

**Univerzita Palackého v Olomouci**

**Přírodovědecká fakulta**

**Katedra geoinformatiky**

**AUTOMATIZACE TVORBY ČÍSELNÝCH  
KARTODIAGRAMŮ V PROSTŘEDÍ ARCGIS**

**Diplomová práce**

**Bc. Pavel SEDLÁČEK**

**Vedoucí práce Mgr. Radek Barvíř, Ph.D.**

**Olomouc 2026**

**Geoinformatika a kartografie**

## **ANOTACE**

Předkládaná diplomová práce se zabývá problematikou automatizace tvorby metody číselných kartodiagramů v prostředí softwaru ArcGIS Pro. Hlavním cílem práce je navrhnout, vyvinout a otestovat nástroj, který zefektivní a zautomatizuje proces vizualizace gradovaných číselných popisků, jenž v tomto nativním prostředí dosud vyžadoval zdlouhavé manuální nastavování.

Teoretická část práce shrnuje klasifikaci metody kartodiagramů, kartografická pravidla pro tvorbu stupnic a číselných kartodiagramů. Také rozebírá technologické limity současných GIS programů v té problematice.

Praktická část se věnuje samotnému vývoji nástroje s názvem *Graduated Numbers Generator* ve formátu geoprocessingového toolboxu (.atbx). Skript je napsán v jazyce Python s využitím knihovny ArcPy a pokročilé manipulace s objekty Cartographic Information Model (CIM). Pro plynulou gradaci vizuálních proměnných (velikost, barva a tloušťka řezu písma) je implementován matematický algoritmus lineární interpolace. Pro interpretaci a formátování dynamických popisků (labelů) na mapovém poli jsou generovány výrazy v jazyce Arcade. Zásadním aspektem navrženého řešení je, že zdrojová data nijak nemodifikuje a pro čtení systémových fontů využívá vlastní modul napojený přímo na registry Windows.

Výsledkem práce je plně funkční a přenositelný nástroj, který umožňuje uživatelům snadno aplikovat různé statistické metody pro tvorbu spojených stupnic. Nástroj prošel uživatelským testováním a jeho praktická využitelnost je demonstrována na sérii ukázkových tematických map. Vytvořené řešení významně redukuje časovou náročnost práce kartografa a úspěšně zaplňuje technologickou mezeru v možnostech platformy ArcGIS Pro.

## **KLÍČOVÁ SLOVA**

Číselný kartodiagram; tematická kartografie; automatizace; ArcGIS Pro; Python

Počet stran práce: 81

Počet příloh: 8

## **ANNOTATION**

This master thesis deals with the automation of the graduated number map method within the ArcGIS Pro software environment. The main objective of the thesis is to design, develop, and test a tool that streamlines and automates the visualisation process of graduated numeric labels, which previously required tedious manual configuration in this native environment.

The theoretical part summarises the classification of proportional symbol maps and cartographic rules for creating scales and graduated number maps. It also analyses the technological limitations of current GIS software regarding these issues.

The practical part focuses on the development of a tool named *Graduated Numbers Generator* in a geoprocessing toolbox format (.atbx). The script is written in Python using the ArcPy library and advanced manipulation of Cartographic Information Model (CIM) objects. For the smooth gradation of visual variables (size, colour, and font weight), a linear interpolation algorithm is implemented. Arcade language expressions are generated for the interpretation and formatting of dynamic labels in the map layout. A crucial aspect of the designed solution is that it does not modify the source data and uses a custom module connected directly to the Windows registry to read system fonts.

The result of the thesis is a fully functional and portable tool that allows users to easily apply various statistical methods for creating continuous scales. The tool has undergone user testing, and its practical applicability is demonstrated on a series of sample thematic maps. The developed solution significantly reduces the time demands placed on the cartographer and successfully fills a technological gap in the capabilities of the ArcGIS Pro platform.

## **KEYWORDS**

Graduated Number Maps; Thematic Cartography; Automation; ArcGIS Pro; Python

Number of pages: 81

Number of appendixes: 8

**Prohlašuji, že**

- Diplomovou práci včetně příloh, jsem vypracoval samostatně a uvedl jsem všechny použité podklady a literaturu.

- jsem si vědom, že na moji diplomovou práci se plně vztahuje zákon č.121/2000 Sb. - autorský zákon, zejména § 35 – využití díla v rámci občanských a náboženských obřadů, v rámci školních představení a využití díla školního a § 60 – školní dílo,

- beru na vědomí, že Univerzita Palackého v Olomouci (dále UP Olomouc) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou práci užívat (§ 35 odst. 3),

- souhlasím, že údaje o mé diplomové práci budou zveřejněny ve Studijním informačním systému UP,

- v případě zájmu UP Olomouc uzavřu licenční smlouvu s oprávněním užít výsledky a výstupy mé diplomové práce v rozsahu § 12 odst. 4 autorského zákona,

- použít výsledky a výstupy mé diplomové práce nebo poskytnout licenci k jejímu využití mohu jen se souhlasem UP Olomouc, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly UP Olomouc na vytvoření díla vynaloženy (až do jejich skutečné výše).

Rád bych vyjádřil upřímné poděkování vedoucímu práce, Mgr. Radku Barvířovi, Ph.D., za jeho aktivní a inspirativní vedení. Velmi si cením jeho mimořádné ochoty, časového rozvržení průběhu práce a bleskové zpětné vazby na mé četné dotazy v průběhu tvorby. Děkuji rovněž doc. Ing. Zdeně Dobešové, Ph.D., za poskytnuté konzultace a její cenné odborné rady a podněty.

# UNIVERZITA PALACKÉHO V OLOMOUCI

Přírodovědecká fakulta

Akademický rok: 2024/2025

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Pavel SEDLÁČEK**  
Osobní číslo: **R240579**  
Studijní program: **N0532A330009 Geoinformatika a kartografie**  
Téma práce: **Automatizace tvorby číselných kartodiagramů v prostředí ArcGIS**  
Zadávající katedra: **Katedra geoinformatiky**

### Zásady pro vypracování

Cílem práce je automatizovat proces tvorby číselných kartodiagramů (graduujících kvantitativních numerických hodnot) v prostředí ArcGIS za účelem efektivizace jejich tvorby. Student nejprve provede rešerši klasifikace kartodiagramů a zaměří se v ní na důkladné srovnání možností tvorby číselných i jiných kartodiagramů a popisu v různých GIS softwarech. Následně algoritmuje proces v prostředí ArcGIS tak, aby došlo k výrazné redukci množství manuálních kroků, které dosud kartograf musí při jejich tvorbě udělat. Při tvorbě nástroje/skriptu vezme v potaz a dle možností implementuje nejen možnost gradace velikosti, ale také barvy a řezu písma. Pomocí nástroje následně vytvoří nejméně tři ukázkové mapy, využívající tuto metodu pro vizualizaci vybraných statistických dat.

Text práce student zpracuje v souladu se závaznou šablonou pro kvalifikační práce KGI. O diplomové práci student vytvoří webovou stránku a poster. Celou práci (text, přílohy, výstupy, zdrojová a vytvořená data, poster a web) odevzdá student v digitální podobě na datové úložiště katedry. Do evidence STAG student odevzdá úplný text práce s přílohami, které určí vedoucí práce. Fyzicky student odevzdá pouze přílohy určené vedoucím práce.

Rozsah pracovní zprávy: **max. 50 stran**  
Rozsah grafických prací: **dle potřeby**  
Forma zpracování diplomové práce: **elektronická**

### Seznam doporučené literatury:

- [1] Korycka-Skorupa, Jolanta & Golebiowska, Izabela. (2020). Numbers on Thematic Maps: Helpful Simplicity or Too Raw to Be Useful for Map Reading?. *International Journal of Geo-Information*. 9 (7). 415. 10.3390/ijgi9070415.
- [2] Hazra, Subhjit & Bora, Kundan & Bora, Kundan & Singh, Preet Amol. (2024). Mapping World: A Comparative Analysis of Functionality and Usability between QGIS and ArcGIS. *Recent Advances in Computer Science and Communications*. 17. 10.2174/0126662558332529240919063643.
- [3] Chen, Taisheng & Chen, Menglin & Tan, Shanshan & Luo, Qi & Wu, Mingguang. (2011). Study on Analysis and Sharing of ArcGIS Symbol and SLD Symbol. 10.1007/978-3-642-23223-7\_7.
- [4] Paul A. Zandbergen. (2013). *Python Scripting for ArcGIS*. Esri Press. First edition. ISBN: 9781589482821.
- [5] Cybulski, Paweł. (2020). Spatial distance and cartographic background complexity in graduated point symbol map-reading task. *Cartography and Geographic Information Science*. 47. 1-17. 10.1080/15230406.2019.1702102.
- [6] VOŽENÍLEK, V. (2002). *Diplomové práce z geoinformatiky*. Olomouc: Univerzita Palackého v Olomouci.

Vedoucí diplomové práce: **Mgr. Radek Barviř, Ph.D.**  
Katedra geoinformatiky

Datum zadání diplomové práce: 9. prosince 2024  
Termín odevzdání diplomové práce: 4. května 2026

L.S.

---

prof. RNDr. Martin Kubala, Ph.D.  
děkan



---

prof. RNDr. Vilém Pechanec, Ph.D.  
vedoucí katedry

V Olomouci dne 10. prosince 2024

# OBSAH

<b>SEZNAM POUŽITÝCH ZKRATEK</b> .....	<b>9</b>
<b>ÚVOD</b> .....	<b>10</b>
<b>1 CÍLE PRÁCE</b> .....	<b>11</b>
<b>2 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY</b> .....	<b>12</b>
2.1 Kartodiagramy a jejich klasifikace.....	12
2.2 Číselné hodnoty na mapách.....	15
2.3 Číselné kartodiagramy .....	19
2.4 Tvorba číselných kartodiagramů v GIS softwarech .....	20
2.5 Technologie fontů a jejich standardy .....	22
2.6 Softwarové nástroje .....	23
2.7 Práce zabývající se automatizací kartografické tvorby.....	24
<b>3 METODY A POSTUP ZPRACOVÁNÍ</b> .....	<b>26</b>
3.1 Použité metody .....	26
3.2 Použité programy.....	26
3.3 Použitá data .....	27
3.4 Použití umělé inteligence (AI) .....	27
3.5 Postup zpracování .....	28
<b>4 NÁVRH A ARCHITEKTURA NÁSTROJE</b> .....	<b>30</b>
4.1 Technické přípravy a specifika .....	30
4.2 Analýza možností ArcGIS Pro .....	30
4.3 Konceptní návrh toolboxu .....	32
4.4 Návrh klasifikačních metod .....	37
4.5 Algoritmus gradací a barevných přechodů .....	39
<b>5 TVORBA TOOLBOXU V ARCGIS PRO</b> .....	<b>42</b>
5.1 Vytvoření .atbx toolboxu .....	42
5.2 Architektura toolboxu.....	42
5.3 Popis všech parametrů nástroje .....	46
5.4 Validace vstupů – třída ToolValidator.....	51
5.5 Klasifikace dat a výpočet tříd .....	58
5.6 Správa a vytváření label classes .....	62
5.7 Pokročilé stylování pomocí CIM API.....	63
5.8 Podpora variabilních fontů a interpolace váhy.....	65
5.9 Parsování barev a interpolace barevných gradientů.....	67
5.10 Konfigurace Maplex Label Engine .....	69
5.11 Optimalizace výkonu, ošetření chyb a diagnostika .....	70
5.12 Textové nápovědy, metadata a formální náležitosti.....	72
5.13 Uživatelské testování nástroje .....	73
<b>6 VÝSLEDKY</b> .....	<b>74</b>
6.1 Toolbox pro tvorbu číselných kartodiagramů.....	74
6.2 Ukázkové mapové výstupy .....	76
<b>7 DISKUZE</b> .....	<b>78</b>
<b>8 ZÁVĚR</b> .....	<b>80</b>
<b>POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE</b>	
<b>PŘÍLOHY</b>	

