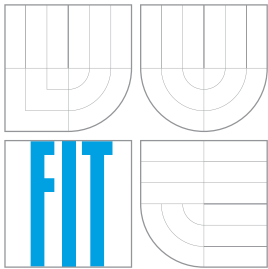


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

# **RASTROVÁ ANALÝZA PRO GIS NÁSTROJ ARCGIS**

SPATIAL ANALYSIS TOOLS FOR ARCGIS

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**Bc. RADIM HUPŠIL**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. MARTIN HRUBÝ, Ph.D.**

BRNO 2008

## Abstrakt

Práce se zabývá studiem geografického informačního systému ArcGIS. Zaměřuje se zejména na možnosti jeho rozšíření vlastními nadstavbami a způsob jejich programování. Dále jsou vysvětleny některé základní nástroje rastrové analýzy. Cílem této práce je návrh a realizace vlastního rozšíření ArcGISu, které bude poskytovat sadu nástrojů rastrové analýzy. Inspirací pro návrh je existující rozšíření – Spatial Analyst od firmy ESRI.

## Klíčová slova

ArcGIS, rastr, analýza, rastrová analýza, prostorová analýza, Spatial Analyst, rozšíření, nadstavba, vlastní rozšíření, vlastní nadstavba, programování, .NET, C#, reklasifikace, převzorkování, mapová algebra, sklon svahů, orientace svahů

## Abstract

This project is about studying geographic information system ArcGIS. It focuses on possibilities of extending ArcGIS by custom extensions and method of their programming. Furthermore some basic tools of raster analysis are illustrated. This project's main objective is to design and implement custom implementation of ArcGIS extension, which provides a set of tools for raster analysis. Design is inspired by an existing extension – Spatial Analyst developed by ESRI.

## Keywords

ArcGIS, raster, analyse, raster analysis, spatial analysis, Spatial Analyst, extension, custom extension, programming, .NET, C#, reclassification, reclass, resample, map algebra, slope, aspect

## Citace

Radim Hupšil: Rastrová analýza pro GIS nástroj ArcGIS, diplomová práce, Brno, FIT VUT v Brně, 2008

# Rastrová analýza pro GIS nástroj ArcGIS

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Martina Hrubého, Ph. D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Radim Hupšil  
13. května 2008

© Radim Hupšil, 2008.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Úvod</b>                                     | <b>3</b>  |
| 1.1      | Seznámení s problematikou                       | 3         |
| 1.2      | Struktura dokumentu                             | 4         |
| 1.3      | Návaznost na Semestrální projekt                | 4         |
| <b>2</b> | <b>Rastrová analýza</b>                         | <b>5</b>  |
| 2.1      | Základy GIS                                     | 5         |
| 2.2      | Rastry  | 6         |
| 2.3      | Uložení rastrů                                  | 8         |
| 2.4      | GeoTIFF   | 9         |
| 2.5      | Analýza rastrů                                  | 10        |
| 2.5.1    | Reklasifikace                                   | 10        |
| 2.5.2    | Mapová algebra                                  | 11        |
| 2.5.3    | Aspect - orientace svahů                        | 13        |
| 2.5.4    | Slope - sklon svahů                             | 14        |
| <b>3</b> | <b>ArcGIS a možnosti jeho rozšíření</b>         | <b>16</b> |
| 3.1      | Co je ArcGIS                                    | 16        |
| 3.2      | Možnosti rozšíření                              | 18        |
| 3.3      | Způsob programování uživatelských nadstaveb     | 18        |
| 3.3.1    | Nároky na programovací jazyk                    | 19        |
| 3.4      | Návod na programování nadstaveb pro ArcGIS      | 19        |
| 3.4.1    | Úvodní problémy při programování                | 19        |
| 3.4.2    | ArcGIS Software Developer Kit                   | 20        |
| 3.4.3    | Úvod do rozšiřování ArcObjects                  | 20        |
| 3.4.4    | První aplikace                                  | 22        |
| <b>4</b> | <b>Návrh vlastní nadstavby ArcGISu</b>          | <b>27</b> |
| 4.1      | Inspirace                                       | 27        |
| 4.1.1    | Přehled funkcí nadstavby ArcGIS Spatial Analyst | 27        |
| 4.2      | Návrh nadstavby                                 | 29        |
| 4.2.1    | Implementované nástroje                         | 29        |
| 4.2.2    | Postup implementace                             | 30        |
| 4.2.3    | Volba programovacího jazyka                     | 31        |
| 4.2.4    | Výkonnost vybraného jazyka                      | 31        |

|          |   |           |
|----------|---|-----------|
| <b>5</b> | <b>Implementace knihovny</b>                                | <b>33</b> |
| 5.1      | Panel nástrojů . . . . .                                    | 33        |
| 5.2      | Reklasifikace rastrů . . . . .                              | 33        |
|          | 5.2.1 Implementace reklasifikace rastru . . . . .           | 34        |
|          | 5.2.2 Problémy při implementaci . . . . .                   | 36        |
| 5.3      | Převzorkování rastru . . . . .                              | 36        |
| 5.4      | Mapová algebra . . . . .                                    | 38        |
|          | 5.4.1 Implementace mapové algebry . . . . .                 | 38        |
|          | 5.4.2 Problémy při implementaci . . . . .                   | 42        |
| 5.5      | Aspect - orientace svahů . . . . .                          | 42        |
|          | 5.5.1 Implementace výpočtu orientace svahů . . . . .        | 42        |
|          | 5.5.2 Problémy při implementaci . . . . .                   | 43        |
| 5.6      | Slope - sklon svahů . . . . .                               | 44        |
|          | 5.6.1 Implementace výpočtu sklonu svahů . . . . .           | 44        |
| 5.7      | Filtrování rastrů . . . . .                                 | 45        |
|          | 5.7.1 Implementace filtrování rastrů . . . . .              | 45        |
|          | 5.7.2 Problémy při implementaci . . . . .                   | 47        |
| <b>6</b> | <b>Testování knihovny</b>                                   | <b>48</b> |
| 6.1      | Porovnání výstupních rastrů a výkonnosti nástrojů . . . . . | 48        |
|          | 6.1.1 Reklasifikace rastru . . . . .                        | 49        |
|          | 6.1.2 Převzorkování rastru . . . . .                        | 49        |
|          | 6.1.3 Mapová algebra . . . . .                              | 49        |
|          | 6.1.4 Aspect - orientace svahů . . . . .                    | 50        |
|          | 6.1.5 Slope - sklon svahů . . . . .                         | 50        |
|          | 6.1.6 Filtrování rastrů . . . . .                           | 50        |
| 6.2      | Shrnutí . . . . .   | 51        |
| <b>7</b> | <b>Závěr</b>  | <b>52</b> |
| 7.1      | Zhodnocení výsledků . . . . .                               | 52        |
| 7.2      | Vlastní přínos . . . . .                                    | 53        |

# Kapitola 1

## Úvod

### 1.1 Seznámení s problematikou

Nástroj ArcGIS, vyvíjený společností ESRI, Inc., je kvalitní komerční geografický informační systém používaný mnoha organizacemi, zabývajícími se správou, analýzou a vizualizací komplexních informací vztahujících se k území. Mezi ně se řadí i naše fakulta, na které je v předmětu GIS (Geografické informační systémy) použit ArcGIS pro osvojení práce s vektorovou analýzou. Pro potřeby výuky rastrové analýzy je pak používán volně dostupný program Grass. Nelze říci, který je lepší nebo horší, oba mají velkou řadu příznivců.

Ovšem nemožnost použít při výuce rastrovou analýzu v ArcGISu neplyne z toho, že by ji tento nástroj nepodporoval. ArcGIS je jednoduše rozšiřitelný a existuje pro něj mnoho nadstaveb, které jsou ve většině případů bohužel placené. Rastrová analýza patří mezi ně. Z toho důvodu, že fakulta nevlastní licence potřebné k jejímu provozování, se zrodil nápad vytvořit vlastní implementaci takové knihovny. Ta by měla poskytnout zdarma některé základní algoritmy rastrové analýzy.

ArcGIS dovoluje programování a použití vlastních nadstaveb a skriptů. Skripty jsou určeny spíše pro jednoduché funkce a často používané sekvence funkcí. Jejich zápis je poměrně snadný a proto jich využívají i obyčejní uživatelé. Některé z nich jsou volně dostupné na různých fórech a stránkách zabývajících se ArcGISem. V mnoha případech se jedná pouze o zřetězení několika funkcí, což uživateli může mnohdy velmi usnadnit práci.

Vlastní nadstavby jsou pak určeny pro složitější, obsáhlejší funkce a ucelenější řešení. Většinou se nejedná o jednu funkci, jako v případě skriptů, ale o celou sadu nějak souvisejících funkcí. Takových nadstaveb poskytuje firma ESRI několik. Je však možné tyto nadstavby také vytvořit a naprogramovat. Pravděpodobně sice nebudou dosahovat takových výkonů, jako oficiální nadstavby, ale jejich výhoda spočívá v ceně.

Tato práce se zaměří na způsoby programování uživatelských nadstaveb a na některé algoritmy rastrové analýzy. Cílem je návrh a implementace vlastního zásuvného modulu, který bude dané algoritmy rastrové analýzy dodávat. Výstupem by pak měla být volně šiřitelná nadstavba ArcGISu dodávající algoritmy rastrové analýzy a také její zdrojové kódy, které bude možné dále upravovat, což by v budoucnu mělo vést ke zdokonalení a rozšíření knihovny o další funkce ze strany jiných osob.

Jako inspiraci lze využít oficiálně dodávané rastrové analýzy od firmy ESRI – Spatial Analyst. Popis této nadstavby je dostupný na internetu v publikaci [2]. Obsahuje více než 150 různých nástrojů pro rastrovou analýzu, jejich seznam je uveden např. v [11]. Tato nadstavba obsahuje velké množství funkcí, které pro základní výuku geografických informačních systémů na naší škole nepotřebujeme, proto implementovaná knihovna bude obsahovat

pouze ty nejpoužívanější funkce.

## 1.2 Struktura dokumentu

Před samotným návrhem knihovny se objeví kapitoly sloužící k vysvětlení základních pojmů a zasazení do problematiky. V první z nich (kap. 2) budou vysvětleny pojmy týkající se geografických informačních systémů. Jedná se zejména o pojmy rastr a rastrová analýza. Budou zde podrobně popsány některé základní funkce (algoritmy), které rastrová analýza používá. Tato kapitola slouží k zasazení do problematiky a základnímu pochopení prostorové analýzy, které je potřebné pro pochopení dalších částí práce.

Kapitola 3 se bude zabývat nástrojem ArcGIS, jeho základním popisem, možnostmi použití a zejména možnostmi jeho rozšíření uživatelskými nadstavbami. Budou přiblíženy způsoby, jakými je možné ArcGIS rozšířit a popsány jejich výhody a nevýhody. Programování nadstaveb pro ArcGIS bude věnována podkapitola 3.4, kde rozeberu programování nadstaveb obecně. Tento text pak může sloužit jako návod dalším programátorům nejen k rozšíření knihovny, která je předmětem této práce, ale i k implementaci dalších. Za tímto účelem tato práce vznikla.

Následovat bude kapitola 4, ve které bude rozebrán samotný návrh vlastní realizace nadstavby. V této kapitole je rozebrán obsah výsledné nadstavby a její předpokládaný vývoj.

Po kapitole s návrhem bude diskutována samotná implementace knihovny v kapitole 5. Jednotlivé funkce rastrové analýzy obsažené ve výsledné knihovně budou podrobněji popsány z pohledu programátora (z pohledu GIS byly popsány v kapitole 2). To by mělo usnadnit navázání dalších řešitelů na tento projekt.

Popis a výsledky testování knihovny v porovnání např. s nadstavbou Spatial Analyst budou obsaženy v kapitole 6.

Poslední kapitolou bude závěr obsahující zhodnocení výsledků. Zmíněn bude také vlastní přínos.

## 1.3 Návaznost na Semestrální projekt

Tato diplomová práce navazuje na Semestrální projekt, který byl řešen o jeden semestr dříve. V rámci Semestrálního projektu bylo prostudováno algoritmy rastrové analýzy a nástroje ArcGIS. V případě nástroje ArcGIS šlo o nastudování materiálů týkajících se zejména způsobu programování uživatelských nadstaveb. Tato témata jsou obsahem kapitol 2 a 3. Tyto kapitoly byly z velké části vypracovány již v Semestrálním projektu a do této práce byly převzaty a rozšířeny. V kapitole 2 byly doplněny informace o formátu GeoTIFF a podrobněji zpracována podkapitola popisující rastrové analýzy. V kapitole 3 byl oproti Semestrálnímu projektu přidán návod na programování vlastních nadstaveb.

Dalším bodem Semestrálního projektu byl návrh vlastní knihovny rozšiřující ArcGIS. Návrhem se zabývá kapitola 4 a byla převzata ze semestrálního projektu s mírnými úpravami.

## Kapitola 2

# Rastrová analýza

Před návrhem a implementací rastrové analýzy pro GIS nástroj ArcGIS je potřeba přiblížit také tyto samotné pojmy. V této kapitole budou vysvětleny základy geografických informačních systémů a jejich rozdělení. Blíže se pak budeme zabývat rastry a jejich analýzou. Nástrojem ArcGIS, jeho popisem, možnostmi použití a rozšíření se pak zabývá další kapitola.

### 2.1 Základy GIS

Jak uvádí [1], v geografických informačních systémech (GIS) se pro popis modelu reálného světa používají geoobjekty. To znamená, že se neukládají záznamy o fyzické realitě, ale pouze její model. Geoobjekty jsou prostorové objekty, většinou se u nich omezuje na tři dimenze – poloha, atributy a čas. Modely v geografických informačních systémech lze rozdělit do dvou skupin:

- **Modely založené na polích** – spojitě rozložená informace.
- **Objektové modely** – seznam diskretních objektů.

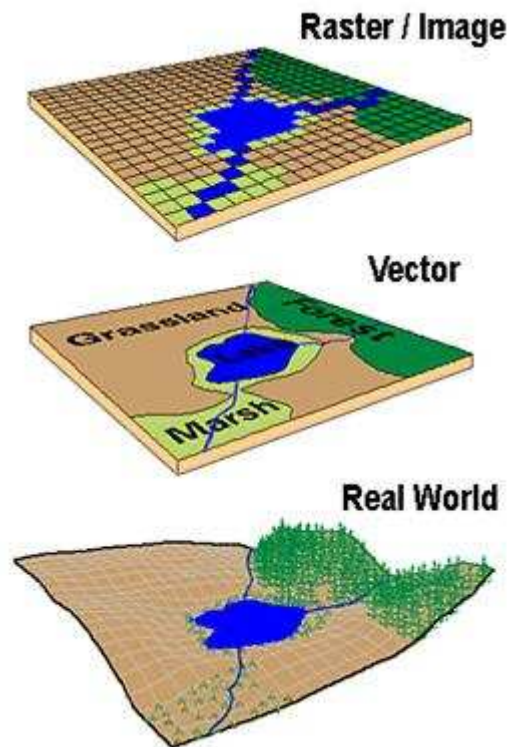
Toto rozdělení vede k použití rastrového a vektorového modelu. Klasickým přístupem k navrhování a vytváření topografických a tematických map v kartografii je vrstvý přístup. Jednotlivé vrstvy obsahují sady tematicky příbuzných dat. Složením několika vrstev vznikne celková mapa. Kombinováním vrstev pomocí analytických postupů lze odvodit nové geoobjekty. Tato vlastnost je v geografických informačních systémech považována za těžiště funkce a odlišuje je tím od ostatních informačních systémů.

Zda se bude vrstva zakládat na rastroch nebo vektorech je důležité rozhodnout již při modelování dané situace. Pro každý typ informací je vhodnější jeden nebo druhý:

- **Vektor** – výčet konečného množství ohraničených geoobjektů. Jejich atributy jsou ukládány do oddělené databáze. Použití například pro modelování cest, řek, vodovodu, umístění bankomatu apod.
- **Rastr** – popis spojitě rozložené informace v ohraničeném prostoru. Atribut je uložen v buňce rastru, rozsah jeho platnosti určují hranice buňky. Používá se např. pro výškový model, půdní pokryv, teploty apod.

Pro jednotlivé hodnoty v rastroch se používá několik různých termínů, např. buňka (anglicky cell), ploška nebo pixel (zejména v dokumentaci firmy ESRI). V této práci se o nich budu jednoduše vyjadřovat jako o buňkách.





Obrázek 2.1: Ukázka vektorového a rastrového modelu [16]

Mezi rastry a vektory lze provádět jisté převody, ale jejich možnosti jsou poměrně omezené. Vektorová analýza není předmětem této práce a proto se v jejich případě omezíme pouze na již řečené vlastnosti, což nám bude pro pochopení postačovat.

## 2.2 Rastry

Podle [1] je rastr systém pod pravým úhlem se protínajících čar, které ohraničují jednotlivé buňky. Je to jedna z možných definic dělení prostoru. Prostor lze dělit pravidelně (tvar buněk je přesně definovaný, např. trojúhelník, čtverec, šestiúhelník) a nepravidelně (různé tvary a velikosti buněk). V praxi se však používá téměř výhradně pravidelného dělení na čtverce se stejným rozlišením (velikostí). Z pohledu počítačů je to přirozené a tudíž efektivní rozložení. V některých případech se používají buňky obdélníkového tvaru, ale taktéž pravidelné. Pouze ve výjimečných případech se setkáme s nepravidelným dělením nebo s jinými než pravoúhlými tvary buněk.

V geografických informačních systémech je rastr uložením čísel, na rozdíl od některých jiných oborů (multimédia), kde se jedná o uložení barev. Rastr lze tedy popsat jednoduše jako matici čísel. Polohou buňky je míněna poloha jejího středu. Její obsah je ale po celé ploše buňky stejný, proto je nutné použít dostatečné rozlišení, aby byl model přesnější.

V té nejjednodušší a nejpoužívanější formě rastr sestává z matice buněk organizovaných do řádků a sloupců (mřížky). Každá buňka obsahuje hodnotu reprezentující informaci o veličině, kterou rastr popisuje (např. teplota, nadmořská výška nebo využití půdy). Rastry mohou být například digitální letecké snímky, skenované mapy, obrazy ze satelitu nebo jiné

digitálně uložené informace. Výhody uložení dat v rastru jsou popsány například v [18].

Data uložená v rastru reprezentují model skutečného světa. Rastrové mapové vrstvy je možné podle [10] dělit na:

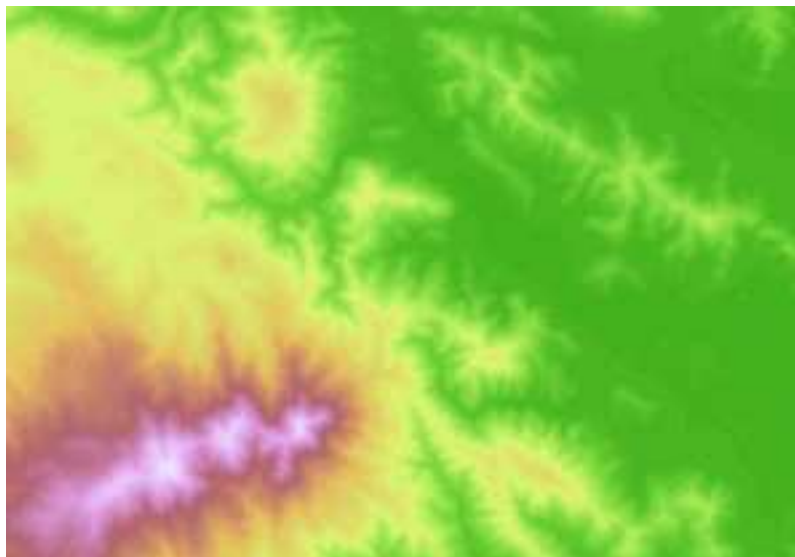
- **Rastrové vrstvy výčtového typu** – Každá buňka rastru obsahuje jistý kód, typicky celočíselný. Tento kód reprezentuje kategorii sledovaného jevu. Součástí rastrové vrstvy tohoto typu je překladová tabulka, která kódy interpretuje. Rastry výčtového typu se používají tam, kde má zkoumaný jev konečný počet hodnot (např. typ vegetace), nebo tam, kde lze spojitou veličinu rozdělit do konečného počtu kategorií (např. nízká, střední a vysoká hustota zalidnění). Tento typ rastru je zobrazen na obrázku 2.2.
- **Rastrové vrstvy hodnotového typu** – Každá buňka rastru nese informaci o diskretizované hodnotě spojitě veličiny, která může teoreticky nabývat nekonečného počtu hodnot. V praxi je samozřejmě omezena rozsahem a přesností použitého datového typu (integer, float). Takto reprezentované veličině se v prostředí GIS někdy říká prostorový proces. Příkladem prostorového procesu může být nadmořská výška, atmosférický tlak, teplota, apod. Na obrázku 2.3 je ukázka rastru, který reprezentuje nadmořskou výšku.



Obrázek 2.2: Ukázka rastrové vrstvy výčtového typu – využití půdy

Hodnoty buněk v rastru mohou nabývat celočíselných nebo desetinných hodnot, stejně tak hodnot kladných nebo záporných. Celočíselná data se používají k vyjádření diskretních dat (kategorií), desetinná čísla popisují ve většině případů data spojitá. Buňka může nabývat také speciální hodnoty označované jako NoData. Tato hodnota reprezentuje absenci dat pro tuto buňku. To může nastat v případě, že hodnota nemohla být zjištěna, nebo je pro dané téma nepotřebná.

Hodnota buňky vždy reprezentuje hodnotu naměřenou (nebo jinak získanou) v jejím středu. Z toho vyplývá, že čím menší jsou rozměry buňky, tím přesnější data budou v rastru zaznamenána. Velikosti reálně používaných rozměrů buněk se různí od centimetrů po kilometry. Je tomu tak proto, že pro různé potřeby jsou nutné různé přesnosti dat. Kilometrové buňky pak sice nejsou dokonale přesné, ale v určitém případě popisují zkoumaný jev dostatečně. Rozměry buněk mají velký vliv také na výslednou velikost souboru s daty. Proto



Obrázek 2.3: Ukázka rastrové vrstvy hodnotového typu – nadmořská výška

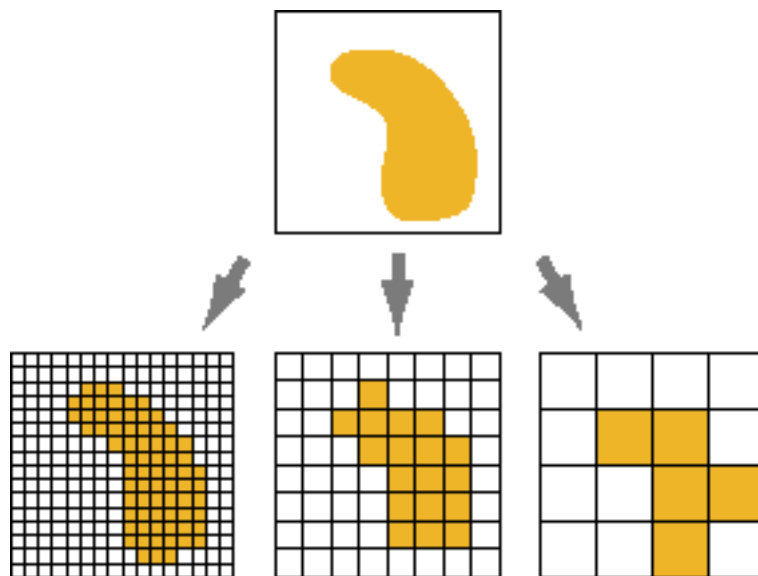
je vždy potřeba zvolit přiměřenou velikost, aby práce s rastrem byla rychlá, a zároveň výsledky dosahovaly co nejlepší přesnosti. Obrázek 2.4 převzatý z [18] znázorňuje vektorovou vrstvu (nahore) a odpovídající rastrové vyjádření s různými rozměry buněk. Buňky mohou nabývat jakýchkoliv rozměrů, obrázek ale pro jednoduchost demonstruje rozdíly na buňkách o rozměrech dvojnásobku (uprostřed) a čtyřnásobku (vpravo) velikosti buněk z rastru vlevo.

