí konstrukce složitější, které je nutí více přemýšlet a také chápat podstatu celé věci. Pochopení této látky již může některým studentům činit potíže a je potřeba jim k pochopení látky pomoci i jinou cestou.

Moje práce představuje didaktickou aplikaci schopnou podat probíranou látku srozumitelně a netradičním způsobem, který zaručí, že i méně zdatní studenti dosáhnou uspokojivého výsledku.

1.1. Zaměření aplikace

Program je primárně určen pro učitele základních škol a jejich studenty, ale dobře poslouží i lidem, kteří si chtějí osvojit základní algoritmy, kterými lze trojúhelník zkonstruovat. Postup je možné procházet po jednotlivých krocích obsahujících přesné popisky a názorné obrázky. Uživatel si může přímo při studiu jednotlivé kroky postupu zkoušet v jednoduchém rýsovacím prostředí.

1.2. Návrh řešení

Názornost a jednoduchost aplikace byly hlavními kritérii, kterými se její návrh a následná implementace ubírala. Hledal jsem na internetu podobné programy, ale úspěšný jsem nebyl a nenašel jsem téměř nic. Návrh jsem tedy udělal čistě podle vlastního uvážení a aplikaci rozdělil do dvou částí - *Výukové* a *Procvičovací*. Části jsou na sobě nezávislé, a tak může uživatel v reálném čase studovat daný algoritmus a jeho postup rovnou rýsovat, stačí překliknout do *Procvičovací* sekce.

Při návrhu *Procvičovací* části jsem přemýšlel, jak rýsování udělat co nejkomfortnější. Stylem rýsování jsem se nechal inspirovat v programu *AutoCad*, který používají zejména architekti. Převzal jsem z něj postup používaný při rýsování nových objektů - a to, že dokud objekt nepotvrdím pravým tlačítkem, mohu jeho vlastnosti měnit jak myší, tak přímým zadáním hodnot.

1.3. Použité technologie

Aplikace byla vyvíjena v programovacím jazyku *C#* pro framework *.NET* od společnosti Microsoft. Jako vývojové prostředí jsem zvolil *Visual Studio 2008* od stejné vývojářské firmy.

2. Uživatelská dokumentace

Tato kapitola podrobně popisuje práci s aplikací a její jednotlivé části.

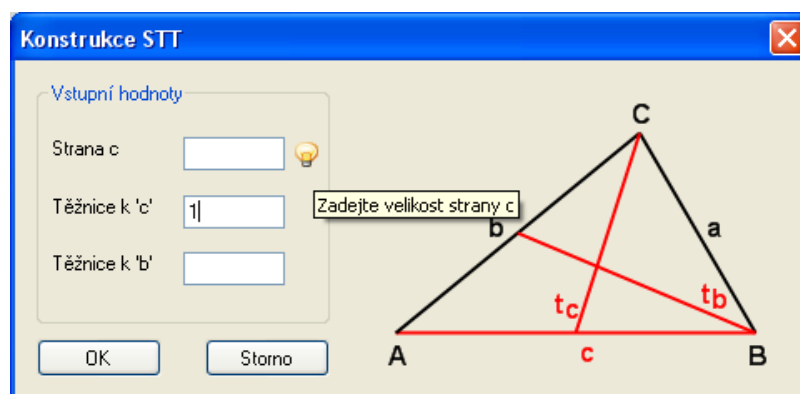
2.1. Systémové požadavky

Aplikace vyžaduje operační systém Microsoft Windows s nainstalovaným prohlížečem webových stránek. Podporovány jsou verze Windows XP, Windows Vista a Windows 7. K běhu programu je také potřeba Microsoft .NET Framework ve verzi 3.5 Service Pack 1 a vyšší. Nároky na procesor, paměť a místo na disku jsou zanedbatelné.

2.2. Ovládání

Program se ovládá myší a klávesovými zkratkami. Tvoří jej dvě stěžejní části: *Výuka* a *Procvičování*. Každá z těchto částí vyžaduje pro svou funkčnost zvolení určitého algoritmu a následné předání validních dat. Bez zvolení algoritmu a předání vstupních dat nebude uživatel moci se zvoleným modulem pracovat.

