

Lekce 1: Úvod do automatizace GIS pomocí Model Builderu

Programování pro GIS

Fakulta životního prostředí, ČZU Praha

2025-01-01

Table of contents

1	Úvod do kurzu	3
1.1	Co se v tomto kurzu naučíte	3
1.1.1	Týdny 1-2	4
1.1.2	Týdny 3-5	4
1.1.3	Týdny 6-9	4
1.1.4	Týdny 10-11	4
1.1.5	Týden 12	4
1.2	Proč tento kurz?	4
1.3	Struktura výuky	5
2	Proč automatizovat GIS úlohy?	5
2.1	Motivační příklad	5
2.1.1	Řešení ručně	6
2.1.2	Řešení s automatizací	6
2.2	Výhody automatizace	6
2.3	Kdy automatizovat?	7
2.3.1	ANO - automatizujte když:	7
2.3.2	NE - neautomatizujte když:	7
3	Model Builder - vizuální programování	7
3.1	Co je Model Builder?	7
3.2	Základní koncepty	7
3.3	Jednoduchý příklad	8
3.4	Jak vytvořit model	8
4	Praktická úloha	9
4.1	Zadání	9
4.1.1	Proč tato úloha?	9

4.2	Data	9
4.2.1	Okresy	9
4.2.2	Dálnice	9
4.2.3	Krajinný pokryv	10
4.3	Analytický postup	10
4.4	Krok za krokem	12
4.4.1	Příprava	12
4.4.2	Krok 1: Výběr okresu Jindřichův Hradec	12
4.4.3	Krok 2: Ořezání dálnic na okres	13
4.4.4	Krok 3: Vytvoření ochranného pásma (buffer)	13
4.4.5	Krok 4: Vytvoření binárního rastru lesů	14
4.4.6	Krok 5: Zonální statistika	14
4.4.7	Uložení a spuštění modelu	15
4.4.8	1. Uložit	15
4.4.9	2. Ověřit	15
4.4.10	3. Spustit	15
4.4.11	4. Počkat	15
4.4.12	5. Výsledek	15
4.5	Interpretace výsledku	15
4.5.1	Manuální dopočítání	16
5	Export modelu do Pythonu	16
5.1	Proč exportovat?	16
5.2	Jak exportovat	17
5.3	Prohlédnutí Python kódu	17
5.3.1	Struktura kódu	17
5.4	Co vidíme v kódu?	17
5.4.1	1. Import ArcPy	17
5.4.2	2. Proměnné	17
5.4.3	3. Volání nástroje	18
5.5	Interaktivní prozkoumání	18
5.6	Co by se dalo snadno změnit v Pythonu?	19
5.6.1	Změna vzdálenosti	19
5.6.2	Jiný okres	19
5.6.3	Automatický výpočet procent	19
5.7	Výhody Pythonu vs. Model Builder	20
6	Možnosti rozšíření	20
6.1	Co kdybychom chtěli víc?	20
6.2	Možnost A: Spustit model víckrát ručně	21
6.3	Možnost B: Použít iterátor (příště!)	21
6.4	Možnost C: Python (za 2-3 týdny)	22
6.5	A co kombinace parametrů?	23
6.5.1	V Model Builderu	23
6.5.2	V Pythonu	24
6.6	Diskuze	25

7 Shrnutí	25
7.1 Co jsme se dnes naučili	25
7.2 Co nás čeká příště (Lekce 2)	26
7.2.1 První část - Iterátory	26
7.2.2 Druhá část - Python Tool	26
7.3 Klíčové pojmy	27
7.4 Domácí úkol (volitelný)	27
8 Volitelné úkoly	27
8.1 Úkol 1: Změna parametrů modelu	27
8.2 Úkol 2: Jiný okres	28
8.3 Úkol 3: Jiný typ krajinného pokryvu	29
8.4 Úkol 4: Export a zkoumání Python kódu	30
8.4.1 A) Najděte	30
8.4.2 B) Experimentujte	31
8.4.3 C) Přemýšlejte	31
8.5 Úkol 5 (BONUS): Vnořené modely - výzva!	31
8.5.1 Přístup A	32
8.5.2 Přístup B	32
8.5.3 Reflexe (důležitější než řešení!)	33
9 Další zdroje	34
9.1 Dokumentace	34
9.2 Video tutoriály (volitelné)	35
9.3 Corine Land Cover	35
9.4 Tipy na další studium	35
10 Kontakt a dotazy	35
11 Poznámky pro další lekci	36

i Informace o lekci

Časová dotace: 90 minut (1,5 hodiny)

Předpoklady: Základní znalost ArcGIS Pro (GIS 1, GIS 2)

Materiály: ArcGIS Pro projekt ke stažení na [odkaz]

1 Úvod do kurzu

1.1 Co se v tomto kurzu naučíte

Tento kurz vás provede cestou od vizuálního programování v Model Builderu k psaní vlastních Python skriptů pro automatizaci GIS úloh.

1.1.1 Týdny 1-2

Model Builder → Python

- Úvod do automatizace
- První pohled na Python kód
- Export modelů

1.1.2 Týdny 3-5

Základy programování v Pythonu

- Proměnné, cykly, funkce
- Práce se soubory
- Algoritmické myšlení

1.1.3 Týdny 6-9

ArcPy - Python pro GIS

- Automatizace analýz
- Tabulkové operace
- Rastrové analýzy

1.1.4 Týdny 10-11

Pokročilé techniky

- Kurzory
- Geometrie
- Tvorba nástrojů s GUI

1.1.5 Týden 12

Závěrečný projekt

- Vlastní nástroj
- Řešení reálného problému

1.2 Proč tento kurz?

Už umíte pracovat s ArcGIS Pro - znáte nástroje, dokážete provádět analýzy, vytvářet mapy. **Nyní se naučíte GIS automatizovat a programovat.**

Příklady z praxe

Opakující se úlohy

“Každý měsíc musíme aktualizovat mapy dostupnosti zdravotnických zařízení pro 14 krajů.”

→ **Řešení:** Napsat skript, který to udělá automaticky za 5 minut.

Velké množství dat

“Potřebujeme zpracovat 500 rastrových snímků z družice.”

→ **Řešení:** Python skript běžící přes noc, vy ráno kontrolujete výsledky.

Složité analýzy

“Chceme optimalizovat umístění větrných elektráren na základě 10 různých kritérií.”

→ **Řešení:** Vlastní nástroj s GUI, který může používat kdokoli v týmu.