Matice hodnot je reprezentována v kartézském souřadném systému, takže umístění buněk v rastru je definováno indexem řádku a sloupce. V geografických informačních systémech se však používají jiné systémy souřadnic pro projekci dat v prostoru, např. UTM (Universal Transverse Mercator). V tom případě je nutné souřadnice v rastru přepočítávat na odpovídající souřadnice v systému, ve kterém je zobrazen a georeferencován. V programech náležících k systému ArcMAP mají rastry definovaný prostor (anglicky extent), který specifikuje horní, spodní, levou a pravou souřadnici rastru v používaném souřadném systému.

Rastr může mít jednu nebo více vrstev, ve kterých jsou data uložena. U vícevrstvých rastrů se jedná většinou o data popisující jednotlivé barevné spektrum (červená, zelená, modrá), které po sloučení dají výsledný rastr. Více vrstev v rastru se používá např. u snímků ze satelitů, které měří nejen viditelné spektrum, ale i infračervené nebo ultrafialové záření. Většina aplikací řešených v této práci se ale zabývá pouze jednovrstvými rastry, které jsou pro jejich potřeby nejčastěji používané.

## 2.3 Uložení rastrů

Možností, jak uložit rastry do souboru je nepřehledné množství. Program ArcGIS podporuje více než 30 formátů uložení a u některých formátů je navíc možné ukládat data v různých přesnostech a datových typech, takže celkový počet přesahuje 50 typů uložení rastrů. Jedná se o typy, které je možné v programech ArcGIS otevřít, ukládání do všech typů podporováno není. Základní podporované formáty pro výstup jsou již omezenější. Bez komprese a podobných vymožeností je možnost ukládat rastry do mřížky (soubor s hodnotami, navíc buňky musí být čtvercové), do formátu ERDAS IMAGINE nebo do formátu TIFF.



Obrázek 2.4: Ukázka rozdílů v rozměrech buňky v rastru

Ukládání rastrů do souborů je poměrně obsáhlé téma. Existuje mnoho různých způsobů, jak jednotlivé buňky ukládat, aby se posléze zvýšila výkonnost komprese, ale ArcGIS pracuje na takové úrovni abstrakce, že tyto detaily o rastroch programátor nemusí ani znát. ArcGIS s nimi vždy pracuje jako s maticemi hodnot, které jsou přístupné pomocí kartézských souřadnic. V jakém pořadí jsou jednotlivé buňky uloženy v souboru tudíž není důležité.

## 2.4 GeoTIFF

Zmíněný formát TIFF jsem si vybral jako výstupní formát pro své aplikace. Je samozřejmě možné ukládat i v ostatních formátech a nebo například ještě v dalších, jako je JPEG2000, které podporují kompresi apod. Export do těchto formátů však podporují všechny verze programů ArcGIS, a proto není třeba je dále řešit. Formát TIFF, zvaný také GeoTIFF, se používá velmi často nejen z důvodu přenositelnosti mezi různými systémy, ale i pro jeho vlastnosti. Umí ukládat celá i desetinná čísla v různých typech a volitelně lze použít i kompresi dat.

Formát GeoTIFF je založen na klasickém rastrovém formátu TIFF používaném pro obrazová data na počítačích. GeoTIFF rozšiřuje jeho vlastnosti o možnosti přidat georeference a další kartografické informace používané převážně v geografických informačních systémech. Jedná se o výsledek snažení asi 160 různých firem a organizací působících v této oblasti, včetně firmy ESRI. Jeho specifikaci lze nalézt v [12]. GeoTIFF se nesnaží nahradit jiné formáty ukládání rastrů, pouze přidává k existujícímu a často používanému formátu informace popisující polohu rastru v rámci zvoleného souřadného systému.

TIFF je formát založený na tzv. tagu (v českém překladu značka). Svou strukturou poskytuje možnosti pro ukládání doplňujících informací. Všechn obsah, organizaci dat, jejich datový typ, velikost i použitou kompresi lze popsat pomocí tagů. Těchto tagů je možno použít více než 65 000. První polovina je rezervována pro samotné informace o rastru, druhá polovina je použitelná pro informace zaregistrovaných firem, které si tak mohou ve formátu TIFF uchovávat vlastní informace. Jedním z požadavků GeoTIFFu je také to, aby

zůstal transparentní vůči obyčejným prohlížečům obrázků. GeoTIFF tak používá několik rezervovaných tagů k uložení informací o geografické poloze a použitém souřadném systému. Jednotný systém je definován ve specifikaci, a proto se o tomto formátu lze vyjádřit jako o platformově nezávislém. Všechny detaily jsou uvedeny ve zmíněné specifikaci.

Výhodou tohoto formátu je především zjednodušení práce samotných uživatelů geografických dat. Zavedením tohoto standardu odpadá nutnost pracného zadávání referenčních bodů a rozměrů buněk, v horším případě geometrických transformací pomocí vlíčovacích bodů.

Pro výhody a jednoduchost použití se tak tento formát stal výstupním formátem většiny mnou implementovaných funkcí, o kterých bude řeč v kapitole 5. Jak již bylo řečeno, ArcGIS podporuje konverzi (export) do ostatních formátů, což mohou využít uživatelé, kteří potřebují ukládat data v jiném formátu. Pro většinu v praxi používaných rastrů je formát GeoTIFF dostačující, což vypovídá o jeho robustnosti a použitelnosti. Proto se také stal jedním z nejpoužívanějších rastrových formátů v geografických informačních systémech.

V souvislosti s tímto formátem jsem při programování narazil na problém, který zmíním v kapitole 3.4.1, protože je poměrně důležitý. Nejedná se však o chybu ve formátu, ale ve zpracování programem ArcGIS, proto se tomuto tématu budu věnovat v příslušné kapitole.

## 2.5 Analýza rastrů

Po vysvětlení základních pojmů v geografických informačních systémech můžeme přistoupit k podrobnějšímu popisu analýzy rastrů. Analýza v geografických informačních systémech je podle [1] odvozování nových znalostí ze znalostí již existujících a uložených. Jde o experimenty s existujícím modelem za účelem zjištění nových skutečností.

Některé důležité analytické funkce týkající se rastrů:

- **Reklasifikace** – selekce a klasifikace buněk splňujících podmínku kladenou na jejich atributové hodnoty
- **Mapová algebra** – základ pro zpracování rastrových vrstev. Do této kategorie spadají veškeré početní operace nad rastry
- **Vzdálenostní analýzy** – zkoumání frikčních povrchů (rastr zde udává tzv. nákladovou vzdálenost) a různých vyjádření vzdáleností. Používá se například pro vyhledání optimální trasy
- **Analýza modelů terénu** – zkoumání výškových modelů. Jde o výpočty sklonů svahů, povodí, eroze, viditelnosti, proudění vzduchu apod.
- **Analýza obrazů** – interpretace, zpracování, filtrace a podobné operace pracující s obrazy například z dálkového průzkumu Země. Patří zde i georeferencování obrazu

Podrobnější popis některých zmíněných funkcí bude následovat v dalších podkapitolách. Jedná se především o ty funkce, které budou implementovány ve výsledné knihovně, jejíž návrh popisuje kapitola 4.

### 2.5.1 Reklasifikace

Jedná se o selekci a klasifikaci buněk, které splňují podmínku kladenou na jejich atributové hodnoty. Lze tím dosáhnout například sloučení původních kategorií buněk do kategorií nových. Reklasifikační funkce mění hodnoty buněk použitím různých metod.

Přiřazení nových hodnot buňkám se provádí na základě reklasifikační tabulky. Ta může obsahovat některé nebo všechny možné hodnoty buněk v rastru, případně může slučovat několik původních hodnot do jedné nové. Takto lze dosáhnout záměny konkrétních hodnot za jiné. Nejlépe tuto situaci vysvětlí obrázek 2.5, převzatý z nápovědy k programu ArcGIS z publikace [3].

Většinou se reklasifikace provádí nad celočíselnými rastry, u kterých má každé číslo svůj přesný význam, jako např. les, rybník, řeka, pole, zastavěné území (viz obrázek 2.2). Tyto rastry se nazývají klasifikované a význam jednotlivých čísel se popisuje metadaty, která většinou k rastru náleží (bez nich je klasifikovaný rastr v podstatě nepoužitelný). Pokud pak potřebujeme zobrazit například pouze jednu z kategorií, můžeme ji pomocí reklasifikace vynést do nového rastru a ostatní hodnoty budou nastaveny na hodnotu NoData. Tato hodnota označuje prázdné buňky, u kterých není určeno, jaké hodnoty nabývají a pro další výpočty se nepoužívají. Lze samozřejmě spojit více kategorií do jedné, takže výsledkem pak může být rastr znázorňující všechny vodní plochy.

Reklasifikaci lze použít také na neceločíselné rastry, jako jsou snímky DEM apod. (Tyto snímky se zpracovávají jinak, ale jako příklad postačují.) V tomto případě se novou hodnotou nahrazuje většinou celý interval čísel, což ve výsledku sníží přesnost, ale také objem dat rastru. Pro zmíněný výškový model ale takového použití nemá valný význam.

Výstupem bývá až na výjimky rastr celočíselný, v mé implementaci je to tak vždy, stejně jako v nadstavbě Spatial Analyst. Je to tak proto, že samotná reklasifikace je určena převážně k analýze celočíselných rastrů. Pokud nejsou do reklasifikační tabulky zadány všechny možné hodnoty, kterých buňky rastru nabývají, lze to řešit dvěma způsoby. Jeden je ten, že se původní buňky s nezadanými hodnotami nezmění, druhý je ten, že se nezadané hodnoty změň na hodnotu NoData.

## 2.5.2 Mapová algebra

Mapová algebra by se dala nazvat překrýváním vrstev spojeným s nějakou operací. Přesněji je mapová algebra popsána v [1]. Funguje výhradně nad rastry, pro vektory existují různé jiné funkce. Obecně ji lze popsat jako matematický výraz nad rastrovými mapami:

$$vysledek = expr(mapa_1, mapa_2, \dots, mapa_n),$$

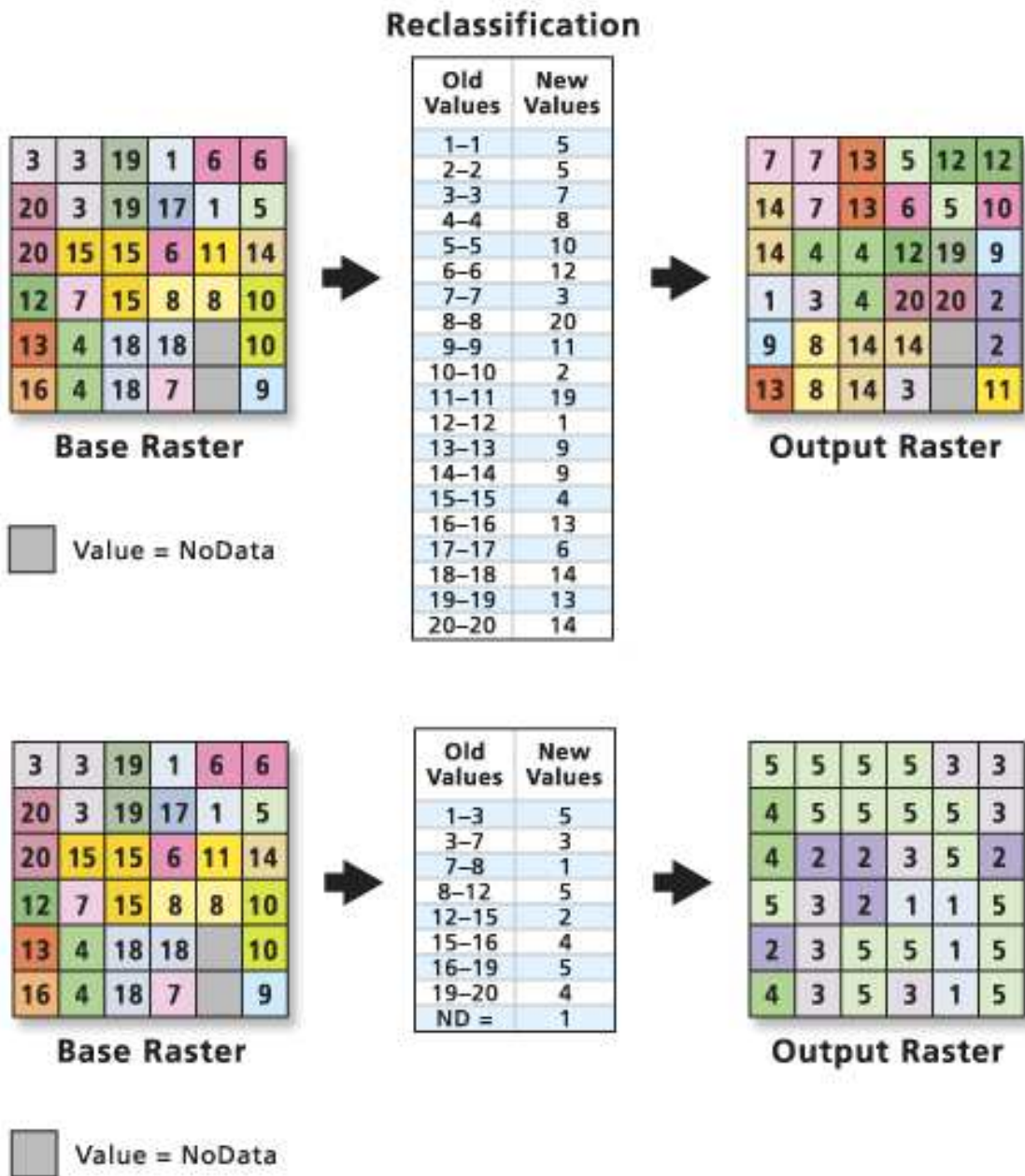
kde  $mapa_i$  pro  $i = 0..n$  jsou vstupní rastrové mapové vrstvy. Výraz se převádí (podobně jako všude jinde) na sérii jednodušších operací, většinou unárních nebo binárních. Operaci nad mapami lze pak brát jako operaci s maticemi. Rastrové mapy jsou totiž pouhým uložením čísel do buněk, stejně jako matice.

V mapové algebře se používají různé matematické operace:

- **Aritmetické** – např. +, −, \*, /...
- **Boolovské** – např. AND, OR, XOR, NOT...
- **Relační** – např. <, >, <=...
- **Logické operace, bitové posuvy apod.**

Protože k rastrům lze přistupovat jako k maticím čísel, existují pro ně podobné způsoby propojení dvou či více rastrů v jedné operaci. Zavádí se proto operace:

- **Lokální** – pracuje se nad hodnotou konkrétní buňky



Obrázek 2.5: Ukázka reklasifikace rastru

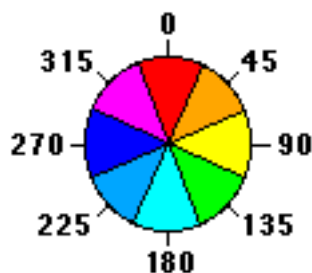


- **Fokální** – pracuje se nad okolím buňky
- **Zonální** – vykonávají se na specifické oblasti mapové vrstvy
- **Globální** – týkají se všech buněk rastru

Nejčastěji používané jsou binární lokální operace. Názorný příklad algoritmu je vyobrazen ve zmíněné publikaci, ale je opravdu velmi zjednodušený. Mapová algebra z tohoto pohledu vypadá poměrně jednoduše, ale bohužel je nutné počítat s dalšími problémy, jako např. neúplné překrytí dvou rastrových vrstev, rozdílné velikosti buněk apod. To vše je potřeba vyřešit před tím, než se přistoupí k samotnému výpočtu matematické operace. Dále se tímto tématem zabývám v části s popisem implementace v kapitole 5.4.

### 2.5.3 Aspect - orientace svahů

Výpočet orientace svahů je jednou z několika analýz modelů terénu. Výsledkem je úhel (azimut) určující směr, kterým je svah orientován (viz obrázek 2.6, převzatý z [3]). Tento směr je udáván ve stupních a funguje podobně jako kompas. Dalším filtrováním a zpracováním lze získat například linie hřebenů a údolí. To dále vede k výpočtům povodí, řečišť apod. Také je možno vypočítat z orientace svahů viditelnost mezi dvěma body nebo osvětlení terénu.



Obrázek 2.6: Směr orientace svahů

Výpočet orientace svahů se zakládá na okolí bodů. Nejedná se tedy už o lokální funkci, ale o funkci fokální. Počítá se nejen z hodnoty samotné buňky, ale i z jejího přímého okolí, tzn. v pravoúhlé síti buněk z okolních osmi buněk. Jedná se tedy o výpočet nad maticí o velikosti 3x3. Výpočet není nijak zvláště náročný, ale jeho implementace se dá provést několika různými způsoby.

Jedním ze způsobů výpočtu je procházení rastrem po buňkách, podobně jako v mapové algebře. Prochází se sice po buňkách, ale počítá se i s jejich okolím, a to podle následujících vzorců (i s obrázkem 2.7 převzato z [3]):

$$\frac{\Delta z}{\Delta x} = \frac{(c + 2f + i) - (a + 2d + g)}{8} \quad (2.1)$$

$$\frac{\Delta z}{\Delta y} = \frac{(g + 2h + i) - (a + 2b + c)}{8} \quad (2.2)$$

$$aspect = 180 + 180/\Pi * atan\left(\frac{\Delta z}{\Delta y}, \frac{\Delta z}{\Delta x}\right) \quad (2.3)$$

Vzorec 2.3 byl oproti udávanému zdroji upraven, ale jeho výsledek je stejný. Jenom se tak zjednodušil jeho zápis.



|   |   |   |
|---|---|---|
| a | b | c |
| d | e | f |
| g | h | i |

Obrázek 2.7: Ilustrace okna (matice) nad kterým se provádí výpočet

Pro výpočet orientace svahů lze však použít ještě jiný přístup. Je založen na stejném principu a vzorec 2.3 v něm také figuruje. Jde o to, že hodnoty okolních buněk není nutné počítat ručně, ale na vstupní rastr použijeme dva speciální filtry, a z jejich výsledků vypočteme stejnou výslednou orientaci svahů pomocí vzorce 2.3. Použité filtry a způsob výpočtu jsou vysvětleny blíže v kapitole 5.5, protože tento způsob používám ve své implementaci. Dále je řešení vysvětleno například v [13]. Ve stejné publikaci je vysvětlen také princip výpočtu sklonu svahů, který je velmi podobný.

#### 2.5.4 Slope - sklon svahů

Sklon svahů patří, stejně jako orientace svahů, mezi analýzy modelů terénu. Jeho výsledkem je číslo udávající poměr převýšení k určité vzdálenosti, neboli míra změny souřadnice  $z$  v rastru (například výškový model).

Pro každou buňku se určí sklon jako maximální míra změny hodnot mezi touto buňkou a jejími sousedy. Je možné počítat buď přímé sousedy (pouze 4), nebo i sousedy diagonální (celkem 8). Ve většině případů se počítá se všemi osmi sousedními buňkami, protože tato metoda dává přesnější výsledky. Zjednodušeně se počítá velikost průměrného převýšení sousedních buněk vůči buňce aktuální. Podobně jako orientace svahů se sklon počítá z matice 3x3, což již vyplynulo z předchozích úvah. Čím menší je hodnota sklonu, tím rovnější je terén, který tato hodnota popisuje. Čím je hodnota vyšší, tím prudší svahy se v daném terénu vyskytují.

Protože se tato funkce nepočítá lokálně ale fokálně, je nutné řešit počítání u okrajů rastru. To lze řešit buď tak, že hodnota stoupání na okraji se vynechá úplně, nebo se vypočte průměr tak, že hodnota buněk, které neznáme, se nastaví na stejnou hodnotu, jako má buňka ve středu matice 3x3. Obě metody se chovají podivně a na okrajích rastrů vznikají hodnoty, které neodpovídají realitě, což je při pohledu na rastr okamžitě vidět (obarvený rastr jasně ukáže, že tyto hodnoty se velmi liší od vedlejších hodnot a vzniká tak viditelný okraj).

Sklon svahů může být počítán ve dvou různých typech jednotek – stupně nebo procenta. Sklon v procentech si lze představit jako poměr převýšení a vzdálenosti vynásobený stem. Například sklon při převýšení 5 metrů na vzdálenosti 5 metrů je  $slope[\%] = 5/5 * 100 = 100\%$ . Na první pohled je jasné, že v případě úhlu je to  $45^\circ$ . Vzdálenost, nad kterou se sklon počítá, se v angličtině označuje jako *run*, převýšení jako *rise*. Vzorce pro výpočet vypadají následovně:

$$slope[\%] = \frac{rise}{run} * 100 \quad (2.4)$$

$$slope[^\circ] = atan\left(\frac{rise}{run}\right) \quad (2.5)$$

Z uvedeného příkladu vyplývá, že pro úhel  $0^\circ$  je sklon zároveň  $0\%$ . V opačném případě, kdy sklon je  $90^\circ$ , je velikost sklonu v procentech nekonečná.

Algoritmus použitý při výpočtu sklonu v matici je ale trochu odlišný, počítá totiž s osmi hodnotami, ne pouze s jednou. Tento výpočet prohíhá velmi podobně jako výpočet orientace. Vzorce pro  $\frac{\Delta z}{\Delta x}$  a  $\frac{\Delta z}{\Delta y}$  zůstávají stejné jako v předchozím případě, jedná se tedy o vzorce 2.1 a 2.2. Vzorce pro výpočet sklonu ve stupních a v procentech vypadají takto:

$$slope[\%] = \sqrt[2]{\left(\frac{\Delta z}{\Delta x}\right)^2 + \left(\frac{\Delta z}{\Delta y}\right)^2} * 100 \quad (2.6)$$

$$slope[^\circ] = atan\left(\sqrt[2]{\left(\frac{\Delta z}{\Delta x}\right)^2 + \left(\frac{\Delta z}{\Delta y}\right)^2}\right) * 180/\Pi \quad (2.7)$$

Stejně jako při výpočtu orientace svahů lze i zde uplatnit druhý přístup výpočtu, kdy nejprve ze vstupního rastru pomocí filtrů vytvoříme dva další, které odpovídají výpočtu rovnic 2.1 a 2.2. Tyto dva rastry lze poté zpracovat pomocí lokální funkce, která vypočte výsledný sklon podle vzorce 2.6 nebo 2.7, v závislosti na požadovaném typu výstupních hodnot. Použité filtry a způsob výpočtu jsou vysvětleny blíže v kapitole 5.5. Ta se sice týká výpočtu orientace svahů a ne jejich sklonu, ale tyto výpočty mají stejný základ, a proto není třeba zmiňovat se o stejných filtrech dvakrát.

