

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO  
KATEDRA INFORMATIKY

## BAKALÁŘSKÁ PRÁCE

Univerzální grafické uživatelské rozhraní s editorem  
formulářů pro řádkové utility



2010

Michal Václavský

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně.

30. července 2010

Michal Václavský

## **Anotace**

*Cílem této bakalářské práce bylo vytvořit aplikaci pro editaci předloh pro spouštění tzv. řádkových utilit.*

Děkuji rodičům za podporu při studiu.

# Obsah

<b>1. ÚVOD</b>	<b>1</b>
1.1. Zadání bakalářské práce . . . . .	1
1.2. Rozdělení práce . . . . .	1
1.3. Motivace . . . . .	1
1.4. Cíle práce . . . . .	2
1.5. Existující řešení . . . . .	2
<b>2. ODBORNÁ ČÁST</b>	<b>3</b>
2.1. Případy užití . . . . .	3
2.1.1. Základní model aplikace . . . . .	3
2.1.2. Práce s programem . . . . .	4
2.1.3. Funkce programu . . . . .	5
2.2. Struktura aplikace . . . . .	6
2.2.1. Schema aplikace . . . . .	6
2.2.2. Grafické uživatelské rozhraní . . . . .	7
2.2.3. Třídy uživatelského rozhraní . . . . .	7
2.2.4. Odvozené třídy parametrů . . . . .	8
2.2.5. Pomocné třídy pro parametry . . . . .	9
2.2.6. Další použité třídy . . . . .	9
2.3. Funkce komponenty CLIEditorControl . . . . .	9
2.4. Testování aplikace . . . . .	10
<b>3. UŽIVATELSKÁ ČÁST</b>	<b>11</b>
3.1. Systémové požadavky . . . . .	11
3.2. Instalace aplikace . . . . .	11
3.2.1. Před instalací aplikace . . . . .	11
3.2.2. Instalace aplikace . . . . .	12
3.2.3. Odinstalování aplikace . . . . .	15
3.3. Popis aplikace . . . . .	16
3.3.1. Hlavní okno aplikace . . . . .	16
3.3.2. Strom parametrů programu . . . . .	17
3.3.3. Property editor . . . . .	18
3.3.4. Editační část . . . . .	19
3.3.5. Ovládací tlačítka aplikace . . . . .	19
3.3.6. Parametry programu . . . . .	19
3.3.7. Doplnkové funkce pro parametry programu . . . . .	20
3.3.8. Vytvoření nového programu . . . . .	20
3.3.9. Editace existujícího programu . . . . .	21
3.3.10. Nastavení aplikace . . . . .	22
3.3.11. O aplikaci . . . . .	22
3.4. Popis vyhodnocování podmínek . . . . .	23

<b>4. POJMY A ZKRATKY</b>	<b>24</b>
<b>5. ZÁVĚR</b>	<b>25</b>
<b>Reference</b>	<b>26</b>
<b>A. Struktura přiloženého CD</b>	<b>27</b>
<b>B. Seznam utilit</b>	<b>28</b>

## Seznam obrázků

1.	Základní model systému . . . . .	3
2.	Práce s programem . . . . .	4
3.	Funkce programu . . . . .	5
4.	Schema aplikace . . . . .	6
5.	Grafické uživatelské rozhraní aplikace . . . . .	7
6.	Odvozené třídy parametrů . . . . .	8
7.	Hlavní okno aplikace . . . . .	16
8.	Strom parametrů programu . . . . .	17
9.	Property editor - vlastnosti parametru programu . . . . .	18
10.	Editační část programu . . . . .	19
11.	Vytvoření nového programu . . . . .	21
12.	Otevření existujícího programu . . . . .	21
13.	Dialog nastavení aplikace . . . . .	22
14.	Dialog zobrazení informací o aplikaci . . . . .	22

# 1. ÚVOD

Aplikace *Editor pro řádkové utility*, vznikla jako závěrečná práce vysokoškolského studia na Univerzitě Palackého v Olomouci, obor Aplikovaná informatika v kombinované formě. Podle zadání bakalářské práce měla být vytvořena aplikace, která umožní vytvoření a editaci definice pro řádkové utility. Další část práce [1] pak řádkovou utilitu s parametry podle definice spustí.

## 1.1. Zadání bakalářské práce

Zatímco běžní uživatelé počítačů ovládají většinu programů pomocí vizuálních ovládacích prvků, mezi informatiky jsou velmi oblíbené především programy ovládané pouze zadáváním parametrů na příkazové řádce, tzv. řádkové utility.

Tento způsob spouštění a konfigurace programů umožňuje velmi pohodlnou práci zkušeným uživatelům, o to méně příjemný je však pro začátečníky.

Úkolem této bakalářské práce je navrhnout univerzální systém, který by umožnil definovat uživatelská rozhraní pro nejběžnější řádkové utility bez nutnosti cokoliv programovat. Každý program bude mít svou šablonu, která bude jednoduchým způsobem definovat, jaké parametry lze u daného programu na příkazové řádce zadávat. Vytvořený systém umožní pohodlně vytvářet tyto šablony a potom je použít pro spouštění programů pracujících na příkazové řádce.

## 1.2. Rozdělení práce

Práce byla zadána pro 2 studenty. První část řeší editaci šablon pro řádkové utility a druhá část aplikace pak vybranou definici spustí. Tato práce se zabývá pouze řešením editační části. Druhá část spouštěče byla řešena kolegou [1]. Při analýze i řešení práce jsme společně části konzultovali, ale každá práce je naprosto samostatná a nezávislá. Spolupráce mezi aplikacemi je řešena na úrovni DLL knihoven.

## 1.3. Motivace

Zadané téma bakalářské práce mě zaujalo, protože sám řádkové utility v praxi používám. Řešení v jazyce C# mi připadalo zajímavé a s využitím nových programovacích technik se dá realizovat i velice efektivní řešení. V minulosti jsem žádnou podobnou aplikaci nedělal, tudíž jsem očekával, že mi realizace této práce rozšíří mé dosavadní zkušenosti. Další motivací byla možnost spolupráce s kolegou a vzájemné provázání obou aplikací. To je další věc, kterou uplatním ve své praxi.



## **1.4. Cíle práce**

Cílem této práce je vytvořit novou, případně editovat existující definici pro spouštěč řádkových utilit. Některé utility používají velké množství parametrů a práce s nimi může být nepřehledná. Popisovaný editor se snaží přiblížit práci s definicemi řádkových utilit většímu okruhu uživatelů tak, aby i laik byl schopný definici bez velikých znalostí vytvořit.

## **1.5. Existující řešení**

V minulosti na katedře informatiky bylo toto téma již několika studenty zpracovááno. Nenašel jsem však žádné komplexní řešení spouštěče i editoru dohromady. Samostatný nástroj pro editování šablon jsem nenašel vůbec. To však nevylučuje existenci nějakého řešení.