Algoritmus se zvolí zaškrtnutím požadované volby a následným kliknutím na tlačítko „Zadej hodnoty“. Objeví se okno, v němž je přehledně popsáno, jaké hodnoty se po uživateli chtějí. Pro větší názornost jsou zvýrazněny červenou barvou v obrázku ilustrativního trojúhelníku. Zadané hodnoty se kontrolují, pokud nejsou korektní, objeví se vedle pole ikonka se žárovkou. Má informativní charakter a pokud se na ni najede myší, objeví se zpráva s chybou, která nastala. Zadaná data potvrdíme tlačítkem „Ok“. Aplikace předané hodnoty opět překontroluje a je-li například podíl dvou stran ve velkém poměru, opět se objeví informativní žárovka a je potřeba některou z hodnot změnit.



Obrázek 1. Dialogové okno pro vložení hodnot zvoleného algoritmu.

2.3. Výuková část

Jakmile uživatel aplikaci předá korektní data, zobrazí se informativní náhled s trojúhelníkem a zadanými hodnotami. Na kreslicím plátně se zobrazí obrázek korespondující s prvním bodem postupu. V tabulce napravo od plátna jsou kroky stručně popsány dle matematických standardů pro rýsování a pod tlačítky určených ke krokování konstrukce je textové pole s podrobným popisem kroku.

2.3.1. Krokování konstrukce

Změna zobrazení aktuálního kroku algoritmu se provede pomocí tlačítek a klávesových zkratk. O krok dopředu se posune kliknutím na kulaté tlačítko se dvěma modrými trojúhelníky jejichž vrchol směřuje doprava a také klávesou zkratkou *Ctrl-U*. O krok dozadu tlačítkem s trojúhelníky tvořícími šipku opačného směru a klávesovou zkratkou *Ctrl-D*. Není-li možné provést žádané posunutí, jsou tlačítka šedé barvy a nelze na ně kliknout jak v toolbaru, tak v položce menu. Jakmile je krokování algoritmu v posledním kroku, objeví se v sekci *Diskuze o počtu řešení* přehledný rozbor všech situací, které mohou nastat. Tyto rozборы jsou z části dílem mým, ale z větší části dílem [7], od něhož jsem čerpal i většinu zadání konstrukčních úloh.

2.3.2. Pohyb obrázku

Obrázek, na kterém je zobrazen aktuální krok konstrukce, se posune pomocí šipek v sekci *Posunování obrázku*. Ten se posune vždy ve směru šipky, na kterou se kliklo. Pokud není možné obrázek posouvat, šipky jsou šedé barvy.

2.3.3. Změna měřítka

Změna měřítka obrázku neboli zoom se vyvolá v toolbaru v sekci obsahující ikonky s lupou. Zvětšení obrázku se provede kliknutím na lupu se symbolem $+$, zmenšení se symbolem $-$. Zvětšování obrázku není možné donekonečna, maximální přípustná hranice zvětšení je 2999% a zmenšení 25%. Zvětšení či zmenšení je rovněž možné zadat přesným požadavkem do pole vedle ikonky. Změna měřítka se promítne na všechny obrázky znázorňující jednotlivé kroky algoritmu.

2.4. Procvičovací část

Hlavním úkolem procvičovací části je vyzkoušet si nabyté vědomosti. Aby bylo možné rýsovat je stejně, jako ve výukové části nutné zvolit algoritmus a předat mu správná data.

2.4.1. Zásady pro rýsování objektů

Kreslicí editor poskytuje čtyři možné objekty - *úsečka, kružnice, bod a přímka*. Objekt určený k narýsování se zvolí kliknutím na jedno z tlačítek, která jsou vedle plátna. Objeví se okno se základními tipy, jak daný objekt kreslit. Aby se okno již nezobrazovalo, stačí zaškrtnout volbu *Příště nezobrazovat*. Zobrazování informativních oken lze také nastavit v hlavním menu aplikace. Každý objekt je určen svými nastavitelnými vlastnostmi, některé jsou pro všechny společné.

Souřadnice bodů jsou uváděny v reálných souřadnicích vzhledem ke kreslicímu plátnu. Délka úsečky nebo poloměr kružnice jsou vždy udávány v relativních jednotkách a jejich hodnoty odpovídají 100% zobrazení narýsovaného obrázku. Kreslené objekty mají implicitně nastaveno, aby se body, které je tvoří, přichytávaly k již nakresleným bodům¹ a také průsečíkům mezi objekty. Toto chování lze zapnout/vypnout v menu pod položkou *Přichytávat body*.