## Kapitola 3

# ArcGIS a možnosti jeho rozšíření

### 3.1 Co je ArcGIS

ArcGIS je sada několika softwarových produktů pro vytvoření kompletního geografického informačního systému, jak je uvedeno v [5]. Výrobcem a dodavatelem je firma ESRI, která je dnes největším výrobcem software pro GIS. Celý ArcGIS je rozdělen do několika menších podskupin:

- **ArcGIS Desktop** – kompletní software pro desktopový GIS. Tato část bude rozebrána dále, protože právě o ni v této práci jde.
- **ArcGIS Engine** – začlenitelný GIS. Poskytuje soubor knihoven, které lze použít pro programování vlastních GIS aplikací nebo rozšiřování stávajících informačních systémů o podporu GIS nástrojů.
- **ArcGIS Server** – GIS na serveru. Pokrývá většinu serverových služeb potřebných pro GIS, ať už databázové služby, nebo samotné aplikační servery, které poskytují plnou funkcionalitu GIS pro intranet i internet.
- **ArcGIS Mobile** – mobilní GIS. Poskytuje vylepšenou podporu mobilních technologií, jako jsou PDA, pro sběr dat v terénu.

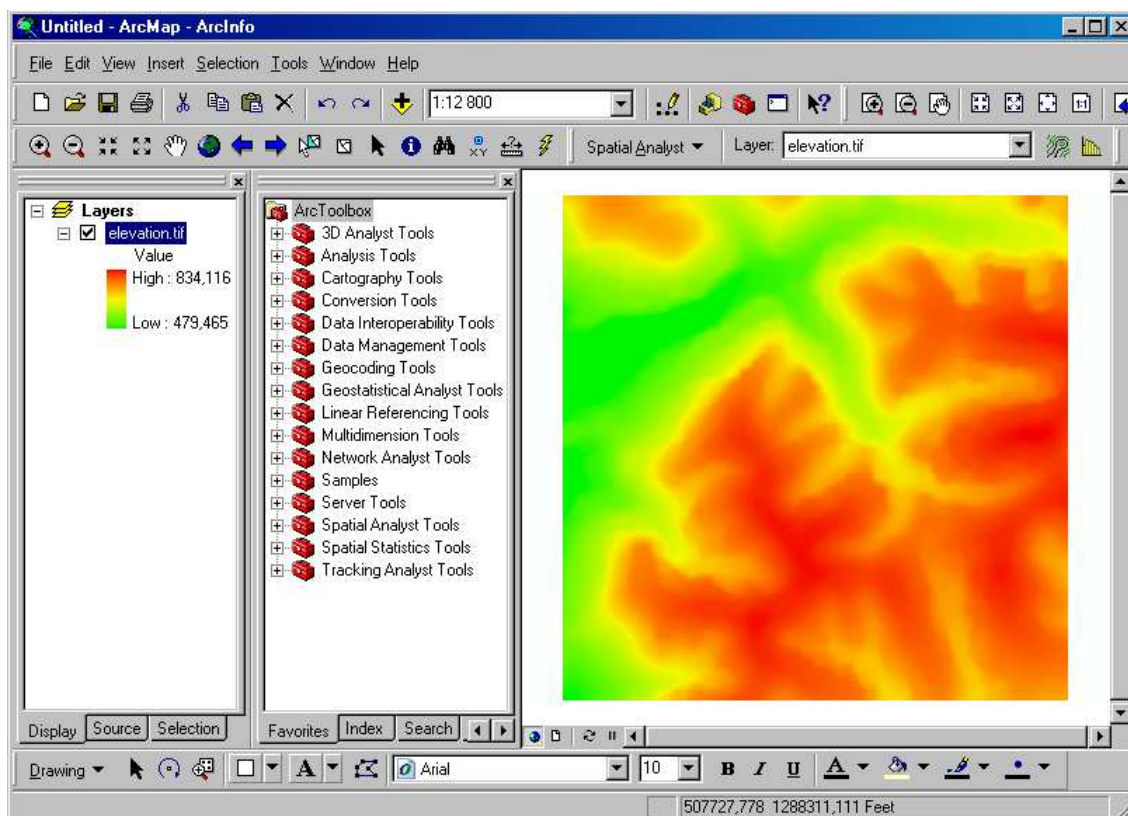
ArcGIS je založen na společné knihovně sdílených softwarových komponent *ArcObjects*, což výrazně usnadňuje jeho další vývoj a rozšiřování. Jak již bylo zmíněno, zde bude předmětem pouze část ArcGIS Desktop. Tato část se sama ještě dělí do tří podskupin (úrovní), které jsou rozděleny podle stupně funkcionality. Každá organizace si tak může vybrat právě ten produkt, který dostačuje potřebné úrovni zpracování dat.

- **ArcView** – poskytuje rozsáhlé nástroje pro tvorbu map, získávání informací z map a jednoduché nástroje pro editaci a prostorové operace (pro tuto verzi vlastní škola licence)
- **ArcEditor** – má stejné funkce jako ArcView a navíc dodává rozšířené editační možnosti pro shapefile a geodatabáze
- **ArcInfo** – rozšiřuje funkcionalitu obou předchozích produktů o rozšířené prostorové operace

Všechny tři produkty mají jednotnou architekturu, uživatelé tedy mohou sdílet výsledky své práce s ostatními i mezi různými produkty. Další výhodou pro uživatele je jednotný vzhled uživatelského prostředí. Možnosti přizpůsobení vzhledu jsou také velmi dobře propracovány, což uživatel jistě ocení.

Ve všech úrovních jsou obsaženy následující aplikace:

- **ArcMap** – centrální aplikace, která poskytuje kompletní funkcionalitu pro tvorbu map (ukázka na obrázku 3.1)
- **ArcCatalog** – aplikace vhodná pro organizaci, vyhledávání a využití GIS dat, stejně jako pro tvorbu dokumentace geografických dat pomocí metadat odpovídajících standardům
- **ArcToolbox** – aplikace ArcToolbox obsahuje kompletní sadu funkcí pro zpracování prostorových dat. Každá úroveň produktu obsahuje různý počet nástrojů (ArcView obsahuje v aplikaci ArcToolbox přes 80 nástrojů, ArcEditor jich nabízí více než 90 a ArcInfo poskytuje přibližně 250 nástrojů pro zpracování prostorových dat).
- **ModelBuilder** – poskytuje grafické modelovací prostředí pro návrh a implementaci modelů zpracování prostorových dat



Obrázek 3.1: Ukázka programu ArcMap

Funkcionalita těchto produktů je rozšiřitelná nejrůznějšími nadstavbami pro ArcGIS Desktop, které vyvíjí nejen firma ESRI, ale také jiné organizace. Vzhledem k tomu, že celý

nástroj je postaven na společné knihovně *ArcObjects*, mohou si uživatelé vyvinout vlastní nadstavby. Těm bude věnována další podkapitola. Všechny nadstavby je samozřejmě možné využívat ve všech třech zmíněných úrovních ArcGIS Desktop: ArcView, ArcEditor a ArcInfo.

## 3.2 Možnosti rozšíření

ArcGIS je možné rozšiřovat pomocí přídatných rozšíření (extensions), v češtině také zvaných například jako balíčky nebo nadstavby. Některé menší a ukázkové rozšiřující pomůcky lze nalézt volně ke stažení na internetu, ale oficiální balíčky poskytované firmou ESRI jsou placené. Ceny jednotlivých rozšíření závisí jak na využitelnosti, tak na objemu dodaných funkcí. Rozhodně ale nelze říci, že se jedná o zanedbatelné částky.

Informace v tomto, ale i v ostatních geografických informačních systémech, jsou rozděleny na dvě skupiny – data rastrová a data vektorová. Pro běžnou práci s vektorovými daty v ArcView není potřeba doinstalovat žádné nadstavby. Rastrovou analýzu ovšem základní instalace neposkytuje, takže je nutné ji dokoupit. Druhou možností, jak takové rozšíření získat, je jeho naprogramování.

Aby bylo možné rastrovou analýzu používat pro účely předmětu GIS, není tedy potřeba kupovat nadstavbu Spatial Analyst, což je rastrová analýza dodávaná firmou ESRI. Je možné vytvořit její zdarma dostupnou verzi, která může poskytovat potřebné funkce.

Existují dvě varianty, jak ArcGIS rozšířit (viz [9]):

- **Uživatelské skripty (makra)** – pro rychlý zápis jednoduchých rozšíření přímo v prostředí ArcGIS. Jako programovací jazyk je použit VBA (Visual Basic for Applications). Tento způsob je podpořen jednak tím, že je podrobně popsán v nápovědě programu, jednak velkým množstvím ukázkových příkladů na internetu. Jednoduchost spočívá také v možnosti kopírování a vkládání skriptu, takže uživatel vlastně nemusí programování rozumět a pokud skript někde sežene, může jej okamžitě začít používat.

Tento způsob je určen pro jednoduché programy, např. opakování jednoduchých sekvencí, které by uživatel musel stále zadávat ručně. Pro vytváření složitějších rozšíření a grafického uživatelského rozhraní je vhodné použít následujícího způsobu.

- **Uživatelské nadstavby (rozšíření, extensions)** – lze vytvářet libovolné rozšíření od jednoduchých komponent po komplexní řešení celého okruhu souvisejících rozšíření. Po vytvoření nadstavby je snadné sdílet ji s ostatními uživateli, stačí přidat příslušnou ActiveX DLL knihovnu do dané aplikace. Po jejím přidání již lze její funkce používat stejně, jako kdyby byly do programu vestavěny.

Této možnosti rozšíření se týká celá tato práce a proto o ní bude řeč ještě v dalších podkapitolách, kde bude přiblížen způsob, jakým lze rozšíření programovat.

## 3.3 Způsob programování uživatelských nadstaveb

Ihned na začátku tohoto tématu je potřeba upozornit, že je velmi obsáhlé a vydalo by na celou knihu. Zde však není prostor pro popsání celé problematiky programování uživatelských rozšíření a proto budou uvedeny jen základní informace. Podrobnější informace budou vysvětleny v následující podkapitole, která se tímto tématem blíže zabývá. Celou dokumentaci vytvořenou za tímto účelem lze nalézt v [9] v sekci Extending ArcObjects, odkud také tato kapitola čerpá (bohužel se tato publikace zabývá především programováním ve Visual C++ a Visual Basic 6).

Na uvedeném odkazu jsou podrobně vysvětleny příklady ukazující možnosti vytváření rozšíření na jednoduchých aplikacích. Důležitý je také manuál popisující jednotlivé objekty v *ArcObjects*, což je pro programátora nepostradatelná pomůcka. Je sice poměrně nepřehledný, ale obsahuje pravděpodobně vše potřebné. Bohužel popis jednotlivých komponent není nijak zvlášť obsáhlý, často se jedná spíše pouze o seznam parametrů a metod.

### 3.3.1 Nároky na programovací jazyk

Uživatelské nadstavby a nástroje mohou uživatelé vyvíjet ve standardních programovacích jazycích pro Microsoft Windows, jako je VBA (Visual Basic for Applications), .NET, Java a Visual C++. Jedinou nutnou podmínkou je podpora programovacího jazyka pro Microsoft Component Object Model (COM), na kterém je naprogramována celá knihovna *ArcObjects* od firmy ESRI. V COM je veškerá komunikace mezi klientem a serverem zajištěna pomocí rozhraní. Má to své výhody a samozřejmě i nevýhody. Výhodou je, že není potřeba zasahovat do zdrojových kódů původního systému a přesto je možné programovat až po nejnižší možnou úroveň. Nevýhodou je pak například nutnost registrovat COM objekty v dané aplikaci.

## 3.4 Návod na programování nadstaveb pro ArcGIS

Jedním z důležitějších bodů v této práci je návod k programování komponent pro nástroj ArcGIS. Zde se objeví tzv. tutorial, který bude pomáhat dalším programátorům začít s vývojem vlastních rozšíření, případně těm, kteří budou tato rozšíření dále upravovat. Protože mám vlastní zkušenosti pouze s programováním rozšíření v jazyce C#, nebudu se o ostatních jazycích zmiňovat. Ve většině případů ale není rozdíl až tak velký a použití je podobné.

### 3.4.1 Úvodní problémy při programování

V průběhu vývoje jsem narazil na mnoho problémů, u některých jsem si vůbec nevěděl rady a řešení jsem nenašel ani na internetu. Jejich řešení nakonec nebylo až tak obtížné, jak by se zdálo. Nejhorší je na takových problémech to, že programátor, který dané prostředí nezná, nemůže vědět, čím je problém způsoben. Aby byli další programátoři ušetřeni podobných zbytečných problémů, považuji za důležité uvést jejich řešení na začátku návodu.

Přestože jsem postupoval podle návodů uvedených přímo na stránkách firmy ESRI určených pro dokumentaci, podporu a komunikaci s programátory, některé programové konstrukce z neznámých důvodů nefungovaly tak, jak by měly. Nebyl jsem si ale jistý, jestli je to chyba systému, že výsledky jsou jiné, než bych očekával. Protože jsem nikde o takových problémech nenašel ani zmínku, snažil jsem se hledat jiná řešení.

Uvedu zde jeden příklad z praxe, na jehož konci se zmíním i o již slibovaném jednoduchém řešení. Jako první funkci jsem implementoval reklasifikaci rastru. Než jsem se k ní dopracoval, zkoušel jsem různé příklady a zjišťoval, jak co funguje. Protože většina příkladů ukládá do formátu TIFF, i já jsem si na tento formát přivykl. Znepokojovalo mě ale to, že při uložení rastru do souboru se neukládaly georeference a další detaily, jako souřadný systém, jednotky vzdálenosti apod. Tento problém jsem nedokázal vyřešit velmi dlouho, a proto jsem pokračoval v implementaci dalších funkcí, kde jsem ovšem stále narážel na stejný problém.

Na řešení jsem přišel, dalo by se říci, náhodou. Od nainstalování opravného balíčku Service Pack 3 fungovalo vše tak, jak mělo. Opravný balíček jsem instaloval shodou okolností

kvůli úplně jinému problému, který měl tento balíček opravovat. Chyba byla opravdu v systému a ne na straně programátora. Proto uvádím tuto jinak nepříliš důležitou podkapitulu na začátku návodu na programování.

Z posledních několika vět plyne jedno ponaučení, které je dobré znát ihned na začátku: **Je nutné mít vždy nainstalované co nejnovější verze opravných balíčků**, které Vám, jako programátorům, mohou nejednou pomoci vyhnout se nepříjemným problémům.

### 3.4.2 ArcGIS Software Developer Kit

Po úvodním upozornění na možné komplikace při programování se zmíním o nástrojích, které nabízí sama firma ESRI programátorům, kteří chtějí vyvíjet vlastní nástroje. Jedná se především o ArcGIS Software Developer Kit, který v mnoha ohledech zjednodušuje práci programátora. Jeho instalací získá nejen soubor nástrojů a utilit, ale také mnoho příkladů v programovacím jazyce, který si při instalaci vybere. Podporovány jsou nejpoužívanější programovací jazyky, jako již zmíněné VBA6, C#, Visual C++ nebo Java. Výhodou je také integrace do vývojového prostředí, za pomoci které se programátorovi usnadní spousta práce, protože jsou obsaženy šablony pro jednotlivé komponenty a nástroje pro ArcGIS.

Další důležitou pomůckou je dokumentace, ve které je obsažen popis všech objektů. Dokumentaci je také možno integrovat přímo do vývojového prostředí, pokud je podporováno (např. Microsoft Visual Studio .NET 2005). Pro začátečníky (jako jsem byl i já) doporučuji využít právě zmíněných výhod integrace do vývojového prostředí Microsoft Visual Studio .NET 2005 a jako programovací jazyk si v tomto případě vybrat C#.

Pro ostatní jazyky je pravděpodobně možné také integrovat dokumentaci, šablony i další pomůcky do některého vývojového prostředí. Většina příkladů na internetu je sice pro jazyk VBA6, ale pokročilejší programátor zvládne tyto příklady přepracovat do C# bez větších problémů a odměnou mu bude minimálně přehlednější kód, než jaký nabízí VBA6.

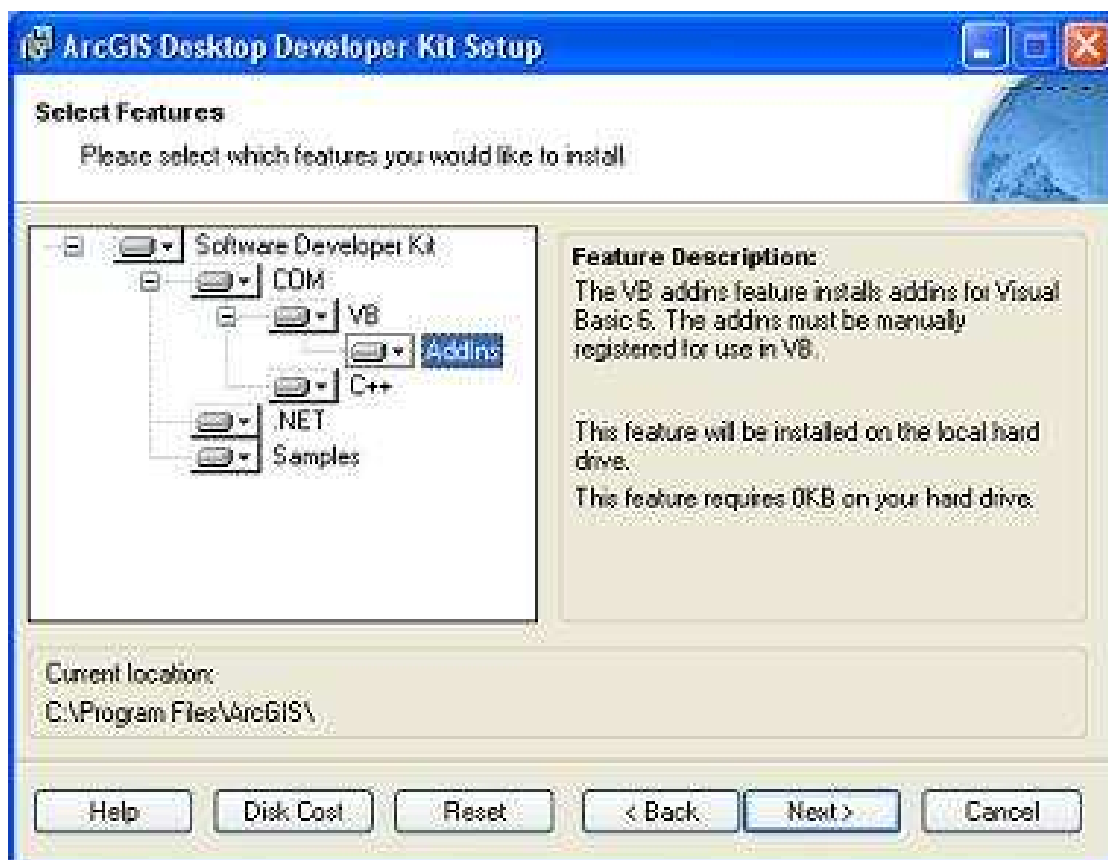
ArcGIS Software Developer Kit (dále jen SDK) je obsažen na instalačních médiích s programem ArcGIS. Standardně se ale SDK neinstaluje, proto je potřeba jej doinstalovat, nebo přímo při instalaci nástroje ArcGIS vybrat i instalaci SDK pro požadovaný programovací jazyk. Pokud chcete integrovat funkce a pomůcky SDK do vývojového prostředí, je nutné nainstalovat nejprve toto prostředí. Průběh instalace znázorňuje obrázek 3.2.

Je vhodné se také ujistit, zda Vámi vybraná platforma splňuje minimální hardwarové požadavky. Uvedené minimální požadavky jsou opravdu minimální a je lepší mít hardware alespoň na úrovni požadavků doporučených. ArcGIS je velmi náchylný k nestabilitě, pokud nejsou tyto požadavky dodrženy, i pokud je spouštěn zároveň s dalšími výkonově náročnými aplikacemi.

### 3.4.3 Úvod do rozšiřování ArcObjects

Velmi častým pojmem v této kapitole bude *ArcObjects*. Jedná se o vývojovou platformu pro rodinu programů ArcGIS. Protože staví na COM technologii, je možné ji rozšiřovat pomocí programování COM komponent. Zde uvedu pouze základy programování nástrojů a rozšíření pro ArcGIS. V případě, že se budete tomuto tématu věnovat hlouběji, budete muset dříve či později využít dokumentace *ArcObjects* (viz např. [9]) pro zjištění podrobnějších detailů.

Rozšiřování *ArcObjects* je založeno na implementaci některého z rozhraní, které poskytuje. Samozřejmostí je možnost použití funkcí, které jsou v nich obsaženy. Po instalaci systému ArcGIS jsou do systému přidány nové knihovny objektů, které začínají písmeny ESRI. Stejně jako v jiných případech je nutné importovat tyto knihovny do zdrojového



Obrázek 3.2: Ukázka průběhu instalace SDK

kódu programu, aby bylo možné pracovat s danými objekty. Objekty jsou rozděleny do několika knihoven podle příslušnosti k danému tématu (ne vždy je jednoduché odvodit, kde dané objekty najít). Jak již bylo zmíněno, vše funguje pomocí rozhraní, a proto o nich bude řeč v souvislosti s implementací velmi často. Rozhraní v *ArcObjects* jsou označeny počátečním písmenem I (z anglického interface). Tak lze jednoduše rozeznat rozhraní od ostatních objektů.

Před samotným programováním je vhodné se rozhodnout, o jak složité rozšíření nebo nástroj půjde a jestli bude obsahovat jeden nebo několik příkazů, tlačítek, panelů nástrojů, nebo celých oken. Vždy však existuje možnost zakomponovat již existující jednoduchý nástroj například do panelu nástrojů. Je pak nutné pouze přidat implementaci tohoto panelu, do kterého se jednoduše daný nástroj přidá. To může být výhodou, pokud na začátku projektu není znám jeho konečný rozsah.

Následuje výběr vhodného rozhraní pro rozšíření. Je několik možností, ze kterých si lze vybrat. Například pro naprogramování jednoduchého příkazu postačuje pouze implementace rozhraní *ICommand*. Samozřejmě je nutné zakomponovat vlastní funkční kód, který se po zavolání z hlavní aplikace provede.

Nástroj lze vytvořit podobně, je ale potřeba implementovat navíc rozhraní *ITool*. Tak je možné vytvořit například nástroj pro změnu přiblížení (lupa) apod. Podobných rozhraní je několik, pro různé úrovně složitosti rozšíření nebo pro různé typy komponent. Kromě jednoduchých příkazů a nástrojů lze vytvářet také menu (rozšířením třídy *BaseMenu*), panely



nástrojů (rozšířením třídy `BaseToolbar`) a v neposlední řadě také celé komplexní nadstavby (implementací rozhraní `IExtension`). Ne vždy je nutné přímo implementovat daná rozhraní, v některých případech stačí využít dědičnosti a použít již připravené třídy, které implementaci usnadní tím, že daná rozhraní implementují za nás. Tím naši práci velmi zjednoduší a urychlí.

Klíčem k vytváření vlastních komponent je dokonalé pochopení struktury *ArcObjects*. Podporována jsou nejen rozšíření v podobě příkazů nebo nástrojů, existuje možnost doplnit do ArcGISu podporu vlastních formátů dat, tvorbu vlastních rozšířených vrstev apod.

*ArcObjects* obsahují mnoho různých funkcí pro všechny možné oblasti použité v ArcGIS, například pro práci s vektory, uživatelským rozhraním, vrstvami atd. Práce s rastry je v *ArcObjects* zabudována také, to znamená, že máme k dispozici různé funkce pro práci s nimi. Samozřejmě se nejedná o funkce rastrové analýzy, ale o funkce pro přístup k datům rastru, čtení a zápis do rastru, vytváření nových rastrů, jejich ukládání a podobně. Jejich použití je názorně vysvětleno v publikacích [8] nebo [7]. Seznam všech takových funkcí lze nalézt v dokumentaci i s popisem očekávaných parametrů a návratových typů. Bohužel je nutné často hledat pomocí fulltextových vyhledávačů, protože pokud nevíme, že daná funkce vůbec existuje, tak ji jen těžko můžeme hledat v dokumentaci podle jména. Podobné je to u většiny objektů a tříd. To je asi největší nedokonalost dokumentace k *ArcObjects*.