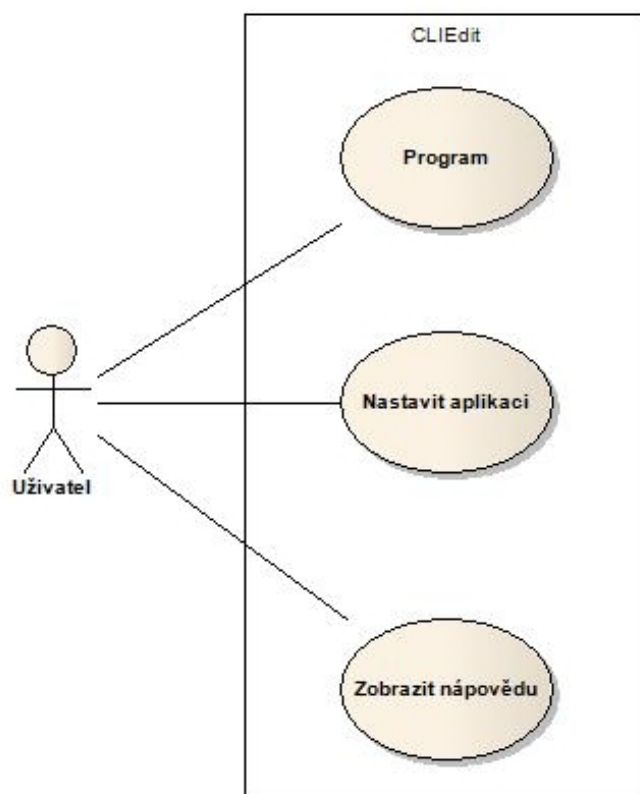
## 2. ODBORNÁ ČÁST

### 2.1. Případy užití

Podle zadání bakalářské práce byla provedena základní analýza a poté byly vytvořeny následující diagramy případů užití.

#### 2.1.1. Základní model aplikace

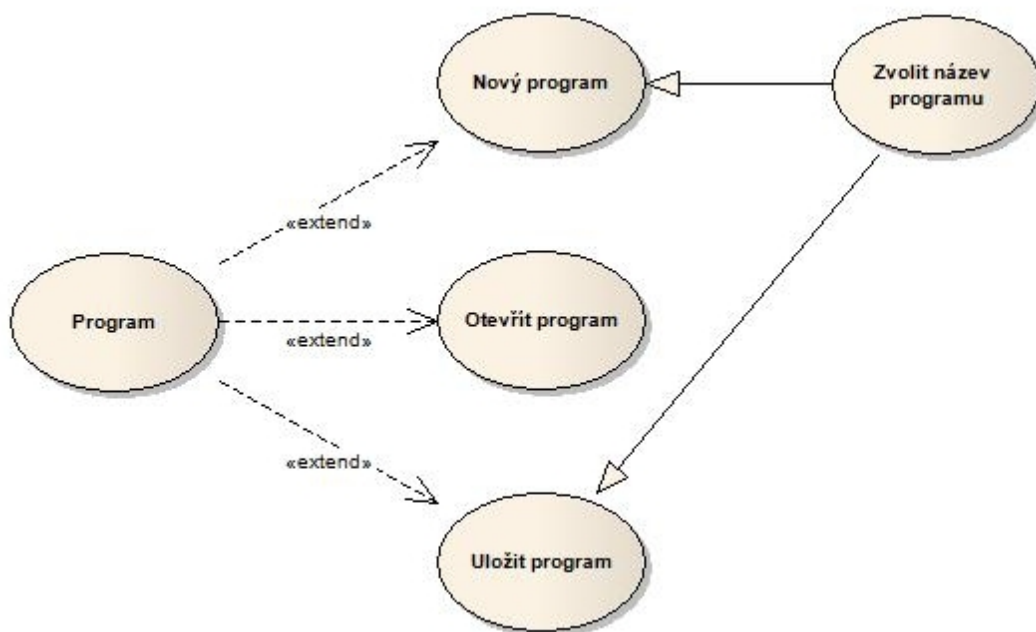
Základní model systému je velmi jednoduchý a je rozdělen do několika částí. Tyto části jsou vidět v diagramu (obrázek č. 1.).



Obrázek 1. Základní model systému

### 2.1.2. Práce s programem

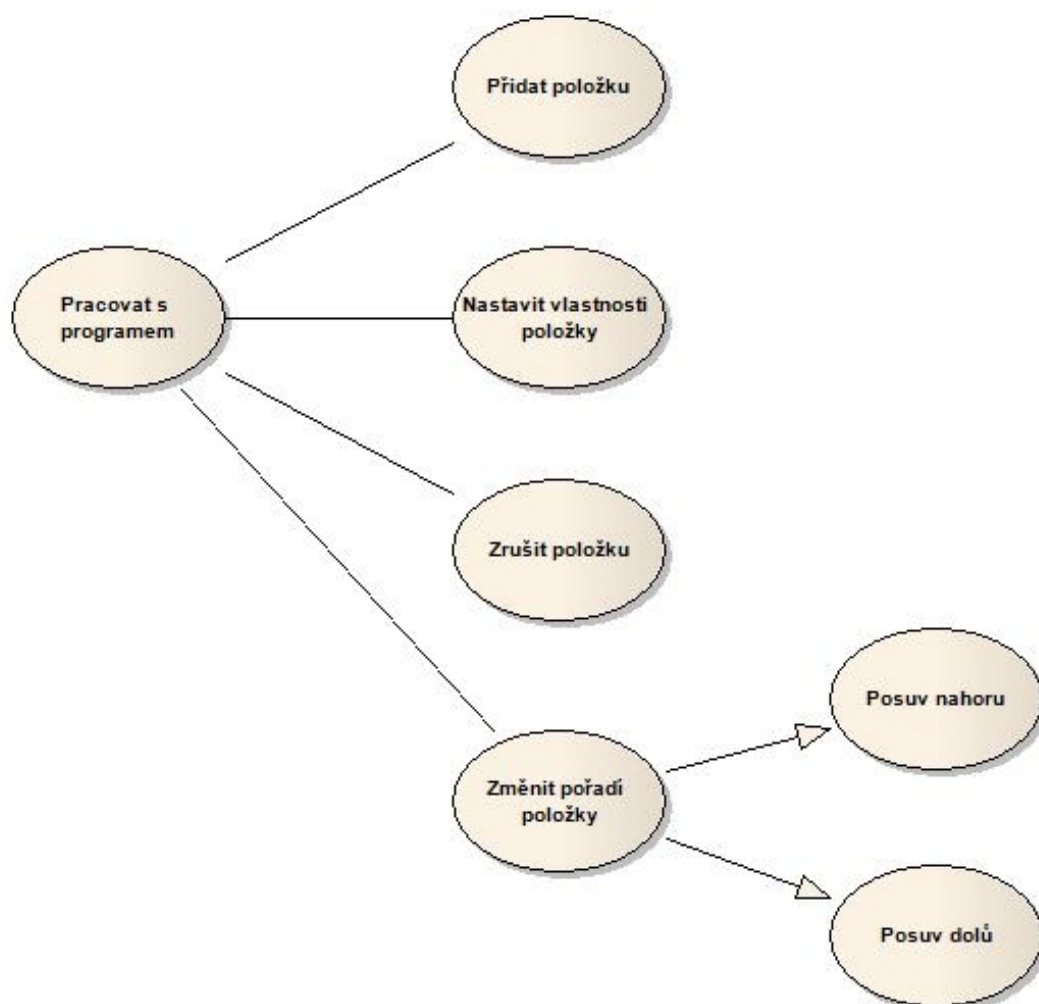
Na obrázku č. 2. je vidět případ užití pro část aplikace, která zajišťuje práci s programem.



Obrázek 2. Práce s programem

### 2.1.3. Funkce programu

Detail funkcí pro práci s programem je na obrázku č. 3..