1.3 Struktura výuky

Každý týden:

- 2 × 1,5 hodiny praktických cvičení
- Kombinace výkladu + samostatné práce
- Volitelné úkoly k procvičení

Hodnocení:

- Aktivita na cvičeních
- Průběžné úkoly (malé, týdenní)
- **Závěrečný projekt** (hlavní část hodnocení)
 - Funkční Python nástroj
 - Řeší reálný GIS problém
 - S dokumentací

Nástroje:

- ArcGIS Pro (máte nainstalované?)
- Python 3.x (součást ArcGIS Pro)
- Textový editor (dnes ukážeme)
- Později: Jupyter Notebook, VS Code

2 Proč automatizovat GIS úlohy?

2.1 Motivační příklad

Představte si, že máte tuto úlohu:

Zadání

Spočítejte vzdálenost od každého okresu v ČR k nejbližší dálnici a vytvořte mapu s výsledky.

2.1.1 Řešení ručně

1. Vybrat okres Praha
2. Spustit nástroj Near
3. Zapsat výsledek
4. Vybrat okres Benešov
5. Spustit nástroj Near
6. Zapsat výsledek
7. ... opakovat 75×
8. Vytvořit mapu

Čas: 2-3 hodiny

Riziko chyby: Vysoké (únava, překlep, zapomenutý okres)

2.1.2 Řešení s automatizací

1. Vytvořit model/skript
2. Spustit

Čas: 5 minut běhu + 30 minut na přípravu modelu

Riziko chyby: Minimální

Bonus: Příště to použijete znovu!

2.2 Výhody automatizace

Table 1: Přehled výhod automatizace

Výhoda	Popis	Příklad
Časová úspora	Tisíce opakování za vás	Analýza pro 77 okresů × 10 parametrů = 770 opakování
Přesnost	Žádné chyby z nepozornosti	Vždy stejný postup, stejné výsledky
Dokumentace	Workflow je viditelný	Model = dokumentace postupu
Opakovatelnost	Snadno spustit znovu	Aktualizace dat? Jeden klik.
Parametrizace	Snadná změna nastavení	Změnit vzdálenost z 500m na 1000m
Sdílení	Kolegové mohou použít	Vytvořit nástroj pro celý tým

2.3 Kdy automatizovat?

2.3.1 ANO - automatizujte když:

- Úlohu budete opakovat víc než 2-3×
- Úloha má jasné kroky
- Pracujete s mnoha datovými sadami
- Potřebujete konzistentní výsledky
- Chcete ušetřit čas (i přes přípravu)

2.3.2 NE - neautomatizujte když:

- Úloha je jednorázová a jednoduchá
- Trvá 2 minuty
- Příprava automatizace by trvala déle než ruční práce
- Není jasný postup (ještě experimentujete)

3 Model Builder - vizuální programování

3.1 Co je Model Builder?

Model Builder je nástroj v ArcGIS Pro pro grafické sestavování workflow (pracovních postupů). Je to forma “**programování bez psaní kódu**” - místo textových příkazů skládáte nástroje jako puzzle.

3.2 Základní koncepty

Model se skládá ze tří typů prvků:

```
flowchart LR
    A[Vstupní data] --> B[Nástroj/Proces]
    B --> C[Výstupní data]

    style A fill:#6baed6
    style B fill:#fee391
    style C fill:#6baed6
```

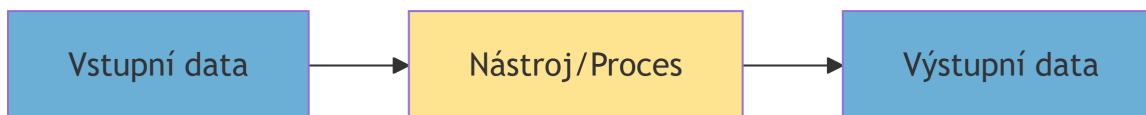


Figure 1: Základní struktura modelu

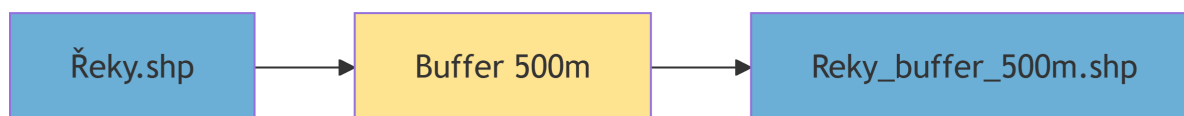
- **Modrý ovál** = Data (vrstvy, tabulky, rastry)
- **Žlutý obdélník** = Nástroj (Buffer, Clip, Intersect, ...)
- **Šipky** = Tok dat (co se používá kam)

3.3 Jednoduchý příklad

Úloha: Vytvořit 500m buffer kolem řek

```
flowchart LR
    A[Řeky.shp] --> B[Buffer 500m]
    B --> C[Reky_buffer_500m.shp]

    style A fill:#6baed6
    style B fill:#fee391
    style C fill:#6baed6
```



Jednoduchý model - Buffer řek

Co se děje:

1. Model načte vrstvu řek
2. Spustí nástroj Buffer s parametrem 500 metrů
3. Uloží výsledek

3.4 Jak vytvořit model

i Krok za krokem - Vytvoření modelu

1. Otevřít ModelBuilder:

- V ArcGIS Pro: Insert → New Toolbox
- Pravý klik na toolbox → New → Model

2. Přidat vstupní data:

- Přetáhnout vrstvu z Contents pane do modelu
- NEBO: Insert → Variable → [typ dat]

3. Přidat nástroj:

- Přetáhnout z ArcToolbox
- NEBO: Insert → Tool → vybrat nástroj

4. Propojit:

- Přetáhnout šipku z dat do nástroje
- Nastavit parametry nástroje (dvojklik)

5. Spustit:

- Nejdřív uložit! (File → Save)
- Pak Run (tlačítko)

4 Praktická úloha

4.1 Zadání

! Analytická úloha

Jaké je zastoupení lesů v ochranném pásmu 500 metrů kolem dálnic v okrese Jindřichův Hradec?

4.1.1 Proč tato úloha?

- **Realistická** - ochranná pásma, hlukové mapování, dostupnost
- **Jednoduchá** - pochopitelné kroky
- **Rozšiřitelná** - později přidáme další parametry
- **Ukáže limity** Model Builderu → motivace pro Python

4.2 Data

Všechna data jsou připravena v projektu ArcGIS Pro, který si stáhnete z [odkaz].

4.2.1 Okresy

okresy - polygony okresů ČR

- Souřadnicový systém: EPSG:3035
- Klíčové pole: NAZ_LAU1 (název okresu)

4.2.2 Dálnice

dalnice - linie dálnic ČR

- Souřadnicový systém: EPSG:3035
- Geometrie: polyline

4.2.3 Krajinový pokryv

clc_2018 - rastr Corine Land Cover

- Souřadnicový systém: EPSG:3035
- Hodnoty:
 - 1 = Urbanizované plochy
 - 2 = Zemědělská půda
 - 3 = Lesy
 - 4 = Mokřady
 - 5 = Vodní plochy

i Note

Poznámka: Data jsou už transformována do jednotného souřadnicového systému ETRS89 LAEA (EPSG: 3035).