Objekt se narýsuje kliknutím pravým tlačítkem myši. Než se tak stane, má červenou barvu a je možno měnit jeho atributy. Pokud je potvrzen k narýsování nekorektní objekt (např. kružnice s nulovým poloměrem), objeví se varování o chybě.

2.4.2. Rýsování úsečky

Prvním kliknutím na plátno se získají souřadnice prvního bodu tvořícího úsečku a do té doby, než se druhým kliknutím potvrdí pozice druhého bodu, bude mít druhý bod pozici aktuální pozice kurzoru myši. Souřadnice bodů lze poté měnit v tabulce nebo stylem „drag and drop“; tedy nakliknutím bodu určeného ke změně, následným tažením a potvrzením pozice puštěním tlačítka myši.

Nekreslí-li se úsečka dvěma body, je nutné ji zadat jedním bodem a její délkou. Volitelně je možné definovat úhel, pod jakým bude úsečka vzhledem k **vodorovné** ose plátna. Souřadnice bodů takto zadané úsečky je rovněž možné měnit stylem „drag and drop“.

¹Narýsovaným bodem se rozumí *tlustá tečka*.

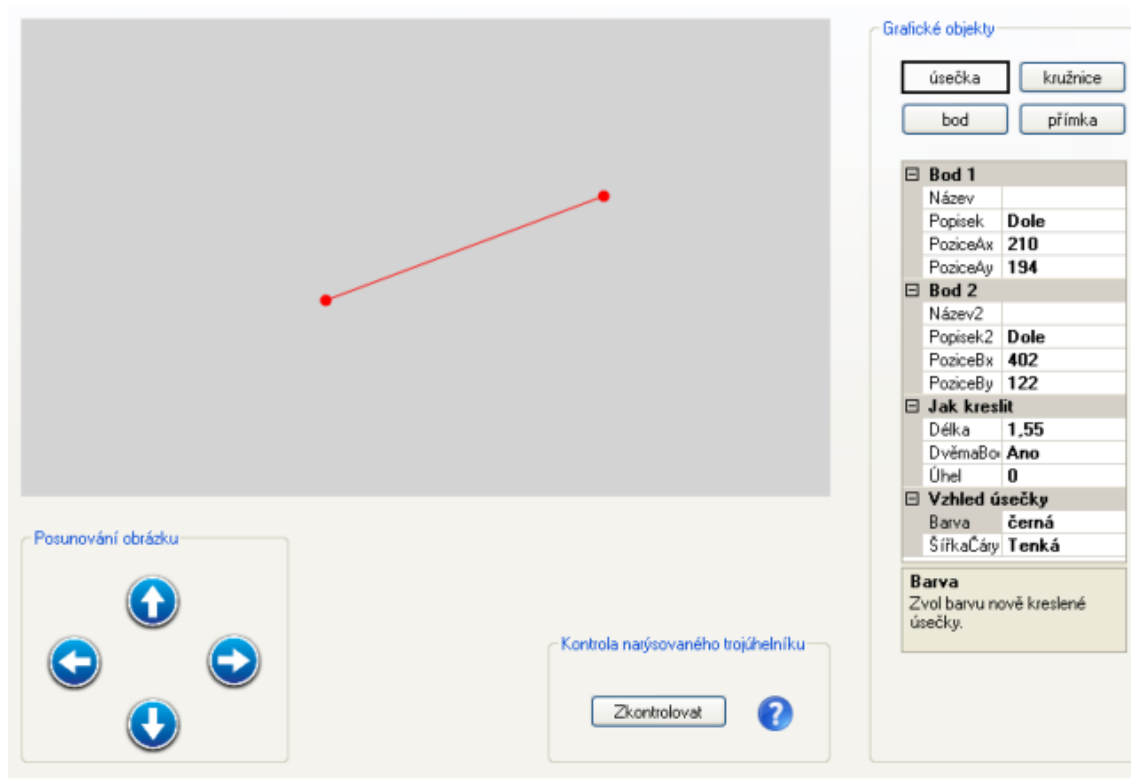
Atributy úsečky			
Bod 1	Bod 2	Jak kreslit	Vzhled úsečky
Název	Název2	Délka	Barva
Popisek	Popisek2	DvěmaBody	ŠířkaČáry
PoziceAx	PoziceBx	Úhel	
PoziceAy	PoziceBy		

Tabulka 1. Možnosti nastavení úsečky.

