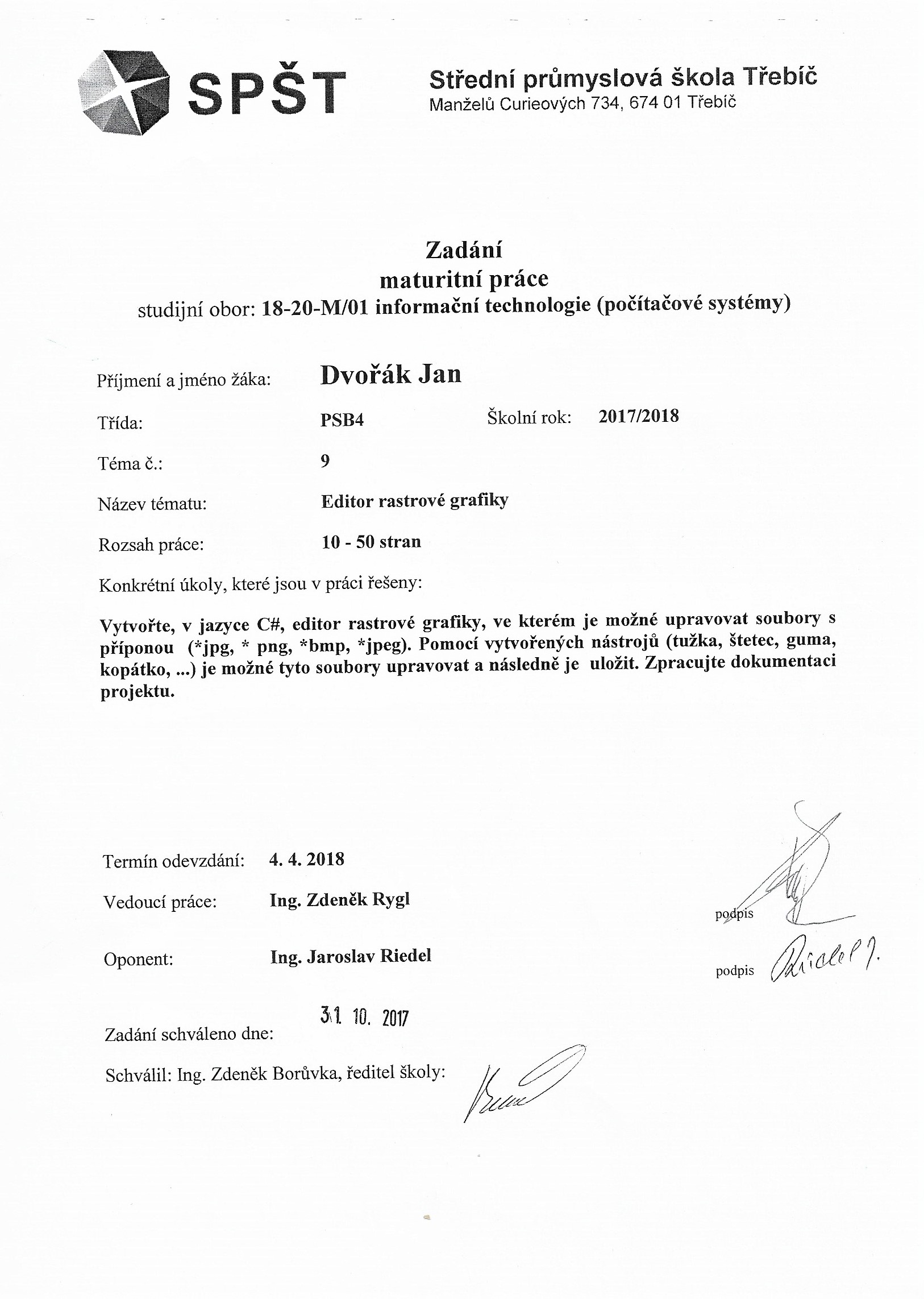
**Maturitní práce**

Editor rastrové grafiky

Studijní obor: počítačové systémy

Třída: PSB4

Školní rok: 2017/2018 Jméno: Jan Dvořák



# Poděkování

Chtěl bych poděkovat svému vedoucímu práce Ing. Zdeňku Ryglovi, za odborné vedení, za pomoc a rady při zpracování této práce.