## SEZNAM POUŽITÝCH ZKRATEK

<b>Zkratka</b>	<b>Význam</b>
AI	Artificial Intelligence (umělá inteligence)
API	Application Programming Interface (rozhraní pro programování aplikací)
BIM	Building Information Modeling (digitální model stavby)
CAD	Computer aided design (počítačem podporované projektování)
CIM	Cartographic Information Model
CMYK / CMYKA	Cyan, Magenta, Yellow, black / Alpha (barevný model)
ČR	Česká republika
ČSÚ	Český statistický úřad
GIS	geografický informační systém
Git	systém pro správu verzí (např. GitHub)
GUI	Graphical User Interface (grafické uživatelské rozhraní)
GUS	Główny Urząd Statystyczny (polský Hlavní statistický úřad)
HEX	hexadecimální (šestnáctková) soustava / kód barvy (barevný model)
HSL / HSLA	Hue, Saturation, Lightness / Alpha (barevný model)
ICAO	International Civil Aviation Organization
IDE	Integrated Development Environment (integrované vývojové prostředí)
MS	Microsoft
OSM	OpenStreetMap
OTF	OpenType Font
PLN	polský zlotý (měnový kód)
RGB / RGBA	Red, Green, Blue / Alpha (barevný model)
SQL	Structured Query Language
TTC	TrueType Collection
TTF	TrueType Font
UI	User Interface (uživatelské rozhraní)
UP / UPOL	Univerzita Palackého v Olomouci
ZÚ	Zeměměřický úřad

# ÚVOD

Tematická kartografie neustále prochází technologickým vývojem, jehož cílem je co nejpřesněji a nejefektivněji vizualizovat prostorová data. K často využívaným metodám patří kartodiagramy, které reprezentují absolutní hodnoty kvantitativních jevů. Mezi jeho specifické varianty patří tzv. číselné kartodiagramy, které umožňují přímé vyjádření kvantitativních hodnot pomocí alfanumerických znaků umístěných v mapovém poli. Tato metoda je ceněna zejména pro svou schopnost zachovat přesnost numerických údajů a současně umožnit jejich rychlé prostorové porovnání. Gradace velikosti, barvy či tloušťky řezu písma takového znaku zobrazované hodnoty uživateli umožňuje rychlé odečítání hodnot bez nutnosti srovnávání s legendou (Korycka-Skorupa a Gołębiowska, 2020).

S rostoucím množstvím dostupných dat a požadavků na jejich efektivní prezentaci se zvyšuje význam automatizace kartografických postupů. Přestože moderní geografické informační systémy poskytují široké možnosti tvorby metod tematických map, některé specifické kartografické metody stále vyžadují značný podíl manuální práce. V prostředí softwaru ArcGIS Pro dosud neexistuje plně integrovaný nástroj, který by umožňoval efektivní tvorbu číselných kartodiagramů. Kartograf je proto nucen postupovat prostřednictvím opakovaného nastavování jednotlivých tříd popisků (anglicky label classes), úprav jejich grafických parametrů a definování intervalů, což je časově náročný a málo flexibilní proces.

Motivací pro zpracování této diplomové práce je snaha zefektivnit a zjednodušit proces tvorby metody číselných kartodiagramů prostřednictvím automatizace vybraných kroků. Automatizace těchto postupů může významně snížit časovou náročnost práce kartografa, minimalizovat riziko chyb vznikajících při manuálním nastavování a současně umožnit vytváření vizuálně konzistentních mapových výstupů. Význam této problematiky je podpořen také současným trendem rozvoje skriptování a automatizace v prostředí GIS, zejména s využitím programovacího jazyka Python a knihovny ArcPy, které umožňují efektivní manipulaci s kartografickými prvky a jejich vlastnostmi.

Tato diplomová práce se proto zaměřuje na návrh a implementaci nástroje umožňujícího automatizovanou tvorbu číselných kartodiagramů v prostředí ArcGIS Pro. Součástí řešení je nejen analýza současných možností tvorby těchto vizualizací ve dvou nejpoužívanějších GIS softwarech, ale také návrh algoritmů pro gradaci vizuálních proměnných, jako je velikost, barva a typografické vlastnosti písma. Výsledkem práce je vytvoření funkčního nástroje ve formátu geoprocessingového toolboxu (.atbx), který umožňuje flexibilní a uživatelsky přívětivou tvorbu tematických map s metodou číselných kartodiagramů.

Navržené řešení reaguje na existující technologickou mezeru v možnostech současných GIS nástrojů a přispívá k rozvoji automatizace kartografických postupů a rozšíření použití této metody tematické kartografie. Vytvořený nástroj má potenciál nalézt praktické uplatnění nejen ve vzdělávacím prostředí, ale také v odborné praxi, kde může sloužit jako prostředek pro rychlou a efektivní vizualizaci kvantitativních dat v mapách.

# 1 CÍLE PRÁCE

Cílem diplomové práce je automatizovat proces tvorby číselných kartodiagramů, tedy metody tematické kartografie, při které je množství jevu v mapě vyjádřeno alfanumerickým znakem s graficky gradující číselnou hodnotou za účelem efektivizace jejich tvorby. Zaměření práce vychází z potřeby minimalizovat množství manuálních kroků, které musí kartograf při tvorbě číselných kartodiagramů vykonávat, a přispět tak ke zvýšení jeho efektivity.

Práce se soustředí na vytvoření nástroje založeném na skriptu, který umožní uživatelsky přívětivou a flexibilní tvorbu číselných kartodiagramů s možností nejen gradace velikosti, ale také gradace barvy a ductu písma při možnosti nastavení regionálních specifik, jako je např. typografie popisků. Nedílnou součástí řešení je i analýza současných možností tvorby kartodiagramů v různých GIS softwarových prostředích a následná implementace poznatků do automatizovaného postupu v ArcGIS Pro.

Hlavním cílem diplomové práce je co největší **automatizace procesu tvorby číselných kartodiagramů**, a to specificky v prostředí ArcGIS. Pro splnění tohoto hlavního cíle je potřeba splnit jednotlivé dílčí cíle, kterou vypsány zde:

K teoretickým dílčím cílům práce patří rešerše věnující se porovnání definic a klasifikací kartodiagramů, jak v českém, tak zahraničním prostředí. Poté bude popsáno užití samotných **alfanumerických znaků na mapách** jakožto vizualizačního prostředku. Prolnutím těchto dvou zmíněných částí dojde k zaměření na číselné kartodiagramy a bude provedeno srovnání možností tvorby číselných, ale i jiných typů kartodiagramů v různých GIS softwarech. Následuje rešerše a analýza možností skriptování a způsobů automatizace v prostředí ArcGIS. Tato část bude věnována programovacímu jazyku Python, prostředí ArcGIS Pro a především knihovně ArcPy, která umožňuje automatizaci procesů, jako jsou klasifikace dat, úprava symbologie, nastavování popisků a další kroky nutné pro automatizaci procesu vytvoření číselného kartodiagramu.

V praktické části budou nasbírané vědomosti z teoretické části aplikovány pro vytvoření **nástroje (toolboxu)** za účelem automatizace tvorby číselných kartodiagramů v prostředí ArcGIS Pro.

Praktické dílčí cíle práce jsou:

- Návrh a rozvržení funkcí nástroje (toolboxu)
- Implementace vlastního nástroje, který umožní vytvořit číselný kartodiagram nejen s možností gradace velikosti, ale také barvy a řezu písma
- Vytvoření uživatelsky přívětivého prostředí, nápověd a dokumentace k nástroji
- Vytvoření několika ukázkových map z vybraných statistických dat

Automatizace tohoto procesu má za cíl výrazně snížit množství ruční práce, kterou musí kartograf v současnosti vykonávat při vytváření vizualizací pomocí této metody, což by vedlo k zefektivnění a zjednodušení jeho práce. Výstupy práce budou zveřejněny na webové stránce a vytvořeném posteru.

## 2 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

Teoretický základ pro vlastní řešení práce představuje rešerše dosavadních poznatků v oblasti tematické kartografie a souvisejících technologií. Následující text se zaměřuje především na vymezení pojmu kartodiagram a jeho detailní klasifikaci z pohledu předních českých i zahraničních autorů. Kromě toho nabízí podrobný rozbor specifik spojených s využíváním samotných číselných hodnot přímo v mapovém poli. Zásadní částí je rovněž kritické zhodnocení limitů i možností, které pro tvorbu těchto vizualizací poskytují současné GIS softwary, doplněné o vzhled do technologických standardů fontů a přehled nedávných automatizačních postupů aplikovaných na UPOL.

### 2.1 Kartodiagramy a jejich klasifikace

Podle Voženilka a Kaňoka (2011) jsou kartodiagramy mapová díla pro znázorňování kvantity, především pro znázorňování absolutních hodnot jevu, mající široké uplatnění v mnoha oborech, které pracují s prostorovými daty. Jsou vhodné pro srovnávání konkrétních hodnot v dílčích územních jednotkách na mapě, jako jsou např. počet obyvatel, objem produkce, spotřeby, nebo jiné socioekonomické údaje. Základní podmínkou metody kartodiagramu je, že se používá pro znázornění kvantitativních dat, konkrétně absolutních hodnot jevu a to tak, že se vypočtená velikost diagramu přiřadí v mapě k náležitému bodu, linii nebo areálu (Voženílek a Kaňok, 2011).

Zpracování dat se provádí komplexně a pro celou plochu mapy, nikoliv pro jednotlivé kartodiagramy. Umístění diagramů se u bodových kartodiagramů řeší umístěním na konkrétní pozice bodů, jako jsou např. jejich vztahné body, těžišti malých areálů, které v mapě vystupují jako body. Ve výjimečných případech je možné je posunout, například z důvodu překrytí nebo pro zdůraznění důležitosti dat. (Voženílek a Kaňok, 2011).

Voženílek a Kaňok (2011) klasifikují kartodiagramy v základním dělení na:

**Bodový** – ten znázorňuje kvantitativní charakteristiky jevů pomocí diagramů vztahených k bodům (např. k meteorologickým stanicím)

**Liniový** – jak již z názvu vyplývá, vztahuje se k liniím a umožňuje vyjádřit nejen velikost jevu, ale také směr jeho pohybu. Je pro něj důležité zvolit vhodně počáteční a koncový bod, směr a délku pohybu. (např. v ekonomii dopravy, nebo hydrologii)

**Plošný** – znázorňuje kvantitativní charakteristiky jevů pomocí diagramů vztahených k ploše (např. světadílů, regionů, administrativní jednotce)

Dále kartodiagramy rozdělují podle počtu znázorňovaných jevů, způsobu konstrukce, nebo účelu znázornění na:

**Jednoduchý** – zobrazuje pouze jeden jev pomocí diagramu

**Složený** – současně znázorňuje více jevů, přičemž každý z nich je znázorněn buď jiným typem diagramu, nebo stejným typem, ale kvalitativně rozlišenou barvou, nebo rastrem.