Význam některých proměnných.²

- **Popisek** – nastaví, kde bude nakreslen název bodu. Nabývá hodnot **Dole**, **Nahoře**, **Vlevo**, **Vpravo**.
- **DvěmaBody** – určuje, jestli kreslíme úsečku dvěma body. Nabývá hodnot **Ano**, **Ne**. V případě, že se kreslí dvěma body, je údaj **Délka** pouze informativního charakteru, protože délka úsečky je tvořena vzáleností mezi těmito body.
- **Barva** – nastaví barvu výsledné úsečky, nabývá hodnot **černá**, **modrá**, **zelená**.
- **ŠířkaČárky** – určí šířku pera, jakým bude potvrzená úsečka narýsovaná. Obsahuje hodnoty **Tenká**, **Tlustá**.

²V dalším textu již nebudou znovu popisovány.



Obrázek 2. Kreslení úsečky.

2.4.3. Rýsování kružnice

Prvním kliknutím na plátno se získají souřadnice středu kružnice a do té doby, než se druhým kliknutím potvrdí velikost poloměru, bude jeho velikost rovna vzdálenosti kurzoru myši od středu. Poté je možné střed i poloměr měnit v tabulce nebo opět kliknutím. Střed stylem „drag and drop“, poloměr kliknutím kdekoli na plátno - jeho velikost bude určena vzdáleností středu od kliknutého místa.

V případě, že je rýsována kružnice zadaná dvěma body, prvním kliknutím se umístí střed, druhým bod, kterým má kružnice procházet. Následně lze střed měnit opět stylem „drag and drop“, druhý bod jednoduše dalším kliknutím na plátno.

Atributy kružnice			
Jak kreslit	Název	Střed	Vzhled kružnice
Poloměr	Název	PoziceSx	Barva
PoziceBx	Popisek	PoziceSx	ŠířkaČáry
PoziceBy			
StředPoloměr			

Tabulka 2. Možnosti nastavení kružnice.

Význam dosud neznámých proměnných

- **StředPoloměr** – určuje, jestli kreslíme kružnici zadanou středem a poloměrem. Nabývá hodnot `Ano`, `Ne`. Nekreslí-li se středem a poloměrem, kružnice prochází určeným bodem a `Poloměr` je pouze informativního charakteru, protože poloměr kružnice je tvořen vzdáleností mezi středem a daným bodem.

2.4.4. Rýsování bodu

Kliknutím na plátno se získají souřadnice bodu. Ty lze poté měnit v tabulce nebo kliknutím na jiné místo. Funguje také přesun bodu stylem „drag and drop“.

Atributy bodu					
Bod	Barva	Název	Popisek	PoziceAx	PoziceAy

Tabulka 3. Možnosti nastavení bodu.

2.4.5. Rýsování přímky

Pokud se rýsuje přímka určená dvěma body, postup konstrukce je analogický s rýsováním úsečky určené dvěma body (viz podkapitola 2.4.2.). V opačném případě se musí nejprve označit již nakreslená přímka nebo úsečka a na ní bod, z kterého bude nově konstruovaná přímka vycházet. U bodu je důležité, aby byl na daném objektu tvořen „tečkou“. Poté je nutné nastavit velikost úhlu mezi přímkou a vybraným objektem. Zrušení označení objektu a bodu se provede klávesou `Esc`.

Atributy přímky			
Bod 1	Bod 2	Jak kreslit	Vzhled přímky
Název	Název2	DvěmaBody	Barva
Popisek	Popisek2	Úhel	ŠířkaČáry
PoziceAx	PoziceBx		
PoziceAy	PoziceBy		

Tabulka 4. Možnosti nastavení přímky.

2.4.6. Pohyb obrázku

Narýsovaný obrázek je možné posouvat stejným způsobem, jako tomu bylo ve výukové části programu, viz. 2.3.2.. Posune se i náhled dosud nepotvrzeného objektu (je-li zrovna nějaký rýsován).

2.4.7. Změna měřítka

Změna měřítka narýsovaného obrázku je analogická té ve výukové části, viz. 2.3.3. Měřítka změní i červený náhled dosud nepotvrzeného objektu.