## Prohlášení

Prohlašuji že, jsem tuto práci vypracoval samostatně a uvedl v ní všechny prameny, literaturu a ostatní zdroje, které jsem použil.

Podpis: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_­\_\_\_\_\_

Jan Dvořák

# Abstrakt

Tato práce se zabývá vytvořením editoru rastrové grafiky, který obsahuje základní nástroje pro editaci rastrové grafiky. Cílem této práce bylo vytvořit počítačový program, který dokáže pomocí základních nástrojů editovat rastrovou grafiku. Během tvorby této práce byl kladen důraz na funkčnost.

## Klíčová Slova

Editor rastrové grafiky, C#, Microsoft Visual Studio, Počítačová grafika

## Abstract

This project’s focus is making an editor for raster graphics. The goal of this project was to create a computer program that is capable of editing raster graphics by basic editor tools. In the making, a great emphasis has been given on the projects functionality.

## Key Words

Raster graphics editor, C#, Microsoft Visual Studio, Computer graphics

Obsah

Úvod …………………………………………………………………...10

Slovník pojmů 11

Teoretická část 12

1 Bitmapový grafický editor 13

2 Microsoft Visual Studio 14

2.1 Architektura Microsoft Visual Studia 15

3 Rastrová grafika 17

3.1 Výhody rastrové grafiky 17

3.2 Nevýhody rastrové grafiky 18

4 Existující formáty rastrové grafiky 19

4.1 Podporované formáty rastrové grafiky 19

5 C# 20

5.1 Výhody jazyka C# 20

Praktická část 22

6 Nástroje 23

6.1 Metoda OnMouseDown 23

6.2 Metoda OnMouseMove 23

6.3 Metoda OnMouseUp 24

6.4 Nástroj tužka 24

6.5 Nástroj štětec 24

6.6 Nástroj guma 24

6.7 Nástroj ukazovátko 24

6.8 Nástroj kyblík 24

6.9 Nástroj kapátko 25

6.10 Nástroj pohyb 25

7 Menu 26

7.1 Položka File 26

7.2 Položka Edit 26

7.3 Položka Image 26

8 Třída ActionManager 27

9 Formulář MainForm 28

9.1 Volba barvy 28

10 Formulář NewFileForm 29

11 Třída MouseListener 30

12 Třída KeyHandler 31

Závěr ……………………………………………………….…………..32

13 Seznam obrázků 33

14 Použité zdroje 34

14.1 Tištěné dokumenty 34

14.2 Elektronické dokumenty 34

# Úvod

Editor rastrové grafiky je počítačový program umožňující uživateli prostřednictvím grafického rozhraní vytvářet a upravovat soubory s rastrovou grafikou.

Jedním z cílů projektu je vytvořit několik nástrojů, pomocí kterých bude uživatel moci provádět základní úpravy.

Výsledkem projektu bude spustitelný program, ve kterém bude uživatel moci vytvořit nebo otevřít již existující rastrový soubor, provádět v něm změny pomocí vytvořených nástrojů a následně jej uložit do několika z podporovaných formátů rastrové grafiky.

Při tvorbě projektu budu klást velký důraz na funkčnost a na modulárnost, díky které bude možné nadále program rozšiřovat.

Slovník pojmů

IDE – Integrated Development Enviroment

GUI – Graphics User Interface

GC – Garbage Collector

HTML – Hyper Text Markup Language

GIMP - GNU Image Manipulation Program

GNU – GNU is Not UNIX (rekurzivní akronym)

Teoretická část

Tato část obsahuje popis použitých vývojových prostředí, nastínění topologie aplikace.