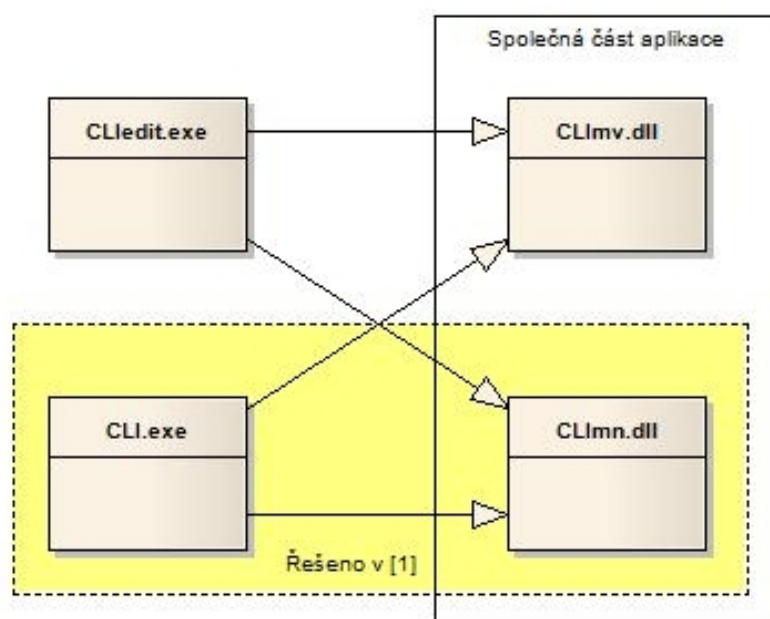


Obrázek 3. Funkce programu

## 2.2. Struktura aplikace

### 2.2.1. Schema aplikace

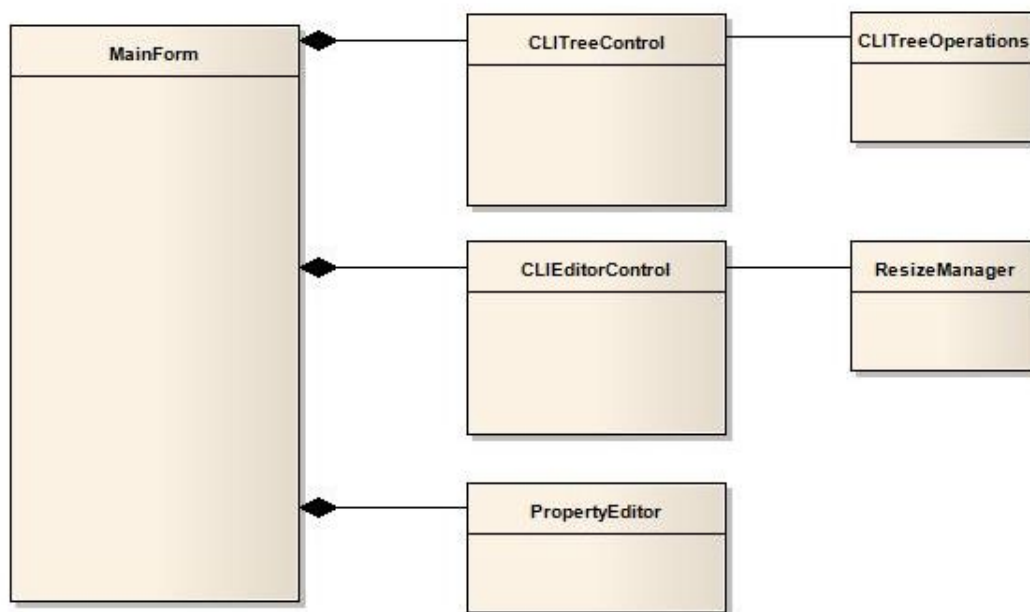
Na obrázku č. 4. je vidět rozdělení aplikace na část spouštěče [1] a část editoru. Jak již bylo zmiňováno v úvodu, aplikace využívá společné části, které jsou umístěné v knihovnách DLL. Jiný důvod pro umístění těchto částí do DLL není a pokud by nebylo potřeba je sdílet zřejmě by byly zakomponovány přímo do EXE aplikace. Další popis se týká části editoru.



Obrázek 4. Schema aplikace

### 2.2.2. Grafické uživatelské rozhraní

Na obrázku č. 5. je znázorněno schema grafického uživatelského rozhraní aplikace (GUI). GUI aplikace se skládá z několika částí, které jsou tvořeny pomocí tzv. *UserControl*. Každá z částí má svoji roli a jednotlivé části mezi sebou komunikují pomocí eventů. Podrobný popis jednotlivých částí je v následující kapitole.



Obrázek 5. Grafické uživatelské rozhraní aplikace

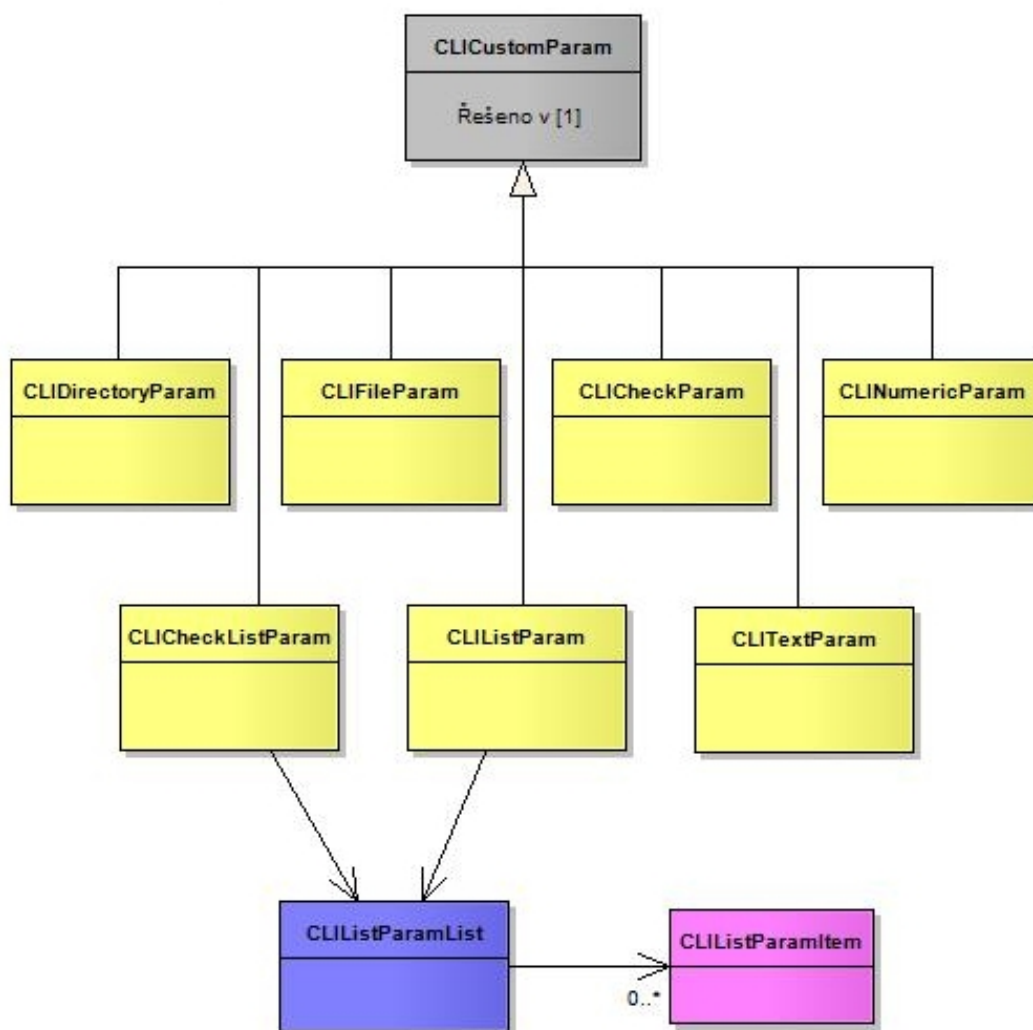
### 2.2.3. Třídy uživatelského rozhraní

Pro práci s definicí programu jsem definoval následující *UserControl*.