4.3 Analytický postup

Cíl: Zjistit, kolik procent plochy okresu Jindřichův Hradec v pásmu 500m od dálnic tvoří lesy.

```
flowchart TD
    A[Okresy] --> B[Vybrat JH]
    B --> C[Okres JH]
    D[Dálnice] --> E[Oříznout okresem]
    C --> E
    E --> F[Dálnice v JH]
    F --> G[Buffer 500m]
    G --> H[Pásmo 500m]
    I[CLC rastr] --> J[Binární rastr lesů]
    H --> K[Zónální statistika]
    J --> K
    K --> L[Výsledek]

    style C fill:#6baed6
    style F fill:#6baed6
    style H fill:#6baed6
    style J fill:#74c476
    style L fill:#fd8d3c
```

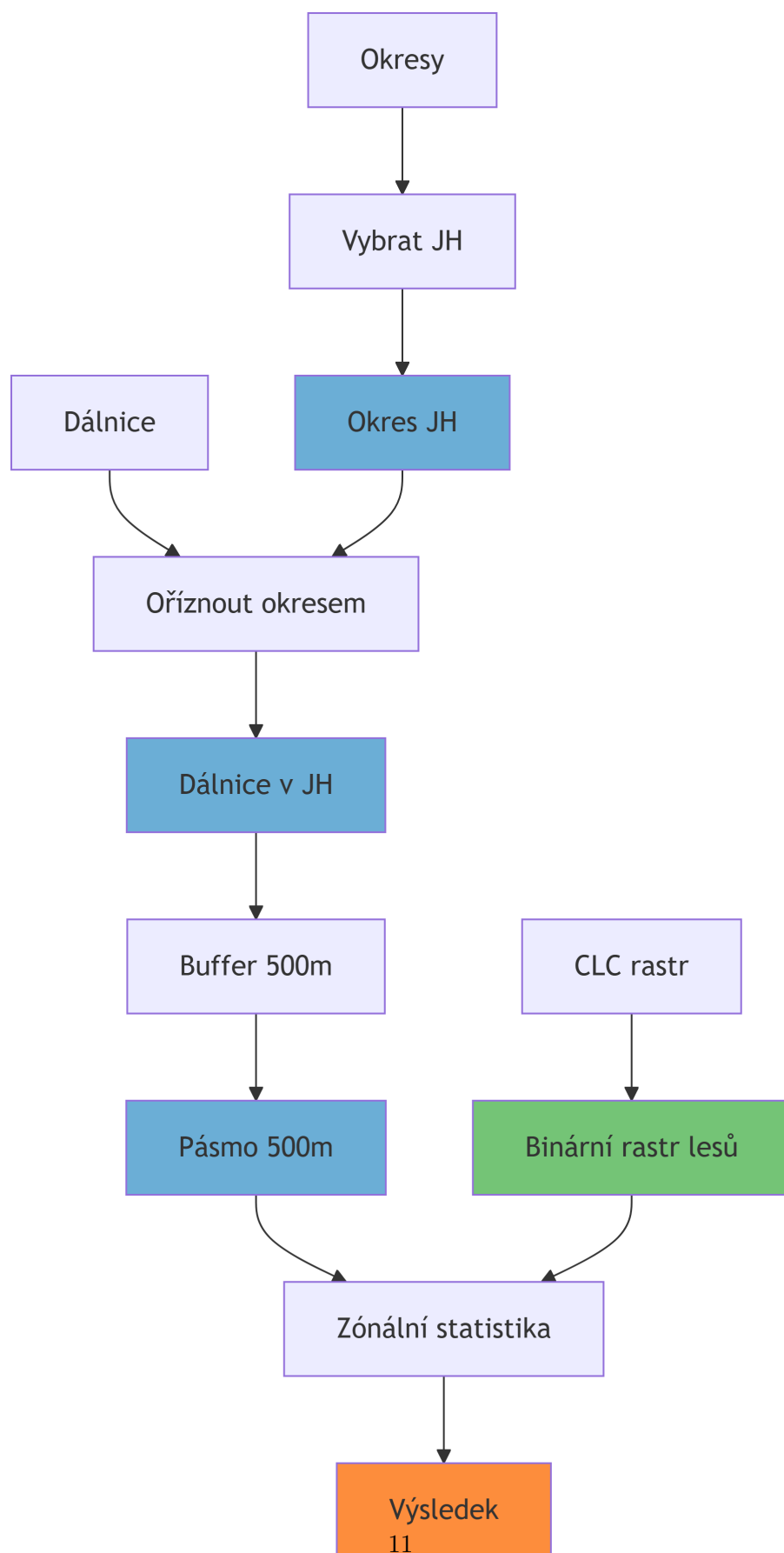


Figure 2: Workflow analytického postupu

4.4 Krok za krokem

4.4.1 Příprava

1. Otevřete ArcGIS Pro projekt `Lekce1_AutomatizaceGIS.aprx`
2. Prohlédněte si data v mapě
3. Vytvořte nový toolbox:
 - Právý klik v Catalog Pane → New → Toolbox
 - Pojmenujte: `Lekce1_Tools.atbx`
4. Vytvořte nový model:
 - Právý klik na toolbox → New → Model
 - Pojmenujte: `Analyza_Lesu_v_Pasmu`

4.4.2 Krok 1: Výběr okresu Jindřichův Hradec

Nástroj: Make Feature Layer

💡 Proč ne Select?

Nástroj *Select* vytváří novou datovou sadu na disku. *Make Feature Layer* vytváří pouze dočasnou vrstvu v paměti, což je rychlejší a efektivnější.

Postup:

1. V modelu: Insert → Tool → vyhledat “Make Feature Layer”
2. Přetáhněte vrstvu `okresy` do modelu
3. Propojte `okresy` s nástrojem Make Feature Layer
4. Dvojklik na nástroj → nastavit parametry:
 - **Input Features:** `okresy`
 - **Output Layer:** `okres_jh_layer`
 - **Expression:** Klikněte SQL

```
NAZ_LAU1 = 'Jindřichův Hradec'
```

5. OK

Kontrola: `okresy` → Make Feature Layer → `okres_jh_layer`

4.4.3 Krok 2: Ořezání dálnic na okres

Nástroj: Clip

Účel: Z celé vrstvy dálnic chceme jen úseky, které jsou v okrese JH.