# Bitmapový grafický editor

Bitmapový grafický editor nebo také rastrový grafický editor je počítačový program umožňující uživateli prostřednictvím grafického rozhraní vytvářet a upravovat soubory s rastrovou grafikou. Data jsou zaznamenávána v některém z formátů vhodných pro bitmapovou grafiku jako např. JPEG, PNG, GIF a TIFF.

Mezi nejznámější bitmapové editory patří Adobe Photoshop, GIMP, nebo také Microsoft Paint

# Microsoft Visual Studio

Microsoft Visual Studio je vývojové prostředí (IDE) od Microsoftu. Může být použito pro vývoj konzolových aplikací a aplikací s grafickým rozhraním spolu s aplikacemi Windows Forms, webovými stránkami, webovými aplikacemi a webovými službami jak ve strojovém kódu, tak v řízeném kódu na platformách Microsoft Windows, Windows Mobile, .NET Compact Framework a Microsoft Silverlight.

Visual Studio obsahuje editor kódu podporující IntelliSense a refaktorování. Integrovaný debugger pracuje jak na úrovni kódu, tak na úrovni stroje. Další vestavěné nástroje zahrnují designer formulářů pro tvorbu aplikací s GUI, designer webu, tříd a databázových schémat. Je možné přidávat rozšíření, což vylepšuje funkčnost na téměř každé úrovni – od doplnění podpory pro verzovací systémy (jako Subversion a Microsoft Team Foundation Server) po nové nástroje jako editory a vizuální designery pro doménově specifické jazyky nebo nástroje pro další aspekty návrhu programu (jako klient Team Foundation Serveru Team Explorer).

Visual Studio podporuje jazyky prostřednictvím jazykových služeb, což umožňuje, aby editor kódu a debugger podporoval jakýkoliv programovací jazyk. Mezi vestavěné jazyky patří C/C++ (použitím Visual C++), VB.NET (použitím Visual Basic .NET) a C# (použitím Visual C#). Podpora dalších jazyků jako Oxygene, F#, Python a Ruby spolu s ostatními může být přidána jazykovými službami, které musí být nainstalovány zvlášť. Také je podporováno XML/XSLT, HTML/XHTML, JavaScript a CSS. Existují i verze Visual Studia pro určitý jazyk, které uživateli poskytují omezenější jazykové služby. Tyto individuální balíčky jsou Microsoft Visual Basic, Visual J#, Visual C# a Visual C++.

Aktuálně je na trhu Visual Studio 2013 spolu s omezenými balíčky Visual Studio 2013 Express Editions.

## Architektura Microsoft Visual Studia

Visual Studio nepodporuje žádný programovací jazyk nebo nástroj samo o sobě. Místo toho je mu možno přidat různá rozšíření funkčnosti. Každá funkčnost je zabalena do balíčku VSPackage. Když je nainstalována, je dostupná jako služba IDE a poskytuje tři služby: SVsSolution, která umožňuje očíslovat projekty a sestavy; SVsUIShell, který poskytuje rozdělování na okna a UI funkce (jako panely, nástrojové lišty a okna nástrojů); a SVsShell, který se stará o registraci balíčků VSPackage. IDE je také odpovědné za koordinaci služeb a umožnění komunikace mezi nimi. Všechny editory, designery, typy projektů a další nástroje jsou implementovány jako balíčky VSPackage. Visual Studio používá COM pro přístup k balíčkům. SDK také obsahuje Managed Package Framework (MPF), což je sada spravovaných obalů okolo rozhraní COM, které umožňují, aby mohly být balíčky psány v jazycích .NET. Nicméně, MPF neposkytuje veškerou funkčnost představenou COM rozhraními Visual Studia. Služby mohou být použity pro tvorbu dalších balíčků, což přidá funkčnost do IDE Visual Studia.