#### 3.4.4 První aplikace

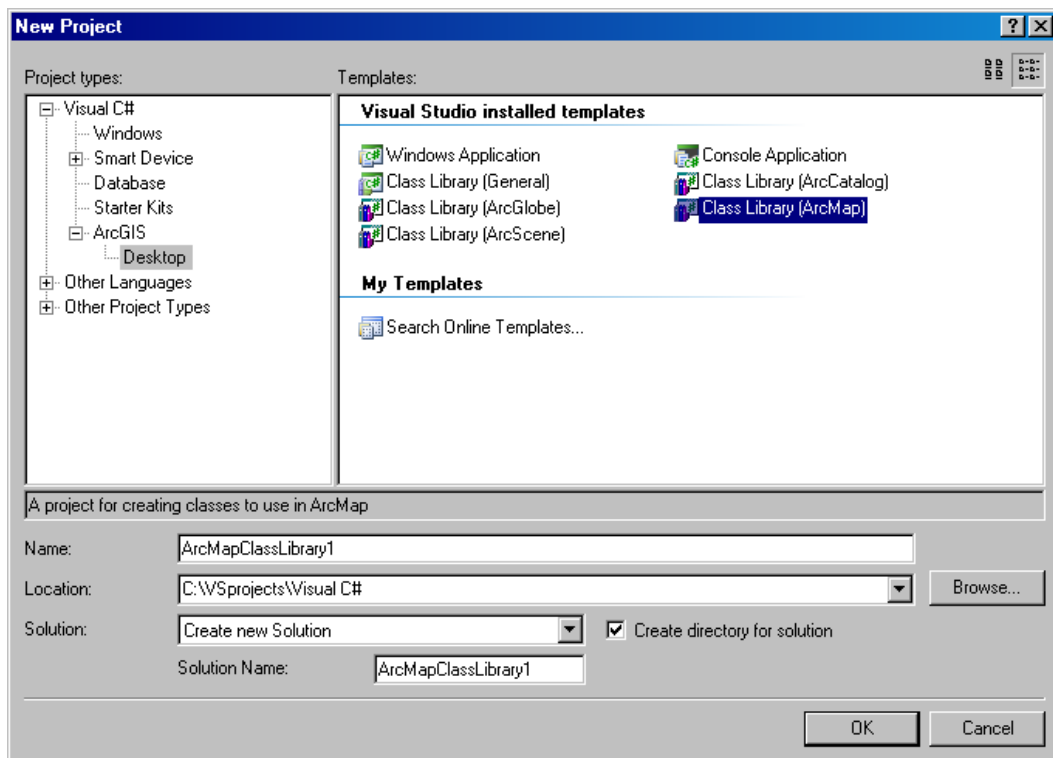
Na internetových stránkách s dokumentací k *ArcObjects* je několik jednoduchých ukázkových příkladů, jak vytvořit jednoduché příkazy, nástroje apod. Jedním z nich je i příklad popisující vytvoření jednoduchého příkazu zděděním třídy `BaseCommand` uvedený na internetové stránce [6]. Zde tento příklad bude využit a zkráceně popsán. Protože já sám jsem programoval v C#, bude se další výklad vztahovat především k tomuto jazyku. V ostatních programovacích jazycích funguje vše velmi podobně.

Předpokladem je nainstalované vývojové prostředí Microsoft Visual Studio .NET 2005 a ArcGIS Software Developer Kit. Pro vyzkoušení výsledného příkazu je potřeba mít nainstalován také ArcGIS 9.2.

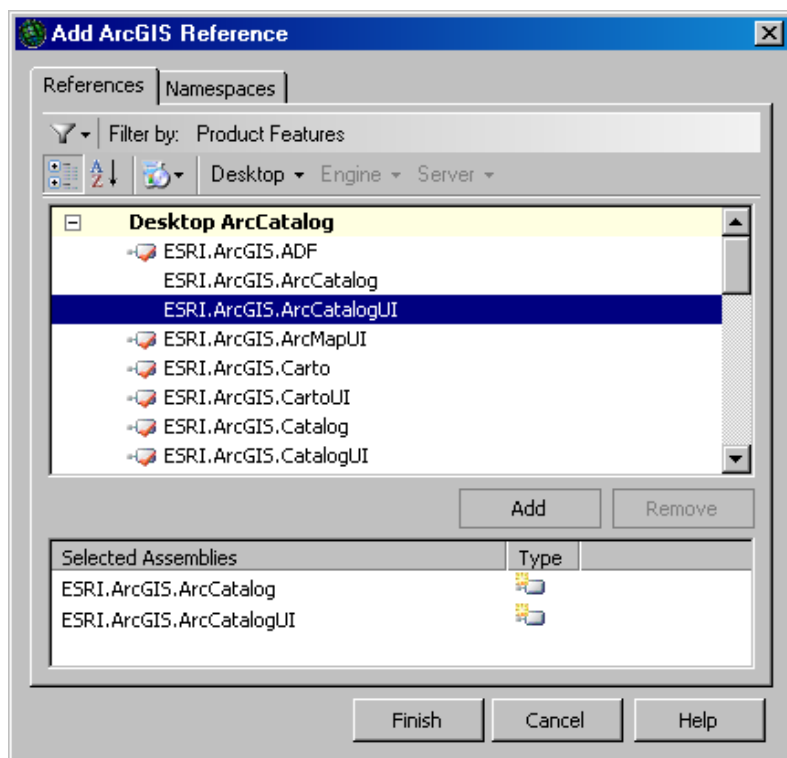
Na začátku je potřeba vytvořit nový projekt. Instalací SDK přibyly do nabídky typů projektů některé nové, které se týkají vývoje aplikací pro ArcGIS. Jsou zobrazeny na obrázku 3.3. Na výběr je několik možností, rozděleny jsou především podle programu, pro který bude daný příkaz či nástroj vyvíjen. V našem případě to bude příkaz pro program ArcMap. Lze vytvořit také obecnou knihovnu, která bude dostupná ve všech programech rodiny ArcGIS. Po výběru položky *Class Library (ArcMap)* a zadání jména projektu vytvořte nový projekt.

Aby bylo možné využívat funkcí *ArcObjects*, je potřeba přidat do projektu reference na knihovny objektů firmy ESRI. K tomu se otevře nástroj, ve kterém je možné požadované knihovny jednoduše přidat do projektu (viz obrázek 3.4). Knihovny lze samozřejmě přidat kdykoliv později, což je většinou potřeba, protože málokdo předem ví, které knihovny bude ve svém projektu používat.

Visual Studio při vytvoření projektu automaticky vytvoří také soubor s novou třídou. Tento soubor však není vytvořen ze šablony, kterou chceme použít, a proto jej můžeme odstranit. Místo něj přidáme nový soubor, pro který nám bude nabídnuto několik šablon přímo pro program ArcGIS. Jejich výhodou je to, že programátor nemusí sám vytvářet všechny potřebné funkce, jako je registrace COM komponent do registrů apod. Ty jsou totiž v šabloně obsažené. Na výběr je několik možností, mezi nimi i *Base Command*. Tuto



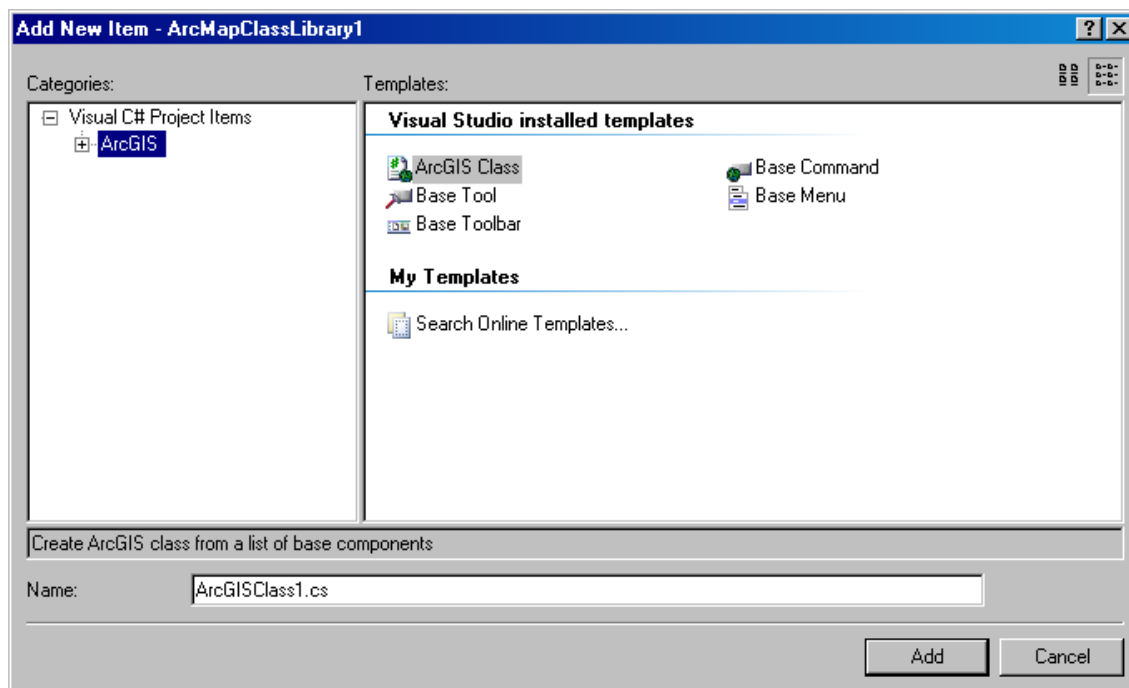
Obrázek 3.3: Ukázka obrazovky pro vytváření nového projektu



Obrázek 3.4: Ukázka obrazovky pro přidávání referencí na knihovny ESRI

šablonu použijeme pro vytvoření našeho příkazu. Ostatní šablony jsou podobné a lze pomocí nich vytvořit nástroje, panely nástrojů nebo menu, jak ukazuje obrázek 3.5. Po výběru dané šablony se utilita integrovaná do vývojového prostředí z SDK zeptá, pro který program z rodiny ArcGIS chceme nový příkaz vytvořit.

Podobně jako při vytváření nového projektu lze vybrat některý z rodiny programů ArcGIS, nebo například univerzální příkaz, který bude fungovat ve všech programech. Tato situace je zobrazena na obrázku 3.6.



Obrázek 3.5: Ukázka obrazovky se seznamem šablon pro ArcObjects

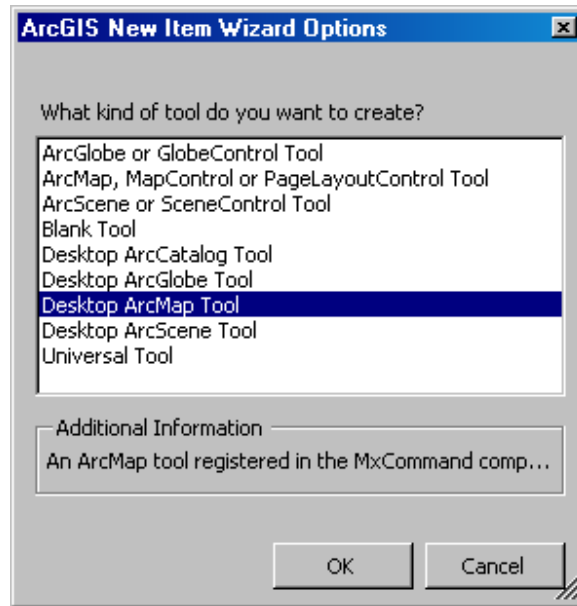
Třída **BaseCommand** (a některé další podobné) byla vytvořena vývojáři firmy ESRI tak, aby zjednodušila práci ostatním programátorům, protože pomocí ní lze vytvořit příkaz rychleji a jednodušeji, než implementací samotného rozhraní  **ICommand**. Některé tyto pomůcky jsou dostupné až od verze 9.2.

Přidáním nového příkazu se vytvoří soubor s jeho implementací, soubor s jeho ikonou a soubor s kurzorem. Poslední dva zmíněné lze upravit podle uvážení. V souboru se zdrojovým kódem příkazu byly automaticky vytvořeny některé potřebné metody a konstruktor. Dědění třídy **BaseCommand** je zajištěno řádkem

```
public sealed class ZoomToLayer : BaseCommand.
```

V konstruktoru lze pak nastavit některé podrobnosti, jako soubor s ikonou a kurzorem, nebo názvy příkazu zobrazované v programu (jméno příkazu, bublinovou nápovědu, řetězec zobrazovaný ve stavovém řádku a podobně).

Další důležitou funkcí je funkce **onClick()**, která je volána po kliknutí na ikonu našeho příkazu. Zde se ve většině případů provádí kód, který má příkaz vykonávat. Tato metoda patří mezi ty, které je nutné implementovat pro rozhraní  **ICommand**. Další metody se také týkají používání myši, protože některé příkazy nebo nástroje nemusejí reagovat na použití tlačítka, ale například až při kliknutí do mapy, tažení myši v mapě apod.



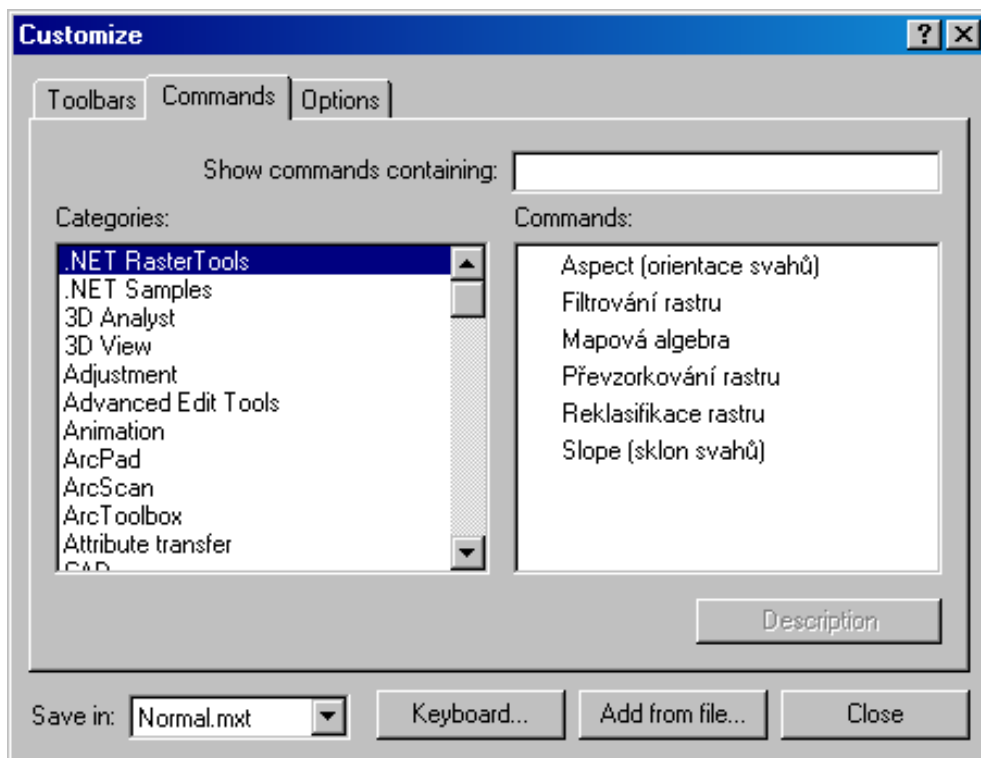
Obrázek 3.6: Ukázka obrazovky s výběrem vlastností nového příkazu (nástroje)

```
public override void OnClick()
{
    myImplementationOfCommand(); // vlastní kód prováděný příkazem
}
```

Po zapsání vlastního kódu, který má náš příkaz vykonávat bude vše připraveno k přeložení projektu a spuštění příkazu v programu ArcMap. Příkaz může vykonávat například jednoduché výpočty, zobrazování údajů o mapě, zvětšování měřítka, nebo i složitější funkce, jako vyvolání dialogu pro získání vstupních parametrů pro výpočty apod. Vše záleží na programátorovi. Pro začátek doporučuji vyzkoušet přeložit připravené příklady, než se sami pustíte do psaní vlastních. V těchto příkladech také budete často hledat různé konstrukce.

U projektu v jazyce C# je potřeba nastavit, aby se automaticky registrovaly vytvořené COM komponenty do registrů systému Windows. Někdy je potřeba také zviditelnit komponentu COM (*COM Visible* nastavit na `true`). Po implementaci potřebných funkcí a přeložení projektu lze jednoduše v některé z aplikací ArcGIS přidat vytvořenou komponentu pomocí jednoduchého dialogu k tomu určenému a okamžitě ji začít používat. K tomuto dialogu se lze dostat výběrem položky *Customize* v nabídce *Tools*. Zde se zobrazí seznam dostupných nástrojů (příklad viz obrázek 3.7), ze kterých si vybereme námi požadovaný a myší jej přetáhneme do některého z panelů nástrojů v programu. Aby se v této nabídce zobrazil náš vytvořený příkaz, je nutné jej zaregistrovat do registrů Windows, což v našem případě zařídí automaticky při překladu vývojové prostředí. Nezapomeňte, že u některých nástrojů je možné nastavit, aby fungovaly pouze v daných programech rodiny ArcGIS, např. pouze v ArcMap. V tomto případě v jiných aplikacích nepůjdou spustit a nebudou ani zobrazeny v seznamu dostupných nástrojů.

V některých případech (jako například v mojí práci) není potřeba přidávat tlačítka na panel ručně přetahováním jednotlivých příkazů a nástrojů na panel. Pokud vytvoříme vlastní panel nástrojů pomocí třídy `BaseToolbar`, přibude do nabídky panelů nástrojů v progra-



Obrázek 3.7: Ukázka obrazovky pro úpravy panelů nástrojů

mech rodiny ArcGIS automaticky. Do tohoto panelu se pak velmi snadno přidávají jednotlivé položky (nástroje) nebo roletové nabídky apod. Aby mohl ArcGIS takový panel zobrazit, musí být panel zaregistrován, stejně jako všechny ostatní komponenty, které chceme používat. Při kompilování se to děje automaticky, na ostatních počítačích je nutné registrovat komponenty ručně (pomocí \*.reg souboru nebo pomocí instalátoru, viz literatura [14]).

Pro ladění programů lze využít standardní nabídky Debug ve vývojovém prostředí, kde stačí nastavit, ve kterém programu rodiny ArcGIS se bude náš vytvořený příkaz ladit.

Jak bylo ukázáno, samotné vytvoření příkazu není nic obtížného. Podobně se vytvářejí i nástroje, panely nástrojů a další komponenty. Vše potřebné zajistí vývojové prostředí samo, takže programátorovi samotnému zbývá naimplementovat pouze samotný kód příkazů a nástrojů. Ten se samozřejmě u jednotlivých nástrojů liší rozsahem i funkcí, takže zde nelze jednoduše popsat, jak vytvořit samotný takový nástroj. Pro jeho implementaci lze využít všechny třídy a metody obsažené v knihovnách ESRI a všechny ostatní, které jsou podporovány vybraným programovacím jazykem. V knihovnách ESRI je zpracováno i několik prvků grafického rozhraní, které může programátor libovolně používat. Jejich výhodou je nejen zjednodušení práce, ale i zachování standardního rozhraní pro komunikaci s uživatelem.

Pokud má programátor (a tím také zadavatel projektu) k dispozici i některá oficiální rozšíření pro ArcGIS, může využívat i jejich výhod. V mém případě je to samozřejmě nemožné, protože používání objektů z rozšíření je vázáno na licenci, což je v tomto případě nežádoucí.

## Kapitola 4

# Návrh vlastní nadstavby ArcGISu

### 4.1 Inspirace

Inspirací této práce je již existující nadstavba Spatial Analyst od firmy ESRI. Ta je ale velmi rozsáhlá a obsahuje přes 150 funkcí (nástrojů). Je zřejmé, že naprogramování takového množství funkcí nemůže zvládnout jediný člověk za přijatelnou dobu. Proto budou vybrány pouze některé funkce, a to podle jejich důležitosti a použitelnosti ve výuce. ArcGIS Spatial Analyst umožňuje například vytváření, zobrazování a analýzu rastrových dat. Dovoluje také provádět kombinované analýzy, kdy se zpracovávají jak data rastrová, tak i data vektorová.

Spatial Analyst umožňuje využití dat, která popisují spojitě se měnící veličiny, jako např. nadmořská výška, sklon, teplota, srážky a podobně. Zároveň lze v rámci nadstavby Spatial Analyst pracovat i s klasifikovanými rastry (např. rastr vyjadřující způsob využití půdy apod.) a také takové rastry vytvářet (převodem z vektorového formátu nebo kategorizací spojitých dat). Kombinací různých rastrových i vektorových dat, logických dotazů a nejrůznějších algoritmů zpracování prostorových dat se získávají nové informace o území. Příkladem výstupu mohou být oblasti vybrané na základě nějakých kritérií, např. zalesněné oblasti v nadmořské výšce vyšší než 1000 m.

Na Spatial Analyst navazují některé další nadstavby, např. Geostatistical Analyst a 3D Analyst.