**Strukturní** – obsahuje diagramy stejné velikosti strukturně dělené pro znázornění jevů v bodech, liniích, nebo plochy. Nelze z nich však zjistit absolutní hodnotu jevu. U diagramů dává součet všech dílčích částí vždy 100 %.

**Součtový** – reprezentuje soubor diagramů, kde každý z nich zobrazuje velikost sledovaného jevu v absolutních hodnotách a zároveň zobrazuje vnitřní strukturu jevu. Jedná se tedy o diagramy, kde celkové velikosti jevu narůstají a jsou znázorňovány součtem jednotlivých dílčích složek.

**Srovnávací** – Skládá se dvou dílčích diagramů, konkrétně z jednoho o konstantní velikosti, který znázorňuje obvykle střední hodnotu jevu na sledovaném území, optimální, výchozí, nebo perspektivní hodnotu. Velikost diagramu vyjadřuje velikost jevu v dané pozici, nebo dílčím území celku.

**Dynamický** – vyjadřuje časově proměnlivé jevy. Využívají se diagramy, které znázorňuje alespoň tři časové údaje o jevu. Mohou být bodové, liniové, plošné, jednoduché, složené, kruhové, čtvercové, trojúhelníkové, čárové, nebo sloupcové.

**Segmentový** – překládá čtenáři statické údaje v segmentech uspořádaných do větších pravidelných obrazců, např. řádků, čtverců, obdélníků. Obrazce jsou vztaheny k plochám a méně často k bodům. Ve stupnici je k různě velkým segmentům přiřazena jiná hodnota jevu. Právě za tyto segmenty se nejčastěji používají geometrické, nebo symbolické znaky. Čtenář vnímá počet segmentů a jejich součtem odhaduje celkovou hodnotu jevu.

**Liniové kartodiagramy**, které jak již bylo zmíněno dříve, vyjadřují dvě informace o jevu, a to jeho velikost a směr. Podle toho se dále dělí na:

- **Vektorový** – Mají svůj střed, směry světových stran a délky vektorů reprezentující intenzitu jevu. Typickým příkladem jsou tzv. větrné růžice u meteorologických stanic.
- **Vektorový proudový** – na rozdíl od obyčejného vektorového kartodiagramu, tento nemá centrální bod a skládá se z proudu, nebo trsu vhodně lokalizovaných a podle sledovaného jevu správně směrově orientovaných šipek. Slouží např. pro znázornění mořských a vzdušných proudů.
- **Vektorový dosahový** – konstruuje se tak, že z určitého centrálního bodu, nebo ploše se rýsují rovné, dle potřeby i zaoblené čáry směrem k jiným bodům a plochám, se kterými je centrální bod v nějaké souvislosti.
- **Stuhový** – zachovává reálný průběh čar a vyjadřuje i podíly přemísťované kvantity jevu. Číselná hodnota jevu je vyjádřena celkovou šířkou stuhy, zatímco směr linie znázorňující průběh přemísťování jevu se mění. Má spoustu variant: jednoduchý, složený, strukturní, součtový, srovnávací, dynamický, izochronický, jednosměrný, dvousměrný apod.

Podle Miklína a kol. (2018) metoda kartodiagramu (value-by-size/count maps) vyjadřuje číselnou hodnotu geometrickým parametrem, tedy buď velikostí znaku, nebo počtem symbolů rozmístěných v určité ploše.

Kartodiagramy Miklín a kol. (2018) klasifikují:

Podle geometrie:

- **Figurální** – diagram, nebo graf umístěný v mapě, vztahený k bodovému či plošnému prvku. Mohou být neorientované, což je většina, nebo také orientované, tedy jejich orientace, nebo orientace části má geografický význam (např. větrné růžice).
- **Liniové** – liniová značka se šířkou jako parametrem
- **Areálové** – diagramy vyplňující celou plochu jednotky.

Podle složitosti, tedy množství vyjadřovacích parametrů na:

- **Jednoduché** – Mezi ně patří prosté, vícenásobné a specifické.
- **Komplexní** – Zde patří strukturní a složené, které se ještě dělí na jednoduché, strukturní a součtové.

S pojmem „kartodiagram“ (resp. „cartodiagram“) se však v anglicky psané literatuře nesetkáme. V angličtině se pro tuto kartografickou metodu používá označení „proportional symbol map“, „graduated symbol map“ nebo „proportional point symbol map“ (Voženilek a Kaňok, 2011).

Podle Denta a kol. (2009) se jedná o jednu z nejflexibilnějších a neoblíbenějších technik tematického mapování. Kartodiagramy, v této literatuře nazvány právě „proportional point symbol map“, jsou založeny na jednoduché myšlence, kdy kartograf vybere tvar symbolu (například kruh, čtverec, nebo trojúhelník) a mění jeho velikost v poměru ke kvantitě jevu, který představuje v daném místě. V této metodě jsou tvar a velikost dvě nejdůležitější vizuální proměnné. Důležitou roli může hrát také odstín nebo sytost barev v použitých znacích, zejména, pokud se na mapě zobrazuje více jevů najednou. Použitelnost této jednoduché metody může být narušena přílišným zaplněním mapy, nebo špatným výběrem měřítka pro zobrazení zvolených jednotek.

Slocum a kol. (2022) definují kartodiagramy jako metodu, která má za účel reprezentovat číselná, tedy kvantitativní data asociovaná s body a plochami. Rozděluje je na:

- **True Point data** – „Pravá“ data, která jsou skutečně naměřená v lokaci bodu (např. počet barelů ropy vyprodukuje ropný vrt).
- **Conceptual point data** – Data, která jsou sesbírána napříč celými plochami, ovšem umístěna do bodů za účelem symbolizace na mapě. (např. počet barelů ropy vyprodukovaných ropnými vrty v celém státě).

Dále se o kartodiagramy Slocum a kol. (2022) zajímá spíše skrze pravidla vizualizace a samotné symboly, které kategorizuje na: **geometrické** (např. tvary jako čtverce, kruhy, trojúhelníky, krychle) a **piktografické** (např. ilustrace obilného snopu, silueta člověka)

Podle Robinsona a kol. (1978) se jedná o široce využívanou kartografickou metodu. Rozdílné velikosti znaků se používají k symbolizaci buď kvantity jevu na konkrétních místech, nebo celkový součet hodnot vztahujících se k územním jednotkám. Jsou tedy užitečné tehdy, když se bodová data vyskytují v těsné blízkosti, ale mají vysoké souhrnné hodnoty, např. populace města. Dále např. pro znázorňování celkových hodnot veličin, jako je tonáž, náklady, intenzita dopravy, nebo pro vyjádření souhrnných hodnot vztahujících se k relativně velkým územím. Kruhový znak je jedním z nejstarších kvantitativních bodových znaků používaných pro statistické znázornění. Již na počátku 19. století byl využíván v grafech, které znázorňovaly tehdejší nová sčítací data. Jeho první výskyt na mapách, v podobě znaku na kartodiagramu se datuje do 30. let 19. století. Od té doby se pravidelně využívá, jako jedna z nejčastějších metod tematické kartografie (Robinson a kol., 1978). Dále se zabývá stupňováním velikosti kruhových znaků v kartodiagramech. Uvádí několik metod škálování znaků, jako jsou:

- **Proporcionální plošné (Proportional Area Scaling)/Metoda druhé odmocniny (Square Root)** – Velikost poloměru, nebo průměru kruhových znaků odpovídá druhé odmocnině hodnoty pozorovaného jevu. Díky tomu odpovídá velikost plochy/obsah kruhu proporcionálně velikosti hodnoty jevu, kterou představují a jsou tak k ní lineárně úměrné.
- **Rozsahově odstupňované (Range-Graded Scaling)** – Kategorizace jevů do odstupňovaných tříd/intervalů, pro které jsou velikosti znaků v jednotlivých třídách jednotné.

- **Psychologické (Psychological Scaling)** – Rozsáhlý výzkum v oblasti psychofyzických aspektů kartografických symbolů ukázal, že vnímaná reakce na rozdíl mezi velikostmi znaků není lineární funkcí. Běžný pozorovatel podceňuje velikost větších symbolů ve srovnání s menšími. Důkazy zvláště ukazují, že podceňování nastává mezi velikostmi kruhových znaků. Pokud jsou vytvořeny striktně úměrné hodnotám jevů, které představují, snížíme tím vizuální vnímanou velikost větších kruhů ve srovnání s menšími, tedy zvýšíme vizuální význam menších kruhů vůči větším. To je ovšem nežádoucí, neboť primárním cílem těchto map je efektivní vizualizace kvantitativních údajů tak, aby uživatel mapy mohl získat přesnou a věrnou představu o prostorovém rozložení zobrazovaných hodnot (Robinson a kol., 1978).

Kartodiagramy tedy představují metodu tematické kartografie určenou k vizualizaci kvantitativních dat, zejména absolutních hodnot (Voženílek a Kaňok, 2011). Přestože jednotliví autoři přistupují ke klasifikaci rozdílně – např. podle geometrie a složitosti (Miklín a kol., 2018), typu dat (Slocum a kol., 2022) či historického vývoje a metod škálování znaků (Robinson a kol., 1978) – shodují se na tom, že hlavním principem je využití velikosti, tvaru a případně barvy symbolu k vyjádření prostorově vázaných jevů. Rozdíly mezi přístupy spočívají zejména v terminologii a v důrazu na určité aspekty, jako jsou pravidla vizualizace (Dent a kol., 2009), psychologické vnímání velikostí znaků (Robinson a kol., 1978) či řešení směrových vztahů u liniových kartodiagramů (Voženílek a Kaňok, 2011). Společným jmenovatelem všech pohledů však zůstává důraz na čitelnost a věrné zprostředkování kvantity jevů v prostoru.

## 2.2 Číselné hodnoty na mapách

Přestože se číselné hodnoty na mapách používají celkem často, jen zřídka jsou předmětem rozsáhlého empirického výzkumu (Słomska, 2018), a proto jsou jen zřídka zahrnuta do různých klasifikací spolu s dalšími způsoby kartografické prezentace. Čitelnost čísel na mapách se může zlepšit, pokud splňují určité požadavky. Doporučuje se, aby se neskládala z příliš mnoha číslic, a proto aby byla někdy záměrně upravena (Brewer a Campbell, 1998). Nedoporučuje se například uvádět čísla s desetinnými čísly, která by mohla snížit čitelnost mapy. Naopak doporučuje se používat jednoduchý typ písma, jasně viditelné barvy, která je zvýrazní na pozadí mapy a rozumný počet čísel na mapě, aby nedošlo k přehučení a zhoršené interpretaci.