Podpora programovacích jazyků je přidána balíčkem zvaným Language Service (jazyková služba). Jazyková služba definuje různá rozhraní, která může implementace VSPackage implementovat pro přidání podpory různé funkčnosti. Funkčnost, která může být tímto způsobem přidána, zahrnuje zvýraznění syntaxe, doplňování příkazů, zvýrazňování párů závorek, tipy parametrů informací, seznamy členů a chybové značky pro kompilaci na pozadí. Pokud je rozhraní implementováno, funkčnost bude pro jazyk dostupná. Implementace jazykových služeb mohou použít kód z překladače nebo editoru jazyka. Jazykové služby mohou být implementovány jak ve strojovém kódu, tak v řízeném kódu. Pro strojový kód mohou být použity jak rozhraní COM, tak Babel Framework (součást SDK). Pro spravovaný kód obsahuje MPF obaly pro psaní spravovaných jazykových služeb.

Visual Studio neobsahuje žádnou vestavěnou podporu verzování, ale definuje MSSCCI (Microsoft Source Code Control Interface) implementováním toho, které verzovací systémy mohou být integrovány s IDE. MSSCCI definuje sadu funkcí, které jsou použity pro implementaci různé funkčnosti verzování. MSSCCI bylo poprvé použito pro integraci Visual SourceSafe s Visual Studiem 6.0, ale později bylo zpřístupněno přes Visual Studio SDK. Visual Studio .NET 2002 používalo MSSCCI 1.1 a Visual Studio .NET 2003 používalo MSSCCI 1.2. Visual Studio 2005 i 2008 používají MSSCCI verze 1.3, která přidává podporu přejmenovávání a mazání propagace, stejně jako asynchronní otevírání.

Visual Studio podporuje spuštění více instancí prostředí (každé s vlastní sadou balíčků). Instance používají jiné větve registrů (viz definici MSDN pojmu „registry hive“ použité v tomto smyslu) k uložení stavu konfigurace a jsou rozlišeny AppId (Application ID). Instance jsou spuštěny specifickými exe, které vyberou AppId, nastaví kořenovou větev a spustí IDE. Balíčky registrované pro jedno AppId jsou integrovány s ostatními balíčky pro toto AppId. Různé edice Visual Studia používají jiná AppId. Produkty edice Visual Studio Express jsou nainstalovány se svými vlastními AppId, ale produkty Standard, Professional a Team Suite sdílí stejné AppId. Proto mohou být edice Express nainstalovány souběžně s ostatními edicemi, na rozdíl od ostatních edic, které aktualizují stejnou instalaci. Profesionální edice obsahuje supersadu balíčků ve standardní edici a Team Suite obsahuje supersadu balíčků v obou ostatních edicích. Systém AppId je ovlivněn systémem Visual Studio Shell ve Visual Studiu 2008.

# Rastrová grafika

Bitmapová grafika (rastrová grafika) je jeden ze dvou základních způsobů, jakým počítače ukládají a zpracovávají obrazové informace (druhý způsob je vektorová grafika).

V bitmapové grafice je celý obrázek popsán pomocí jednotlivých barevných bodů (pixelů). Body jsou uspořádány do mřížky. Každý bod má určen svou přesnou polohu a barvu v nějakém barevném modelu (např. RGB). Tento způsob popisu obrázků používá např. televize nebo digitální fotoaparát. Kvalitu záznamu obrázku ovlivňuje především rozlišení a barevná hloubka.

Rozmístění a počet barevných bodů obvykle odpovídají zařízení, na kterém se obrázek zobrazuje (monitor, papír). Pokud se obrázek zobrazuje na monitoru, stačí rozlišení 72 DPI, pro tisk na tiskárně 300 DPI.

Pro převod obrazových předloh (klasické fotografie, kreseb a dalších) do bitmapové grafiky slouží zařízení nazývané skener nebo digitální fotoaparát.

Další možností pro ukládání a zpracování obrazů je vektorová grafika, kde se obrázek popisuje pomocí geometrických objektů - křivek a mnohoúhelníků.