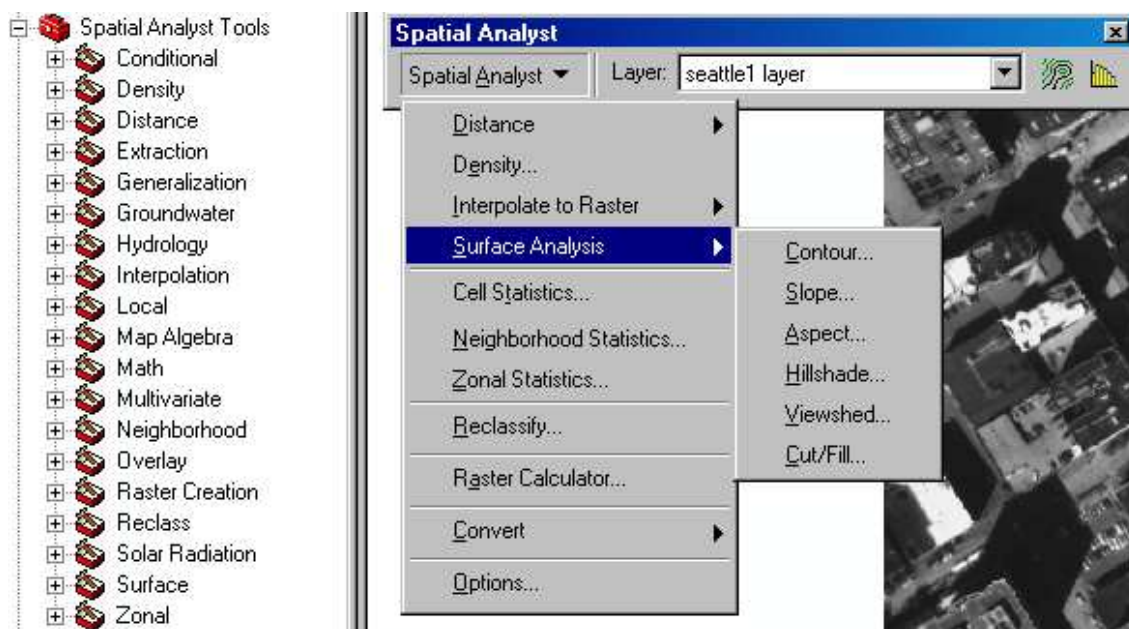
#### 4.1.1 Přehled funkcí nadstavby ArcGIS Spatial Analyst

Funkce jsou rozděleny do několika odvětví, podle toho, jak spolu souvisí:

- **Interpolace měření** – na základě měření v diskrétních bodech pomocí interpolačních algoritmů vytvoří rastr, který informuje o spojitém rozložení měřené hodnoty na celém zájmovém území
- **Dotazování** – výběr oblastí odpovídajících jedné či více daným podmínkám (např. které oblasti jsou určeny pro bytovou zástavbu a zároveň mají vysokou hladinu podzemní vody a leží na svahu se sklonem nad 15 %)
- **Statistika rastru** – porovnání několika rastrů, statistika sousedství, výpočet průměrných hodnot apod.
- **Analýza terénu** – informace o svažitosti, orientaci svahů ke světovým stranám, stínování reliéfu, analýza viditelnosti, tvorba vrstevnic

- **Vzdálenostní analýza** – nalezení nejvhodnější trasy s uvážením odporu prostředí, výpočet ceny dané trasy apod.
- **Analýza hustoty** – přehled o tom, jaká je v různých oblastech území koncentrace linií či bodů
- **Hydrologické modelování** – modelování hladiny spodní vody, nalezení oblastí kumulace odtékající vody, zjištění směru odtoku z každého bodu terénu, rozdělení oblasti na povodí
- **Dynamické modelování** – modeluje šíření dynamických jevů, např. požárů či znečištění ovzduší a vody na základě ovlivňujících parametrů (směr a síla větru, hořlavost apod.)
- **Mapová algebra** – tvorba nových vrstev pomocí jednoduchých operátorů a vestavěných algoritmů
- **Konverze formátů** – vektorová data mohou být převedena do rastrového formátu včetně zachování informací z atributové tabulky a do jisté míry naopak.
- **Reklasifikace** – klasifikace rastru podle požadavků
- **Projekce (zobrazení)** – volba způsobu vizuální prezentace dat

Toto rozdělení je uvedeno v [4] a nepoužívá zcela standardní terminologii, proto se v některých detailech mírně liší. Na obrázku 4.1 je záběr na nastavbu Spatial Analyst v programu ArcMap.



Obrázek 4.1: Ukázka nastavby Spatial Analyst

## 4.2 Návrh nadstavby

Plánem bylo především vytvoření základních funkcí používaných v předmětu GIS na naší škole, ale také přiblížení možnosti programování vlastních nadstaveb. První část je spíše praktická, ve druhé jde o zdokumentování první části a vysvětlení základních principů programování nadstaveb pro případné další zájemce, kteří tak budou již mít na čem stavět při programování dalších funkcí nebo rozšíření.

### 4.2.1 Implementované nástroje

Jak již bylo zmíněno, není pravděpodobně možné vytvořit všechny funkce, které obsahuje komerční nadstavba Spatial Analyst. Proto byly vybrány některé nejdůležitější a nejpoužívanější funkce, které byly implementovány. Základem jsou algoritmy reklasifikace a mapové algebry, ke kterým přibýly analýzy terénu. V plánu byly také vzdálenostní analýzy, které se nepodařilo implementovat.

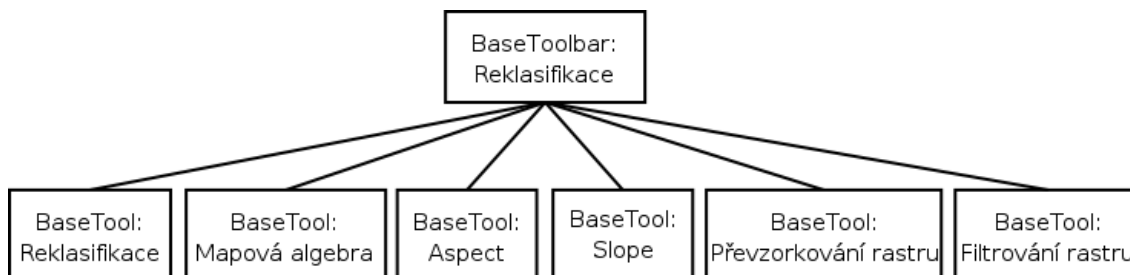
Zde bude uveden stručný popis nástrojů, které byly v plánu v diplomové práci implementovat. Protože se ale složitost takových algoritmů špatně odhaduje, zvláště bez zkušeností s programováním a rozšiřováním *ArcObjects*, výsledná nadstavba obsahuje pouze některé z uvedených funkcí.

- **Reklasifikace** – pokud uživatel vybere nástroj reklasifikace, zobrazí se okno, ve kterém bude možnost výběru vstupního rastru a výstupního rastru. Dále bude obsahovat reklasifikační tabulku. K tomu patří také zobrazení rozsahu možných hodnot, které rastr obsahuje (minimální a maximální hodnota obsažená v buňkách, pokud je možné ji zjistit). Uživatel samozřejmě bude moci zadávat jak jednotlivé hodnoty, tak celé rozsahy hodnot, na základě kterých bude vytvořena nová vrstva s výsledným rastrem. Reklaifikační tabulku bude možné uložit do souboru a v případě pozdější potřeby kdykoliv znovu načíst. Syntaxe ukládaného souboru bude jednoduchá a tudíž bude soubor možné vytvářet i ručně (se zachováním některých pravidel).
- **Mapová algebra** – umožní uživateli provádět různé matematické (aritmetické, booleanové a relační) operace nad rastry. Mezi ně patří například sčítání, odčítání, logické operace AND a OR, operace porovnání apod. Podobně jako v případě reklasifikace bude nabídnuta možnost výběru vstupních a výstupních rastrů (vrstev). Dále uživatel nastaví operace, které se mají mezi jednotlivými vstupními rastry provést.
- **Analýza modelů terénu** – pomocí algoritmů obsažených v této skupině lze počítat sklony a orientaci svahů (anglicky slope a aspect). Patří sem i další nástroje, které však nebyly naprogramovány (např. tvorba vrstevnic nebo analýza viditelnosti).
- **Vzdálenostní analýzy** – soubor nástrojů týkajících se vzdálenostní analýzy. Mezi nejdůležitější algoritmy patří výpočet okolí, tzv. obalové zóny (buffering). Další funkce jsou např. nákladová vzdálenost nebo funkce šíření a proudění (v nadstavbě Spatial Analyst jsou tyto algoritmy rozděleny do více skupin, např. hydrologické modelování). Tyto nástroje se bohužel nepodařilo implementovat.

Zjednodušený diagram na obrázku 4.2 zobrazuje uspořádání nadstavby. Všechny nástroje jsou vytvořeny na základě třídy `BaseTool` a jsou přístupné přes jediný panel nástrojů, který je vytvořen pomocí třídy `BaseToolBar`. Každý další nový nástroj je jednoduché přidat do tohoto panelu. Všechny nástroje jsou však vytvořeny a registrovány také tak, aby byly



použitelné samostatně. Pokud tedy bude uživatel chtít používat pouze některý z nich, může si jej přidat na libovolný panel nástrojů.



Obrázek 4.2: Diagram uspořádání rozšíření

Po stisknutí tlačítka na panelu nástrojů si uživatel vybere některý z dostupných nástrojů. Poté bude ve většině případů v dialogovém okně vyzván k zadání vstupních souborů (s rastry) a případně dalších dat. Program vytvoří na základě zadaných parametrů novou rastrovou vrstvu, která bude po zpracování obsahovat výsledek. Tu nakonec zobrazí a uloží do vybraného souboru. V některých případech lze rastr uložit také pouze v paměti, ale ne vždy je chování takového rastru při další práci vhodné, a proto budou vždy rastry uloženy na disk. Pokud uživatel nezadá přímo cestu k výstupnímu souboru, uloží se výstup do adresáře pro dočasné soubory. Pokud uživatel tento výstup nepotřebuje jen jako dočasný, musí si jej pro jistotu uložit později jinak, protože ze složky dočasných souborů může být rastr odstraněn i jinými programy.

#### 4.2.2 Postup implementace

Nejdůležitějším krokem je implementace prvního algoritmu a jeho začlenění do programu ArcGIS. Programování dalších funkcí je již podobné. Tento předpoklad stavěl především na tom, že jsem byl v tomto oboru začátečník a nadstavby pro ArcGIS pro mě byly zcela novým tématem. Jelikož reklasifikace patří mezi nejzákladnější a nejdůležitější algoritmy rastrové analýzy, byla implementována jako první. Ostatní nástroje byly vytvářeny postupně v pořadí, v jakém byly uvedeny výše. Je to dáno například tím, že mapová algebra může být použita pro výpočet dalších funkcí, a proto byla implementována mezi prvními. Některé další algoritmy ji pak využívají a tím se zjednodušují, protože není potřeba použít operace implementovat zbytečně vícekrát.

Jak bylo vysvětleno v kapitole 3, nadstavba vznikne rozšířením *ArcObjects*, respektive implementací rozhraní *IExtension*. Nadstavbu však není potřeba implementovat pomocí rozhraní *IExtension*, to je nutné pouze v případě, že je potřeba zakomponovat například závislosti na jiných rozšířeních a podobně. Z toho vyplývá, že pokud chceme do některého z programů rodiny ArcGIS přidat např. pouze jednoduché příkazy nebo nástroje, není vůbec nutné implementovat je jako rozšíření (nadstavbu). Toho využívá i moje práce. Protože nepotřebuje kontrolovat závislosti na jiných rozšířeních, není nutné ji implementovat jako rozšíření. V tomto případě stačí přidat do nabídky panelů nástrojů položku s vlastním panelem, který poskytuje přístup ke všem implementovaným nástrojům. To, že uživatel nemůže zakázat používání nadstavby tím, že by ji v nastavení vypnul, není na škodu. Tyto situace jsou nutné spíše v případech, kdy má firma omezený počet licencí na danou nadstavbu. V našem případě nikdy nic takového nenastane.

Základem mojí nadstavby je tedy třída, která rozšiřuje třídu `BaseToolbar`. Ta poskytuje přístup k implementovaným funkcím, které jsou vytvořeny pomocí třídy `BaseTool`. Jednotlivé nástroje se pak zobrazují v roletovém menu. Funkčnost nástrojů odpovídá jejich klasickému pojetí v geografických informačních systémech, grafické rozhraní je do určité míry inspirováno existující nadstavbou `Spatial Analyst`.

Původní návrh obsahoval požadavek také na vytvoření sady nástrojů – toolboxu. V programu ArcGIS je to balíček s několika spolu souvisejícími nástroji, který se zobrazuje jako červený kufřík v daném panelu. Mnou implementované funkce lze použít z toolbaru, což je panel v okně programu. Bohužel se mi prozatím nepodařilo zjistit, zda je možné propojit toolbox a toolbar dohromady. Je to dáno i tím, že oficiální rozšíření `Spatial Analyst` se v těchto dvou rozdílných místech také liší. Obsahuje i toolbox i toolbar (panel), jak je vidět na obrázku 4.1. Oba mají ve svém repertoáru podobné funkce, ale jejich implementace je pokaždé jiná (stačí zkusit spustit nástroj se stejným jménem a porovnat).

### 4.2.3 Volba programovacího jazyka

Uživatelské nadstavby a nástroje mohou uživatelé vyvíjet ve standardních programovacích jazycích pro Microsoft Windows, jako je VBA (Visual Basic for Applications), .NET, Java a Visual C++. Jedinou podmínkou je podpora programovacího jazyka pro Microsoft Component Object Model (COM). Tu splňují všechny uvedené jazyky. Každý jazyk, který je v tomto případě možné použít, má své výhody i nevýhody. Pro vývoj je firmou ESRI doporučován jazyk VBA, pro který je pravděpodobně největší podpora z jejich strany. Proti jeho použití mluví zase v dnešní době poměrně nepřehledný kód. Ostatní jazyky jsou ale také dobře podpořeny, existuje mnoho názorných příkladů a pro většinu uvedených jazyků i návody přímo na internetových stránkách [9]. Příklady jsou si naštěstí podobné i v rámci různých programovacích jazyků, takže lze většinou bez větších problémů přepsat příklad z jednoho jazyka do jiného. Problémem jsou většinou typové konverze, které se jinak hlídají v Javě a C# než ve VBA.

Já jsem si pro implementaci vybral platformu .NET. Není zde přímo předepsáno, který programovací jazyk se musí použít, je tedy možné si vybrat. Můj výběr padl na C#. Tento jazyk jsem si vybral proto, že má poměrně dobrou podporu ze strany firmy Microsoft a mnoho různých funkcí, které je jinde potřeba doplnit. Pro jeho použití mluví také kvalita programovacího nástroje Microsoft Visual Studio .NET, který jednoduchost programování podpoří. Firma ESRI vytvořila pro tento jazyk příklady a také integrovala do zmíněného prostředí základní knihovny a několik utilit pro použití *ArcObjects*, což je také výhodou. Dalším důvodem, proč jsem si vybral právě tento programovací jazyk je ten, že s ním mám více zkušeností než s ostatními. Je velmi podobný jazyku Java, ve kterém pracuji často, ale podpora COM v Javě není tak ideální a jednoduchá, jak by bylo potřeba.

Významnou roli sehrála také příjemnost prostředí Microsoft Visual Studio .NET, které programátorovi usnadní programování, například v podobě inteligentní nápovědy.

### 4.2.4 Výkonnost vybraného jazyka

Některé zdroje uvádějí, že výkonnost programů napsaných v jazyci C# může být v některých případech nižší, než při použití např. VBA nebo Visual C++. To může být jistou nevýhodou, protože efektivita bude mít velký vliv při zpracování rastrů, zvláště pak těch velkých. Na druhou stranu podle některých porovnávaní v [9] není rozdíl tak markantní a projevuje se pouze při některých specifických operacích. Základem vytvářené knihovny je ale implementace

algoritmů tak, aby se daly použít zdarma, ne tak, aby byly při zpracování obrovských objemů dat efektivnější než knihovna dodávaná firmou ESRI.

Jak se později při implementaci ukázalo, je použití tohoto jazyka pravděpodobně lepší než použití například jazyka Java, protože ten (alespoň pokud je mi známo) neumí pracovat s ukazateli na funkce. V C++ je tento případ běžný, a protože C# je ze stejné rodiny jazyků, jeho vývojáři tuto vlastnost zanechali. Ukazatele na funkce se programují pomocí delegátů, blíže se o tom zmíním v příští kapitole při popisu implementace knihovny, přesněji mapové algebry. Vzhledem k efektivitě je tento přístup lepší, než jaký dovoluje programovací jazyk Java.

Výhodou výběru je také například možnost jednoduchého vytvoření instalátoru, který by jinak sám o sobě mohl být předmětem diplomové práce. Jeho vytvoření je velmi pěkně popsáno přímo na stránkách firmy ESRI [14].

Naopak nevýhodou použití jazyka C# a obecně platformy .NET je to, že je nutné registrovat všechny součásti výsledného programu do registrů Windows. Na počítači, kde se program překládá, se zápis do registrů provede automaticky, na ostatních počítačích to tak už ale není. V případě použití jazyka Visual C++ údajně stačí výsledný *dll* soubor překopírovat a v programu ArcGIS k němu najít cestu. Pokud je pro vývoj použita platforma .NET, je nutné nejdříve registrovat součásti, a poté lze teprve spustit knihovnu v programu ArcGIS. Registraci lze provést více způsoby, nejvýhodnější je ale použít již zmíněný instalátor, který ji provede automaticky.

Výkonem výsledné knihovny s rozšířením oproti oficiální nadstavbě Spatial Analyst se bude blíže zabývat kapitola 6 o testování knihovny.

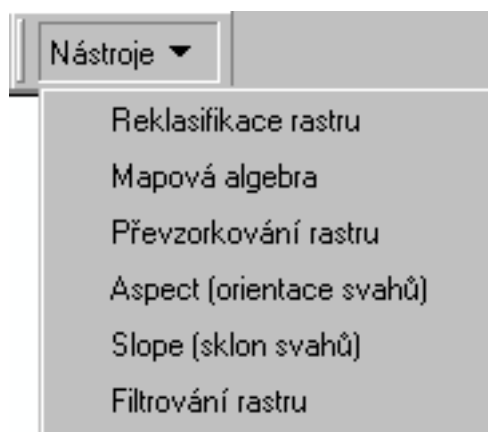
## Kapitola 5

# Implementace knihovny

Tato kapitola se bude zabývat implementací jednotlivých nástrojů rastrové analýzy. Budou zde podrobně popsány jednotlivé algoritmy, jak lze při jejich implementaci využít knihovny *ArcObjects* a také si dovoluji zmínit problémy, na které jsem při implementaci narazil. Obrázky grafického uživatelského rozhraní použitého u nástrojů se budou vyskytovat v blízkosti popisu daného nástroje, jako ukázka výsledku.

### 5.1 Panel nástrojů

Po nainstalování nadstavby se do nabídky použitelných panelů nástrojů v programu ArcGIS přidá položka „Rastrová analýza“. Po jejím zatržení se objeví panel, na kterém je menu pro výběr požadovaného nástroje. Tento panel je zobrazen na obrázku 5.1. Po stisknutí některé z položek menu se spustí odpovídající funkce. Jednotlivé funkce jsou probrány podrobně dále v této kapitole.



Obrázek 5.1: Ukázka panelu nástrojů dodaného knihovnou

### 5.2 Reklasifikace rastrů

Prvním nástrojem, který jsem se po konzultaci rozhodl implementovat, je reklasifikace rastru. Jedná se o selekci a klasifikaci buněk, které splňují podmínku kladenou na jejich

atributové hodnoty. Lze tím dosáhnout například sloučení původních kategorií buněk do kategorií nových. Reklasifikační funkce mění hodnoty buněk použitím různých metod. Podrobný popis reklasifikace byl uveden již v kapitole 2.5.1.

### 5.2.1 Implementace reklasifikace rastru

Na implementaci reklasifikace rastru je důležité to, jak samotná reklasifikace proběhne. To je zajištěno pomocí speciálního filtru, který dodává knihovna *ArcObjects*. Jedná se přesněji o filtr *RemapFilterClass*, který implementuje rozhraní *IRemapFilter*.

Tento filtr pomocí několika metod dovoluje zadávat rozsahy vstupních hodnot, pro které má být změněna hodnota na zadanou výstupní hodnotu. Rozsahy by se neměly překrývat, protože pak není jednoznačně určeno, kam bude hodnota z překrývajících se intervalů zařazena. Pokud je počátek a konec rozsahu stejné číslo, aplikuje se toto pravidlo filtru pouze na dané číslo. Přidávání pravidel do filtru se provede takto:

```
IRemapFilter filter = new RemapFilterClass (); // vytvoření filtru
filter.AllowUnmatched = false; // nezadané hodnoty budou změněny na NoData
filter.AddClass(50,50,100); // hodnota 50 bude nahrazena hodnotou 100
filter.AddClass(100, 200, 150); // čísla mezi 100 a 200 se změní na 150
filter.AddNoDataClass (200, 300); // čísla mezi 200 a 300 budou NoData
```

Do filtru je teoreticky možno zadat neomezené množství pravidel. Jak je vidět, zvládá filtr jak reklasifikaci konkrétních hodnot, tak jejich rozsahů. Čísla se filtru zadávají v typu *double*. Horní hranice není zahrnuta do intervalu, spodní ano. Pro seznam (výčet) hodnot filtr žádné pravidlo nenabízí, takže tento výčet musí být zpracován programem a rozdělen na jednotlivé hodnoty (případně rozsahy hodnot), které jsou pak vloženy jako pravidla do filtru.

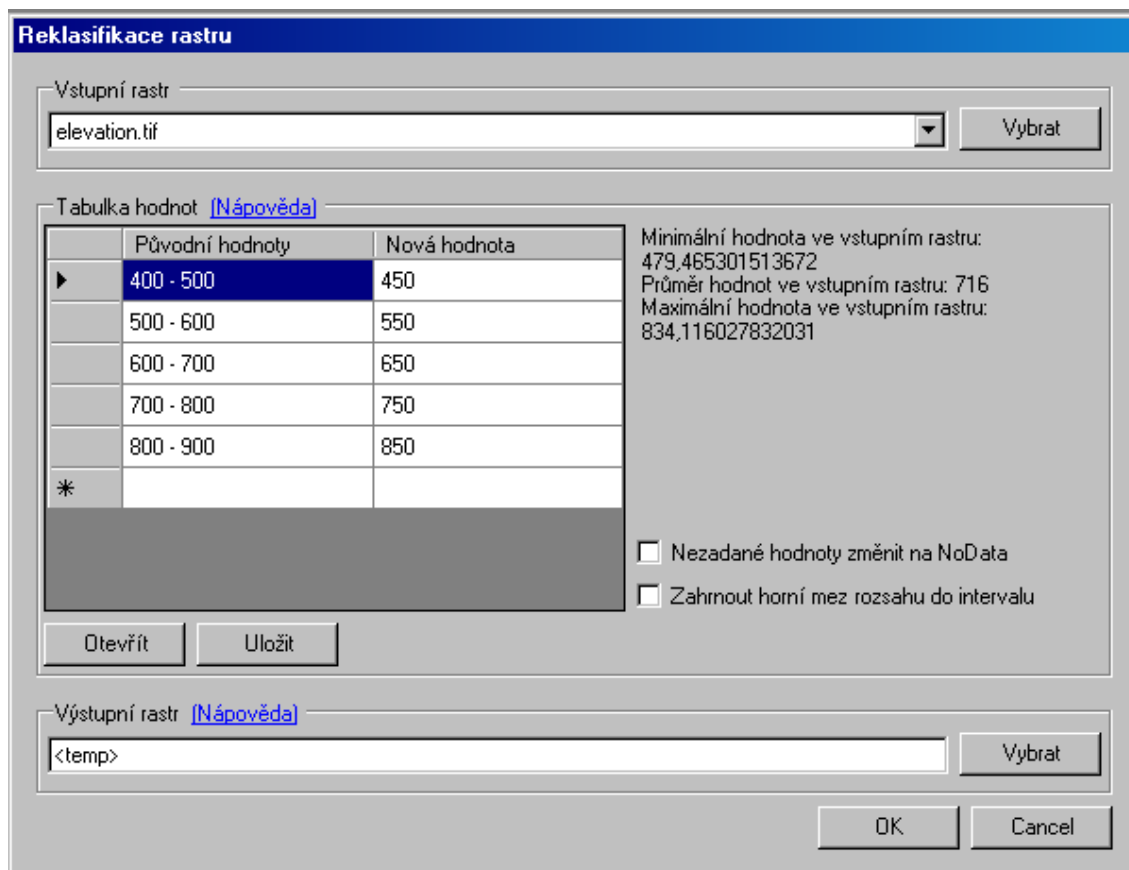
Vytvořený filtr lze jednoduše použít na rastr otevřený v paměti pomocí rozhraní *IRaster*, které lze přetypovat na rozhraní *IPixelOperation*. Toto rozhraní pak dovoluje nastavit vlastnost *PixelFilter*. Přiřazením vytvořeného filtru do této vlastnosti se s původním rastrem nic nestane, ale pokud z něj budeme číst data (zobrazovat nebo jinak používat) nebo jej uložíme, výsledkem budou reklasifikovaná data. Zjednodušeně lze postup zapsat takto:

```
IRaster inputRaster; // před použitím se sem musí načíst rastr
IPixelFilter pf = (IPixelFilter) filter; // přetypování filtru
IPixelOperation pixelOperation = (IPixelOperation) inputRaster;
pixelOperation.PixelFilter = pf; // nastavení filtru vstupnímu rastru
```

To znamená, že pokud po přiřazení filtru rastr uložíme, uloží se rastr s novými hodnotami. Samozřejmě nelze přepsat původní rastr (teoreticky ano, ale to stejně nesmyslné), musí být vytvořen nový soubor. Protože *ArcObjects* nedokáže přepsat existující soubor automaticky, musí být v případě existence nejprve odstraněn. Programy *ArcGIS* nedovolují přepisování existujících souborů vůbec.