2.4.8. Mazání narýsovaných objektů

Objekty lze z obrázku mazat nebo je vracet v toolbaru v sekci s tmavě modrými šipkami. Tlačítko s šipkou doleva nebo klávesová zkratka *Ctrl-Z* smaže poslední narýsovaný objekt. Lze smazat i více objektů za sebou. Tlačítko s šipkou doprava nebo zkratka *Ctrl-Y* nakreslí poslední smazaný objekt. Smazat kompletně celý obrázek umožňuje tlačítko s bílým listem papíru a klávesová zkratka *Ctrl-N*. Protože je tato volba značně destruktivní, objeví se okno, v němž je možné vyvolanou akci zrušit.

2.4.9. Kontrola narýsovaného trojúhelníku

Ke kontrole narýsovaného trojúhelníku slouží tlačítko *Zkontrolovat*. Aby bylo možné trojúhelník v obrázku jednoznačně určit, musí v něm být pojmenované body A, B, C. Nezáleží, jestli jsou pojmenované velkým nebo malým písmenem, rozhodující je, aby byl v obrázku z každého názvu pouze jeden. Tyto body potom představují trojúhelník určený ke kontrole. Kontroluje se, jestli narýsovaný trojúhelník odpovídá tomu, který by vznikl při přesném dodržení postupu dle algoritmu a všech délek stran, těžnic, výšek a velikostí úhlů.

2.4.10. Nápověda

Spuštění nápovědy se vyvolá klávesou *F1*. Zobrazí se v novém okně nebo nové kartě uživatelem používaného prohlížeče.

2.5. Seznam klávesový zkratek

- F1 – zobrazení nápovědy
- Ctrl-U – posun o krok vpřed při demonstraci algoritmu ve výukové části
- Ctrl-D – posun o krok zpět při demonstraci algoritmu ve výukové části
- Ctrl-Z – smazání posledního narýsovaného objektu v procvičovací části
- Ctrl-Y – narýsování posledního smazaného objektu v procvičovací části
- Ctrl-N – smazání celého narýsovaného obrázku

3. Programátorská část

Kapitola popisuje použité technologie, návrh tříd a některé stěžejní metody.

3.1. Teoretická část

Aplikace při operaci s obrázky hojně využívá geometrických transformací. Podrobný popis je v [9], níže uvádím transformace, které jsou v programu aplikovány.

3.1.1. Změna měřítka

Změna měřítka (scale, zoom) je změnou velikosti objektu ve směru souřadnicových os. Pokud je absolutní hodnota koeficientu v intervalu $(0, 1)$, dochází ke zmenšení transformovaného objektu. Je-li absolutní hodnota koeficientu větší než jedna, dojde k prodloužení, je-li znaménko záporné, dochází k prodloužení či zmenšení v opačném směru. Rovnice pro změnu měřítka bodu $P = [X, Y]$ mají tvar $X' = S_x X$, $Y' = S_y Y$, kde S_x je koeficient změny měřítka ve směru souřadnicové osy x , S_y je koeficient změny měřítka ve směru souřadnicové osy y , X, Y jsou původní souřadnice bodu P a X', Y' jsou nové souřadnice bodu P .

3.1.2. Translace

Transformace posunutí neboli *translace* bodu $P = [X, Y]$ je určena vektorem posunutí $p = (X_T, Y_T) = (X' - X, Y' - Y)$. Aplikací této transformace na bod P se získá bod P' o souřadnicích $X' = X + X_T$, $Y' = Y + Y_T$, kde X, Y jsou původní souřadnice bodu P a X', Y' jsou nové souřadnice bodu P' .

V aplikaci jsou obě metody zkombinovány. Jakmile uživatel nazoomuje obrázek, všechny jeho body se posunou o translační vektor směrem k počátku souřadnicových os, na ně se aplikuje *scale* transformace a výsledný obrázek se translací o opačný translační vektor posune zpět. *Translační vektor* je určen počátkem souřadnicových os (v levém horním rohu kreslicího plátna) a bodem přesně uprostřed plátna vypočítaným při defaultní velikosti okna aplikace.

3.1.3. Analytický popis objektů

Grafické objekty typu *přímka*, *úsečka*, *kružnice* uchovávají ve svých slotech i analytické vyjádření.