## Výhody rastrové grafiky

* jediná rozumná volba pro digitální fotografii (a některé další typy grafiky)
* pořízení obrázku je velmi snadné například pomocí fotoaparátu nebo pomocí skeneru.

## Nevýhody rastrové grafiky

* velké nároky na paměťové zdroje (při vysokém rozlišení a barevné hloubce velikost obrázku dosahuje i jednotek megabytů, v profesionální grafice se běžně operuje i s podklady o desítkách megabytů)
* změna velikosti (zvětšování nebo zmenšování) vede ke zhoršení obrazové kvality obrázku
* zvětšování obrázku je možné jen v omezené míře, neboť při větším zvětšení je na výsledném obrázku patrný rastr¨

# Existující formáty rastrové grafiky

* APNG
* BMP
* FLIF
* GIF
* HDP
* JPEG
* JPEG 2000
* MNG
* PCX
* PNG
* TIFF
* WBMP
* WebP
* XPM

## Podporované formáty rastrové grafiky

* BMP
* JPEG
* PNG
* TIFF

# C#

C# (vyslovované anglicky jako C Sharp, doslova to označuje notu cis) je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft zároveň s platformou .NET Framework, později schválený standardizačními komisemi ECMA (ECMA-334) a ISO (ISO/IEC 23270). Microsoft založil C# na jazycích C++ a Java (a je tedy nepřímým potomkem jazyka C, ze kterého čerpá syntaxi).

C# lze využít k tvorbě databázových programů, webových aplikací a stránek, webových služeb, formulářových aplikací ve Windows, softwaru pro mobilní zařízení (PDA a mobilní telefony) atd.

## Výhody jazyka C#

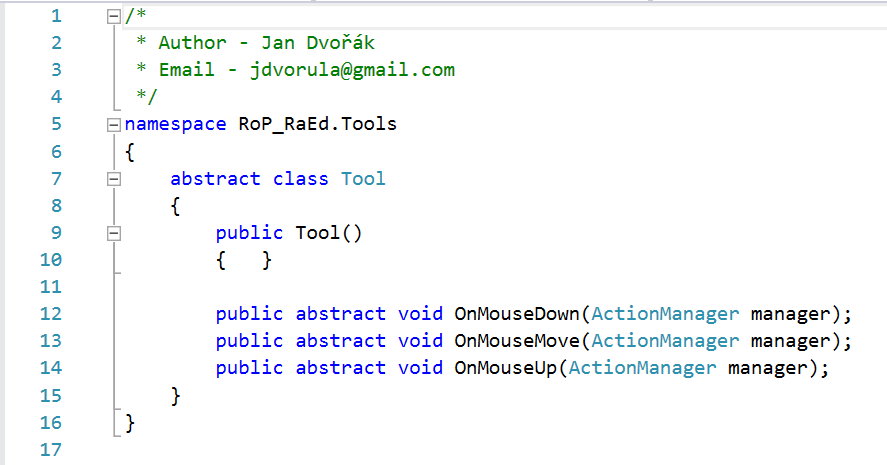
* C# je jednoduchý, moderní, mnohoúčelový a objektově orientovaný programovací jazyk.
* Jazyk a jeho implementace poskytuje podporu pro principy softwarového inženýrství, jakými jsou kupř. hlídání hranic polí, detekce použití neinicializovaných proměnných a automatický garbage collector. Důležité jsou také jeho vlastnosti jako robustnost, trvanlivost a programátorská produktivita.
* Jazyk je vhodný pro vývoj softwarových komponent distribuovaných v různých prostředích.
* Přenositelnost zdrojového kódu je velmi důležitá, obzvláště pro ty programátory, kteří jsou obeznámeni s C a C++.
* Mezinárodní podpora je též velmi důležitá.
* C# je navržen pro psaní aplikací jak pro zařízení se sofistikovanými operačními systémy, tak pro zařízení s omezenými možnostmi.
* Přestože by programy psané v C# neměly plýtvat s přiděleným procesorovým časem a pamětí, nemohou se měřit s aplikacemi psanými v C nebo jazyce symbolických adres.