Takto, poměrně jednoduše, lze shrnout funkčnost algoritmu pro reklasifikaci. Tohle byla ale pouze ta část, která samotný převod provede. Bohužel pro programátory, uživatelům není radno nikdy důvěřovat, a proto kontrola vstupů zadaných uživatelem do reklasifikační tabulky je mnohem náročnější, než samotná reklasifikace (kterou provede dobře nastavený filtr).

Implementace grafického uživatelského rozhraní včetně reklasifikační tabulky (viz obrázek 5.2) také není nijak složitá, vzhled je založen na reklasifikačním nástroji z nadstavby Spatial Analyst. Po spuštění se zobrazí okno, ve kterém je potřeba vybrat vstupní rastr, zadat parametry reklasifikace a jméno výstupního rastru. Je zde možnost načíst nebo uložit reklasifikační tabulku z/do souboru. V případě, že byly všechny parametry zadány správně, se po stisknutí tlačítka OK provede reklasifikace rastru.



Obrázek 5.2: Ukázka okna pro reklasifikaci rastru

Nápověda k použití se zobrazí po stisknutí modrého odkazu *Nápověda*. Zde je podrobně popsán formát vstupních dat pro reklasifikační tabulku. Položka původní hodnota může nabývat těchto podob:

- **Konkrétní hodnota** – celé nebo desetinné číslo
- **Rozsah hodnot** – rozsah se zadává jako číslo – číslo, např. 1 – 6, okolo pomlčky musí být z obou stran vždy mezera
- **Seznam (výčet)** – jednotlivé položky seznamu jsou odděleny znakem ; (středník), hodnoty mohou nabývat jedné ze dvou dříve uvedených podob
- **Řetězec „NoData“** – označuje prázdné buňky

Nová hodnota musí být vždy celé číslo, nebo řetězec „NoData“. Jinou hodnotu program nedovolí zadat. Nastavení pro zahrnutí horní meze rozsahu do intervalu se použije v případě,

že horní mez intervalu má být počítána jako součást intervalu. Normálně horní mez nenáleží do intervalu pro reklasifikaci. Nastavení pro změnu nezadaných hodnot na NoData bylo již diskutováno dříve.

### 5.2.2 Problémy při implementaci

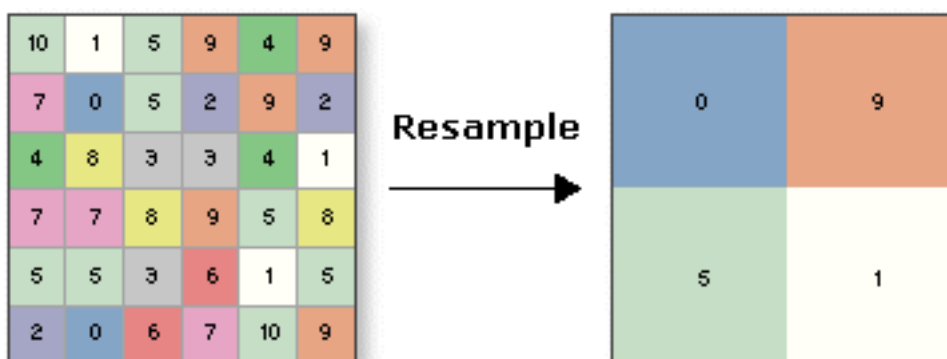
Jelikož byla reklasifikace prvním implementovaným algoritmem, narazil jsem při ní na mnoho problémů, hlavně s neznalostí struktury knihovny *ArcObjects* a práci s ní. Jedním z problémů bylo zjistit, jestli knihovna obsahuje nějaké objekty či funkce, pomocí kterých lze provést reklasifikaci. Řešení jsem nakonec našel v podobě zmíněného filtru.

Dalším problémem byl problém již dříve zmíněný v kapitole 3.4.1. Pokud byl vstupní rastr georeferencován, na výstupním rastru se toto nastavení nepromítlo zcela správně a ve výstupním GeoTIFF souboru zcela chyběly všechny hodnoty s tím související. Tento problém byl odstraněn instalací opravného balíčku pro ArcGIS.

Zdaleka nejtěžší bylo zajistit kontrolu a vyhodnocení uživatelských vstupů, což na vývoji této funkce zabralo většinu času. Projevilo se ale také počáteční potýkání se s neznámým prostředím *ArcObjects*. Dokumentace není tak propracovaná, jak by si při tak velkém rozsahu projektu zasloužila, a v uvedených příkladech také není všechno, takže občas je nutné spustit Object Browser a jednotlivé knihovny dll prostě a jednoduše prohledat a zkoušet. Například dialog pro výběr rastru není obyčejný dialog pro výběr souboru, ale dělá se pomocí rozhraní *IGxDialog*, o kterém v příkladech není zmínka, takže je poměrně složité zjistit, že vůbec existuje.

## 5.3 Převzorkování rastru

Převzorkování rastru se používá v případě, kdy je potřeba změnit rozměry buněk rastru. Jako výsledek se očekává rastr se stejnými geografickými vlastnostmi, ale jinou velikostí buněk. Jednoduchá ukázka převzorkování je na obrázku 5.3, převzatém z nápovědy k programu ArcGIS [3].



Obrázek 5.3: Ukázka převzorkování rastru

Přestože v ArcView existuje nástroj pro převzorkování, pro vlastní účely jsem vytvořil jiný. Jeho využití je především v mapové algebře, ale když už vznikly funkce pro různé možnosti převzorkování, tak k nim vzniklo i uživatelské rozhraní. To sice nepodporuje takové množství nastavení jako nástroj v ArcView, ale dovoluje převzorkovat rastr nastavením

velikosti buněk, nastavením nového rozlišení nebo nastavením měřítka, které se má aplikovat na aktuální velikost buňky.

Obrázek 5.3 ukazuje pouze zjednodušené převzorkování, kdy je velikost nové buňky trojnásobkem původní velikosti buňky. Hodnota nové buňky tak bude odvozena například podle původní hodnoty buňky ležící nad středem nové buňky. Tak to ale nemusí být vždy, velikosti buněk mohou být zvoleny nezávisle a to i v neceločíselných hodnotách. Obecně mohou mít buňky jakékoliv rozměry, a právě proto je potřebná funkce převzorkování, která přepočítá podle různých metod hodnoty původních buněk na hodnoty buněk nových.

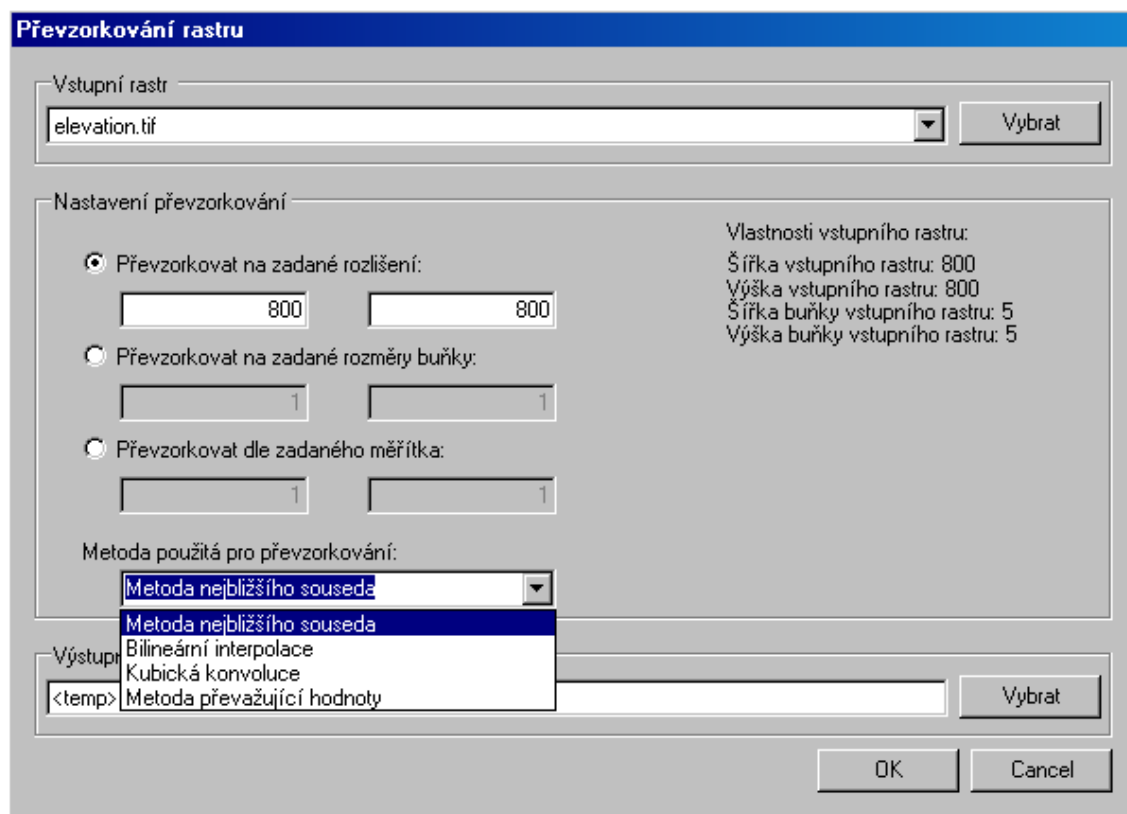
Při převzorkování mohou vznikat nové buňky nebo se mohou původní buňky spojovat v nové. Do těchto buněk je nutné dopočítat hodnotu, která bude reprezentovat co nejvěrněji hodnotu buňky, která byla na daném místě před převzorkováním. Protože hodnota buňky reprezentuje hodnotu jejího středu, není tento úkol vždy jednoduchý. ArcGIS ale poskytuje několik způsobů, jak hodnotu nových buněk vypočítat. Přesněji jsou to 4 metody:

- **Metoda nejbližšího souseda (Nearest neighbor)** – hodnota buňky je určena podle hodnoty nejbližší sousední buňky. Tato metoda je nejrychlejší a nevznikají při ní žádné nové hodnoty, proto se používá zejména na klasifikované rastry.
- **Metoda převažující hodnoty (Majority)** – má podobné vlastnosti jako předchozí metoda. Je výpočetně trochu náročnější, ale v jistých případech může dávat lepší výsledky. Výpočet hodnoty nové buňky je založen na převažující hodnotě původních buněk v daném místě. Stejně jako v předchozí metodě nevznikají žádné nové hodnoty buněk v rastru.
- **Metoda bilineární interpolace** – výpočet nové hodnoty probíhá interpolací původních buněk v daném bodě. Tato operace je výpočetně náročnější než předchozí a při jejím použití mohou vznikat v rastru nové hodnoty, takže se používá například pro výškové modely.
- **Metoda kubické konvoluce** – tato metoda je obdobou předchozí, ale její výsledky by měly být přesnější, respektive reálnější. To se podepisuje na její časové náročnosti při výpočtu.

Samotné převzorkování funguje zase velmi jednoduše, protože je na něj připravena funkce v *ArcObjects*. Otevřenému vstupnímu rastru nastavíme nové rozměry buňky a metodu, pomocí které se mají buňky převzorkovat. Při ukládání se (jako v případě filtru u reklasifikace) uloží převzorkovaný rastr s novými rozměry buněk, původní rastr zůstane nezměněn. Jak je vidět, v některých případech je programování nových funkcí opravdu poměrně snadné.

Uživatelské rozhraní (viz obrázek 5.4) sestává z prostoru pro výběr vstupního a výstupního rastru a nastavení převzorkování. Na výběr je několik možností podle čeho se má rastr převzorkovat a také jaká metoda se má použít. Pro programátory je přínosem to, že kód této knihovny poskytuje funkce pro převzorkování rastru podle zadaných parametrů a programátor je tedy nemusí znovu programovat, když je potřeba využít v dalších nástrojích. Já sám převzorkování používám v mapové algebře, kde je potřeba při výpočtech nad dvěma rastry zajistit, aby výpočet probíhal nad stejně velkými buňkami. Popis mapové algebry navazuje na tuto podkapitolu.





Obrázek 5.4: Ukázka okna pro převzorkování rastru

## 5.4 Mapová algebra

Dalším implementovaným nástrojem se stala mapová algebra. Bylo to tak z důvodu, že je možné ji dále použít při výpočtu jiných funkcí, při kterých se používají početní operace nad rastry, jako je například sklon svahů nebo jejich orientace. Podrobný popis mapové algebry z pohledu funkce byl uveden v kapitole 2.5.2.

Mapová algebra je pravděpodobně nejrozsáhlejší implementovanou funkcí, protože pro ni již knihovna *ArcObjects* nenabízí žádná podobná zjednodušení, jako to bylo v předchozích případech. Samozřejmě s *ArcObjects* mapovou algebru lze používat, ale pouze v případě, že máme k dispozici licence na rozšíření Spatial Analyst. V našem případě se tak opravdu počítá s jednotlivými buňkami, které firma ESRI nazývá pixely (i když název pixel se používá spíše v terminologii multimédií a ne geografických informačních systémů).

### 5.4.1 Implementace mapové algebry

Ve vlastní implementaci jsem se rozhodl mapovou algebru udělat jednodušší, než je mapová algebra dodávaná rozšířením Spatial Analyst. Je to tak hlavně proto, že mapová algebra může používat obrovské množství funkcí, které jsou mnohdy pouze do počtu, a jejich využití je naprosto minimální. Repertoár funkcí lze jednoduše rozšířit, jak bude vysvětleno dále. Dále jsem se rozhodl implementovat pouze funkce a binární operátory, protože pomocí nich je možné vytvářet další složitější operace.

Protože zpracování složitějších výrazů s možností použití závorek je poměrně složité

a není přímo předmětem této diplomové práce, je možné pracovat pouze s jednoduchými výrazy nad dvěma rastry nebo rastrem a číslem, v případě funkcí pouze nad jedním rastrem. Samozřejmě, nebylo by až tak složité zpracovat i složitější výrazy, když už lze zpracovávat výrazy jednoduché. Stejně tak lze ale pomocí těchto jednoduchých výsledků dosáhnout stejného cíle, i když bohužel za cenu větší pracnosti ze strany uživatele. Zpracování výrazů nad více rastry by bylo možno provést dvěma způsoby. Buď vytvořit funkce pro jakýkoliv počet parametrů (rastrů i čísel), nebo zpracovávat jednotlivé výrazy podle priority operátorů zvláště po maximálně dvou položkách. První možnost je v tomto případě v podstatě nemožné použít, druhá možnost je reálná a implementovaný nástroj lze poměrně snadno přizpůsobit tak, aby byla funkční. Bylo by potřeba pouze analyzovat výraz a rozdělit jej na sekvenci jednoduchých operací, které by se potom ve správném pořadí provedly.

Samotnému výpočtu různých funkcí nad rastry předchází několik důležitých algoritmů. V případě, že se operace provádí nad dvěma rastry, je nutné nejprve zjistit, jestli se tyto rastry překrývají. Pokud ano, má smysl pokračovat dále ke kontrole shodnosti rozměrů buněk. V případě operace nad jedním rastrem, respektive jedním rastrem a číslem, je překrytí nesmyslný požadavek, takže se pokračuje přímo dále i bez kontroly velikosti buněk, která v tomto případě také nemá význam.

Pokud mají rastry shodně velké buňky, pokračuje se dále. Pokud ne, musí být jeden nebo oba převzorkovány (např. pomocí předchozího nástroje) na stejnou velikost buněk. Převzorkování se může týkat obou rastrů v případě, že jeden má vyšší a užší buňky a druhý nižší a širší. V obou směrech se vždy vybere nejmenší možná hodnota, která je v rastrech obsažena a na tuto velikost buněk se rastry převzorkují. Pokud má jeden rastr menší obě hodnoty rozměrů buněk, neděje se s ním nic, kdežto druhý rastr musí být převzorkován. Pro převzorkování je automaticky použita metoda nejbližšího souseda. Pokud chce uživatel využít jinou, musí převzorkovat rastr ručně (ale zakomponovat nastavení typu převzorkovací funkce není složité).

Po případné kontrole překrytí a převzorkování rastrů následuje ještě výpočet, který odvodí datový typ dat, která budou ve výsledném rastru uložena. V případě operace dělení to bude například vždy typ `double`. U ostatních operací jako je sčítání, odčítání, násobení apod. se typ určí dynamicky podle toho, jaká maxima se mohou v novém rastru vyskytnout. Ze statistik vstupních rastrů se zjistí minimální a maximální hodnoty. Nad nimi se provede požadovaná operace a podle minimální a maximální hodnoty z výsledků se určí potřebný typ, aby byla data uložena bez ztráty informace. Toto odvození ale není vždy správné, protože počítá nejhorší možné varianty, které však nemusí ve výsledku vůbec nastat. Vždy je ještě provedena kontrola, jestli nemá některý ze vstupních rastrů datový typ s vyšší přesností, který se případně použije. Toto řešení je ale stále nedokonalé, takže byla přidána ještě jiná možnost. Tato možnost dovoluje uživateli samotnému vybrat datový typ, jaký si pro výsledný rastr představuje. Tím se vynechají zmíněné výpočty a rastru se předem nastaví vybraný datový typ (typ se nastavuje při vytváření rastru, takže je potřeba jej znát dříve, než proběhne samotný výpočet).

Teprve nyní lze provést první krok k vytvoření výsledného rastru. Vytvoří se rastr (v paměti nebo na disku) a nastaví se mu georeference, rozměry buněk a šířka a výška v buňkách (pixelech). Do tohoto rastru budeme dále při výpočtu ukládat výsledné hodnoty buněk. Rastr v paměti funguje dobře, ale nakonec je nutné jej stejně uložit na disk, protože jinak s ním má ArcGIS problémy a nedokáže zjistit statistiky, histogram apod., nehledě na to, že není rastr perzistentní.

Operace, která se bude provádět, se nastaví pomocí delegáta (obdoba odkazu na funkci v C++) na odpovídající funkci, což zvyšuje efektivitu výpočtu. To se ukazuje být výhodou

při přidávání dalších funkcí a operátorů do mapové algebry, které se tím velmi zjednoduší. Všechny operátory a funkce jsou přitom definovány na jednom místě a lze je jednoduše změnit a přidat.

Na druhou stranu se tím snižuje efektivita, protože výpočet probíhá vždy nad datovým typem `double`, což samo o sobě není příliš efektivní. Bohužel není jednoduché zajistit všechny možnosti datových typů, které mohou nastat při výpočtu, a proto se používá vždy tento typ. Následuje ukázka kódu jedné takové funkce včetně definice delegáta:

```
delegate double MyOperation (double a, double b);
static double op_plus (double a, double b)
{
    return a + b;
}
```

Po provedení předchozích kroků lze konečně přistoupit k samotnému výpočtu nových hodnot. Pokud rastry nemají stejné georeference, musí se nejprve zjistit, ve kterém bodě se začínají navzájem překrývat a odtud výpočet začne. Připomeňme, že velikosti buněk jsou již stejné, takže lze procházet oběma rastry v cyklech a číst hodnoty jednotlivých buněk, které si tím pádem odpovídají v prostoru. Pro procházení jednotlivých buněk rastru poskytuje *ArcObjects* rozhraní `IPixelBlock`, které načte blok buněk ze zadaných souřadnic o zadaných rozměrech. Tento přístup by měl být efektivnější než přímý přístup k jednotlivým buňkám rastru, protože se načte najednou větší pole hodnot, které lze pak procházet v paměti.

Z přečtených hodnot je vypočtena nová hodnota, která je pak uložena v požadovaném datovém typu do výsledného rastru. Zápis v požadovaném typu se provádí pomocí dalšího odkazu na funkci. Tak lze totiž jednoduše zajistit, aby výsledek mohl být ve správném datovém typu aniž by se daný typ musel vyhodnocovat při každém zápisu hodnoty.

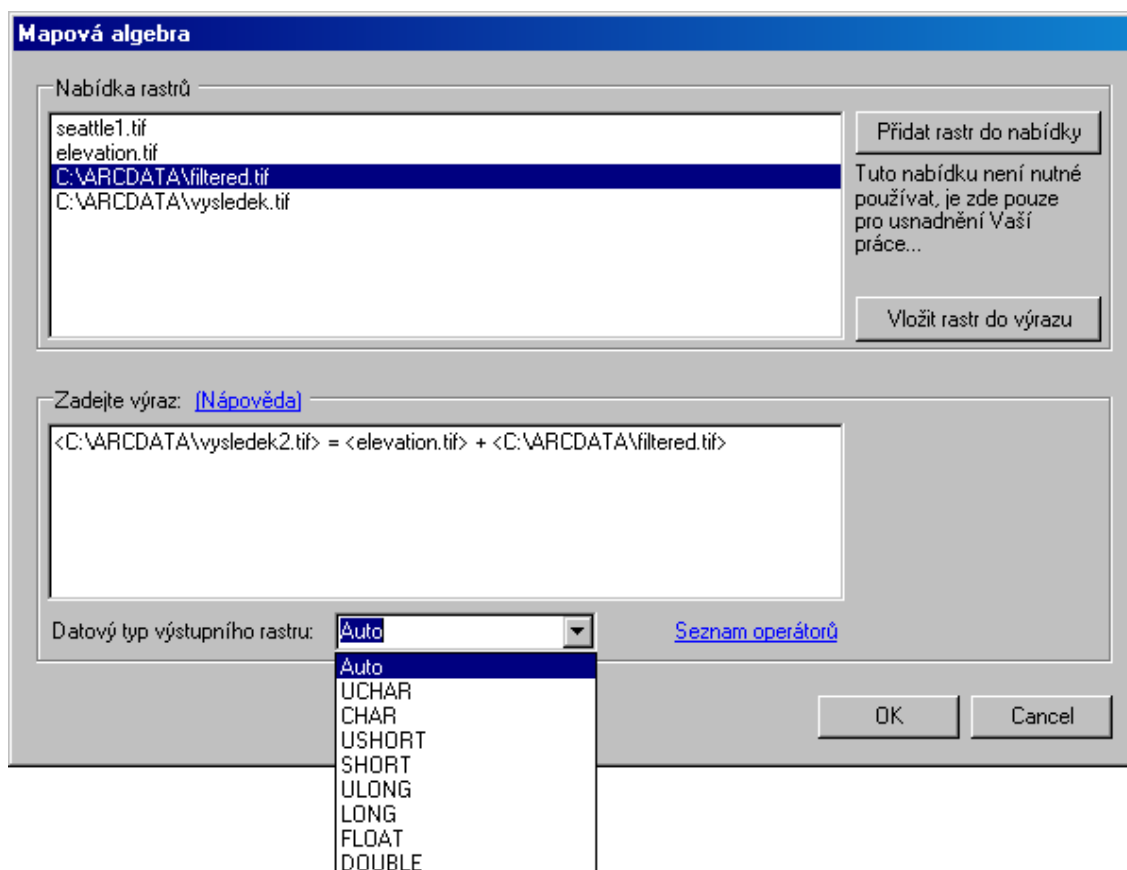
Po dokončení cyklů procházení rastry a výpočtu nového rastru je rastr uložen na disk a zobrazen do aktivní mapy v programu ArcMap.