Postup:

1. Insert → Tool → “Clip”
2. Přetáhněte vrstvu **dálnice** do modelu
3. Propojte:
 - **dálnice** → Clip (jako Input Features)
 - **okres_jh_layer** → Clip (jako Clip Features)
4. Dvojklik na Clip → parametry:
 - **Input Features:** dálnice
 - **Clip Features:** okres_jh_layer
 - **Output:** dálnice_clip
5. OK

4.4.4 Krok 3: Vytvoření ochranného pásma (buffer)

Nástroj: Buffer

! Důležité nastavení

Dissolve Type = ALL (spojí všechny buffery do jednoho)

Postup:

1. Insert → Tool → “Buffer”
2. Propojte **dálnice_clip** → Buffer
3. Dvojklik na Buffer → parametry:
 - **Input Features:** dálnice_clip
 - **Output:** buffer_500m
 - **Distance:** 500 Meters
 - **Dissolve Type: ALL** ← důležité!
 - Side Type: FULL
 - End Type: ROUND
4. OK

i Proč Dissolve ALL?

Bez dissolve bychom měli desítky překrývajících se bufferů (jeden pro každý úsek dálnice). S ALL se všechny spojí do jednoho (multi)polygonu.

4.4.5 Krok 4: Vytvoření binárního rastru lesů

Nástroj: Equal To (Spatial Analyst)

Účel: Z CLC rastru (hodnoty 1,2,3,4,5) vytvořit rastr s hodnotami 0/1, kde 1 = les.

Postup:

1. Insert → Tool → “Equal To” (v kategorii Spatial Analyst → Math → Logical)
2. Přetáhněte rastr `clc_2018` do modelu
3. Propojte `clc_2018` → Equal To
4. Dvojklik na Equal To:
 - **Input raster:** `clc_2018`
 - **Input value:** 3 (kód pro lesy)
 - **Output:** `lesy_binarni`
5. OK

i Co se stane?

Rastr bude mít hodnotu 1 tam, kde je les (CLC=3), a hodnotu 0 všude jinde.

4.4.6 Krok 5: Zonální statistika

Nástroj: Zonal Statistics as Table

Účel: Spočítat průměr z binárního rastru v rámci bufferu.

💡 Matematický trik

Průměr z nul a jedniček = podíl jedniček = relativní plocha lesů!
Pokud je průměr 0.35, znamená to, že 35% pixelů má hodnotu 1 (les).

Postup:

1. Insert → Tool → “Zonal Statistics as Table”
2. Propojte:
 - `buffer_500m` → Zonal Statistics (jako Input Zone Data)
 - `lesy_binarni` → Zonal Statistics (jako Input Value Raster)
3. Dvojklik na Zonal Statistics:
 - **Input Zone Data:** `buffer_500m`
 - **Zone Field:** OBJECTID
 - **Input Value Raster:** `lesy_binarni`
 - **Output Table:** `vysledek_lesy.dbf`
 - **Statistics Type:** MEAN (průměr)

- **Ignore NoData:** zaškrtnuto

4. OK

4.4.7 Uložení a spuštění modelu

4.4.8 1. Uložit

File → Save (Ctrl+S)

4.4.9 2. Ověřit

Model → Validate Entire Model

Pokud je vše OK, všechny nástroje budou barevné (ne šedé)

4.4.10 3. Spustit

Klikněte na Run

4.4.11 4. Počkat

Model běží, sledujte progress

4.4.12 5. Výsledek

Otevřete `vysledek_lesy.dbf`

4.5 Interpretace výsledku

V tabulce `vysledek_lesy.dbf` najdete sloupec **MEAN**.

Význam:

- $\text{MEAN} = 0.354 \rightarrow 35.4\%$ plochy v pásmu tvoří lesy
- $\text{MEAN} = 0.205 \rightarrow 20.5\%$ plochy v pásmu tvoří lesy

i Proč to funguje?

Průměr z binárního rastru (0/1) v dané zóně = podíl pixelů s hodnotou 1 = relativní plocha lesů.
Pro procenta: $\text{MEAN} \times 100$

4.5.1 Manuální dopočítání

Výslednou hodnotu MEAN musíme **ručně vynásobit 100**, abychom dostali procenta.

Postup:

1. Otevřít tabulku `vysledek_lesy.dbf`
2. Najít hodnotu MEAN (např. 0.354)
3. Kalkulačka: $0.354 \times 100 = 35.4\%$
4. Zapsat si výsledek

Problém?

Ano! Musíme počítat ručně. Co kdybychom chtěli analyzovat všech 77 okresů? 77× otevřít tabulku a počítat...

Diskuze:

“Dalo by se to automatizovat v Model Builderu?”

→ Ano, pomocí Calculate Field, ale komplikované.

→ V Pythonu? Jednoduchý výpočet!

```
# Python řešení
mean_value = 0.354
procento = mean_value * 100
print(f"Podíl lesů: {procento:.1f}%")
```

5 Export modelu do Pythonu

5.1 Proč exportovat?

Model Builder je skvělý pro vizualizaci workflow, ale má limity:

- Těžko se kopírují části modelu
- Složitě sdílení (musíte sdílet toolbox)
- Omezené možnosti logiky (podmínky, cykly)

Python nám dává:

- Textový soubor (snadno sdílitelný, verzovatelný)
- Možnost úprav v textovém editoru
- Přidání vlastní logiky
- Spuštění mimo ArcGIS Pro

5.2 Jak exportovat

1. V Model Builderu: **Model** → **Export** → **To Python Script**
2. Uložit jako: `model_export.py`
3. Vybrat lokaci a uložit

5.3 Prohlédnutí Python kódu

Otevřete exportovaný soubor v textovém editoru:

- **Notepad++** (doporučeno - zvýrazňuje syntax)
- **VS Code** (pokud máte)
- **Poznámkový blok** (funguje, ale bez barev)

5.3.1 Struktura kódu

5.4 Co vidíme v kódu?

5.4.1 1. Import ArcPy

```
import arcpy
```

Note

Význam: “Chci použít nástroje ArcGIS v Pythonu”

Analogie: Jako když v ArcGIS Pro otevřete ArcToolbox - získáte přístup k nástrojům.

5.4.2 2. Proměnné

```
okresy = "okresy"  
vzdalenost = "500 Meters"
```

Tip

Výhoda proměnných: Můžeme snadno změnit na jednom místě:

```
vzdalenost = "1000 Meters" # Změna parametru!
```

5.4.3 3. Volání nástroje

```
arcpy.Buffer_analysis(  
    in_features=dalnice_clip,  
    out_feature_class=buffer_500m,  
    buffer_distance_or_field="500 Meters",  
    dissolve_option="ALL"  
)
```

Srovnání s Model Builderem:

Table 2: Srovnání Model Builder vs. Python

Model Builder	Python
Žlutý obdélník “Buffer”	<code>arcpy.Buffer_analysis()</code>
Dialog s parametry	Parametry v závorkách
Propojení šipkou	Proměnné jako parametry
Kliknutí na Run	<code>python script.py</code>

! Klíčové poznání

Je to STEJNÉ, jen jinak zapsané!