Praktická část

Tato část obsahuje popis řešení jednotlivých problémů a průběh vývoje aplikace.

# Nástroje

Pro vytváření nástrojů jsem vytvořil abstraktní třídu, která obsahuje tři abstraktní metody pro akce tlačítek myši, které jsou požívány jednotlivými nástroji.



Obrázek - Abstraktní třída nástrojů

## Metoda OnMouseDown

Tato metoda řeší použití nástroje při stisku levého tlačítka myši.

Jako parametr této metody je celá nadřazená třída ActionManager, která obsahuje veškerá data pro správnou funkci nástroje.

## Metoda OnMouseMove

Tato metoda řeší použití nástroje mezi stisknutím levého tlačítka myši až do chvíle, kdy uživatel tlačítko pustí.

Jako parametr této metody je celá nadřazená třída ActionManager, která obsahuje veškerá data pro správnou funkci nástroje.

## Metoda OnMouseUp

Tato metoda řeší použití nástroje při puštění levého tlačítka myši

Jako parametr této metody je celá nadřazená třída ActionManager, která obsahuje veškerá data pro správnou funkci nástroje.

## Nástroj tužka

Při stisku myši s tímto zvoleným nástrojem se na plátno začne vykreslovat čára zvolené barvy, která následuje myš až do uvolnění levého tlačítka.

## Nástroj štětec

Po stisku levého tlačítka myši vykreslí kruh okolo kurzoru na plátně a následně, při držení tlačítka, sleduje pohyb kurzoru myši a v intervalu akce OnMouseMove na plátně spojuje aktuální a poslední pozici kurzoru křížem a zvolenou barvou. Při uvolnění levého tlačítka vykreslí další kruh, kterým překryje křížovou dráhu pohybu myši.

## Nástroj guma

Po stisku levého tlačítka myši vykreslí čtver okolo kurzoru na plátně bílou barvou (0xFFFFFF) a spojuje úseky pohybu myši.

## Nástroj ukazovátko

Tento nástroj je zvolen jako defaultní nástroj při spuštění programu.

## Nástroj kyblík

Tento nástroj při stisknutí tlačítka nad plátnem bude od tohoto místa hledat sousední, stejně barevná místa a bude je vykreslovat zvolenou barvou.

Algoritmus pro tento nástroj (Flood Fill) byl vytvořen dle internetového tutoriálu z portálu simpledevcode, který poskytuje tutoriály a kódy pro výukové účely.

## Nástroj kapátko

Při stisku na určité místo na plátnu tento nástroj z tohoto místa zjistí barvu a tuto barvu uživateli zvolí pro další používání jinými nástroji.

## Nástroj pohyb

Tento nástroj slouží k libovolnému  posouvání plátna po vyhrazené pozici ve formuláři.

# Menu

Ve formuláři aplikace se nachází hlavní menu programu, při horní liště formuláře.

## Položka File

Po otevření této položky můžeme najít volbu pro vytvoření nového plátna, otevření již existujícího obrázku, rychlé uložení obrázku, uložení obrázku s jinou cestou, a možnost ukončení programu.

## Položka Edit

Při otevření této položky můžeme najít veškeré nástroje editoru.

## Položka Image

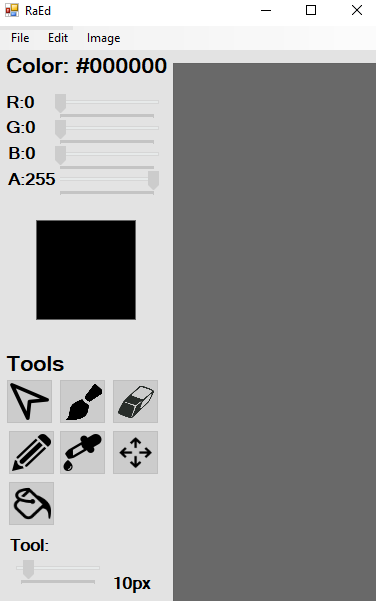
Při otevření této položky můžeme najít možnosti otáčení obrázku o 90 stupňů nalevo či napravo, otočení obrázku o 180 stupňů, možnost obrázek zrcadlově převrátit horizontálně, či vertikálně a možnost vyčistit celé plátno bílou barvou. Dále se zde nachází možnost vycentrovat plátno do formuláře, v případě že jej uživatel, pomocí nástroje posun, posunul mimo formulář.