Grafické rozhraní pro tyto funkce je poměrně jednoduché. Je zobrazeno na obrázku 5.5. Horní okénko slouží jako pomůcka, aby uživatel nemusel vypisovat jména rastrů, ale mohl je vložit poklikáním na jejich jméno, stisknutím klávesy Enter, nebo tlačítka *Vložit rastr do výrazu*. Do tohoto prostoru se automaticky přidávají všechny rastrové vrstvy otevřené v aktuální mapě. Pro přidání dalších je zde tlačítko *Přidat rastr do nabídky*. Po jeho stisknutí se otevře dialog pro výběr rastru.

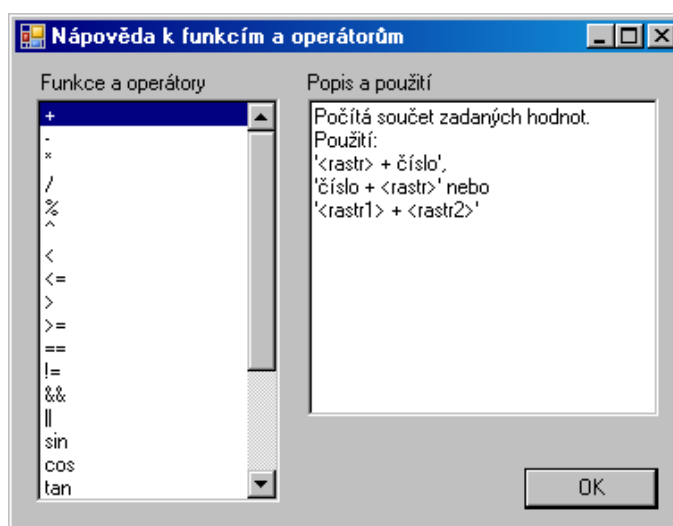
Po stisknutí modrého odkazu *Nápověda* se zobrazí nápověda, ve které jsou popsány jednotlivé možné tvary zápisu výrazu, které lze použít. Vstupem musí být vždy minimálně jeden rastr. Druhý vstup u funkcí nelze zadat, u binárních operátorů se očekává buď číslo, nebo další rastr. Číslo může být zadáno jako celé i jako desetinné, a může být prvním nebo druhým parametrem operace. Aby bylo možné rozlišit jméno rastru od jména funkce, musí být jméno rastru vždy uzavřeno v ostrých závorkách (např. `<elevation.tif>`).

Další modrý odkaz s popiskem *Seznam operátorů* slouží k zobrazení okna s nápovědou k operátorům a funkcím. V okně se zobrazí seznam všech použitelných operátorů a funkcí a po kliknutí na danou položku se zobrazí textový popis včetně návodu k použití (syntaxe). Ukázka okna s nápovědou je na obrázku 5.6.

Jméno výstupního rastru je od zbytku výrazu odděleno znakem `=`. Z obou stran rovnítka musí být zadány mezery, stejně jako u všech dalších operátorů. Vstupní rastr nemusí být uveden vůbec, což vede k uložení výsledku do adresáře pro dočasné soubory. Teoreticky by bylo možné ponechat výstupní rastr pouze v paměti, ale jak již bylo dříve zmíněno, ArcGIS s takovým rastrem neumí zcela korektně pracovat. Datový typ výstupního rastru



Obrázek 5.5: Ukázka okna mapové algebry



Obrázek 5.6: Ukázka okna s nápovědou k funkcím a operátorům mapové algebry

lze vybrat z vysouvací nabídky ve spodní části okna. Ten může být pevně určen uživatelem nebo vyhodnocen automaticky ze vstupních dat.

### 5.4.2 Problémy při implementaci

Při implementaci některých funkcí, které používají filtry, jako je např. funkce  $slope(rastr)$ , jsem narazil na malý problém. Pokud zadáme této funkci rastr otevřený v mapě, je předán pouze odkazem. Protože se na daný rastr používají dva různé filtry a poté probíhá výpočet nad dvěma rastry, které jsou v tomto případě stejné, výsledek nebyl správný a navíc se změnil původní rastr v mapě. Všechny rastry se tak musely načítat z disku a každý zvlášť, i když byly stejné. To lze ale ošetřit pomocí rozhraní `IClone`, které zajistí vytvoření identického rastru do paměti. Tím se jednoduše ošetří problém s přepisováním původního rastru v mapě.

## 5.5 Aspect - orientace svahů

Dalším algoritmem implementovaným v knihovně je algoritmus pro výpočet orientace svahů. Ten je často používaný jako základ dalších analýz, jako je například hledání linií horských hřebenů, údolí apod.

### 5.5.1 Implementace výpočtu orientace svahů

Princip výpočtu použitého v implementaci byl již zmíněn v kapitole 2.5.3, zde jej trochu více upřesním. Výhodou vybraného přístupu je to, že rastrové filtry nabízí knihovna *ArcObjects* (na podobném principu funguje i reklasifikace), a na výsledky filtrovaného rastru lze použít již implementovanou rastrovou algebru, aniž by bylo nutné znovu programovat průchod rastrem po buňkách. Zmíněné filtry vypočítají to, co vzorce 2.1 a 2.2, ale výsledek uloží zpět do rastru (v paměti, ne na disk). Do mapové algebry bylo nutné pouze přidat novou funkci, která výpočet zajistí. Tato funkce je obdobou vzorce 2.3. Protože se tak z fokální funkce stane lokální, nepracuje se dále s okolím buněk, to je totiž zaneseno do samotných hodnot v rastrech po filtrování.

Zmíněné filtry mají podobu matic o rozměrech 3x3 (viz obrázek 5.7), ve kterých jsou zadána čísla podobná, jako v uvedených vzorcích. Jde o filtry, které počítají sklon svahu ve směru na jih (vlevo) a na západ (anglicky gradient south a gradient west). Tyto filtry jsou implementovány v *ArcObjects* jako konstanty ve výčtu `esriRasterFilterTypeEnum`: `esriRasterFilterGradientSouth` a `esriRasterFilterGradientWest`.

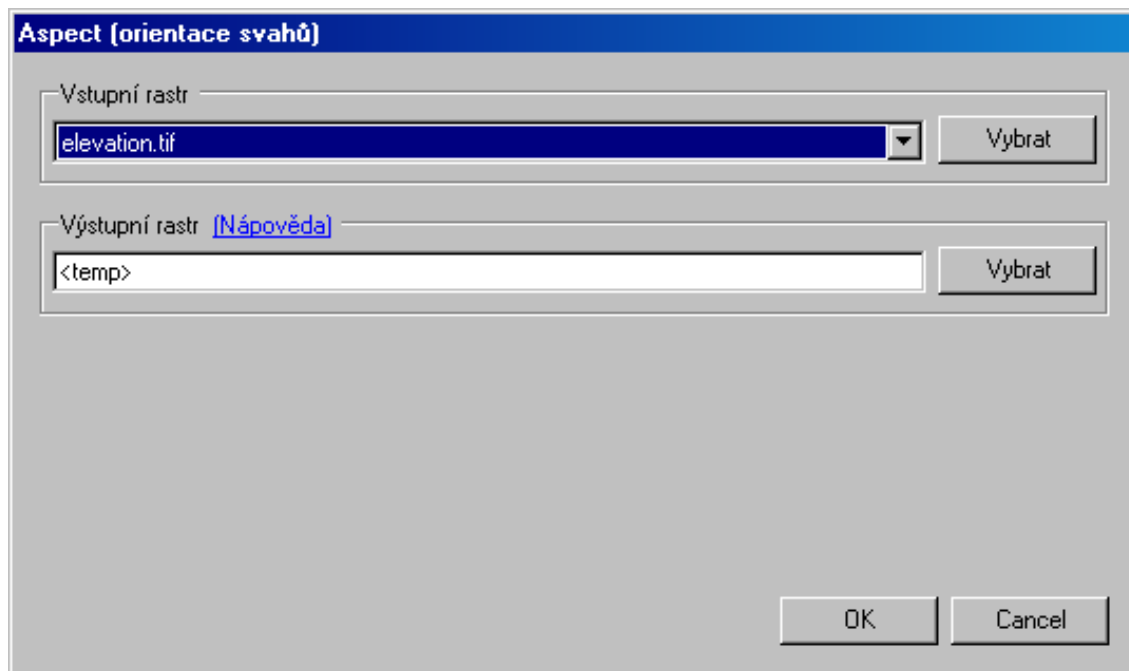
|    |    |    |
|----|----|----|
| 1  | 2  | 1  |
| 0  | 0  | 0  |
| -1 | -2 | -1 |

|    |   |   |
|----|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

Obrázek 5.7: Matice filtrů použitých při výpočtu sklonu a orientace svahů

Při výpočtu jde o rozdělení jedné operace na dvě, které budou provedeny sekvenčně. Nejprve se vypočítají rastry odpovídající výsledkům vzorců, a na ně se potom aplikuje lokální funkce mapové algebry pro výpočet orientace svahů.

Funkce pro výpočet sklonu svahů je přímo začleněna do mapové algebry (jako funkce `aspect()`), ale její použití je možné také přes samostatné okno (viz obrázek 5.8) věnované pouze této funkci. Je to totiž často používaná funkce a tento přístup je jednodušší, než použití v mapové algebře.



Obrázek 5.8: Ukázka okna pro výpočet orientace svahů

Uživatelské rozhraní je velice jednoduché. Jediné, co je pro tuto funkci potřebné, je zadat cestu ke vstupnímu a výstupnímu souboru. Výpočet proběhne po stisknutí tlačítka *OK*. Pokud nebyl zadán správný vstupní rastr, nebo byl jako výstupní rastr zadán existující soubor, funkce na to upozorní a požaduje změnu. Tato vlastnost je odvozena z chování programu ArcGIS, není to pouhé programátorské zjednodušení, jak obejít složité mazání souborů souvisejících s daným rastrem.

### 5.5.2 Problémy při implementaci

Při implementaci jsem narazil na problém, který vznikl na okrajích rastru. Po filtrování se procházely všechny buňky lokálně, na okrajích rastrů ale vznikaly hodnoty, které neodpovídaly realitě. To proto, že uvedené filtry na okrajích rastru počítají nesmyslné hodnoty. Bylo tedy nutné upravit pro tento případ mapovou algebru tak, aby počítala sklon a orientaci svahů pro všechny buňky rastru mimo těch, které se nacházejí na jeho okrajích. Na okraje se vkládá konstanta. V nastavbě Spatial Analyst na okrajích také vznikají odlišné hodnoty, než se očekává, ale nejsou konstantní, nýbrž vypočtené z dostupných dat. Protože jsou ale stejně nepoužitelné, rozhodl jsem se je pro jednoduchost nahradit konstantou. Výsledky obou implementací jsou téměř shodné.

## 5.6 Slope - sklon svahů

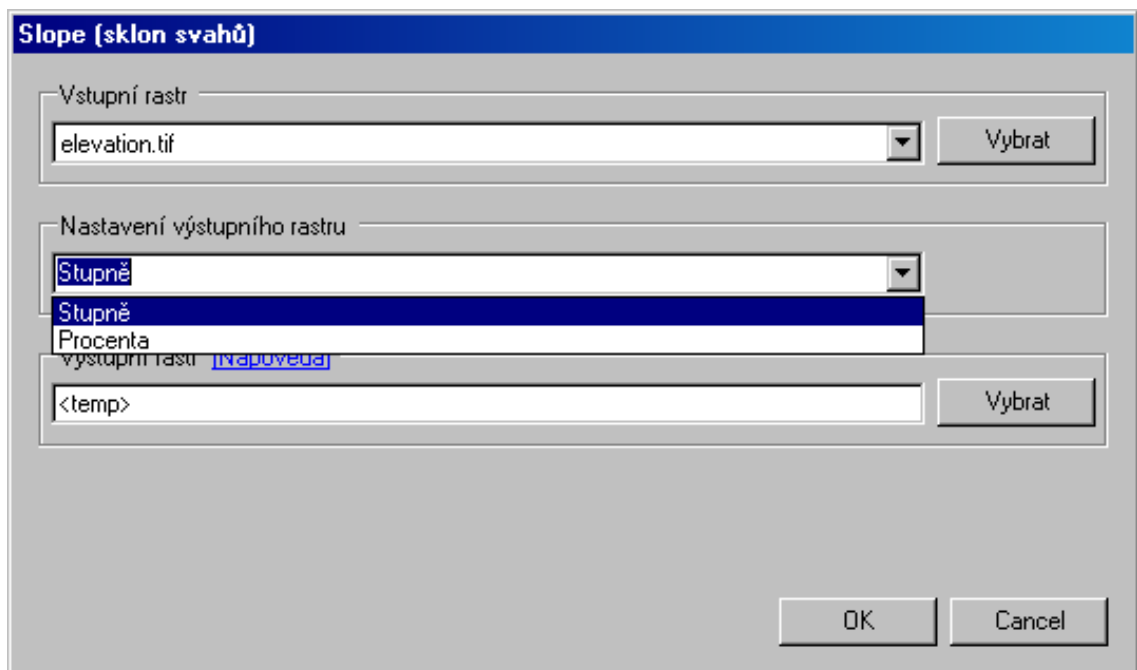
Na algoritmus pro výpočet orientace svahů navazuje velmi podobný algoritmus pro výpočet sklonu svahů. Sklon svahů lze využít při výpočtech směru odtoku vody, pravděpodobnosti šíření požárů, proudění apod.

### 5.6.1 Implementace výpočtu sklonu svahů

Princip výpočtu byl probrán v kapitole 2.5.4. Protože je velmi podobný předchozímu algoritmu, nebude zde popsán tak podrobně, aby se údaje neopakovaly. Oba výpočty by bylo teoreticky možné spojit do jednoho, ale mapová algebra není naprogramována tak, aby mohla ukládat do dvou různých rastrů najednou, takže jsou obě funkce prováděny zvlášť.

Algoritmus funguje tak, že na vstupní rastr aplikuje dva filtry (viz algoritmus 5.5), které vypočtou to, co vzorce 2.1 a 2.2. Protože filtry fungují na fokálním principu, není dále nutné počítat s okolím buněk, protože to je již započteno použitím filtrů. Na takto filtrované rastry potom aplikujeme jednoduchý výpočet podle vzorců 2.6 nebo 2.7. První zmíněný počítá sklon svahů v procentech, druhý ve stupních. Který z nich se bude počítat záleží na uživateli, který si může vybrat.

Výpočet sklonu svahů je implementován v mapové algebře. Pro každý ze dvou možných způsobů výpočtu je nadefinována zvláštní funkce, protože se pokaždé počítají trochu odlišné věci. Jelikož se ale využívá odkazů na funkce, je tento přístup ve výsledku trochu efektivnější. Tím, že jsou funkce pro výpočet implementovány v mapové algebře, je možné je použít jak tam, tak i v dalších funkcích mimo mapovou algebru. V mapové algebře se sklon svahů vypočte pomocí funkce `slope_deg()` ve stupních, případně pomocí `slope_perc()` v procentech. V závorkách se pak uvede jméno vstupního rastru.



Obrázek 5.9: Ukázka okna pro výpočet sklonu svahů

Aby měl uživatel přístup k výpočtu sklonu svahů i bez použití mapové algebry, vytvořil

jsem k této funkci také grafické rozhraní, které je na obrázku 5.9. Prostředí je poměrně jednoduché. Pro výpočet je potřeba nastavit vstupní a výstupní rastr a vybrat výsledný typ hodnot – stupně nebo procenta. Pokud je zadáno vše správně, po stisknutí tlačítka *OK* se provede výpočet. Problémy, na které jsem při implementaci narazil jsou shodné s předchozím algoritmem, protože oba fungují na podobném principu.

## 5.7 Filtrování rastrů

Posledním implementovaným nástrojem je nástroj, pomocí kterého lze na rastry použít filtry. Filtrem je zde myšlena matice čísel s udaným středem. Tato matice se pak aplikuje na každou buňku rastru, a to přiložením středu matice filtru (jádra) na danou buňku a výpočtem nad buňkou samotnou i jejím okolím. Hodnotu výsledné buňky ovlivňují původní hodnoty ve spojení s koeficienty matice. Takové filtrování rastrů je již použito v předchozích nástrojích pro výpočet sklonu a orientace svahů.

Teoreticky mohou být matice libovolných rozměrů, ale nejčastěji jsou používány matice čtvercového tvaru s lichým počtem řádků a sloupců. Je to tak proto, že takové matice jsou symetrické a mají jednu středovou buňku (jádro filtru). Protože použití matic jiných tvarů nemá příliš význam a takové matice lze nahradit i čtvercovými, není jejich použití v implementovaném nástroji dovoleno.

### 5.7.1 Implementace filtrování rastrů

Standardní rozměry matic filtrů jsou 3x3 nebo 5x5. Několik takových filtrů lze využít přímo v *ArcObjects*, ty nejpoužívanější jsou již přednastaveny. Pro jejich použití je vytvořena třída *IStockConvolutionFilter*, ukázka následuje níže. Dále poskytuje knihovna *ArcObjects* možnost definice uživatelského filtru, takže je možné vytvořit téměř libovolný filtr. Tento filtr je vytvořen třídou *IRasterConvolutionFilter* a jeho vytvoření a nastavení je následovné:

```
// přednastavený filtr:
IStockConvolutionFilter filter;
filter = (IStockConvolutionFilter) new RasterConvolutionFilterClass();
filter.Type = esriRasterFilterTypeEnum.esriRasterFilterGradientWest;

// uživatelský filtr:
IRasterConvolutionFilter pf = new RasterConvolutionFilterClass ();
double[] coefList = new double[9] {-1, 0, 1, -2, 0, 2, -1, 0, 1};
pf.PutCoefficients (ref coefList); // nastavení koeficientů filtru
pf.SetCenterPosition (1, 1); // indexy se počítají od 0
pf.ScaleBy (1.0/8.0); // měřítko
```

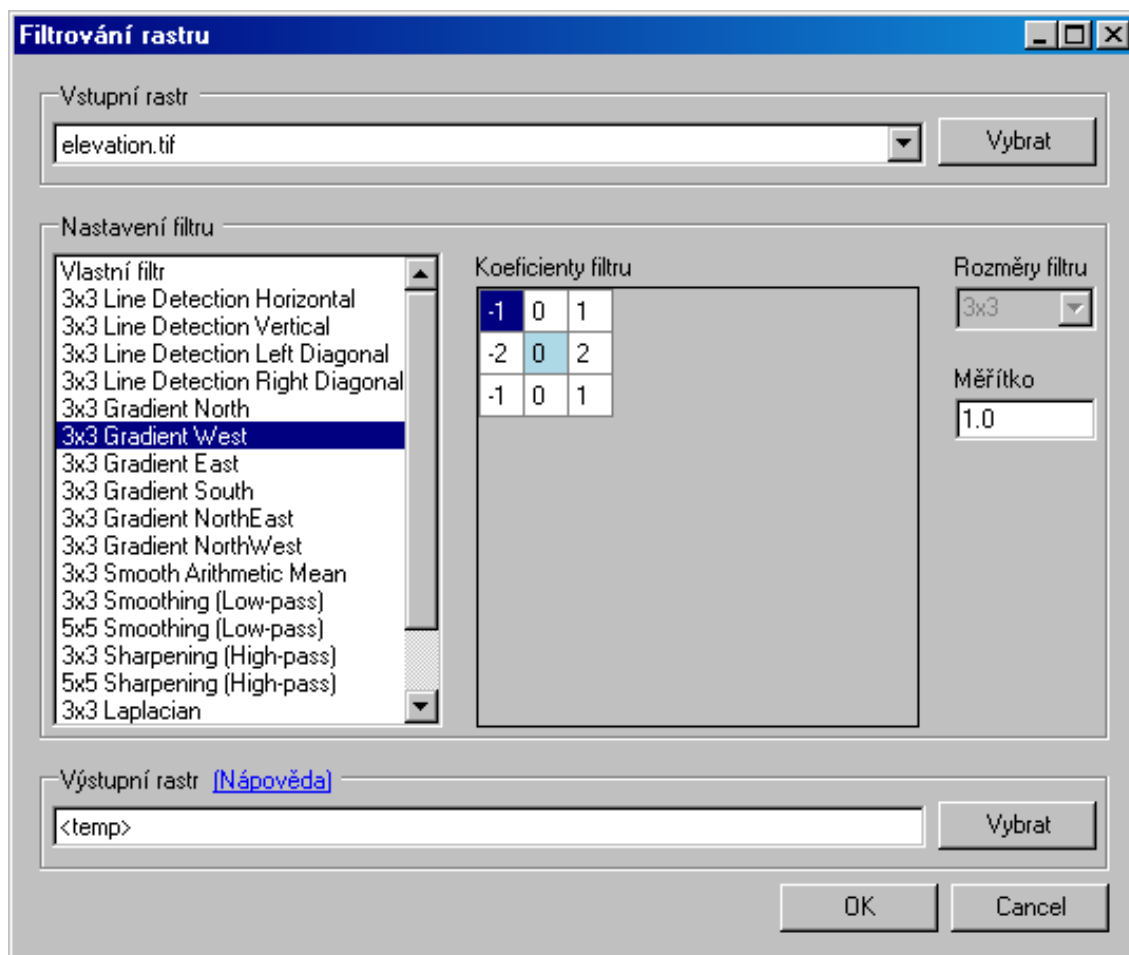
Pro správnou funkci uživatelského filtru je nutné mít nainstalovaný ArcGIS Service Pack 3 nebo vyšší. Výsledek výpočtu filtru nad rastrem je dán součtem jednotlivých součinů buněk filtru a buněk rastru. Po přiložení středu filtru nad danou buňku se vynásobí obsah koeficientů matice filtru s odpovídajícími buňkami rastru a tyto součiny se pak sečtou. Dále je většinou požadováno, aby byl tento součet upraven měřítkem. To udává např. součet všech buněk (vah) ve filtru, takže výsledkem je vážený průměr. Tím se potom výsledné hodnoty drží v intervalu hodnot přibližně odpovídajícím původnímu rastru.

Nevýhodou přednastavených filtrů je nemožnost nastavit jim měřítko, s jakým mají počítat výsledky. Jde o to, že jejich výsledky je nutné většinou ještě dále upravovat. Lze



to ale obejít pomocí uživatelských filtrů, kterým toto měřítko již nastavit lze. Na těch je vystavěn můj vlastní nástroj pro filtrování rastrů. Ten nabízí stejné filtry jako jsou ty přednastavené a navíc možnost vytvořit vlastní filtr.

Grafické rozhraní se jako obvykle skládá ze seznamu pro výběr vstupního rastru a políčka pro zadávání jména výstupního rastru. Obě zmíněné komponenty fungují stejně jako ve všech ostatních nástrojích.



Obrázek 5.10: Ukázka okna nástroje pro filtrování rastrů

Ve střední části okna je seznam pro výběr některého z předdefinovaných rastrů. Jako první je uveden uživatelsky definovatelný filtr, ostatní jsou konstantní. Po výběru některého z filtrů se jeho koeficienty zobrazí v tabulce vedle seznamu, aby uživatel věděl, o jaký filtr se jedná. Středová buňka je zvýrazněna odlišnou barvou. Protože jsou předdefinované filtry konstantní, nelze u nich měnit hodnoty ani rozměry filtru. Tyto hodnoty lze měnit pouze v případě uživatelského filtru. Dovolené rozměry filtru jsou 3x3, 5x5, 7x7 a 9x9. (Není složité v případě potřeby přidat jiné rozměry, ale je potřeba to udělat ve zdrojových kódech programu, které pro jednoduchost nabízejí pouze tyto rozměry.) U všech filtrů lze samozřejmě nastavit požadované měřítko. Po stisknutí tlačítka *OK* se provede výpočet a výsledný přefiltrovaný rastr se zobrazí jako nová vrstva v aktuální mapě v programu.