5.5 Interaktivní prozkoumání

i Úkol 1: Najděte v kódu

Kde je napsáno “Jindřichův Hradec”?

```
where_clause="NAZ_LAU1 = 'Jindřichův Hradec'"
```

i Úkol 2: Najděte v kódu

Kde je vzdálenost bufferu?

```
buffer_distance_or_field="500 Meters"
```

i Úkol 3: Najděte v kódu

Kde se vytváří binární rastr lesů?

```

arcpy.gp.EqualTo_sa(
    in_raster_or_constant1=clc_2018,
    in_raster_or_constant2="3", # ← tady je kód pro lesy
    out_raster=lesy_binarni
)

```

5.6 Co by se dalo snadno změnit v Pythonu?

5.6.1 Změna vzdálenosti

```

# Místo:
buffer_distance_or_field="500 Meters"

# Můžeme:
vzdalenost = 1000 # metry
buffer_distance_or_field=f"{vzdalenost} Meters"

```

5.6.2 Jiný okres

```

# Místo:
where_clause="NAZ_LAU1 = 'Jindřichův Hradec'"

# Můžeme:
okres = "Praha"
where_clause=f"NAZ_LAU1 = '{okres}'"

```

5.6.3 Automatický výpočet procent

```

# Po Zonal Statistics přidat:
import arcpy

# Přečíst výsledek
with arcpy.da.SearchCursor(vysledek_lesy, ["MEAN"]) as cursor:
    for row in cursor:
        mean_value = row[0]
        procento = mean_value * 100
        print(f"Podíl lesů: {procento:.1f}%")

```

💡 Vidíte?

V Pythonu můžeme snadno:

- Měnit parametry
- Přidávat výpočty
- Automatizovat opakování

5.7 Výhody Pythonu vs. Model Builder

Table 3: Srovnání Model Builder vs. Python

Aspekt	Model Builder	Python
Vizualizace	Výborná	Žádná (jen text)
Rychlé vytvoření	Drag & drop	Musíte psát
Sdílení	Toolbox soubor	Textový .py soubor
Verzování (Git)	Binární formát	Textový formát
Podmínky (IF)	Omezené	Plná podpora
Cykly (FOR)	Jen iterátory	Plná flexibilita
Výpočty	Calculate Field	Jakékoli operace
Debugging	Obtížné	Snadné
Rychlost běhu	Pomalejší	Rychlejší

! Závěr

Model Builder = skvělý start

Python = mocný nástroj pro pokročilé úlohy

6 Možnosti rozšíření

6.1 Co kdybychom chtěli víc?

Nyní máme řešení pro:

- 1 okres (Jindřichův Hradec)
- 1 vzdálenost (500m)
- 1 typ krajinného pokryvu (lesy)

Co když chceme:

- Všechny okresy? (77×)
- Různé vzdálenosti? (100m, 300m, 500m, 1000m)
- Všechny typy pokryvu? (lesy, pole, města, voda, ...)

6.2 Možnost A: Spustit model víckrát ručně

Pro různé vzdálenosti:

1. Otevřít model
2. Změnit parametr bufferu na 100m
3. Spustit
4. Změnit na 300m
5. Spustit
6. ...

Čas: 10-15 minut

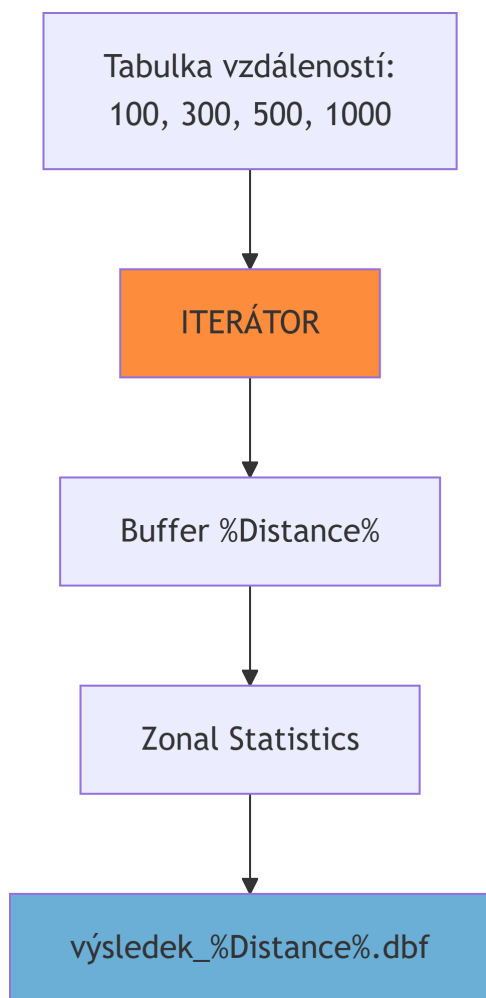
Problém: Nudné, opakující se, náchylné k chybám

6.3 Možnost B: Použít iterátor (příště!)

V Lekci 2 se naučíme:

```
flowchart TD
    A["Tabulka vzdáleností:<br/>100, 300, 500, 1000"] --> B["ITERÁTOR"]
    B --> C["Buffer %Distance%"]
    C --> D["Zonal Statistics"]
    D --> E["výsledek_%Distance%.dbf"]

    style B fill:#fd8d3c
    style E fill:#6baed6
```



Model s iterátorem

Výsledek: 4 tabulky (jedna pro každou vzdálenost)

⚠ Warning

Stále problém: Jak spojit 4 tabulky do jedné?

6.4 Možnost C: Python (za 2-3 týdny)

```
# Elegantní řešení v Pythonu
vzdalenosti = [100, 300, 500, 1000]
vysledky = []

for vzd in vzdalenosti:
```

```

# Buffer
buffer = vytvor_buffer(vzd)

# Analýza
procento = vypocitej_lesy(buffer)

# Uložit
vysledky.append({
    'Vzdalenost': vzd,
    'Procento_lesu': procento
})

# Jedna tabulka se všemi výsledky!
uloz_tabulku(vysledky, 'vysledky_vsechny.csv')

```

Výhoda

- Jeden skript
- Jedna výsledná tabulka
- Snadno rozšířit o další parametry
- Rychlé

6.5 A co kombinace parametrů?

Scénář: Chceme analyzovat:

- 2 typy komunikací (dálnice, silnice I. třídy)
- 3 vzdálenosti (100m, 500m, 1000m)
- = **6 kombinací**

6.5.1 V Model Builderu

Potřebovali bychom **vnořené iterátory** = 2 modely propojené dohromady → velmi složité!