Vektor

Vektorem se nazývá množina všech souhlasně orientovaných úseček stejné délky. Značí se \vec{u} , a skládá se ze dvou složek u_1, u_2 . V rovině je určen body $A[A_x, A_y]$ a $B[B_x, B_y]$ a pro složky vektoru \vec{u} platí $u_1 = (B_x - A_x)$, $u_2 = (B_y - A_y)$. Velikost vektoru \vec{u} se značí $|u|$ a vypočítá se pomocí vztahu $\sqrt{u_1^2 + u_2^2}$. *Skalárním*

součinem vektoru se rozumí operace, kdy je dvěma vektorům přiřazeno číslo. Je definován pro vektory $\vec{u} = (u_1, u_2)$, $\vec{v} = (v_1, v_2)$ jako $\vec{u}\vec{v} = (u_1v_1 + u_2v_2)$.

Parametrické vyjádření přímky v rovině

Pro určení parametrického vyjádření přímky v rovině je potřeba vektor $\vec{u} = (u_1, u_2)$, který na ní leží, a bod $A[A_x, A_y]$. Parametrickou rovnicí přímky se rozumí vztah $X = A + t\vec{u}$, kde X je libovolný bod přímky a $t \in R$ je parametr.

Řekneme, že rovnice je rovnicí $\begin{cases} \text{úsečky} & \text{jestliže } t \in \langle 0, 1 \rangle \\ \text{polopřímky} & \text{jestliže } t \in \langle 0, \infty \rangle \end{cases}$

Obecné vyjádření přímky v rovině

Obecnou rovnicí přímky p v rovině dostaneme z jejích parametrických rovnic tak, že se zbavíme parametru t . Obecná rovnice přímky p je poté dána vztahem $ax + by + c = 0$, kde x, y jsou souřadnice libovolného bodu přímky p , čísla a, b, c jsou reálná čísla, pro něž musí platit $[a, b] \neq [0, 0]$.

Obecné vyjádření kružnice v rovině

Obecná rovnice kružnice je definována vztahem $x^2 + y^2 - 2mx - 2ny + p = 0$, kde $p = m^2 + n^2 - r^2$. Platí přitom $m^2 + n^2 - p > 0$, v opačném případě nejde o kružnici.

Uvedené rovnice se hojně využívají k výpočtu průsečíků mezi objekty. Nalézt průsečík v analytické geometrii znamená vyřešit soustavu dvou rovnic reprezentujících dané objekty.

Výpočet průsečíku			
	Kružnice	Přímka	Úsečka
Kružnice	Kva x Kva	Kva x Lin	Kva x Lin
Přímka	Lin x Kva	Lin x Lin	Lin x Lin
Úsečka	Lin x Kva	Lin x Lin	Lin x Lin

Tabulka 5. Přehled soustav použitých při hledání průsečíku.

pozn. *Kva* reprezentuje kvadratickou rovnici, *Lin* rovnicí lineární.

3.2. Vývojové prostředí

Program byl vyvíjen v prostředí Microsoft Visual Studio 2008 pro platformu Microsoft .NET Framework 3.5 SP1 v jazyce Visual C#. GUI bylo zpracováno za pomoci knihovny tříd Windows Forms. Aplikace byla testována na počítačích s operačními systémy Windows XP, Vista i 7.

3.3. Grafické uživatelské rozhraní

Grafické uživatelské rozhraní (GUI) bylo zpracováno v design editoru prostředí Microsoft Visual Studio 2008 za využití knihovny tříd Windows Forms, což se vzhledem k programovacímu jazyku Visual C# a charakteru aplikace jeví jako nejvýhodnější kombinace. Při návrhu rozhraní programu byl kladen důraz zejména na celkovou přehlednost, dodržení standardů pro formulářové aplikace technologie Windows Forms a snadné ovládání.

Uživatelé jsou v daném momentě povoleny jen ty akce, které je oprávněn provést. Například, není-li v procvičovací části nakreslen obrázek, není možné s ním pohybovat, nebo jej zoomovat.

GUI obsahuje velké množství grafických prací které jsou pod licencí [10] a byly staženy z [11]. Tento web je databází free ikoněk a obrázků. Studia, která ikonky v aplikaci použité vytvořila jsou [12], [13] a [14]. Hlavní motiv aplikace a nápovědy vytvořil [15].