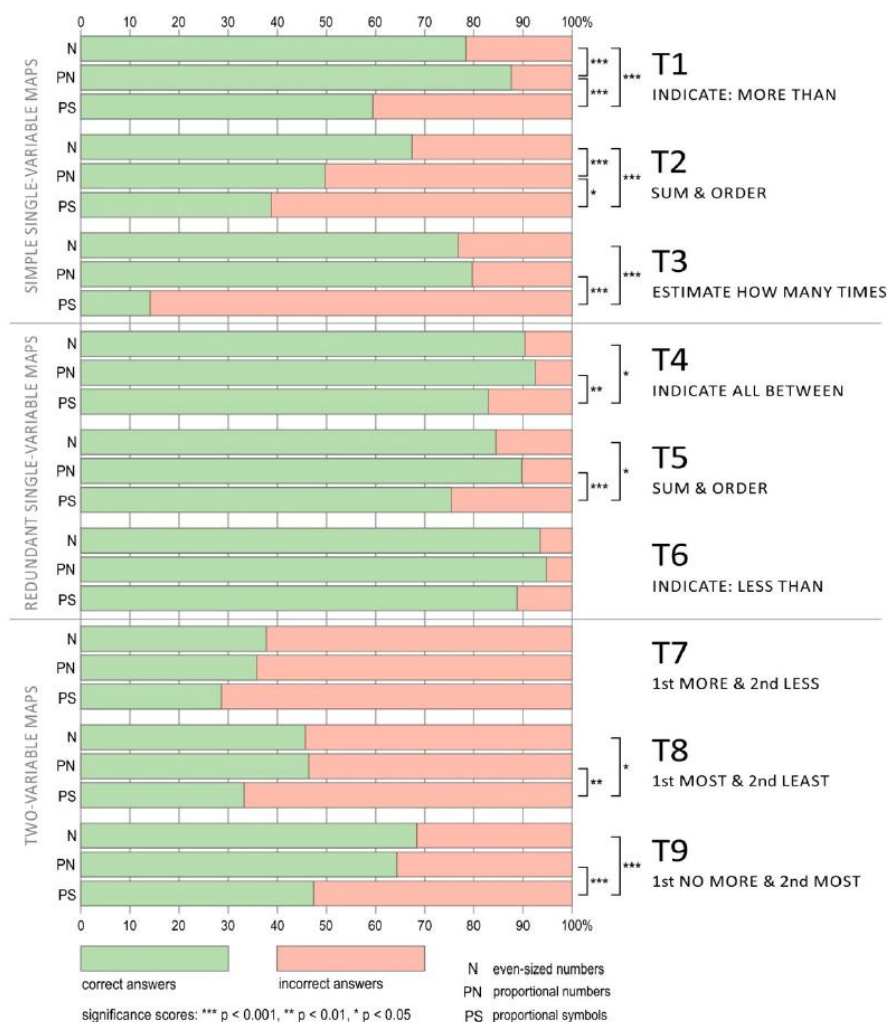
Čísla na mapách plní podobnou úlohu, jako jednoduchý kartodiagram, tedy v každé jednotce představují individuální hodnotu jevu v daném bodě, nebo ploše. Pokud se čísla na celé mapě prezentují stejnou grafickou podobou a hlavně velikostí, nazýváme je anglicky „**even-sized numbers**“. Česky tím rozumíme **alfanumerické znaky bez gradace velikosti**. Právě takové alfanumerické znaky mohou být doplněny jinou kartografickou metodou, např. jednoduchým kartodiagramem, který vyjadřuje stejný jev. V případě použití číselných hodnot přímo v mapovém poli, ať už ve formě popisu či samostatných alfanumerických znaků, pozbývá legenda mapy význam, jelikož hodnoty jsou explicitně uvedeny bez nutnosti je dekodovat z parametrů mapových znaků (Korycka-Skorupa a Gołębiowska, 2020).

Použití další kartografické metody, jako např. jednoduchý kartodiagram, nebo kartogram, jsou v tomto případě na mapě redundantní, tedy zbytečné, neboť vytvářejí „vizuální zátěž“ mapy. Někdy ale mohou mít naopak pozitivní dopad na čitelnost mapy (Shortridge a Welch, 2013).

Když se však vrátíme k metodě číselného kartodiagramu, kvantitativní charakteristiky se obvykle prezentují pomocí proměnných velikosti, barevné škály a jiných grafických metod, takže čtenář mapy si může rychleji všimnout vysokých hodnot, které prezentují větší velikosti písma, a získat tak rychlý přehled o rozložení kvantity jevu v mapovaném území.

## Uživatelské testování

Pro ověření využitelnosti samotných čísel v mapové vizualizaci byl v roce 2020 proveden experiment se 580 studenty z 20 polských středních škol. Jak je vidět na obrázku 1, účastníci řešili 9 úloh (**T1-T9**) založené na použití různých metod tematických map – konkrétně kartodiagramy s alfanumerickými znaky bez gradace velikosti (**N**), kartodiagramy s proporcionálními alfanumerickými znaky (**PN**) a jednoduché kartodiagramy s proporcionálními znaky (**PS**). Výsledky ukázaly, že nejvyšší úspěšnost měli studenti u map s redundancí (tj. kombinací více metod zobrazení stejného jevu), zatímco nejslabších výsledků dosahovali u map s proporcionálními symboly. Jak ukazuje obrázek 1, statisticky významné rozdíly byly zaznamenány mezi různými typy vizualizace, přičemž čistě číselné varianty často vedly k přesnějším odpovědím než grafické znaky (Korycka-Skorupa a Gołębiowska, 2020).



Obr. 1 Výsledky uživatelského testování čitelnosti různých metod vizualizace tematických map (zdroj: Korycka-Skorupa a Gołębiowska, 2020)

## Tvorba stupnic

Jedním ze základních předpokladů použití metody kartodiagramu je správná konstrukce stupnice, tedy systému převodu číselných hodnot na grafické znaky. Stupnice v tematické kartografii nepředstavuje pouze technický prostředek, ale je zásadním nástrojem, který rozhoduje o čitelnosti, přesvědčivosti a interpretaci mapového díla. Jak upozorňuje Kaňok (1999), „náhodně sestavená stupnice (někdy záměrně, někdy z neznalosti) nesprávně ovlivňuje čtenářovu interpretaci sledovaného jevu“. Stejně tak Kaňok a Voženílek (2008) zdůrazňují, že právě nesprávně konstruované stupnice patří mezi časté chyby v tematických mapách, protože buď potlačují regionální rozdíly, nebo naopak zdůrazňují odlišnosti, které v datech reálně neexistují.

Účelem dobře zvolené stupnice je podle Kaňoka (1999) zvýraznit prostorovou homogenitu sledovaného jevu a umožnit jeho regionální interpretaci. Kaňok a Voženílek (2008) v této souvislosti připomínají, že stupnice má nejen statistický, ale i vizuální rozměr – musí být pro uživatele srozumitelná a nesmí vést k dezinterpretaci. Proto je nezbytné, aby konstrukce stupnic vycházela z analýzy rozdělení četností, statistických charakteristik dat i účelu mapy.

Podle Kaňoka (1999) lze stupnice rozdělit do dvou hlavních skupin: **intervalové stupnice** a **funkční stupnice**. Jak doplňují Kaňok a Voženílek (2008), toto členění není pouze teoretické, ale má zásadní dopad na výslednou mapu, protože každý typ stupnice je vhodný pro jiný statistický soubor i typ vizualizace. Oba typy mají své podvarianty, které jsou dále rozebrány:

**Intervalové stupnice** – rozdělují variační rozpětí hodnot do třídních intervalů. Podmínkou jejich správného užití je, aby každý interval byl zastoupen alespoň jedním prvkem mapovaného souboru.

- **Plynule navazující intervalové stupnice**

- *Konstantní (aritmická)* – variační rozpětí je rozděleno na stejně široké intervaly. Výhodou je jednoduchost a snadná čitelnost, nevýhodou nízká schopnost vystihnout asymetrická rozdělení (Kaňok, 1999; Kaňok a Voženílek 2008).
- *Pravidelně rostoucí (klesající)* – zahrnují i geometrickou a logaritmickou stupnici. Podle Kaňoka (1999) se v praxi používají jen výjimečně, a to při silně asymetrických datech.
- *Nepřavidelné stupnice* – hranice intervalů jsou stanoveny podle průběhu rozdělení četností. Kaňok (1999) uvádí:
  - stupnice pro normální rozdělení (odvozené z průměru a směrodatné odchylky),
  - stupnice exponenciální (vhodné, pokud se četně vyskytují nízké hodnoty),
  - stupnice sedlové (hranice vycházejí z minima a maxima četností),
  - stupnice kvartilové, decilové či pentilové (odvozené od mediánu nebo průměru).

Kaňok a Voženílek (2008) zdůrazňují, že právě nepřavidelné stupnice mají největší potenciál respektovat skutečné vlastnosti dat, ale zároveň vyžadují nejvyšší míru statistické i kartografické znalosti.

- **Skokové intervalové stupnice** – jejich zvláštností je existence **hiátu**, tedy vynechání jednoho nebo více intervalů. Podle Kaňoka (1999) je hiát přípustný pouze tehdy, pokud se hodnoty z vynechaného intervalu v datech vůbec nevyskytují. Kaňok a Voženílek (2008) upozorňují, že hiát sice zvyšuje přehlednost legendy, ale musí být vždy jasně vyznačen, jinak by mohl čtenáře vést k mylnému závěru, že určitý rozsah hodnot byl autorem záměrně potlačen.

**Funkční stupnice** – nepoužívají třídní intervaly, ale převádějí hodnoty na velikost kartografického znaku pomocí matematického vztahu.

- sloupec – lineární vztah,
- čtverec a kruh – kvadratický vztah,
- krychle – kubický vztah.