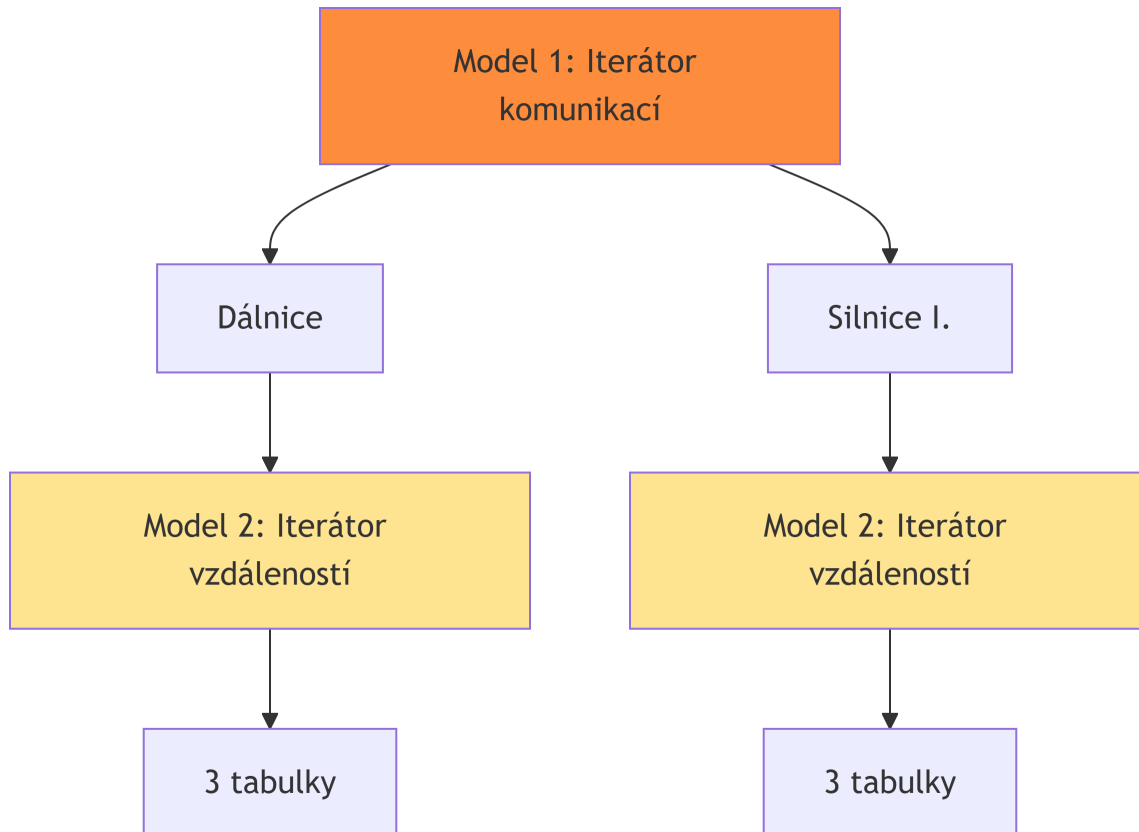
```

flowchart TD
    A[Model 1: Iterátor komunikací] --> B[Dálnice]
    A --> C[Silnice I.]
    B --> D[Model 2: Iterátor vzdáleností]
    C --> E[Model 2: Iterátor vzdáleností]
    D --> F[3 tabulky]
    E --> G[3 tabulky]

    style A fill:#fd8d3c

```

```
style D fill:#fee391
style E fill:#fee391
```



Vnořené modely (složitě!)

6.5.2 V Pythonu

```
komunikace = ['dálnice', 'silnice1']
vzdalenosti = [100, 500, 1000]

for kom in komunikace:
    for vzd in vzdalenosti:
        analyzuj(kom, vzd)
```

! 3 řádky!

Vidíte rozdíl?

6.6 Diskuze

Otázky k zamyšlení

1. Kolik času byste strávili vytvořením vnořených modelů v Model Builderu?
2. Jak byste spojili 6 výsledných tabulek do jedné?
3. Co kdyby zadání změnilo: “Teď to chceme pro 5 typů komunikací a 10 vzdáleností” (50 kombinací)?

Odpověď: Proto se učíme Python!

7 Shrnutí

7.1 Co jsme se dnes naučili

Přehled lekce

Proč automatizovat GIS úlohy

- Úspora času
- Eliminace chyb
- Opakovatelnost
- Dokumentace

Model Builder - základy

- Vizualní programování
- Skládání nástrojů do workflow
- Vytvoření funkčního modelu

Praktická úloha

- Analýza krajinného pokryvu v ochranných pásmech
- 5 kroků: výběr → clip → buffer → binární rastr → zonální statistika
- Interpretace výsledků

Export do Pythonu

- Model = Python kód
- První pohled na Python syntax
- Srovnání Model Builder vs. Python

Limity Model Builderu

- Složité operace (vnořené cykly)
- Spojování výsledků
- Motivace pro Python

7.2 Co nás čeká příště (Lekce 2)

7.2.1 První část - Iterátory

- Jak automatizovat opakování v Model Builderu
- Iterate Field Values - procházení různých vzdáleností
- Problém: mnoho výstupních tabulek

7.2.2 Druhá část - Python Tool

- Z modelu vytvoříme nástroj s GUI
- Parametry: uživatel si vybere okres, vzdálenost
- Nástroj můžete sdílet s kolegy

7.3 Klíčové pojmy

Table 4: Klíčové pojmy z lekce

Pojem	Význam
Automatizace	Opakované spouštění úloh bez lidského zásahu
Model Builder	Nástroj pro vizuální tvorbu workflow
Workflow	Posloupnost kroků vedoucí k výsledku
Iterator	Mechanismus pro opakování (příště)
ArcPy	Python knihovna pro ArcGIS
Zonální statistika	Výpočet statistik v definovaných zónách
Binární rastr	Rastr s hodnotami 0/1 (ano/ne)

7.4 Domácí úkol (volitelný)

Procvičte si látku pomocí [volitelných úkolů](#) níže.

Doporučení

- Začněte Úkolem 1 (lehký) - určitě zvládnete!
- Pokud vás to baví, zkuste Úkol 2
- Úkol 3 je pro ty, kdo chtějí výzvu

8 Volitelné úkoly

8.1 Úkol 1: Změna parametrů modelu

Obtížnost: Lehká

Cíl: Naučit se měnit parametry v modelu a vidět, jak to ovlivní výsledky.