3.4. Návrh tříd

Složitost celé aplikace si vyžadovala zodpovědný přístup k návrhu tříd a jejich vzájemné interakci. Třídy jsou tematicky rozděleny do několika sekcí.

3.4.1. Rozdělení tříd

- DataClass - třídy uchovávající data
- GraphicsClass - třídy reprezentující jednotlivé rýsované objekty
- PropertyGridClass - třídy definující vlastnosti rýsovaných objektů pro komponentu *PropertyGrid*³
- Forms – třídy s (grafickými) formuláři
- Nezařazené – statické a pomocné třídy

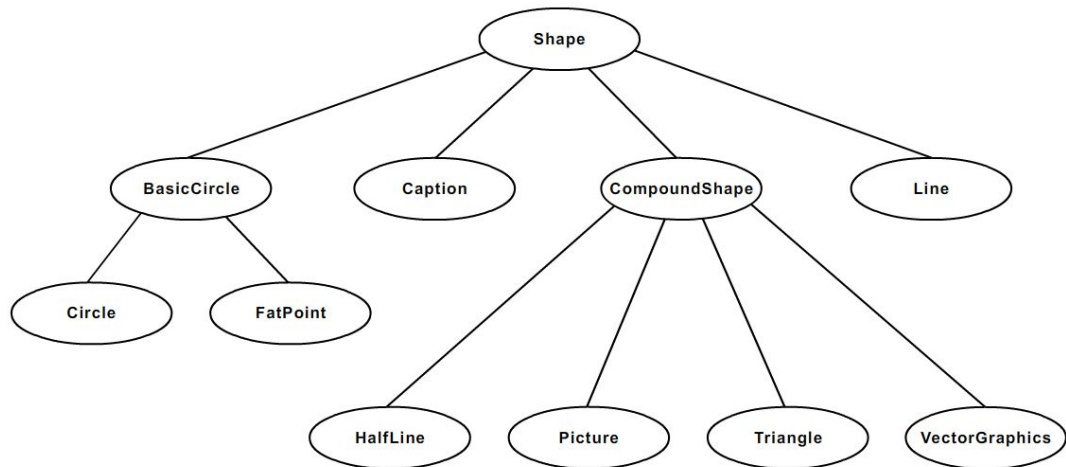
3.4.2. Grafické třídy

Grafické třídy jsou navrženy dle [8] a jde o učebnicový příklad použití dědičnosti. Stěžejní je abstraktní třída *Shape*, kterou dědí její potomci.

Metody, které má každý grafický objekt implementovány

- void Move(float x, float y) – posune objekt po x-ové a y-ové ose
- void Paint(PaintEventArgs e) – nakreslí objekt
- void Zoom(double percent) – nazoomuje objekt

³Komponenta reprezentuje tabulku, v níž se nastavují vlastnosti rýsovaných objektů.



Obrázek 3. Strom dědičnosti grafických objektů.

3.4.3. Datové třídy

U navrhování datových tříd je využito návrhového vzoru *Simple Factory Method* podrobně popsaného v [5]. Stručně řečeno, tento návrhový vzor slouží jako náhrada volání konstruktoru. Jeho výhodou je konstrukce jediného objektu (často potomka abstraktní třídy), s kterou se dá v rámci aplikace pracovat. Toho je využito jak ve třídách skupiny `DataClass`, tak skupiny `PropertyGridClass`.

Ve skupině tříd `DataClass` je definována třída `AlgorithmData`. Třída nemá konstruktor a je kořenem stromu dědičnosti datových tříd. Instance třídy se získá voláním statické metody `public static AlgorithmData getAlgorithmData(ConstructionType type, ...)`, která na základě proměnné `type` určující typ konstrukce vrátí potomka třídy `AlgorithmData` obsahující konkrétní data. Kdykoli za běhu programu vznikne požadavek na obměnu dat v instanci třídy `AlgorithmData`, opět se zavolá výše popsaná metoda a instance bude obsahovat aktuální data, s kterými aplikace může pracovat. Tímto se elegantně vyhnulo spravování velkého množství objektů reprezentujících jednotlivé datové třídy (pro informaci jich je celkem 8), kterých by bylo zapotřebí.

3.4.4. Formulářové třídy

Hlavní třídou nejen této skupiny, ale celé aplikace je třída `MainWindow`. Reprezentuje objekt hlavního okna aplikace a stará se o všechno dění. Vytváří objekty, obsluhuje události vyvolané uživatelskými požadavky a mnoho jiného.