Funkční stupnice se dále člení na:

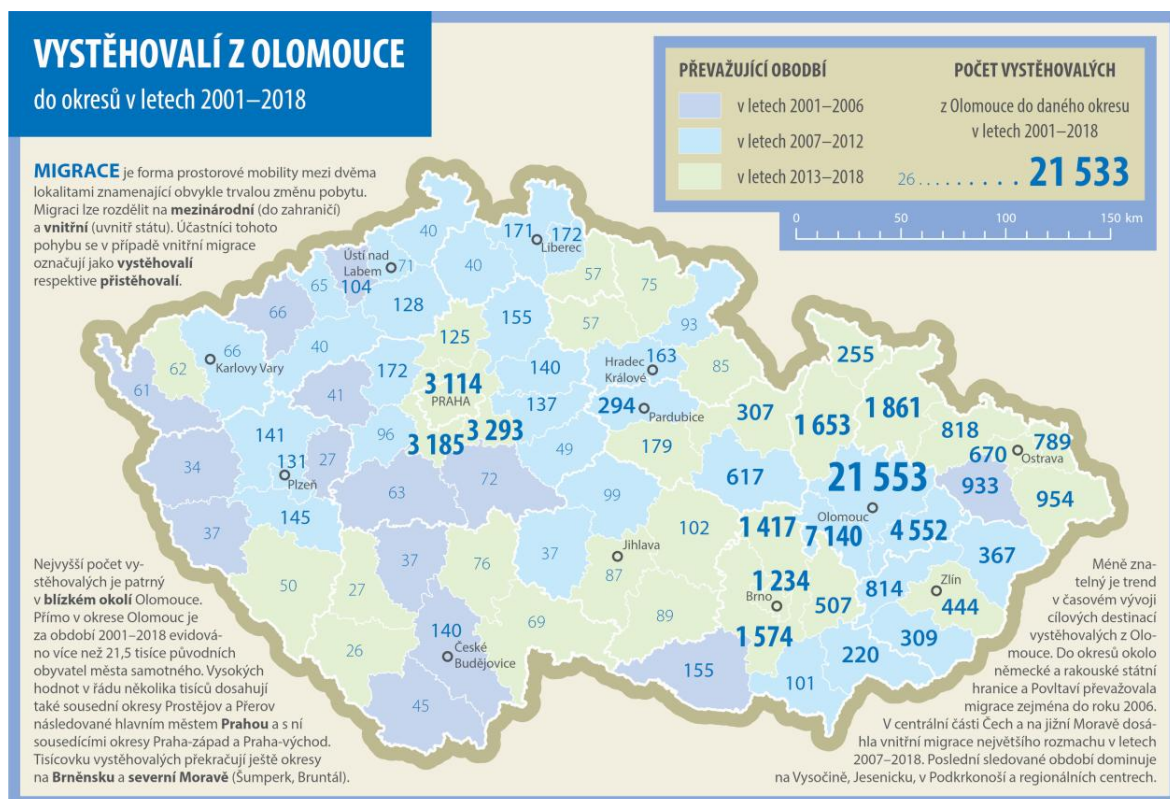
- **Spojitě funkční stupnice** – velikost každého znaku odpovídá hodnotě jevu prostřednictvím jednoznačně definovaného vzorce (Kaňok, 1999).
- **Skokové funkční stupnice** – jejich průběh je narušen. Podle Kaňoka (1999) a také Kaňoka a Voženílka (2008) existují dvě možnosti:
  - *skoková s hiátem* – vynechává se část hodnot, které se v souboru nevyskytují,
  - *skoková se změnou vzorce* – při extrémně vysokých hodnotách se původní matematický vztah nahradí jiným (např. kvadratický lineárním), aby byla zachována čitelnost mapy. Podle Kaňoka a Voženílka (2008) jde o krajní řešení, které musí být v legendě a doprovodném textu vždy transparentně uvedeno.

Jak tedy shrnuje Kaňok (1999), objektivní stupnice jsou nezbytným nástrojem tematické kartografie. Umožňují nejen správně vizualizovat data, ale i zvýraznit regionální rozdíly a homogenity. Kaňok a Voženílek (2008) doplňují, že chyby ve stupnicích patří k nejčastějším nedostatkům tematických map a mohou vést k zásadní dezinterpretaci. Proto je volba stupnice vždy kompromisem mezi statistickou objektivitou a grafickou čitelností mapy, který vyžaduje jak znalost metod, tak kritický přístup k výsledkům.

Tradiční koncepty tvorby stupnic však v dnešní praxi narážejí na limity běžného GIS softwaru, který často neumožňuje automatickou generaci plnohodnotných funkčních stupnic pro přesné dekodování hodnot. Na tento deficit reagují Barvíř a kol. (2025) vývojem online nástroje, který proces tvorby funkčních stupnic automatizuje a činí jej nezávislým na použitém softwaru. Specifické řešení rizik spojených s interpretací stupnic představují číselné kartodiagramy. Jejich výhodou je eliminace stupnice v legendě jako interpretačního prostředníka, neboť čtenář hodnotu vyčítá přímo z numerického údaje v mapě. Tím se předchází vizuálnímu zkreslení či chybám v hodnotovém měřítku, na které upozorňuje Kaňok (1999).

## 2.3 Číselné kartodiagramy

Jak již vyplývá z předešlých kapitol, číselný kartodiagram (anglicky Proportional či Graduated Numbers Map) je kombinací klasické metody kartodiagramu a přímého využití čísel v mapě, které zde vystupují jako vyjadřovací prvek samotného kartodiagramu. Podobně jako u běžných kartodiagramů, kde se kvantitativní data znázorňují pomocí velikosti a tvaru znaku (Voženílek a Kaňok, 2011), je i v tomto případě cílem vizualizace absolutních hodnot jevu, avšak **prostřednictvím alfanumerických znaků**, jak je vidět na obrázku 2. Ty mohou gradovat prostřednictvím jednoho či více grafických parametrů, jako je velikost písma nebo barevná škála (Korycka-Skorupa, Gołębiowska, 2020).



Obr. 2 Příklad využití číselného kartodiagramu s gradací velikosti alfanumerických znaků (zdroj: Néték a kol., 2020)

Číselný kartodiagram se jeví jako velmi intuitivní metoda vizualizace kvantity, protože čtenář je schopen přímo odečíst hodnoty, a v ideálním případě není ani potřebná legenda, pokud je vhodně zvolený název mapy (Słomska, 2018). Oproti klasickému kartodiagramu, kde může dojít ke ztrátě přesné informace důsledkem jejího shlazení, nebo zařazení hodnot do kategorií, umožňuje tato metoda snadné a přesné zjištění hodnot zkoumaného jevu. Významnou předností je tedy přímá interpretace dat, přičemž vizuální gradace znaků zároveň zachovává výhodu prostorového porovnání, která je vlastní i tradičním kartodiagramům (Voženílek a Kaňok, 2011).

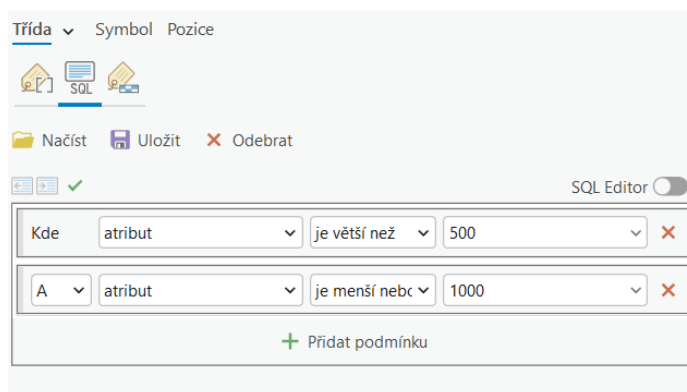
Na druhou stranu literatura upozorňuje i na limity číselných kartodiagramů. Brewer a Campbell (1998) doporučují omezit počet číslic a vyhnout se desetinným místům, které zhoršují čitelnost mapy. Shortridge a Welch (2013) ukazují, že redundantní kombinace čísel a jiných kartografických metod může někdy zvýšit čitelnost, ale jindy naopak rostoucí náplň mapy způsobuje vizuální zátěž pro uživatele.

Při konstrukci číselných kartodiagramů je rovněž nutné respektovat pravidla tvorby stupnic. Podobně jako u klasických kartodiagramů, i zde by gradace velikosti znaků měla vycházet z objektivně definované funkce (Kaňok, 1999). Kaňok a Voženílek (2008) upozorňují, že nevhodně konstruované stupnice mohou vést k potlačení regionálních rozdílů, nebo naopak ke zdůraznění odlišností, které v datech reálně neexistují. To platí i pro velikost písma a symbolů u číselných kartodiagramů – příliš velký rozsah může vést k přehuštění mapy a špatné interpretaci, zatímco příliš malý rozsah sníží kontrast a srozumitelnost.

## 2.4 Tvorba číselných kartodiagramů v GIS softwarech

### ArcGIS Pro

V prostředí softwaru ArcGIS Pro dosud neexistuje systémově integrovaný, automatizovaný a efektivní nástroj pro tvorbu číselného kartodiagramu. V případě zájmu o využití této metody je kartograf nucen postupovat prostřednictvím ruční tvorby a úpravy popisků (labelů), což představuje neintuitivní a časově náročné řešení. Dosavadní postup byl tedy vytvoření několika tříd popisků (labelů), jakožto kategorií a následně pro každou z nich nastavit pomocí SQL výběru (SQL Query) daný interval, který slouží jako hraniční hodnoty mezi jednotlivými kategoriemi. Jedna možnost je nastavení intervalů buďto pomocí přidávání podmínek (obrázek 3), nebo zadáním SQL příkazu, jako například: *(nazev\_atributu > 500 And nazev\_atributu <= 1000*; viz. obrázek 4). Styl písma a jeho gradaci je třeba nastavit rovněž manuálně, tudíž zatím nejefektivnější je font, barvu a řez písma nastavit hned na první vytvořené třídě, tu několikrát duplikovat, abychom dosáhli zamýšleného počtu tříd a následně u každé z nich je nutné manuálně přepsat hodnoty intervalů, měnit velikosti písma a popř. další libovolné grafické úpravy. Pro tento způsob je nejefektivnější, když má kartograf již předem přesnou představu o počtu tříd, hodnotách v jednotlivých intervalech, gradující velikosti písma a jeho celkové grafické vizualizaci. Pokud by se však rozhodnul cokoliv změnit v průběhu, musí danou změnu manuálně přenastavit ve všech třídách jednotlivě, což zabere další čas navíc, který se zvyšuje počtem vytvořených tříd.



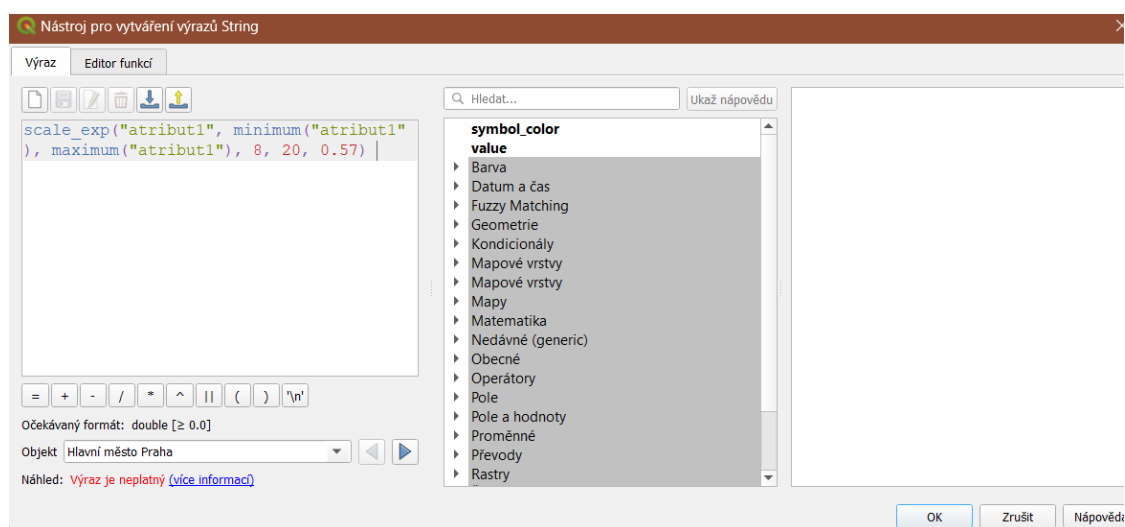
Obr. 3 Vizuální nastavení SQL dotazu pro definici hranic třídy popisků v softwaru ArcGIS Pro



Obr. 4 Přímý zápis filtrovací podmínky pomocí SQL příkazu v ArcGIS Pro

## QGIS

V prostředí softwaru QGIS je možnost vytvořit proporcionálně gradující číselný kartodiagram pomocí **Daty definovaného přepsání** hodnoty velikosti popisek. Přesný postup pro vytvoření jejich jednoduché varianty (bez dalších úprav stylu textu, jako např. různá výraznost barvy, nebo duktus písma) tedy je: Otevření vlastnosti vrstvy, zapnutím popisek v levém menu a vybráním možnosti „Jednotlivé popisky“. V záložce Text se následně zvolí chtěný typ písma, jeho styl, barva atd. Napravo od kolonky „Velikost“ se zvolí ikona Daty definovaného přepsání, kde se zvolí možnost „Upravit...“. Otevře se Nástroj pro vytváření výrazů String, kde se do levého pole zadávají python příkazy. Právě zde se zadá výraz `scale_exp`. Pro názornou ukázkou zvolíme výraz `scale_exp("atribut1", minimum("atribut1"), maximum("atribut1"), 8, 20, 0.57)`, jak je vidět na obrázku 5, kde „atribut“ je název zvoleného atributu vrstvy, minimum a maximum automaticky vyplní minimální a maximální hodnotu atributu, místo čehož by bylo možné zadat manuálně libovolný číselný interval, pro který by gradace fungovala, např. 300, 5000. Následně číslo 8 a 20 představuje min. a max. velikost písma a hodnota 0.57 reprezentuje exponent pro vizuálně vyrovnanější škálování, tedy pokud by byla zvolená hodnota 1, šlo by o čistě lineární škálování, naopak pokud je číslo menší než 1, velikostní rozdíly mezi velkými a malými hodnotami se zmenší.