### 5.7.2 Problémy při implementaci

Při implementaci jsem narazil na několik problémů. Protože jsem přednastavené filtry používal již dříve, zarazilo mě, když nefungovaly uživatelské a dlouho jsem hledal řešení. To mě přivedlo k nainstalování balíčku *Service Pack 3*, který již byl několikrát zmíněn, protože opravuje mnoho dalších chyb, čímž se tento problém vyřešil.

V souvislosti s instalací opravného balíčku nastal jeden další problém, kterého jsem si naštěstí všiml při testování. Při programování funkce *slope* pro výpočet sklonu svahů jsem se řídil výsledky, které dávala oficiální nadstavba pro rastrovou analýzu *Spatial Analyst*. O to větší bylo mé překvapení, když se výsledky těchto výpočtů v samotné oficiální nadstavbě po instalaci opravného balíčku lišily. Chyba byla zřejmá, protože se nepočítalo se šířkou buňky a výsledky odpovídaly tomu, že šířka všech buněk je vždy 1, tím pádem byl sklon vypočten špatně a neodpovídal realitě. Oprava byla snadná, ale neumím si představit, jak takové věci řeší velké firmy, které oficiální nadstavby používají a výsledky např. dále prodávají. Jakou má potom kupující jistotu, že jsou data správná?

Problém, který se mi nepodařilo bohužel vyřešit, je v tom, jak filtry počítají hodnoty na okrajích rastru. Místo aby počítaly pouze s dostupnými daty, berou v potaz i data neexistující (mimo rastr). Tak na okrajích vznikají neočekávané hodnoty jako nekonečno nebo NaN (Not a Number). Tyto hodnoty jsou proto po přefiltrování nahrazeny konstantou.

## Kapitola 6

# Testování knihovny

Testování funkčnosti jednotlivých nástrojů jsem prováděl souběžně s jejich vývojem. Výstup vyvíjeného nástroje jsem vždy porovnal s výstupem oficiální nadstavby pro stejný testovací rastr. Toto průběžné testování probíhalo na malých rastrech, protože operace nad většími rastry jsou výpočetně (a tím i časově) náročnější, a to v průběhu ladění nástrojů není žádoucí. Pro rychlé otestování správné funkčnosti proto stačí i menší rastr. Testy byly prováděny zejména na jednovrstvých rastrech.

Používal jsem některé rastry, které byly obsaženy v dokumentaci SDK, ale později jsem začal používat digitální výškový model, který se více hodí k některým nástrojům. Samozřejmě lze tento rastr použít pro všechny nástroje, ale zpočátku stačily k testování rastry výčtového typu (pro reklasifikaci, převzorkování i základy mapové algebry). Rastr s výškovým modelem, který jsem používal, lze najít na internetové stránce uvedené v literatuře [15]. Jeho rozměry jsou přibližně 800x800 buněk a hodnoty jsou uloženy v datovém typu `Double`.

Na začátku, kdy jsem nebyl příliš seznámen s prostředím ArcGIS se mi výsledné rastry vždy zobrazovaly v mapě na jiném místě, než originální vstupní rastr. To bylo způsobeno tím, že se neukládaly georeference do výstupního souboru, což opravila instalace opravného balíčku. Od této instalace vše funguje jak má.

Všechny nástroje jsem se snažil testovat na rastrech s různými datovými typy buněk. Jejich tvorba nebyla žádný problém po vyvinutí nástroje s mapovou algebrou, který dovoluje vytvořit rastr s možností výběru různých datových typů. Nejsou ale otestovány zdaleka všechny typy rastrů, které ArcGIS podporuje, protože těch jsou stovky. Protože programátor pracuje s rastry na poměrně vysoké úrovni abstrakce, neměl by být ve zpracování jiných typů rastrů žádný problém. Ve většině případů se totiž jedná také o rastry s pravidelným (čtvercovým nebo obdélníkovým) tvarem buněk.

### 6.1 Porovnání výstupních rastrů a výkonnosti nástrojů

Pro závěrečné testování výkonu a funkčnosti nástrojů jsem použil větší rastr, a to rastr o rozměrech přibližně 6500x6000 buněk. To je přibližně 40 milionů buněk. Tento rastr lze nalézt na adrese uvedené v literatuře [17]. Na první pohled je jasné, že výpočty byly zdlouhavější, než s předchozím rastrem a již se nejednalo o výpočty v časech řádově jednotek sekund. Datový typ buněk je v originálu celočíselný, ale z jistých důvodů (uvedených později) jsem používal i jeho kopii v datovém typu `Float`. Výstupní rastry jsem porovnával u všech nástrojů, protože jediné tak lze poznat, jestli fungují naprogramované algoritmy správně. Bylo by vhodné předem říci, že výsledky výpočtu nad rastrem otevřeným v mapě nebo z disku

jsou stejné, ale časy výpočtu se u nich v některých případech mohou lišit i o několik vteřin.

### 6.1.1 Reklasifikace rastru

V případě reklasifikace jsem testoval funkčnost nad celočíselnou verzí testovacího rastru tak, že jsem data ze vstupního rastru rozdělil na několik skupin a ty jsem nahradil výčtovým typem. Bohužel tento rastr se mi nepodařilo reklasifikovat s oficiální nadstavbou, protože ArcGIS odmítl tento rastr z nepochopitelných důvodů zpracovat. Výsledky reklasifikace jiných rastrů, které se mi podařilo zpracovat i v ArcGISu pomocí nadstavby Spatial Analyst byly shodné s mými. Je to dáno především tím, že je použit stejný filtr pro vytvoření reklasifikačních pravidel.

Reklasifikace mým nástrojem trvala 56 sekund, délku trvání reklasifikace s pomocí nadstavby Spatial Analyst nebylo možné z důvodu nemožnosti provedení reklasifikace zjistit. Při testování nad dříve zmíněným menším rastrem byl výpočet reklasifikace ukončen za necelé 2 vteřiny v obou implementacích a výsledné rastry byly shodné. U takto malých rastrů téměř není možné časový rozdíl upozorovat.

Až při testování nad velkým rastrem jsem narazil na jednu velkou nevýhodu mých nástrojů. Protože programování vláken souběžně s použitím *ArcObjects* je poměrně složité a předávání vstupních parametrů vláknům ještě složitější, programoval jsem bez využití vláken. U velkého rastru však výpočet trvá delší dobu a ArcGIS je po tuto dobu jinak nepoužitelný. V oficiální nadstavbě je možné nechat běžet výpočet na pozadí a pracovat dále (i když s obtížemi, protože výpočet zabírá poměrně hodně času procesoru i místa v paměti). To je jednou z největších nevýhod implementované knihovny a také jedním z návrhů na možné vylepšení.

### 6.1.2 Převzorkování rastru

Převzorkování rastru několika různými metodami jsem provedl nad celočíselným velkým rastrem. Rozdíly mezi mým nástrojem a nástrojem z nadstavby Spatial Analyst nebyly v časech výpočtu nijak zvlášť odlišné. To je dáno tím, že k převzorkování rastru využívám standardních metod z *ArcObjects*. Zajímavé je, že výsledné rastry jsou odlišné i při použití stejných metod převzorkování a stejných parametrů výstupního rastru. Hodnoty se ale liší minimálně a převzorkovací metody jsou založeny na rychlosti a ne na přesnosti, takže je pravděpodobné, že rozdíly jsou způsobeny tímto.

Převzorkování rastru z původní velikosti buňky 30 na 10 trvalo 5 minut a výsledný rastr zabíral asi 1,5 GB. To byl pouze pokus pro zjištění, jak dlouho takové převzorkování trvá. Při převzorkování na větší velikost buněk je výpočet daleko rychlejší. Oba nástroje jej zvládají při různých metodách pro velikost buňky 100 do deseti vteřin. Pouze u metody převažující hodnoty je výpočet delší, a to 14 vteřin, shodně u obou nástrojů. V některých případech je můj nástroj o několik vteřin rychlejší. To je pravděpodobně způsobeno nějakou režii vykonávanou nástrojem z oficiální nadstavby navíc. Jinak jsou rychlosti převzorkování srovnatelné.

### 6.1.3 Mapová algebra

Mapovou algebru jsem testoval nad velkým rastrem jak s datovým typem `Int`, tak `Float`. Výsledné rastry odpovídaly pro jednoduchost dvojnásobku vstupního, ale mapová algebra se používá i v dalších algoritmech, takže byla otestována dostatečně.

Výsledné rastry byly v případě oficiální nadstavby i v případě mého nástroje shodné. Avšak čas, za který výpočty jednotlivé implementace provedly, se lišil. Pro velký rastr s `Float` hodnotami počítal mnou implementovaný nástroj 140 vteřin, `Spatial Analyst` pouze 28 vteřin. To mne velmi překvapilo, protože jsem doufal, že rozdíl nebude tak obrovský. Pravděpodobně je to dáno tím, že stále všechna čísla převádím na typ `Double`, a to není efektivní (ani z typu `Float`). V případě celočíselného rastru jsem podobný průběh očekával. Tam můj nástroj počítal 122 vteřin, `Spatial Analyst` pouhých 12, což je desetina času.

Nad malým testovacím rastrem proběhly výpočty do 5 vteřin v obou implementacích, nad některými malými rastry se dokonce dostal do vedení můj nástroj, což ale není u tak malých časů důležité a může to být také chyba měření.

#### 6.1.4 Aspect - orientace svahů

Nástroj pro výpočet orientace svahů z výškového modelu jsem testoval také nad oběma datovými typy. Zezačátku jsem testoval pouze nad celočíselným, ale protože výsledný rastr neodpovídal výsledkům vypočteným pomocí `Spatial Analyst`, provedl jsem pokusy i s rastrem s hodnotami typu `Float`, kde se výsledky již shodovaly.

Do té doby jsem tento nástroj testoval pouze nad datovými typy s plovoucí řádovou čárkou, takže jsem si této chyby ne všiml. Je způsobena tím, že výpočet probíhá pomocí filtrování původního rastru. Při filtrování se ale nemění datový typ rastru, takže hodnoty se zaokrouhlují a výsledky mohou být oříznuty. S takto špatně filtrovaným rastrem se pak provede pomocí mapové algebry výpočet, který samozřejmě bude špatně. Proto pro výpočet orientace i sklonu svahů je potřeba pracovat s rastry, které mají data uložená v datovém typu `Float` nebo `Double` (pokud mají jiný typ, lze je převést např. pomocí mapové algebry).

Z časového hlediska jsou výsledky mé implementace nad malými rastry ve srovnání s výsledky `Spatial Analyst` stále vyšší, ale jedná se o jednotky vteřin. Nad velkým rastrem trval výpočet pomocí mého nástroje 186 vteřin, pomocí `Spatial Analyst` to bylo 54 vteřin.

Výsledky z pohledu dat jsou téměř shodné. Slovo téměř je v tomto případě možná trochu zastrášující. Tento problém byl zmíněn již dříve v části popisující implementaci. Na okrajích rastru vznikají při výpočtu nereálné hodnoty. `Spatial Analyst` na okrajích nechává hodnoty vypočtené pouze z dostupných dat. Moje implementace využívá filtrů a okraje jsou proto problémové, takže v šířce jedné buňky jsou data z okrajů nahrazena konstantou. To je jediný rozdíl mezi výslednými rastry, který je u rozumně velkých rastrů zanedbatelný.

#### 6.1.5 Slope - sklon svahů

Výpočet sklonu svahů je velmi podobný předchozímu, a nese s sebou také stejné problémy. Platí zde tedy vše, co bylo řečeno v minulé podkapitole.

Výpočet nad malým rastrem se opět blíží přibližně 5 vteřinám, u `Spatial Analyst` je to něco přes jednu vteřinu. Rozdíl při výpočtu ve stupních nebo v procentech má na výsledný čas minimální vliv.

Výpočet prováděný nad velkým rastrem pomocí mého nástroje trval 201 vteřin, v případě použití nástroje ze `Spatial Analyst` to bylo 51 vteřin.

#### 6.1.6 Filtrování rastrů

Protože jsem nenašel v nadstavbě `Spatial Analyst` nástroj poskytující filtrování rastrů, nebylo v tomto případě bohužel s čím srovnávat. Pro orientaci jsem tak na velký rastr použil několik různých filtrů a změřil výsledné časy. Ty se liší podle složitosti a rozměrů filtru,

u těch složitějších časy dosahovaly až 160 vteřin. Zde mají vliv jak samotné koeficienty filtru, tak to, že je po přefiltrování nutné projít rastrem a odstranit nesmyslné hodnoty na okrajích.

Zde je potřeba počítat s tím, že datový typ výsledného rastru je stejný, jako datový typ vstupního rastru.

## 6.2 Shrnutí

Výsledky testování z pohledu funkčnosti nástrojů jsou výborné. Vše se zdá být funkční a výsledky odpovídají výsledkům nadstavby Spatial Analyst.

Z časového hlediska již výsledky nejsou tak uspokojující, nicméně již při návrhu bylo vidět, že výkon vlastní knihovny nebude zdaleka dosahovat takových hodnot, jako oficiální nadstavba.

Z výsledků testování vyplývá, že by bylo potřeba zapracovat na efektivitě zpracování. To probíhá nad hodnotami typu `Double` a převody na tento typ a zpět na typ ukládaný do rastru může být zdoluhavý. Další vylepšení, které by zlepšilo práci s knihovnou, by mohlo přidat využití vláken, aby bylo možné nechat běžet složitější výpočty na pozadí a dále pracovat.

# Kapitola 7

## Závěr

### 7.1 Zhodnocení výsledků

Výsledná knihovna s rozšířením dodává několik nejpoužívanějších nástrojů rastrové analýzy. Ty se do samotného nástroje ArcGIS přidávají automaticky instalací této knihovny a je možné je začít okamžitě používat. Bohužel nejsou implementovány všechny nástroje, které byly původně v plánu. Implementováno bylo několik základních algoritmů, ale chybí např. vzdálenostní analýzy, které jsou také velmi používaným nástrojem. Je to způsobeno tím, že složitost programování v poměrně neznámém prostředí se velmi špatně odhaduje a plán proto počítal s několika dalšími algoritmy navíc.

Nicméně výsledky testování nástrojů, které se podařilo implementovat, ukazují, že výsledek je v praxi použitelný, protože dává stejné výsledky jako nadstavba Spatial Analyst. Nutným předpokladem je samozřejmě to, že výsledky oficiální nadstavby jsou správné, protože podle nich jsem porovnával své výsledky i já. Časová náročnost při některých výpočtech je sice větší, ale důležité jsou správné výsledky. Výhodou knihovny je samozřejmě zejména to, že je vyvíjena tak, aby bylo možné ji používat a šířit bezplatně a není závislá na jakýchkoliv jiných rozšířeních nástroje ArcGIS. Protože není vyvíjena tak velkým týmem lidí, nedá se od ní očekávat, že bude dodávat stejný počet stejně rychlých a kvalitních nástrojů. Dá se ale očekávat, že se najdou další lidé, kteří budou pokračovat v rozšiřování a časem by z této knihovny mohl být daleko větší užitek.

Jedním z cílů této práce bylo také vytvořit návod pro ostatní programátory, kteří by se chtěli zapojit do dalšího rozšiřování nástroje ArcGIS, ať už pokračováním v práci na této knihovně, nebo vytvořením vlastní v jiném oboru analýz. Těchto možností je nespočet. Jednoduchý návod na implementaci a zprovoznění prvního vlastního nástroje se v této práci objevil v kapitole 3.4. Několik ukázek použití dalších algoritmů je průběžně v textu popisujícím implementaci, ty mohou být také užitečné. Přesto za největší návod na další programování považuji samotné zdrojové kódy, protože jedině tam programátor opravdu vidí, co se jak dělá. Proto jsou kódy dostatečně podrobně okomentované, aby bylo možné se v nich nejen vyznat, ale také je snadno pochopit.

Veškeré dostupné materiály jsou v angličtině a bohužel zdaleka ne všechna dokumentace je udělána přehledně. Mnoho funkcí implementovaných firmou ESRI v knihovnách *ArcObjects* není smysluplně dokumentováno a popsáno, což značně ztěžuje práci programátorů. Dokumentace tedy existuje, ale je v podstatě použitelná jen pro člověka, který *ArcObjects* zná. Pro začátečníka je nutné následovat příklady a experimentovat. Příklady se ale většinou netýkají rastrové analýzy, a proto není jednoduché odhalit všechny nabízené možnosti *ArcObjects* v tomto oboru.

Další vývoj projektu by mohl pokračovat přidáním některých algoritmů rastrové analýzy, především vzdálenostních analýz. Po jejich implementaci by měl být projekt připraven k plnému nasazení v předmětu Geografické informační systémy. Protože výpočty ve výsledné knihovně trvají poněkud déle, než v případě nadstavby od firmy ESRI, bylo by možné zapracovat na zlepšení efektivity a rozdělení výpočtů na vlákna. Rozšiřování stávající knihovny nebo implementace dalších lze brát jako návrhy na projekty ve zmíněném předmětu nebo (v případě větších projektů) jako další bakalářské či diplomové práce. Z tohoto pohledu je tato práce průkopníkem a tak trochu experimentem, zda je vůbec možné něco takového vytvořit. To se podařilo a je tedy možné dále pokračovat ve vývoji vlastních nástrojů.

## 7.2 Vlastní přínos

Přínos této práce vidím zejména v tom, že dodává několik nástrojů rastrové analýzy zdarma, oproti komerčnímu produktu firmy ESRI. Z tohoto důvodu také práce vznikla. Dalším přínosem je návod k programování podpořený podrobně komentovanými zdrojovými kódy, aby bylo možné dále pokračovat a rozšiřovat knihovnu o další nástroje nebo zvyšovat její efektivity.

Výsledná nadstavba je volně dostupná a to jak pro použití, tak pro úpravy. Z toho důvodu očekávám, že se najdou další programátoři, kteří se pokusí naprogramovat další z nadstaveb pro ArcGIS, a tím se jeho použitelnost v základních verzích může rozšířit bezplatně. Pro firmy s dostatkem financí na zakoupení oficiálních nadstaveb to samozřejmě není ideální náhrada, protože pravděpodobně nikdy nedosáhne takových kvalit. Pro menší firmy nebo podnikatele však může takové bezplatné rozšíření znamenat ušetření až statisícových částek.

Z mého pohledu bylo přínosem také to, že jsem značně prohloubil znalosti týkající se rastrové analýzy a naučil se rozšířit nástroj ArcGIS o vlastní nadstavby a pomůcky. Protože jsem dostal možnost vyzkoušet plnou verzi programu ArcGIS, mohl jsem si vyzkoušet i nadstavby, ke kterým v normální výuce ani ve škole není přístup. Kvalita nástrojů firmy ESRI je opravdu vysoká a jedná se o profesionální systém. Bohužel je občas nestabilní a končí chybou, což uprostřed práce rozhodně nepotěší. Proto je vhodné udržovat systém pomocí dostupných záplat a opravných balíčků, ty v některých případech opravují opravdu závažné problémy. Jak jsem se sám přesvědčil, může dokonce jít o problémy týkající se samotných dat, kdy před instalací a po instalaci opravného balíčku dávaly některé nástroje odlišné výsledky.



# Literatura

- [1] Martin Hrubý. Geografické informační systémy (gis) [studijní opora].  
<http://perchta.fit.vutbr.cz:8000/vyuka-gis/uploads/1/GIS-final2.pdf>, září 2006, (dostupné květen 2008).
- [2] WWW stránky. Arcgis 9.1 desktop functionality matrix.  
<http://download.arcdata.cz/doc/2005/ArcGIS-9.1-functionality.pdf>, (dostupné květen 2008).
- [3] WWW stránky. Arcgis desktop help 9.2.  
<http://webhelp.esri.com/arcgisdesktop/9.2/>, (dostupné květen 2008).
- [4] WWW stránky. Arcgis spatial analyst.  
<http://old.arcdata.cz/download/doc/2006/ArcGIS-Spatial-Analyst.pdf>, (dostupné květen 2008).
- [5] WWW stránky. Co je arcgis.  
<http://download.arcdata.cz/doc/2005/Co-je-ArcGIS-90.pdf>, (dostupné květen 2008).
- [6] WWW stránky. Create a command by inheriting from basecommand.  
<http://edndoc.esri.com/arcobjects/9.2/NET/01c01659-cdf8-4579-9c87-2b965e872d84.htm>, (dostupné květen 2008).
- [7] WWW stránky. Datasourcesraster. <http://edndoc.esri.com/arcobjects/9.2/NET/62937a09-b1c5-47d7-a1ac-f7a5daab3c89.htm>, (dostupné květen 2008).
- [8] WWW stránky. Developing with arcgis raster apis.  
[http://gis.esri.com/library/userconf/devsummit06/papers/raster\\_apis.pdf](http://gis.esri.com/library/userconf/devsummit06/papers/raster_apis.pdf), (dostupné květen 2008).
- [9] WWW stránky. Documentation for developers using arcgis 9.2.  
<http://edndoc.esri.com/arcobjects/9.2/>, (dostupné květen 2008).
- [10] WWW stránky. Geografický informační systém. [http://cs.wikipedia.org/wiki/Geografick%C3%BD\\_informa%C4%8Dn%C3%AD\\_syst%C3%A9m](http://cs.wikipedia.org/wiki/Geografick%C3%BD_informa%C4%8Dn%C3%AD_syst%C3%A9m), (dostupné květen 2008).
- [11] WWW stránky. Geoprocessing tools in arcgis 9.1 extensions. [http://download.arcdata.cz/doc/2006/geoprocessing-tools-extensions\\_1005.pdf](http://download.arcdata.cz/doc/2006/geoprocessing-tools-extensions_1005.pdf), (dostupné květen 2008).
- [12] WWW stránky. Geotiff specification.  
<http://www.remotesensing.org/geotiff/spec/contents.html>, (dostupné květen 2008).

- [13] WWW stránky. Gis and modeling. [http://www.colorado.edu/geography/class\\_homepages/geog\\_4203\\_s08/class6\\_TerrainAnSlopeAspect.pdf](http://www.colorado.edu/geography/class_homepages/geog_4203_s08/class6_TerrainAnSlopeAspect.pdf), (dostupné květen 2008).
- [14] WWW stránky. How to create a custom install program. <http://edndoc.esri.com/arcobjects/9.2/NET/0df20605-b457-42d6-b63c-341a3824474a.htm>, (dostupné květen 2008).
- [15] WWW stránky. Spatial data. [http://spatialdata.iu.edu/DOQQS/inmap05/elev/dem\\_tif\\_stpl\\_e/0508/05081288\\_dem\\_spe.zip](http://spatialdata.iu.edu/DOQQS/inmap05/elev/dem_tif_stpl_e/0508/05081288_dem_spe.zip), (dostupné květen 2008).
- [16] WWW stránky. Spatial data models. [bgis.sanbi.org/GIS-primer/page\\_15.htm](http://bgis.sanbi.org/GIS-primer/page_15.htm), (dostupné květen 2008).
- [17] WWW stránky. U.s. geological survey. [http://edcftp.cr.usgs.gov/pub/data/nedh/tar\\_files/stage3/orig\\_dem.tar](http://edcftp.cr.usgs.gov/pub/data/nedh/tar_files/stage3/orig_dem.tar), (dostupné květen 2008).
- [18] WWW stránky. What is raster data? [http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=What\\_is\\_raster\\_data?](http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=What_is_raster_data?), (dostupné květen 2008).