3.4.5. Nezařazené třídy

Mezi nejvýznamnější nezařazené třídy patří statické třídy `MyMath` a `MyPaint`. Dalšími významnými třídami jsou ty, které analyticky popisují grafický objekt.

Třída `MyMath` poskytuje metody pro výpočty průsečíků mezi objekty a veškeré další matematické výpočty.

Třída `MyPaint` poskytuje metody starající se o kreslení informativních obrázků a různé konverze mezi pery a výtčovými typy.

Do skupiny tříd analyticky popisujících objekt by se daly zařadit třídy

- `LineGeneralEquation` – obecná rovnice přímky
- `LineParametricEquation` – parametrická rovnice přímky
- `CircleGeneralEquation` – obecná rovnice kružnice

Závěr

Cílem práce bylo vytvoření didaktické aplikace pro demonstraci a procvičování algoritmů konstrukcí trojúhelníků, která by mohla být využita při výuce geometrie na základních školách v České republice. Vzniklá aplikace svou funkčností vyhovuje vytyčeným cílům a tudíž nic nebrání jejímu využití v českém školství. Její nevýhodou je malé množství nabídnutých algoritmů k procvičení, mohla by se v budoucnu rozšířit o editor přidání nových. Aplikace by poté nemusela sloužit pouze pro demonstraci výuky trojúhelníků, ale i jiných geometrických objektů, jako jsou například mnohoúhelníky. Rýsovací editor je vytvořen precizně a je vhodný nejen pro rýsování trojúhelníků.

Reference

- [1] Sells, Chriss. *C# a WinForms - programování formulářů Windows* Zoner Press, Brno, 2005.
- [2] Sharp, John. *Microsoft Visual C# 2005 Krok za krokem* Computer Press, Brno, 2006.
- [3] Petzold, Charles. *Programování Microsoft Windows Forms v jazyce C#* Computer Press, Brno, 2006.
- [4] Pomykalová, Eva. *Matematika pro gymnázia - Planimetrie* Prometheus, Praha, 2008.
- [5] Pecinovský, Rudolf. *Návrhové vzory* Computer Press, Brno, 2007.
- [6] Červinková, Petra; Čermák, Pavel *Odmaturuj z matematiky* Didaktis, Brno, 2003.
- [7] Kurka, Štěpán. *Konstrukční úlohy* [online].[cit. 2010-3-4]
Univerzita Karlova, Praha, 2008
- [8] Krupka, Michal. *Paradigmata objektového programování I.* Univerzita Palackého, Olomouc, 2008.
- [9] *Wikipedia - the free encyclopedia* [online].[cit. 2010-6-15]
URL: <<http://en.wikipedia.org/>>
- [10] *Attribution-NonCommercial-ShareAlike license* [online].[cit. 2010-7-26]
URL: <<http://creativecommons.org/licenses/by-nc-sa/3.0/>>
- [11] *Icon Archive* [online].[cit. 2010-3-24]
URL: <<http://www.iconarchive.com/>>
- [12] *Photoshop backgrounds at PSD Graphics* [online].[cit. 2010-4-20]
URL: <<http://www.psdgraphics.com/>>
- [13] *GoSquared - beautiful, powerful and intuitive web apps for real-time traffic monitoring* [online].[cit. 2010-4-22]
URL: <<http://www.gosquared.com/>>
- [14] *VisualPharm - Windows 7 Icons and User Interface design* [online].[cit. 2010-5-12]
URL <<http://www.visualpharm.com/>>
- [15] *ZaKaR on deviantART* [online].[cit. 2010-7-26]
URL <<http://zakar.deviantart.com/>>

4. Obsah přiloženého CD

Součástí práce je CD obsahující zdrojové kódy aplikace i tohoto textu.

bin/

Ve složce jsou soubory nutné k bezproblémovému běhu aplikace VYUKATVORBYTROJUHELNIKU.

doc/

Obsahuje dokumentaci ve formátu .PDF a všechny zdrojové kódy včetně obrázků.

src/

Jsou zde kompletní zdrojové kódy (včetně obrázků a ikon) nutné pro znovusestavení programu VYUKATVORBYTROJUHELNIKU.

readme.txt

Instrukce pro instalaci programu VYUKATVORBYTROJUHELNIKU.