Obr. 5 Využití funkce `scale_exp` pro automatickou gradaci velikosti popisek v softwaru QGIS

## 2.5 Technologie fontů a jejich standardy

V digitálním prostředí je písmo definováno prostřednictvím fontu, což je softwarový soubor obsahující vektorová data glyfů, tabulky metrik a instrukce pro vykreslování. Vývoj těchto technologií byl motivován snahou o odlišení, nezávislost a maximální platformovou kompatibilitu. Pro potřeby automatizované tvorby číselných kartodiagramů v ArcGIS Pro je klíčové porozumět vnitřní struktuře fontů, protože ta přímo určuje možnosti dynamické manipulace s velikostí, váhou a dalšími vizuálními parametry textových symbolů prostřednictvím CIM API (Esri, 2025c).

### Klasifikace a evoluce fontových formátů

Z hlediska datové struktury a způsobu uložení vektorových informací rozlišujeme několik dominantních standardů, které se v průběhu let staly průmyslovými normami:

- **TrueType (TTF):** Vyvinut společností Apple na konci 80. let 20. století a v roce 1991 licencován společností Microsoft. Formát využívá kvadratické Bézierovy křivky pro popis obrysů glyfů (Microsoft, 2020).
- **PostScript (Type 1):** Vyvinut společností Adobe v roce 1984. Využívá kubické Bézierovy křivky a byl dlouho standardem v předtiskové přípravě. Podpora tohoto formátu byla ukončena v novějších verzích Adobe softwaru od roku 2023 (Adobe, 2025).
- **OpenType (OTF):** Společný formát vyvinutý společnostmi Microsoft a Adobe na konci 90. let (první specifikace z roku 1997). Jedná se o univerzální „kontejner“, který může obsahovat jak TrueType, tak PostScript (CFF/CFF2) obrysy a podporuje pokročilé typografické funkce. Od roku 2019 je technicky ekvivalentní mezinárodnímu standardu ISO/IEC 14496-22 (Microsoft, 2024; ISO/IEC, 2019).

### Variabilní fonty (Variable Fonts)

Nejvýznamnější inovací posledních let je technologie OpenType Font Variations, která byla oficiálně představena v září 2016 společnostmi Microsoft, Google, Apple a Adobe jako součást specifikace OpenType 1.8. Na rozdíl od klasických statických fontů, kde je každý řez (např. Thin, Regular, Bold) uložen jako samostatný soubor, variabilní font obsahuje v jediném souboru celý designový prostor definovaný variačními osami (např. wght – weight/tloušťka, wdth – width/šířka, ital – italic, slnt – slant) (Google Open Source Blog, 2016; Microsoft, 2024).

Tento přístup umožňuje plynulou interpolaci mezi extrémními hodnotami bez nutnosti ukládat tisíce variant. Matematicky je založen na ukládání delt (rozdílů mezi výchozím a cílovým stavem osy), což výrazně snižuje velikost souboru a zvyšuje flexibilitu. Právě tato schopnost spojitě změny parametrů tvoří technologický základ pro gradaci tloušťky písma v číselných kartodiagramech.

### Specifika implementace v ArcGIS Pro

Prostředí ArcGIS Pro plně podporuje moderní **OpenType** specifikace včetně variabilních fontů. V CIM API (Cartographic Information Model) lze u variabilních fontů přistupovat přímo k jednotlivým variačním osám prostřednictvím vlastnosti **fontVariationSettings** objektu CIMTextSymbol (Esri, 2025c). Tato funkce je dostupná od verze ArcGIS Pro 3.0 a vyšší a umožňuje programově nastavovat hodnoty os (např. **wght** od 100 do 950) pro každou vytvořenou label class.

▷		Arial	Sample
▷		Arial	<b>Sample</b>
▷		Arial Rounded MT Bold	Sample
▷		ASGARD	<b>SAMPLE</b>
▷		Audiowide	Sample
▷		Avenir Next LT Pro	Sample
▷		Bahnschrift	Sample

Obr. 6 Identifikace typu fontů (TrueType, OpenType, variabilní) v uživatelském rozhraní ArcGIS Pro

Jak je patrné z obrázku 6, v prostředí ArcGIS Pro lze typ fontu snadno rozpoznat již v rozbalovacím seznamu písem. Statické fonty založené na **TrueType** (.ttf) obrysech jsou označeny ikonou „**TT**“, fonty s **PostScript Opentype** (.otf) CFF obrysy ikonou „**O**“ a variabilní fonty ikonou zlatého „**V**“ v levém horním rohu (např. Bahnschrift). V kontextu této diplomové práce je podpora variabilních fontů důležitá, protože umožňuje plynulou gradaci tloušťky písma (**font weight**) souběžně s gradací velikosti a barvy. Tím se dosahuje vyšší vizuální kvality a lepší čitelnosti číselných kartodiagramů.

## 2.6 Softwarové nástroje

### ArcGIS Pro

ArcGIS Pro je vlajkový desktopový GIS software od společnosti Esri, který představuje novou generaci nástrojů pro analýzu, správu a vizualizaci prostorových dat. Představuje budoucnost desktopového GIS ve společnosti Esri, přičemž veškerý vývojářský potenciál je nyní soustředěn výhradně do tohoto platformy (Peebles, 2020). ArcGIS Pro je dostupný ve třech úrovních licencí: Basic, Standard a Advanced. Basic obsahuje základní funkce, kdežto zbylé poskytují další pokročilé a rozšířené funkce, včetně specializovaných analýz (Esri, 2025b). Mezi klíčové vlastnosti patří integrované 2D a 3D prostředí, pokročilé analytické nástroje, cloudovou integraci a další podporu rozšíření různých. ArcGIS Pro byl poprvé představen v roce 2015 jako nástupce ArcMap postavený na 64bitové architektuře. Hluboká integrace s ArcGIS Online a ArcGIS Enterprise usnadňuje sdílení dat a colaboraci. Od své první verze prošel řadou aktualizací a momentálně nejnovější verze 3.6 má již integrované funkce podpory CAD a BIM a je rozšířena o další funkce (Elkins, 2015).

### Python

Jedná se o vysokoúrovňový, interpretovaný programovací jazyk, navržený v roce 1991 holanďanem jménem Guido van Rossum. Je to objektově orientovaný a interaktivní jazyk, který je vyvíjen jako open-source projekt, který je kompatibilní pro většinu běžných platform, jako jsou MS Windows, MacOS, Unix, Android a v systémech GNU a Linux je dokonce součástí základní instalace. Jeho hlavními výhodami jsou jednoduchá a přehledná syntaxe, která klade důraz na čitelnost kódu a díky velkému počtu volně dostupných knihoven umožňuje vývoj různých typů softwaru, od webových aplikací, datovou analýzu, automatizaci úloh až po vývoj umělé inteligence. (General Python FAQ, 2001; The Python Wiki, 2025) Jako instalátor balíčků a knihoven slouží např. takzvaný *pip* (Pip installs packages), které nabízí rozhraní příkazového řádku (General Python FAQ, 2001).

## Knihovna ArcPy

ArcPy je specializovaná knihovna jazyka Python vyvinutá společností Esri, která slouží jako hlavní rozhraní pro geografickou analýzu, automatizaci a správu prostorových dat. Původně byla představena jako nástupce staršího modulu *arcgisscripting* (Elkins, 2010), nicméně v současném prostředí ArcGIS Pro je využívána její modernizovaná verze založená na Pythonu 3. Knihovna poskytuje přímý přístup ke standardním geoprocením nástrojům, jejichž výstupy se pro potřeby řetězení úloh ukládají do objektů typu *Result*. K získávání metadat o prostorových prvcích slouží funkce *Describe*. Architektura ArcPy zahrnuje globální nastavení (environment settings), systémové třídy (např. *Spatial Reference*) a specifické moduly pro dílčí GIS disciplíny – jako je *arcpy.da* pro práci s daty nebo moderní *arcpy.mp* pro automatizaci mapových dokumentů (Esri, 2025a). Využití této knihovny tak zásadně zefektivňuje rutinní úlohy a umožňuje tvorbu reprodukovatelných pracovních postupů.

## Visual Basic Code

Visual Basic Code je integrované vývojářské prostředí vytvořené pro systémy MS Windows, Linux, macOS a pro webové prohlížeče. Obsahuje editor kódu, terminál, debugger, podporu ladění, zvýraznění syntaxe, automatické doplňování kódu, refaktorování, fragmenty a integrovanou správu verzí pomocí **Git** pro velký výběr programovacích jazyků, jako např. Python, Java, JavaScript, Node.js, C, C#, C++, Fortran, Go, Rust a další. Je vytvořen a vyvíjen společností Microsoft a vzniknul v roce 2015 jako freeware software. Novinkou tohoto prostředí je integrace asistenta **GitHub Copilot**, tedy umělé inteligence, která poskytuje návrhy kódu, dokáže do něj sama zasáhnout, vložit, upravovat ho, jednotlivé části vysvětlit a automatizovat implementaci na základě „promptů“ (pokynů) v přirozeném jazyce zvoleného kódu. Může tak pomáhat s většinou programovacích jazyků a frameworků (Visual Studio Code, 2015).

## 2.7 Práce zabývající se automatizací kartografické tvorby

### Automatizace zobrazení turistických tras na mapách

Diplomová práce Jakuba Pospíšila (Univerzita Palackého v Olomouci, 2024) se zaměřuje na automatizaci tvorby map s turistickými trasami. Autor řeší problém efektivního a opakovatelného zobrazení značených tras, které jsou v mapách vykreslovány paralelně s liniemi komunikací. Tento způsob znázornění se běžně uplatňuje v turistických, cyklistických či průvodcovských mapách, avšak manuální tvorba takových map je časově náročná a náchylná k chybám (Pospíšil, 2024).