Zadání:

Upravte svůj model tak, aby analyzoval **pásmo 300 metrů** (místo 500m) kolem dálnic v okrese Jindřichův Hradec.

Očekávaný výsledek:

- Upravený model s bufferem 300m
- Nová výsledná tabulka
- Porovnání: je procento lesů v pásmu 300m vyšší nebo nižší než v 500m? Proč?

Postup

1. Otevřete svůj model `Analyza_Lesu_v_Pasmu`
2. Dvojklik na nástroj Buffer
3. Změňte Distance: `500 Meters` → `300 Meters`
4. Změňte název výstupu: `buffer_500m` → `buffer_300m`
5. Změňte název výsledné tabulky: `vysledek_lesy.dbf` → `vysledek_lesy_300m.dbf`
6. Uložte a spusťte model
7. Porovnejte výsledky (MEAN hodnoty)

Otázky k zamyšlení:

- Je procento lesů v užším pásmu (300m) jiné než v širším (500m)?
- Jak byste to vysvětlili? (Nápověda: rozmístění lesů vs. dálnic)

Bonus

Vytvořte tabulku v Excelu s porovnáním:

Vzdálenost	Procento lesů
300m	X.X%
500m	Y.Y%

8.2 Úkol 2: Jiný okres

Obtížnost: Střední

Cíl: Pochopit, jak změnit atributový dotaz v modelu.

Zadání:

Upravte model tak, aby analyzoval okres **Praha-východ** (místo Jindřichův Hradec).

Očekávaný výsledek:

- Model fungující pro okres Praha-východ
- Výsledná tabulka s procentem lesů
- Porovnání: má Praha-východ více nebo méně lesů v pásmech kolem dálnic než JH?

Postup

1. Nejprve zjistěte přesný název okresu:
 - Otevřete atributovou tabulku vrstvy `okresy`
 - Najděte pole `NAZ_LAU1`
 - Najděte řádek s Prahou-východ (může být “Praha-východ” nebo “Praha - východ”)

2. V modelu: dvojklik na Make Feature Layer

3. Změňte Expression:

```
NAZ_LAU1 = 'Praha-východ'
```

(Pozor na přesný zápis!)

4. Změňte názvy výstupů, aby bylo jasné, že jde o jiný okres

5. Spustíte model

! Nápověda

- Pokud model hlásí “0 features selected”, zkontrolujte přesný název okresu v datech
- Může být potřeba použít LIKE místo =:

```
NAZ_LAU1 LIKE '%Praha%východ%'
```

💡 Bonus

Vytvořte srovnávací tabulku pro 3-5 různých okresů.

8.3 Úkol 3: Jiný typ krajinného pokryvu

i Obtížnost: Střední

Cíl: Naučit se analyzovat různé kategorie dat změnou jednoho parametru.

Zadání:

Analyzujte zastoupení **zemědělské půdy** (CLC kód 2) místo lesů v pásmu 500m kolem dálnic v okrese JH.

Očekávaný výsledek:

- Model analyzující zemědělskou půdu
- Porovnání: je v pásmu více lesů nebo zemědělské půdy?

💡 Postup

1. V modelu: dvojklik na nástroj Equal To
2. Změňte Input value: 3 → 2
3. Změňte názvy výstupů:

- lesy_binarni → zempuda_binarni
- vysledek_lesy.dbf → vysledek_zempuda.dbf

4. Spustíte model

Rozšíření

Vytvořte tabulku se všemi typy krajinného pokryvu:

CLC kód	Typ	Procento
1	Urbanizované plochy	?
2	Zemědělská půda	?
3	Lesy	?
4	Mokřady	?
5	Vodní plochy	?

(Musíte spustit model 5× s různými kódy)

Otázka

Je tento postup efektivní? Co kdybyste chtěli 10 kategorií? 50?
→ Proto se naučíme iterátory a Python!

8.4 Úkol 4: Export a zkoumání Python kódu

Obtížnost: Lehká

Cíl: Seznámit se s Python syntaxí na vašem vlastním modelu.

Zadání:

Exportujte váš model do Pythonu a prozkoumejte kód.

Postup

1. V Model Builderu: Model → Export → To Python Script
2. Uložte jako `muj_model.py`
3. Otevřete v textovém editoru (Notepad++, VS Code, nebo Poznámkový blok)

Úkoly v kódu:

8.4.1 A) Najděte

Najděte a zvýrazněte:

- Řádek s importem arcpy
- Řádek, kde se vytváří buffer 500m
- Řádek s SQL dotazem pro okres
- Řádek s hodnotou pro lesy (3)

8.4.2 B) Experimentujte

Experimentujte (bez spouštění!):

- Zkuste změnit "500 Meters" na "1000 Meters" - na kterém řádku?
- Najděte místo, kde byste změnili okres na jiný
- Kolik řádků by bylo potřeba změnit, abyste změnili vzdálenost? A v modelu?

8.4.3 C) Přemýšlejte


Přemýšlejte:

- Je kód čitelný? Rozumíte alespoň trochu, co dělá?
- Které části jsou jasné, které ne?
- Vidíte výhody textové podoby vs. grafické?

 Warning

Poznámka: Kód zatím nespouštějte - to se naučíme příště. Teď jen pozorujte strukturu.

8.5 Úkol 5 (BONUS): Vnořené modely - výzva!

 Obtížnost: Velmi těžká

VAROVÁNÍ: Tento úkol je záměrně velmi obtížný! Jeho cílem je ukázat vám, že některé věci jsou v Model Builderu nepraktické. Nebojte se, pokud se vám to nepodaří - právě proto se učíme Python!

Zadání:

Vytvořte řešení, které analyzuje **2 okresy** \times **2 vzdálenosti** = **4 kombinace**:

- Okresy: Jindřichův Hradec, Praha-východ
- Vzdálenosti: 300m, 500m

Očekávaný výsledek:

4 tabulky:

- vysledek_JH_300.dbf
- vysledek_JH_500.dbf
- vysledek_Praha_300.dbf
- vysledek_Praha_500.dbf

8.5.1 Přístup A

Přístup A (jednodušší, ale nudný):

- Spusťte model 4× ručně, pokaždé změňte parametry
- Zapište časy: Jak dlouho to celkem trvalo?

8.5.2 Přístup B

Přístup B (složitý - vnořené modely):

- Vytvořte Model 2 (vnitřní) s iterátorem přes vzdálenosti
- Vytvořte Model 1 (vnější) s iterátorem přes okresy, který volá Model 2
- Poznámka: Je to VELMI složité nastavit!