- **CLITreeControl**  
zajišťuje práci se stromovým zobrazením definice programu. Vrací aktivní (vybranou) položku definice programu. Obsahuje metody pro nalezení další a předchozí položky důležité pro přesuny pořadí položek.
- **CLIEditorControl**  
grafická komponenta pro editaci položek definice programu myší. Lze provádět přesuny vybrané položky na jiné místo, změny velikosti a rozvržení.

#### 2.2.4. Odvozené třídy parametrů

Komponenty použité v aplikaci vychází z definice třídy CLIBaseParam definované v [1]. Pro potřeby aplikace editoru byly definovány nové vlastnosti komponent.



Obrázek 6. Odvozené třídy parametrů

- **CLIDirectoryParam** - *parametr* typu adresář
- **CLIFileParam** - *parametr* typu soubor
- **CLICheckParam** - *parametr* typu zaškrtačací políčko

- **CLINumericParam** - *parametr* typu numerické pole
- **CLITextParam** - *parametr* typu textové pole
- **CLICheckListParam** - *parametr* typu zaškrťovací seznam
- **CLIListParam** - *parametr* typu seznam

### 2.2.5. Pomocné třídy pro parametry

- **CLIListParamList**  
pomocná třída pro definice *parametrů* typu *CLIListParam*, *CLICheckListParam*
- **CLIListParamItem**  
pomocná třída pro definice *parametrů* typu *CLIListParamList*

### 2.2.6. Další použité třídy

- **CLITreeOperation**  
třída pro generování stromové struktury definice programu. Při generování je třeba zohlednit *parametry* typu *CLIGroup* a stromovou strukturu správně vytvořit ve všech úrovních vnoření. Zde byl použitý rekurzivní algoritmus.
- **ResizeManager**  
třída pro kreslení ohraničení vybraného *parametru* programu a změny jeho velikosti.

## 2.3. Funkce komponenty CLIEditorControl

Při vývoji aplikace jsem narazil na problém s generováním vizuální podoby *prvků*. Když byla položka označena myší, docházelo k různým nekontrolovaným změnám stavu komponent. Například zaškrtnutí hodnoty u parametru typu *CLICheckBox*. Protože komponenta *CLIEditorControl* byla navržena pouze pro přesuny *prvků* byly změny ostatních vlastností nežádoucí.

Pomocí *CLIParamControlArranger* [1] jsou metodou *Arrange* generovány *parametry* na panel, který není zobrazován. Jeho vzhled je zachycen do bitmapy, zobrazené na komponentě typu *PictureBox* a vlastnosti tedy nelze editovat. Velikost generované plochy panelu a bitmapy jsou identické, takže podle pozice myši lze snadno spočítat pozice *parametrů* a provadět jejich označení.

Při změně velikosti, nebo pozice je pak generována nová bitmapa.



## 2.4. Testování aplikace

Testování aplikace probíhalo souběžně s vývojem aplikace. Pro vývoj byla použita verze ExpressEdition, která nemá žádné speciální nástroje pro testování. Proto testování probíhalo pouze spouštěním aplikace a zadáváním různých vstupů. Při chybě bylo nalezeno místo v programu, které způsobilo problém a chyba byla opravena.

Testování průběhu instalace bylo prováděno na systémech Microsoft Windows XP Professional (Service pack 3) a Microsoft Windows 7 Professional. Předpokládám, že instalace bez problému proběhne i na systému Windows Vista, přestože tento systém nebyl pro testování instalace k dispozici.

## 3. UŽIVATELSKÁ ČÁST

### 3.1. Systémové požadavky

Aplikace byla naprogramována v prostředí Microsoft Visual C# verze 2010, které pracuje s .Net Frameworkem 4.0. Dle [4] by měl mít počítač pro provozování aplikace následující minimální konfiguraci.

- přibližně 5 MB místa pro aplikaci,
- operační systém Microsoft Windows XP a vyšší,
- operační paměť minimálně 512MB,
- frekvence procesoru minimálně 1GHz.
- nainstalovaný .Net Framework verze 4.0
- pokud není nainstalovaný Framework 4.0
- přibližně 850 MB místa na pevném disku (x86),
- přibližně 2 GB místa na pevném disku (x64),

### 3.2. Instalace aplikace

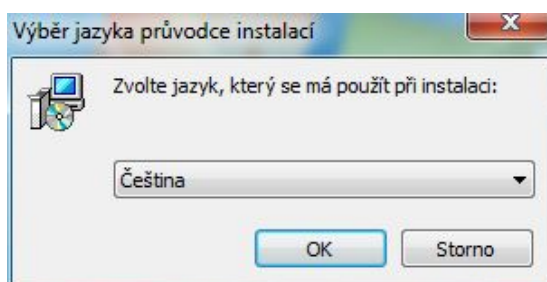
Pro vytvoření instalace aplikace byl použit nástroj *Inno setup*, který je distribuován pod GPL licencí. Instalace byla vytvořena ve verzi 5.3.10, která je přiložena na cd, nebo je možno stáhnout ji z webu [6].

#### 3.2.1. Před instalací aplikace

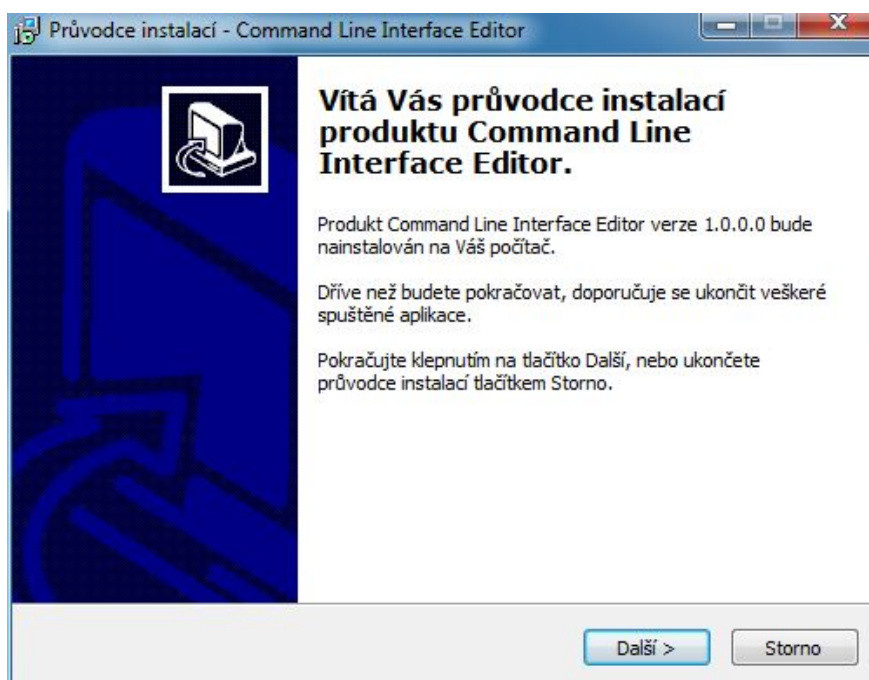
Před spuštěním instalace je třeba prověřit, zda je v systému nainstalovaný framework verze 4. Instalační proces si tuto skutečnost kontroluje a v případě, že instalaci frameworku nenajde přeruší průběh instalace. Verze frameworku potřebná pro běh aplikace je přiložena na cd, nebo je možno ji stáhnout z webu výrobce operačního systému [4].