Cílem práce bylo vytvořit algoritmus a následně Python skript pro prostředí **ArcGIS Pro**, který by proces vizualizace automatizoval. Skript umožňuje načítat data z **OpenStreetMap (OSM)**, analyzovat liniové průběhy turistických tras a generovat jejich posunuté (offsetové) paralelní zobrazení podél silniční sítě. Důraz byl kladen na univerzálnost řešení a minimalizaci manuálních zásahů kartografa (Pospíšil, 2024).

Výsledný skript byl testován na příkladu turistické mapy oblasti Hrubého Jeseníku a ukázal možnost automatizované produkce podkladových dat pro kartografické výstupy. Práce přináší přínos v oblasti automatizace tvorby tematických map a zároveň dokládá potenciál využití jazyka **Python** a knihoven **arcpy** při zefektivnění procesů v prostředí GIS.

## **Automatizace zpracování tras GNSS pro tvorbu map intenzity pohybu**

Diplomová práce Benjamína Šrama (2023) představuje komplexní přístup k **automatizovanému zpracování GNSS záznamů** za účelem tvorby map intenzity pohybu. Autor navrhl a implementoval nástroj pro automatické zpracování dat z GPS/GNSS zařízení (např. Strava, Garmin) a jejich převod do formy kvantitativních vizualizací – konkrétně za užití metod **intenzivních barev** a **kartodiagramů** (Šramo, 2023).

Významným přínosem je automatický **map matching** GNSS trajektorií na silniční síť (OSM), které zajišťuje přesnou prostorovou vazbu mezi naměřenými trasami a digitálním modelem komunikací. Nástroj byl testován na třech případových studiích – město Olomouc, pohoří Malá Fatra a národní park Slovenský ráj. Tato různorodá prostředí umožnila ověřit použitelnost nástroje jak v městském, tak i přírodním kontextu (Šramo, 2023).

Implementace probíhala v prostředí **Google Colab**, což je bezplatné cloudové prostředí **Jupyter Notebook** poskytované společností Google. Umožňuje uživatelům psát a spouštět kód Pythonu prostřednictvím prohlížeče, což je zvláště vhodné pro strojové učení, analýzu dat a vzdělávání (Burke, 2023). Výstupem jsou interaktivní webové mapy a prostorová data připravená k dalšímu zpracování v GIS (Šramo, 2023).

Práce tak přispívá k automatizaci procesu zpracování velkých objemů prostorových dat a demonstruje možnosti propojení datové analýzy, vizualizace a kartografické tvorby pomocí otevřených nástrojů a knihoven v Pythonu.

## **Společné rysy a přínos pro automatizaci kartografie**

Obě práce spojuje snaha **automatizovat vybrané fáze kartografické tvorby** – od zpracování vstupních dat až po vizualizaci.

- **Pospíšil (2024)** se zaměřuje na automatizaci vizualizační části (znázornění značených tras),
- **Šramo (2023)** naopak na automatizaci analytické části (zpracování GNSS trajektorií a tvorbu kvantitativních map).

Společným rysem je využití **Pythonu** a jeho **geoinformačních knihoven** a důraz na praktické využití pro kartografy a GIS specialisty. Tyto práce dokumentují současný trend v kartografii, kdy je cílem minimalizovat manuální práci kartografa a zvýšit tak produktivitu, dále propojit automatizované skriptování s vizuální komplexností mapových výstupů.

## 3 METODY A POSTUP ZPRACOVÁNÍ

Tato kapitola vymezuje metodický rámec diplomové práce a popisuje postup, kterým byl navržen, implementován a ověřen nástroj pro tvorbu **číselných kartodiagramů** v prostředí **ArcGIS Pro**. Nejprve jsou představeny použité metody, software a datové zdroje, následně je krok za krokem rozebrán proces vývoje, validace a testování, který vedl k vytvoření funkčního a prakticky využitelného řešení.

### 3.1 Použité metody

Pro stanovení metodického postupu byla nejprve provedena rešerše dostupné literatury, zaměřená na teoretické i praktické aspekty tvorby kartodiagramů. Studovaná literatura zahrnovala české i zahraniční kartografické publikace, práce z oblasti vizualizace statistických dat a obecné metodiky tematické kartografie. Pozornost byla věnována především:

- definici a klasifikaci popisků (labelů),
- principům číselných kartodiagramů a jejich rozdílům oproti tradičním (např. kruhovým, sloupcovým nebo součtovým),
- zásadám volby a konstrukce stupnic (aritmetických, geometrických, funkčních),
- možnostem a omezením vizualizace číselných hodnot v mapovém poli.

V rámci rešerše byly analyzovány a popsány také současné softwarové přístupy v prostředí **ArcGIS Pro** a **QGIS**. Záměrem bylo identifikovat, jakými způsoby lze v těchto programech realizovat metodu číselných kartodiagramů, a zda již existuje přímá podpora této metody. Výsledkem bylo zjištění, že číselné kartodiagramy nejsou v ArcGIS Pro přímo implementovány, ale lze je napodobit prostřednictvím popisků a jejich stylizace. Tento zjištěný nedostatek se stal hlavní motivací pro návrh vlastního automatizovaného řešení. Metodika dále vychází z principů **iterativního vývoje softwarového nástroje**, kdy jednotlivé funkce byly navrhovány, implementovány a průběžně testovány na vybraných datových sadách. Tento přístup umožnil pružně reagovat na zjištěné nedostatky a postupně rozšiřovat funkcionalitu nástroje.

### 3.2 Použité programy

K realizaci práce bylo využito několik softwarových prostředků:

- **ArcGIS Pro (verze 3.4, 3.5 a 3.6)** – hlavní pracovní prostředí, v němž byl vyvinut vlastní toolbox. ArcGIS Pro umožňuje integraci skriptů v jazyce Python a přístup k objektům mapy prostřednictvím knihovny *ArcPy*.
- **Visual Studio Code** – open-source editor s podporou mnoha jazyků a pokročilou správou kódu, převyšující vestavěný editor v ArcGIS Pro. Významným prvkem využití byla integrace **GitHub Copilot**; tento AI asistent v reálném čase navrhoval optimalizace, doplňoval rutinní pasáže a pomáhal s identifikací logických chyb.
- **GitHub** – platforma pro správu verzí a verzování zdrojového kódu. Slouží k uchování historie vývoje, zálohování a sdílení nástroje pro veřejnost. Díky integraci s Visual Studio Code bylo možné pohodlně provádět změny a sledovat jejich postupný vývoj.
- **MS Excel** – použit pro úpravu a kontrolu dat před jejich importem do GIS.
- **další podpůrné programy** – pro zpracování textu (**MS Word**), tvorbu prezentací (**MS PowerPoint**), tvorbu myšlenkových map a přehledu postupu/kroků (**Xmind**) a finální grafické úpravy mapových výstupů a tvorba posteru (**Affinity**).

### 3.3 Použitá data

Pro naplnění cílů diplomové práce bylo nutné využít data, na kterých bylo možné testovat funkčnost nástroje a tvořit ukázkové mapy. Výběr byl motivován snahou otestovat funkčnost algoritmů na geometricky i tematicky odlišných datech, v různých zobrazeních. Většina datových sad byla čerpána přímo v prostředí **ArcGIS Pro prostřednictvím portálu ArcGIS Online**, což rovněž umožnilo testovat robustnost nástroje při práci s hostovanými webovými vrstvami a cloudovou infrastrukturou.

#### Přehled datových zdrojů

Následující tabulka shrnuje zdroje dat využitých pro tvorbu ukázkových výstupů, důraz byl kladen na různost vizualizovaných území:

Tabulka 1 Přehled využitých datových sad a jejich poskytovatelů pro tvorbu ukázkových map

Poskytovatel / Autor	Název datové sady	Územní rozsah	Typ dat
<b>ARCDATA PRAHA, ZÚ, ČSÚ</b>	Kraje (ArcČR 2024)	Česká republika	Vektorová (ArcGIS Online)
<b>MB Research / Esri Germany</b>	Germany Bundeslander Boundaries	Německo	Vektorová (ArcGIS Online)
<b>Mareike Kortmann</b>	112 Standorte - Feuer- und Rettungswachen	Německo	Bodová (ArcGIS Online)
<b>GUS / Esri Polska</b>	Population Variables Boundaries [Dane demograficzne]	Polsko	Vektorová (ArcGIS Online)
<b>Jennifer Boehnert</b>	country_Europe	Evropa	Vektorová (ArcGIS Online)
<b>OurAirports.com</b>	airports.csv	Globální	Bodová (Tabulka (.csv))

### 3.4 Použití umělé inteligence (AI)

Při zpracování této diplomové práce a vývoji nástroje byly využity systémy generativní umělé inteligence, konkrétně nástroje **GitHub Copilot** a **Google Gemini**. Využití těchto technologií probíhalo v souladu s *Doporučením Univerzity Palackého v Olomouci k využívání generativních modelů AI* a etickými principy UP.

Nástroje byly využity jako pomocné (ediční a programátorské) prostředky v následujícím rozsahu:

- **GitHub Copilot:** Integrovan jako asistent v prostředí *Visual Studio Code* pro zefektivnění zápisu zdrojového kódu. Nástroj byl využit k návrhu syntaktických struktur, doplňování rutinních pasáží kódu a k identifikaci potenciálních chyb při implementaci algoritmů pracujících s ArcGIS Pro.
- **Google Gemini:** Využit k analýze a vysvětlení technické dokumentace ArcGIS Pro a ArcPy a pro korekturu stylistické úrovně vybraných pasáží textu.

Autor práce výslovně prohlašuje, že nástroje **nebyly využity k formulaci vlastních tvrzení, závěrů, interpretaci výsledků ani k argumentaci v diskuzi**. Veškeré výstupy generované AI byly podrobeny lidskému ověření z hlediska faktické správnosti a bezpečnosti. Autor nese plnou a výhradní odpovědnost za konečnou podobu textu i funkčnost nástroje a potvrzuje, že využití AI nenahradilo jeho vlastní tvůrčí činnost nezbytnou pro vznik této práce.

### 3.5 Postup zpracování

Postup práce byl sestaven na základě konzultací s vedoucím práce Mgr. Radkem Barviřem, Ph.D., který poskytoval doporučení a rámcový harmonogram prací, jež provázely zpracování diplomové práce. Postup lze shrnout do několika hlavních kroků:

- **Studium literatury a rešerše**

Prvním krokem bylo prostudování odborných publikací a článků, které se zabývají kartodiagramy a obecně vizualizací kvantitativních dat. Pozornost byla věnována zejména tomu, jak jsou kartodiagramy definovány, tvorbě stupnic, jaké jsou výhody číselných kartodiagramů v porovnání s jinými metodami a jak se s nimi pracuje v současných GIS softwarech.

- **Analýza možností ArcGIS Pro**

Nejprve byla provedena detailní analýza funkcionalit ArcGIS Pro, zejména nástrojů pro tvorbu popisků. Bylo zjištěno, že ačkoli software nabízí široké možnosti úpravy alfanumerických znaků, neexistuje přímý nástroj pro tvorbu číselných kartodiagramů s pokročilou typografií a automatizací. Tyto limity definovaly praktické cíle práce – vytvořit toolbox, který celý proces výrazně usnadní, zpřehlední a zautomatizuje.