 Podrobný postup Přístupu B

Krok 1: Vytvoř Model 2 (vnitřní - “Detail__Analysis”)

Tento model zpracuje JEDNU komunikaci pro různé vzdálenosti

Parametry modelu:

- Input: Komunikace (vrstva linií) [nastavit jako Model Parameter]
- Input: Název komunikace (text) [Model Parameter]

Model:

1. Iterate Field Values přes tabulku vzdáleností (100, 500, 1000)
2. Buffer s distance = %Distance%
3. Intersect s okresem
4. Equal To (lesy)
5. Zonal Statistics → výsledek_%Název%_%Distance%.dbf

Krok 2: Vytvoř Model 1 (vnější - “Master__Analysis”)

Tento model prochází komunikace a volá Model 2

Model:

1. Iterate Feature Classes přes komunikace (dalnice, silnice1)
2. Pro každou komunikaci zavolej Model 2:
 - Nástroj: Detail__Analysis (váš Model 2)
 - Parameter Komunikace: %Name%
 - Parameter Název: %Name%

Krok 3: Spusť Model 1

Náповěda

- Model se volá jako nástroj přes: Insert → Model → [název modelu]
- Aby model šel zavolat z jiného modelu, musí mít definované Model Parameters (pravý klik na proměnnou → Model Parameter)
- Můžeš potřebovat nastavit “Preconditions” (šipka s tečkou)
- Pokud to nefunguje, zkus nejdřív každý model otestovat samostatně

Alternativní přístup (jednodušší, ale nudnější)

Pokud vnořené modely nefungují, můžeš Model 2 spustit 2× ručně:

1. První spuštění: pro dálnice → vytvoří 3 tabulky
2. Druhé spuštění: pro silnice I. → vytvoří další 3 tabulky

8.5.3 Reflexe (důležitější než řešení!)

Po dokončení (nebo pokusu) napište:

1. Kolik času vám to zabralo?
2. Kolikrát jste museli model upravovat?
3. Kolik chyb jste udělali?
4. Jak byste se cítili, kdyby zadání bylo 10 okresů × 10 vzdáleností?

! Ukázka Python řešení

V Pythonu by to vypadalo takto (nemusíte rozumět detailům):

```
okresy = ['Jindřichův Hradec', 'Praha-východ']
vzdalenosti = [300, 500]

for okres in okresy:
    for vzd in vzdalenosti:
        # Výběr okresu
        arcpy.MakeFeatureLayer_management(
            "okresy", "temp_layer",
            f"NAZ_LAU1 = '{okres}'"
        )

        # Buffer
        arcpy.Buffer_analysis(
            "dalnice_clip", "buffer_temp",
            f"{vzd} Meters"
        )

        # Analýza...
        # Uložit: f"vysledek_{okres}_{vzd}.dbf"

print("Hotovo! 4 kombinace zpracovány.")
```

Ponaučení: Vidíte, proč se učíme Python?

9 Další zdroje

9.1 Dokumentace

💡 ArcGIS Pro

- [Model Builder dokumentace](#)
- [Geoprocessing nástroje](#)

💡 ArcPy (Python)

- [ArcPy dokumentace](#)
- [ArcPy Get Started](#)

9.2 Video tutoriály (volitelné)

- [Esri: Introduction to ModelBuilder](#)
- [Esri: Automating Workflows with Python](#)

9.3 Corine Land Cover

- [Corine Land Cover - dokumentace](#)
- [CLC Nomenclature](#)

9.4 Tipy na další studium

1. **Procvičujte:** Čím víc modelů vytvoříte, tím lépe pochopíte workflow
2. **Experimentujte:** Zkuste různé nástroje a parametry
3. **Dokumentujte:** Pište si poznámky k modelům (Description v properties)
4. **Sdílejte:** Diskutujte s kolegy, uče se od ostatních
5. **Připravte se na Python:** Příští lekce už začneme s kódem!

10 Kontakt a dotazy

Kontaktní informace

Vyučující: [Vaše jméno]

Email: [email]

Konzultační hodiny: [čas a místo]

Otázky k lekci:

- Pište na email s předmětem “GIS-L1: [vaše otázka]”
- Nebo přijďte na konzultace

Sdílení úkolů (volitelné):

- Pokud chcete zpětnou vazbu, odevzdejte přes [systém/email]
- Deadline: [datum] (ale není povinné!)

11 Poznámky pro další lekci

Co si přinést příště

- Funkční ArcGIS Pro
- Uložený toolbox s modelem z dnešní lekce
- Nápady: Jaké úlohy byste chtěli automatizovat ve vaší praxi?

Na co se těšit

- Iterátory v Model Builderu
- Automatické procházení různých parametrů
- Tvorba Python Tool s GUI
- Ještě jeden krok blíž k Pythonu!

Gratuluji!

Úspěšně jste dokončili první lekci. Vytvořili jste funkční model, exportovali ho do Pythonu, a pochopili základy automatizace GIS úloh.

Next step: Lekce 2 - Iterátory a Python Tools

Listing 1 model_export.py

```
# -*- coding: utf-8 -*-
# -----
# model_export.py
# Created on: 2025-01-15
# Description: Analýza lesů v pásmu kolem dálnic
# -----

# Import knihovny ArcPy
import arcpy

# Lokální proměnné (cesty k datům)
okresy = "okresy"
dalnice = "dalnice"
clc_2018 = "clc_2018"
okres_jh_layer = "okres_jh_layer"
dalnice_clip = "C:\\Data\\dalnice_clip.shp"
buffer_500m = "C:\\Data\\buffer_500m.shp"
lesy_binarni = "C:\\Data\\lesy_binarni.tif"
vysledek_lesy = "C:\\Data\\vysledek_lesy.dbf"

# PROCES 1: Make Feature Layer - výběr okresu
arcpy.MakeFeatureLayer_management(
    in_features=okresy,
    out_layer=okres_jh_layer,
    where_clause="NAZ_LAU1 = 'Jindřichův Hradec'"
)

# PROCES 2: Clip - ořezání dálnic
arcpy.Clip_analysis(
    in_features=dalnice,
    clip_features=okres_jh_layer,
    out_feature_class=dalnice_clip
)

# PROCES 3: Buffer - ochranné pásmo
arcpy.Buffer_analysis(
    in_features=dalnice_clip,
    out_feature_class=buffer_500m,
    buffer_distance_or_field="500 Meters",
    dissolve_option="ALL"
)

# PROCES 4: Equal To - binární rastr lesů
arcpy.gp.EqualTo_sa(
    in_raster_or_constant1=clc_2018,
    in_raster_or_constant2="3",
    out_raster=lesy_binarni
)

# PROCES 5: Zonal Statistics as Table
```