### 3.2.2. Instalace aplikace

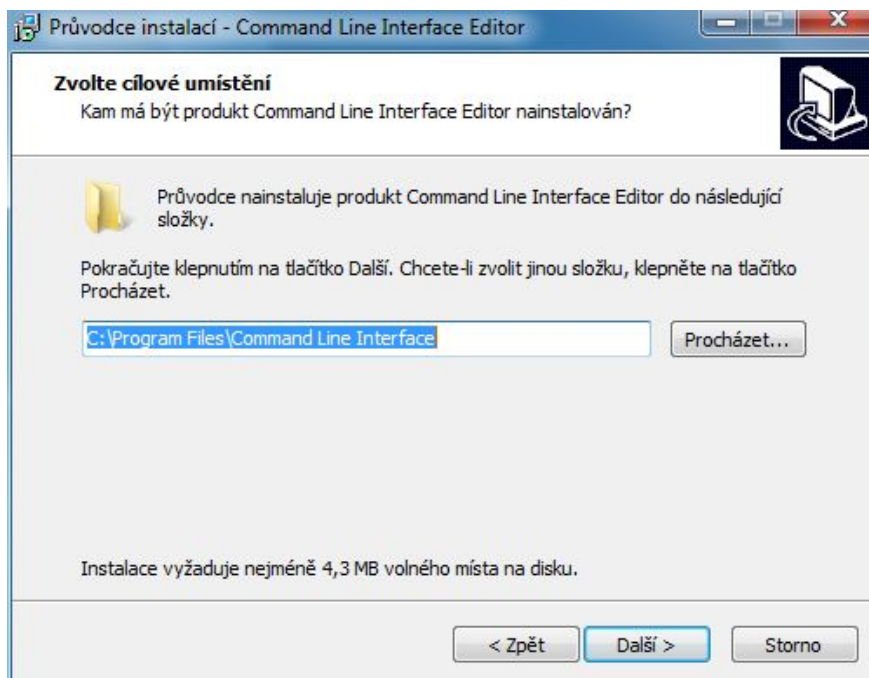
1. Proces instalace aplikace se spustí z adresáře Install na cd pomocí souboru *CLIEditSetup.exe*.
2. Jako první krok instalace je zobrazen výběr jazyka pro průběh instalace. Je možné zvolit češtinu, nebo angličtinu. Podle zvoleného jazyka se nastaví jazyk dalších dialogových oken zobrazovaných v průběhu instalace. Po výběru jazyka instalace klikněte na tlačítko OK.



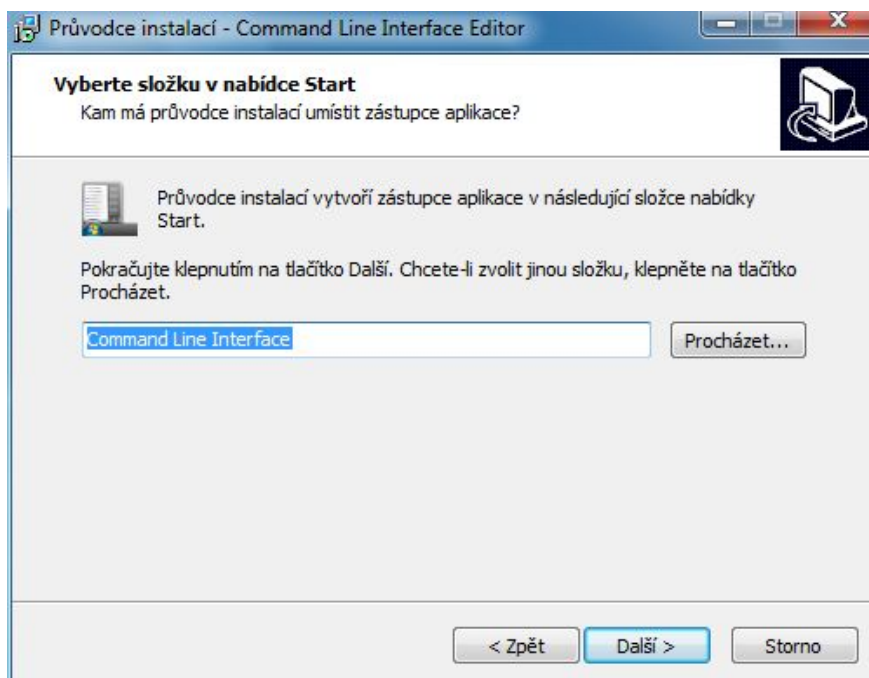
3. Postupujte podle pokynů na obrazovce a klikněte myší na tlačítko Další.



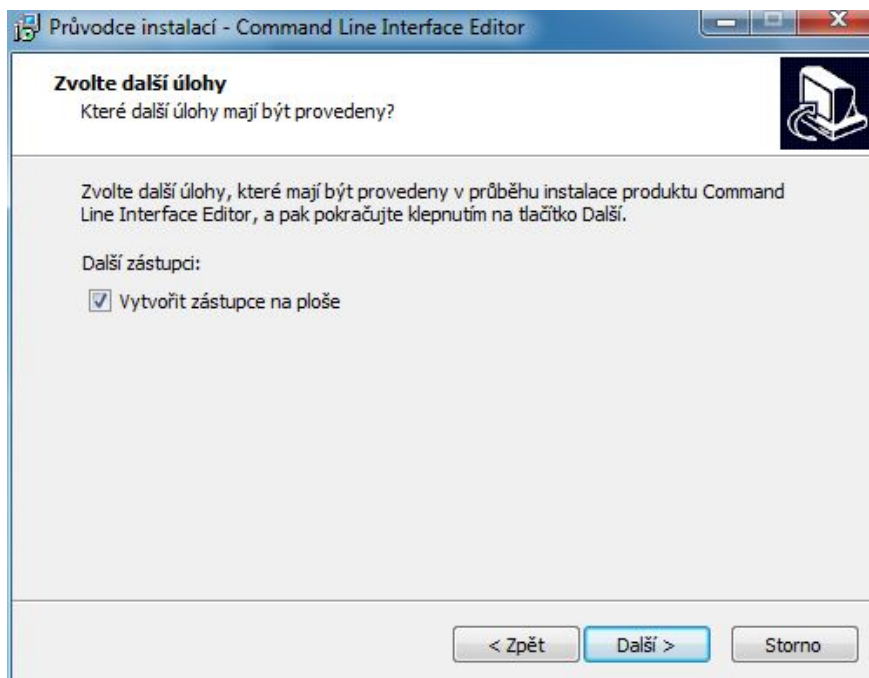
4. Vyberte cílovou složku aplikace. Postupujte podle pokynů na obrazovce a klikněte myší na tlačítko Další.



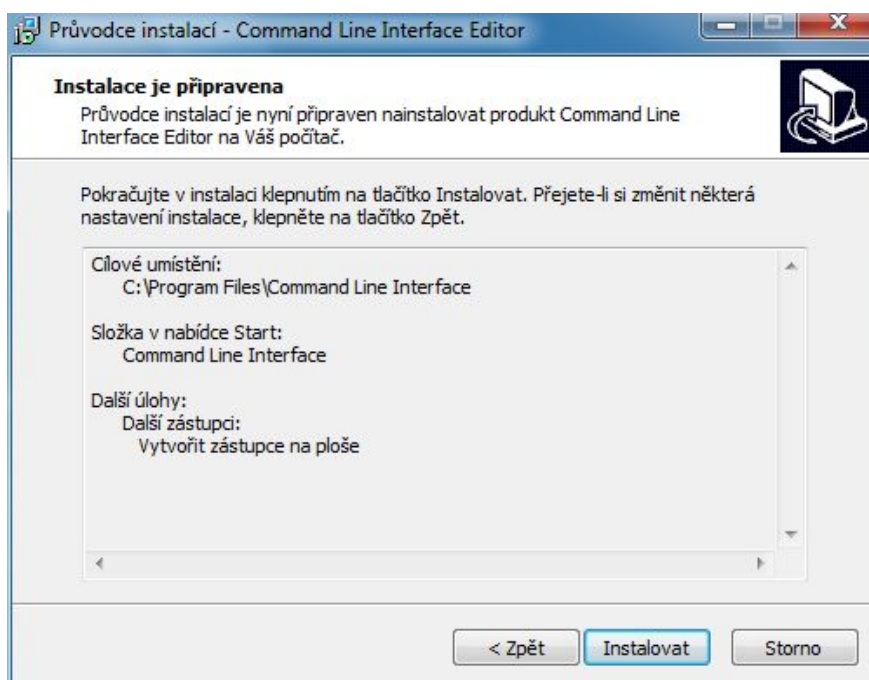
5. Vyberte složku v nabídce Start. Postupujte podle pokynů na obrazovce a klikněte myší na tlačítko Další.