# Třída ActionManager

Tato třída obsahuje většinu správy programu, která zajišťuje úpravy obrázku, uschovává informace o akcích uživatele a tyto akce zajišťuje.

# Formulář MainForm

Tento formulář je hlavní součástí programu, zajišťuje interakci mezi programem a uživatelem, obsahuje zobrazení upravovaného obrázku, aktuální souřadnice kurzoru, volbu barvy a nástrojů.



Obrázek - Náhled rozvržení uživatelského rozhraní

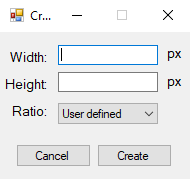
Rozvržení uživatelské prostředí je tvořeno kombinací podobných rastrových editorů (MS Paint, GIMP, …)

## Volba barvy

Při spuštění programu je defaultně zvolena černá barva (0x000000). Uživatel má možnost změnit zvolenou barvu pomocí posuvníků jednotlivých barevných složek, popřípadě pomocí nástroje kapátko.

# Formulář NewFileForm

Tento formulář má na starosti získání dat od uživatele, která následně je předá třídě ActionManager. Na základě těchto dat vytvoří nový obrázek, formátu BMP, dle specifikace uživatele.

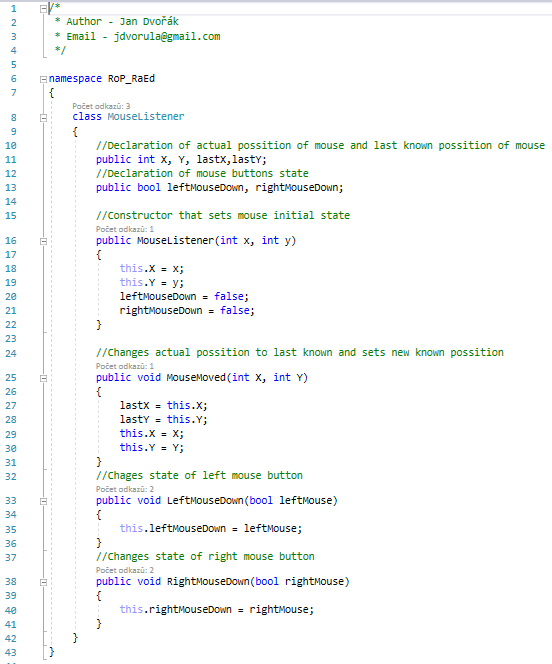


Obrázek - Formulář pro vytvoření nového obrázku

Formulář od uživatele požaduje zadání výšky a šířky vytvářeného obrázku. V případě že se uživatel rozhodne pro některý z předpřipravených poměrů stran obrázku, bude uživatel požádán o zadání pouze šířky obrázku a následná výška bude dopočítána z šířky, pro dodržení poměru.

# Třída MouseListener

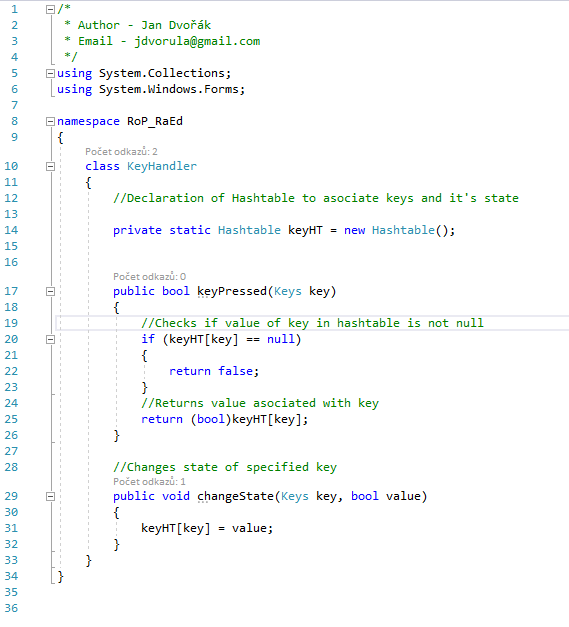
Tato třída obsahuje informace o stavu myši. Starostí této třídy je uchovávání aktuálních souřadnic, na kterých se kurzor nachází a uchovává také souřadnice o předchozí pozici kurzoru, současně se stavem tlačítek myši.



Obrázek : Kód třídy MouseListener

# Třída KeyHandler

Tato třída uchovává aktuální stavy jednotlivých kláves počítače. Třída byla vytvořena pro funkce editoru, které pracují s klávesovými zkratkami a klávesami počítače.



Obrázek 5: Ukázka kódu třídy Key Handler

Závěr

Cílem projektu bylo vytvořit funkční počítačový program pro editaci rastrové grafiky. Tento cíl se podařilo splnit nad má očekávání. Program je schopen vytvořit, nebo otevřít již existující rastrový soubor, ten zeditovat a následně jej uložit. V programu je vytvořeno několik nástrojů, díky kterým je tato editace možná, jako například: tužka, štětec, guma, kapátko… Dále je uživateli umožněno obrázek libovolně otáčet o 90 nebo 180 stupňů a dle horizontální, nebo vertikální osy jej převracet.

Díky tomuto projektu jsem se získal mnoho užitečných znalostí z oblasti objektově orientovaného programování, algoritmizace ale i návrhu uživatelského rozhraní. Přes to že tento projekt nedosáhl mých prvotních představ, tak se jedná o plně použitelný editor rastrové grafiky. Struktura kódu umožňuje program i nadále rozšiřovat novými nástroji anebo měnit uživatelské rozhraní.

# Seznam obrázků

[Obrázek 1 - Abstraktní třída nástrojů 20](#_Toc510515129)

[Obrázek 2 - Náhled rozvržení uživatelského rozhraní 25](#_Toc510515130)

[Obrázek 3 - Formulář pro vytvoření nového obrázku 26](#_Toc510515131)

[Obrázek 4: Kód třídy MouseListener 27](#_Toc510515132)

[Obrázek 5: Ukázka kódu třídy Key Handler 28](#_Toc510515133)

# Použité zdroje

## Tištěné dokumenty

*SELLS, Chris. Windows forms programming in C#. Boston: Addison-Wesley, c2004. ISBN 9780321116208.*

*SHARP, John a John JAGGER. Microsoft Visual C# .NET krok za krokem. Praha: Mobil Media, 2002. iDnes internet knihy. ISBN 8086593274.*

## Elektronické dokumenty

*DOTNETPORTAL [online]. 2018 [cit. 2018-03-29]. Dostupné z: https://www.dotnetportal.cz/*

*OUMGHAR, Karim. Simple dev code: Flood Fill algorithm (using C#.Net) [online]. 2015, December 29, 2015 [cit. 2018-03-29]. Dostupné z: https://simpledevcode.wordpress.com/2015/12/29/flood-fill-algorithm-using-c-net/*

*What's New for Visual C# in Visual Studio 2012 [online]. Microsoft Developer Network [cit. 2013-08-02]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/index*

*?? Operator (C# Reference) [online]. Microsoft [cit. 2008-11-23]. Dostupné z: https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/null-conditional-operator*

*Microsoft Visual Studio. In: Visual studio [online]. [cit. 2018-03-29]. Dostupné z: https://www.visualstudio.com/cs/?rr=https%3A%2F%2Fwww.google.cz%2F*