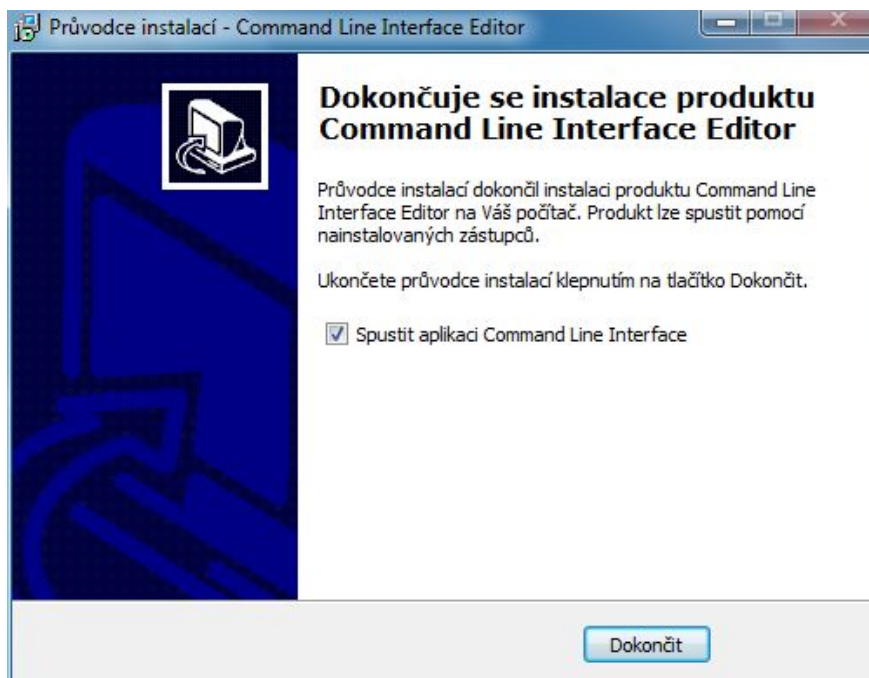
6. Vyberte pokud chcete vytvořit zástupce na ploše. Postupujte podle pokynů na obrazovce a klikněte myší na tlačítko **Další**.



7. Postupujte podle pokynů na obrazovce a klikněte myší na tlačítko **Instalovat**.



8. Vyberte pokud chcete aplikaci po ukončení instalace spustit. Postupujte podle pokynů na obrazovce a klikněte myší na tlačítko **Dokončit**.



### 3.2.3. Odinstalování aplikace

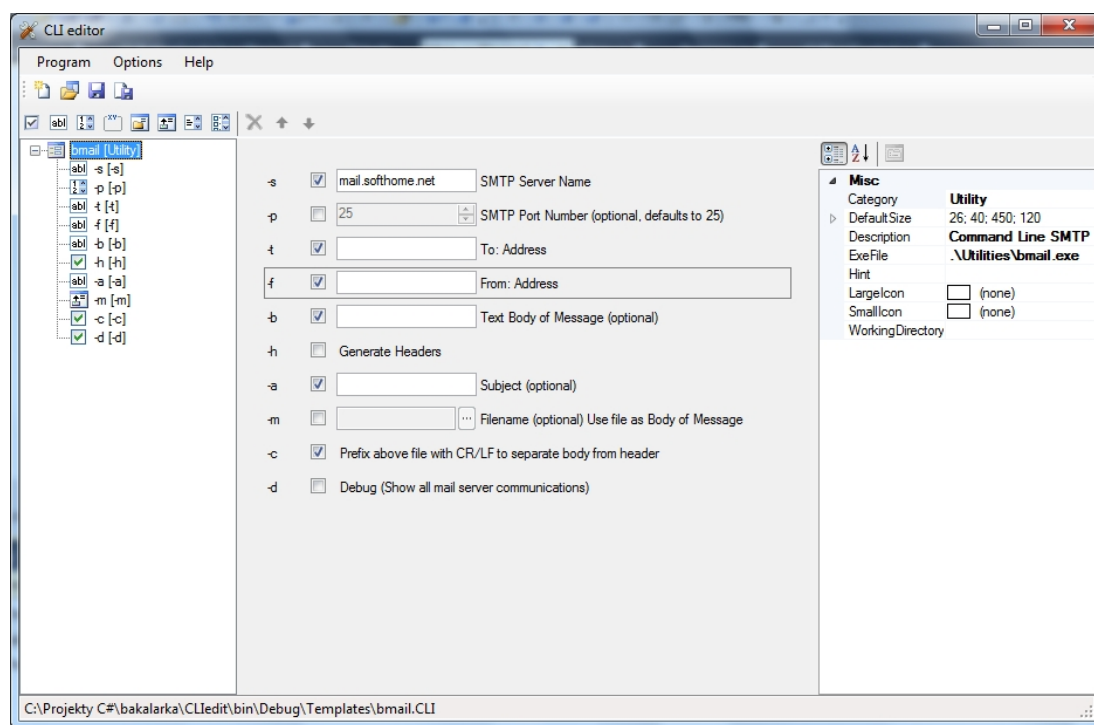
Aplikaci je možné odinstalovat spuštěním zástupce ze skupiny *Command Line Interface*. Odinstalace odstraní soubory a adresáře z PC. Start odinstalace je třeba potvrdit a po dokončení odinstalace se objeví dialog se zprávou, že odinstalace proběhla korektně.

### 3.3. Popis aplikace

#### 3.3.1. Hlavní okno aplikace

Hlavní okno aplikace je rozděleno do několika základních částí. Vlevo je strom se všemi *parametry programu*. V rootu stromu je pak samotný *program*. Uprostřed je editační část se zobrazením všech *parametrů programu*. V editační části je možno pomocí myši přesouvat jednotlivé *parametry programu*. Vpravo je mřížka property editoru, kde je možné přepisovat vlastnosti jednotlivých položek vybraného *parametru*.

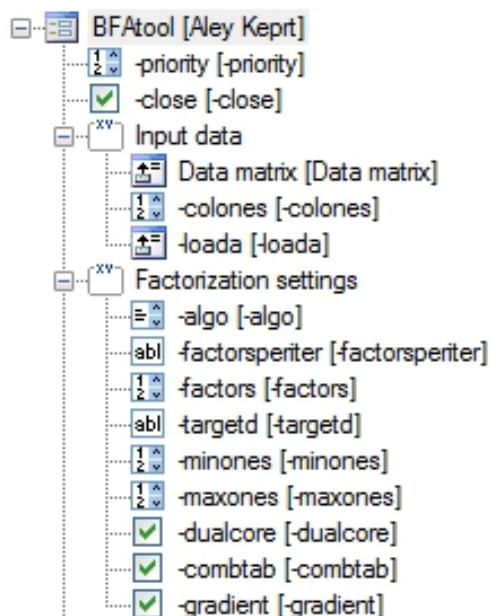
Přístup k funkcím aplikace je možný z tlačítkové lišty, plovoucího menu, nebo z hlavního menu aplikace. Všechny tyto volby jsou funkčně rovnocenné a provádí stejné operace.



Obrázek 7. Hlavní okno aplikace

### 3.3.2. Strom parametrů programu

Na obrázku č. 8. je ukázka stromu s *parametry programu*. Výběrem položky ve stromu se překreslí výčet vlastností v mřížce property editoru. V editační části hlavního okna je pak vidět vybraná položka zvýrazněná. Pokud je aktivní položkou samotný textslprogram, pak není ve střední části zvýrazněna žádná položka.

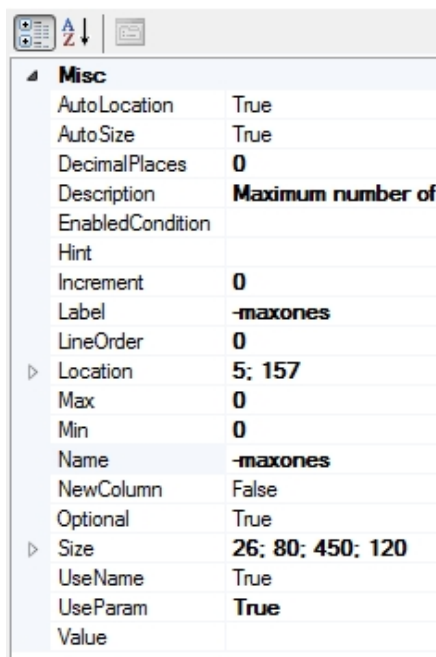


Obrázek 8. Strom parametrů programu



### 3.3.3. Property editor

Na obrázku č. 9. je ukázka mřížky property editoru. Jsou zde všechny vlastnosti *parametru programu*. Každou z těchto vlastností lze editovat. Přepsáním některé z vlastností se překreslí strom a zobrazení položky v editační části hlavního okna aplikace. Například při změně pozice *parametru programu* není změna ve stromovém zobrazení vidět, ale v editační části se změna projeví.

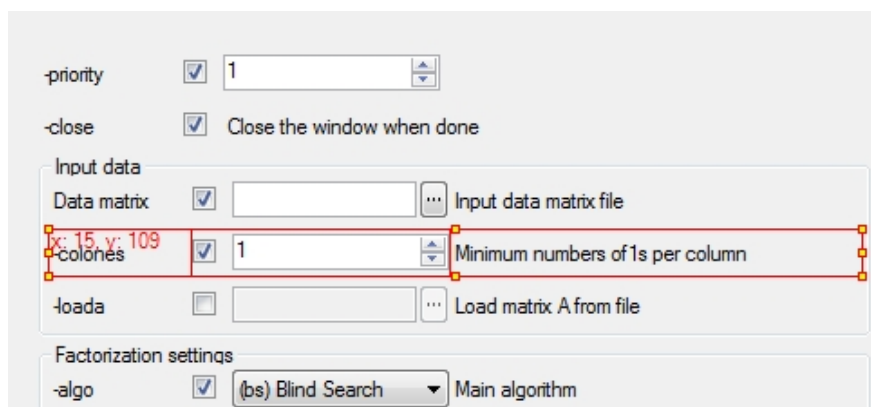


Misc	
AutoLocation	True
AutoSize	True
DecimalPlaces	0
Description	<b>Maximum number of</b>
EnabledCondition	
Hint	
Increment	0
Label	<b>-maxones</b>
LineOrder	0
Location	<b>5; 157</b>
Max	0
Min	0
Name	<b>-maxones</b>
NewColumn	False
Optional	True
Size	<b>26; 80; 450; 120</b>
UseName	True
UseParam	<b>True</b>
Value	

Obrázek 9. Property editor - vlastnosti parametru programu

### 3.3.4. Editační část





Na obrázku č. 10. je ukázka editační části *parametrů programu*. Jsou zde všechny *parametry programu* rozmístěné podle nastavení vlastností jednotlivých *parametrů*. Pozice jednotlivých *parametrů programu* lze měnit přesouváním pomocí myši. Nejprve je třeba *parametr* vybrat najetím myši a poté levým tlačítkem provést jeho označení. Přesun se provádí posunem myši při současném držení levého tlačítka myši. Uvolněním tlačítka myši vybraný objekt přesuneme na požadovanou pozici. Pokud je potřeba pozici upravit je možné ji také přepsat v property editoru.



Obrázek 10. Editační část programu

### 3.3.5. Ovládací tlačítka aplikace

ikona	popis/význam
-------	--------------

	nový program
	otevřít program
	uložit program
	uložit program jako









### 3.3.6. Parametry programu

*Parametry* jsou základní stavební prvky *programu*. Pomocí nich lze sestavit definici řádkové utility a jednotlivých voleb. Každý z *parametrů* má vlastnosti jako je *jméno*, *popisek*, *pozice*, *pořadí na příkazovém řádku*, *povolující podmínka*. Uvedené vlastnosti jsou společné pro všechny *parametry*, jejich výčet je mnohem

větší, ale je nad rámec této dokumentace.

Do definice *programu* je možné vkládat položky z následujícího seznamu. Položku je možné přidat stiskem tlačítka z tlačítkové lišty komponent, nebo z popup menu *Add new control*.




#### ikona popis/význam

	zaškrtačací políčko
	textové pole
	numerická hodnota
	skupina prvků
	adresář
	soubor
	seznam
	zaškrtačací seznam

### 3.3.7. Doplňkové funkce pro parametry programu

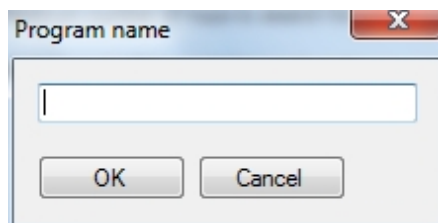
Funkce z následujícího seznamu slouží pro práci s vybraným *parametrem*. Pro rušení vybrané položky je možné v menu **Options** aktivovat volbu kontrolního dotazu. Šipky jsou určeny pro přesuny pořadí vybraného *parametru*.

#### ikona popis/význam

	zrušit vybraný parametr
	přesunout vybraný parametr nahoru
	přesunout vybraný parametr dolů

### 3.3.8. Vytvoření nového programu

Novou definici *programu* lze vytvořit volbou z menu **Program > New program**, případně volbou z tlačítkové lišty, nebo z plovoucího menu. Po provedení této volby se zobrazí dialog se zadáním jména *programu*. Potvrzením **OK**, nebo stiskem klávesy **Enter** se založí nová (prázdná) definice *programu*. Stiskem tlačítka **Cancel**, případně stiskem klávesy **Esc** se dialogové okno zavře a nová definice *programu* se nevytvoří.

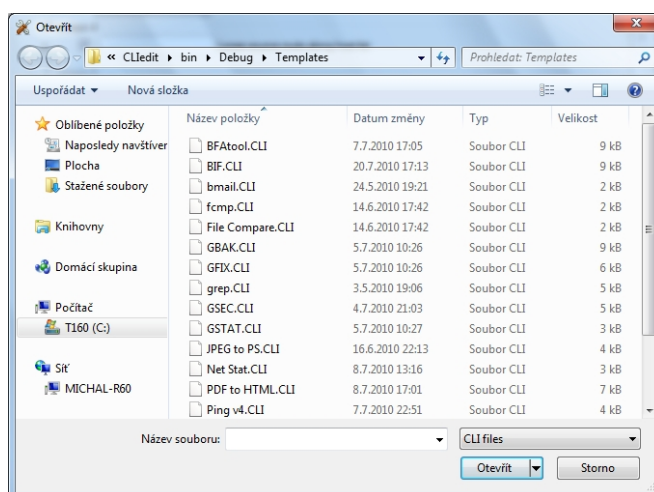


Obrázek 11. Vytvoření nového programu

### 3.3.9. Editace existujícího programu

Existující definici *programu* lze otevřít pro editaci volbou z menu **Program** > **Load program**, případně volbou z tlačítkové lišty, nebo z plovoucího menu. Po provedení této volby se zobrazí systémový dialog se strukturou souborů na pc. Po vybrání a potvrzení požadované definice se otevře definice *programu*, kterou je možné upravovat.

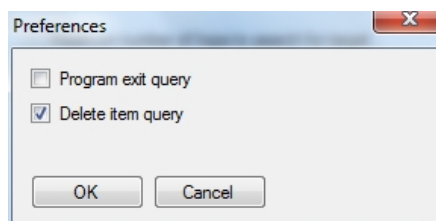
Výchozí adresář pro otevírání definic *programu* je řízen operačním systémem.



Obrázek 12. Otevření existujícího programu

### 3.3.10. Nastavení aplikace

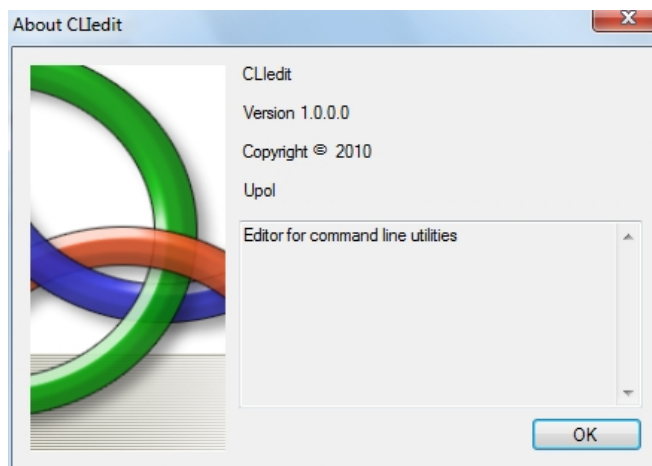
Dialog pro nastavení aplikace lze otevřít volbou z menu **Options**. Lze nastavit zapnutí/vypnutí kontrolního dotazu pro rušení *parametru*. Dále lze nastavit možnost zapnutí/vypnutí kontrolního dotazu pro ukončení aplikace.



Obrázek 13. Dialog nastavení aplikace

### 3.3.11. O aplikaci

Dialog pro zobrazení informací o aplikaci lze otevřít volbou z menu **Help > About CLI editor**.



Obrázek 14. Dialog zobrazení informací o aplikaci

### 3.4. Popis vyhodnocování podmínek

Pro povolování/zakazování jednotlivých *parametrů* je možné napsat do položky *EnableCondition* výraz, který se vyhodnotí a na základě výsledku vyhodnocení tohoto výrazu se *parametr* povolí, nebo zakáže. V této kapitole je popsán způsob použití a povolené operátory pro tyto výrazy.

Výraz je popsán identifikátorem *parametru* a názvem vlastnosti. Jako oddělovač je použita tečka, tzn. celý zápis příklad `-algo.value`.

V tabulce je seznam operátorů pomocí kterých je pak možné porovnávat hodnoty identifikátorů a celý výraz tak vyhodnotit. Například výraz `(-algo.Value == 'gabfa') || (-algo.Value == 'random')` uvedený u položky *parametru* *EnableCondition* povolí položku za předpokladu, že hodnota položky `-algo.Value` nabývá hodnot `random`, nebo `gabfa`.

<b>operátor</b>	<b>význam</b>
+	<i>sčítání</i>
-	<i>odečítání</i>
*	<i>násobení</i>
/	<i>dělení</i>
<	<i>menší</i>
>	<i>větší</i>
<=	<i>menší nebo rovno</i>
>=	<i>větší nebo rovno</i>
!	<i>negace</i>
!=	<i>nerovnost</i>
==	<i>rovnost</i>
&&	<i>logický součin</i>
	<i>logický součet</i>
(	<i>levá závorka</i>
)	<i>pravá závorka</i>

Syntaxe vyhodnocování podmínek je stejná jako v C#. Ve výrazech je dále možné používat závorky pro upřednostnění, nebo seskupení operací.

## 4. POJMY A ZKRATKY

**GUI aplikace** rozhraní mezi uživatelem a aplikací, pomocí kterého se aplikace ovládá

**Program** definice řádkové utility sestavená z *parametrů*, kterou lze editovat v aplikaci CLI editor

**Parametry** části *programu* ze kterých lze sestavit definici řádkové utility

**Vlastnosti parametru** výčet vlastností, které se mohou podle typu *parametru* lišit. Změnou těchto vlastností se nastavuje např. pozice *parametru*, ale i další atributy pro práci s *parametrem*

**Strom parametrů** stromová struktura *programu* se všemi *parametry*

**Property editor** mřížka s kompletním seznamem vlastností *parametru* pomocí které je možné tyto vlastnosti měnit

## 5. ZÁVĚR

Cílem této práce bylo vytvoření editoru pro definice řádkových utilit. Snahou bylo vytvoření aplikace, která by byla uživatelsky přívětivá. Aplikace však zároveň musí umožnit nastavení velkého množství parametrů a jejich vlastností tak, aby bylo možné definovat libovolnou řádkovou utilitu. Při vývoji postupně vzniklo několik verzí, které byly postupně zredukovány do této podoby. Na první pohled se může aplikace jevit jako velmi jednoduchá. Při jejím vzniku však bylo třeba nastudovat velké množství materiálů a zvláště ve společných částech bylo nutné spoustu věcí konzultovat. Myslím, že tato práce obohatila moje znalosti o další nové poznatky, které budou přínosem v mojí další praxi.



## Reference

- [1] MICHAL NĚMEC - řešitel části spouštěče
- [2] ANDREW TROELSEN C# a .NET 2.0 profesionálně, Zoner Press, Brno 2006, ISBN 80-86815-42-0
- [3] CHRIS SELLS C# a WinForms - programování formulářů Windows, Zoner Press, Brno 2005, ISBN 80-86815-25-0
- [4] Webový portál <http://msdn.microsoft.com>
- [5] Webový portál <http://forum.builder.cz>
- [6] Inno setup - instalace aplikace <http://www.jrsoftware.org>

## A. Struktura přiloženého CD

Přiložené CD obsahuje tyto adresáře (složky):

**bin** aplikaci připravenou ve spustitelné podobě

**src** zdrojové kódy aplikace

**doc** všechny dokumenty, které byly použity pro tvorbu této aplikace

**install** instalační balíček programu (podrobnosti k instalaci jsou uvedeny v uživatelské části této dokumentace)

## B. Seznam utilit

- **BFAtool** - program vedoucího práce
- **BIF** - program vedoucího práce
- **BMail** - posílání e-mailů přes SMTP server  
<http://www.beyondlogic.org>
- **File Compare** - porovnání souborů podle obsahu  
<http://www.gammadyne.com/cmdline.htm>
- **Backup9** - zálohování souborů  
<http://www.gammadyne.com/cmdline.htm>
- **Grep** - vyhledávání výrazů v textu  
<http://www.gnu.org/software/grep>
- **NetStat** - statistika TCP/IP spojení - utilita z operačního systému
- **Tracert** - analýza počítačové sítě - utilita z operačního systému
- **PDF to HTML** - převod souborů PDF do HTML  
<http://pdftohtml.sourceforge.net>
- **UnRAR** - dekomprimace souborů  
<http://unxutils.sourceforge.net>