- **Návrh toolboxu**

Na základě zjištěných nedostatků byl navržen vlastní nástroj s důrazem na přehlednost a uživatelskou přívětivost. Parametry byly nakonec rozděleny do pěti kategorií (**Data Statistics, Classification & Size, Font Styling, Format & Units, Advanced Styling**), což umožňuje uživateli snadno najít relevantní nastavení. Klíčovým principem návrhu je dynamická aktivace parametrů – některé volby se zpřístupní až po výběru související možnosti, čímž je minimalizována kognitivní zátěž uživatele a riziko chybného zadání.

- **Implementace v prostředí Python/ArcPy**

Vlastní implementace probíhala v jazyce Python s využitím knihovny arcpy. Toolbox byl vytvořen jako skriptový nástroj ve formátu **.atbx**, plně integrovaný do prostředí ArcGIS Pro. Vývoj byl rozdělen do několika fází:

- Nejprve byly implementovány klasifikační metody a jejich aplikace na data zvoleného atributu.
- Následovalo generování popisků a základní stylizace.
- Postupně byly přidávány pokročilé možnosti grafické a typografické stylizace, včetně podpory proměnných fontů, barevných a halo gradientů, interpolace font-weight a dalších moderních prvků.

- **Validace, metadata a formální náležitosti**

Součástí vývoje bylo pečlivé vyplnění popisů (**description**) u všech parametrů, nastavení výchozích hodnot, přiřazení kategorií a doplnění metadata na úrovni celého toolboxu. Veškeré vstupy uživatele jsou validovány ještě před spuštěním výkonného kódu, což zvyšuje robustnost a uživatelskou přívětivost nástroje.

- **Testování a ladění**

Po vytvoření funkčního prototypu následovalo testování na různých datových sadách. Ověřovala se správnost aplikovaných klasifikačních metod, stabilita nástroje při práci s většími datovými soubory a vizuální kvalita výsledných číselných kartodiagramů. Následovalo uživatelské testování, které bylo provedeno se studenty 1. ročníku magisterského studia v rámci předmětu VIZUL. Na základě tohoto testování byly prováděny dílčí úpravy a optimalizace.

- **Tvorba ukázkových map**

Posledním krokem bylo vytvoření souboru mapových výstupů, které demonstrují možnosti vyvinutého toolboxu. Tyto mapy zahrnují příklady s různým počtem tříd, různou metodou klasifikace a různým stylem vizualizace popisků. Slouží jako ukázka praktického využití nástroje a zároveň jako podklad pro vyhodnocení přínosů práce.

Celý metodický postup kombinuje teoretická východiska tematické kartografie se softwarovou implementací v prostředí ArcGIS Pro. Výsledkem je nástroj, který umožňuje automatizovanou tvorbu metody **číselných kartodiagramů** a rozšiřuje praktické využití této metody v GIS.

## 4 NÁVRH A ARCHITEKTURA NÁSTROJE

Návrh a architektura vyvinutého nástroje pro automatizovanou tvorbu gradovaných číselných kartodiagramů představují téma následujícího textu. Cílem je představit hlavní principy, na kterých je řešení postaveno, a objasnit stěžejní rozhodnutí, která ovlivnila výslednou podobu toolboxu. Pozornost je věnována jak celkové koncepci, tak jednotlivým komponentám a jejich vzájemným vazbám.

### 4.1 Technické přípravy a specifika

Vývoj vlastního geoprocessingového nástroje byl od počátku vázán na specifika architektury softwaru ArcGIS Pro a jeho nativního prostředí. Před samotnou implementací kódu bylo nutné detailně definovat cílové funkce, technologické závislosti a pracovní postup (workflow) tak, aby byla zajištěna přenositelnost a stabilita výsledného řešení.

#### Cílový software a vývojové prostředí (IDE)

Nástroj byl vyvíjen a testován výhradně pro prostředí desktopové aplikace **ArcGIS Pro**. Prvotní návrh, prototypování a implementace základní logiky probíhaly ve verzi **3.4**. V průběhu vývoje byl software aktualizován a finální testování včetně ladění (debugging) proběhlo na verzi **3.6**.

Samotné psaní kódu však neprobíhalo v integrovaném Python okně toolboxu v ArcGIS Pro, které je nedostačující pro vývoj komplexních skriptů, nýbrž v externím vývojovém prostředí **Visual Studio Code**. Toto prostředí umožnilo striktní dodržování formátovacího standardu PEP 8, efektivní kontroly kódu (linting) a izolované testování výrazů.

#### Aktivní prostředí a správa balíčků

Klíčovým rozhodnutím pro zajištění snadné distribuce nástroje koncovým uživatelům bylo zachování výchozího běhového prostředí (run-time environment). Nástroj využívá výchozí systémový profil **arcgispro-py3**, který obsahuje interpreter Pythonu verzi **3.11**, následně **3.13** a také proprietární knihovnu **arcpy**.

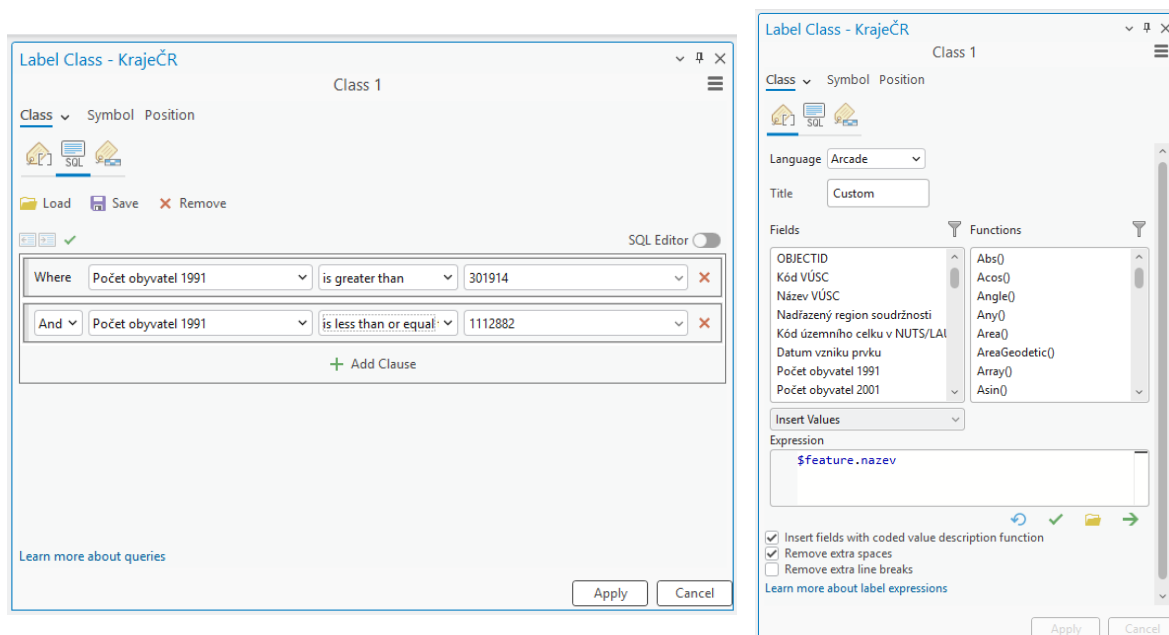
Během vývoje nebylo tedy nutné klonovat výchozí prostředí (Conda environments) a vytvářet izolované instalace. Namísto toho byly maximálně využity takzvané vestavěné (built-in) moduly standardní knihovny jazyka Python.

### 4.2 Analýza možností ArcGIS Pro

V úvodní fázi byla provedena analýza nativních funkcionalit ArcGIS Pro se zaměřením na nástroje pro tvorbu a stylizaci popisků (labeling). Tato fáze zahrnovala testování a experimentování s dostupnými funkcemi panelu *Labeling*, s cílem zjistit, do jaké míry lze číselné kartodiagramy vytvořit pomocí standardních ovládacích prvků softwaru a jak dlouho časově zaberou dílčí manuální kroky.

Během analýzy možností bylo zjištěno, že ačkoliv ArcGIS Pro disponuje robustním rozhraním pro správu popisků (labelů), proces tvorby stupnic je pro účely číselných kartodiagramů **značně uživatelsky nepřívětivý**. Standardní postup vyžaduje pro každou klasifikační třídu manuální vytvoření samostatné **třídy popisků (Label Class)**, přičemž uživatel má pro vymezení intervalů a vizuální úpravu k dispozici dva základní mechanismy, které může využívat buď samostatně, nebo v kombinaci:

- **Vizuální dotazovací rozhraní (SQL Query):** Slouží k filtraci prvků, které do dané třídy spadají. Toto rozhraní je sice intuitivní, ale při požadavku na častou změnu počtu intervalů či klasifikační metody se stává nepraktickým, neboť vyžaduje manuální přepisování podmínek u každé třídy zvlášť, byť je zde možnost uložení nastavení a opětovného načtení (obrázek 7).
- **Modul pro tvorbu výrazů (Label Expression):** Na rozdíl od SQL Query se zde výsledná podoba popisku musí definovat **přímým zápisem kódu**. Uživatel má možnost volby mezi několika skriptovacími jazyky: Arcade, VBScript, JScript nebo Python. Zápisem v tomto kódu se však může kromě filtrování prvků dosáhnout pokročilejších úprav textu, jako například jeho formátování, např. český formát řádů jednotek, desetinné čárky atd (obrázek 7).



Obr. 7 Uživatelské rozhraní pro tvorbu SQL dotazu u třídy popisků (Label Class) v anglické lokalizaci ArcGIS Pro

Zásadní zlom v analýze nastal při zkoumání možností rozhraní ArcGIS Pro **CIM API** (Cartographic Information Model). Bylo zjištěno, že na rozdíl od běžného uživatelského rozhraní, které je primárně navrženo pro individuální nastavení symboliky, skrze CIM API lze k vlastnostem popisků přistupovat na mnohem nižší úrovni. CIM API nepředstavuje přístup k jinak nedostupným funkcím, ale spíše efektivní způsob, jak ovládat detaily, které v běžném uživatelském rozhraní vyžadují opakovanou a zdlouhavou manuální interakci.

Tyto zjištěné limity a následné objevení potenciálu CIM API definovaly praktické cíle práce: namísto pouhého skriptu pro výpočet intervalů vytvořit komplexní toolbox, který proces tvorby číselných kartodiagramů plně automatizuje tím, že přímo manipuluje s objekty CIM, čímž uživatelům zpřístupňuje pokročilou kartografickou stylizaci, ke které by se jinak musel „proklikat“ skrze zdlouhavá manuální nastavení.

## 4.3 Koncepční návrh toolboxu

Před zahájením samotné práce a implementace bylo nezbytné definovat ucelenou představu o podobě a schopnostech vyvíjeného nástroje. Návrh vychází ze zadání práce, které klade důraz na automatizaci a uživatelskou přívětivost při tvorbě číselných kartodiagramů v prostředí ArcGIS Pro i přes jeho technická omezení. Cílem bylo navrhnout rozhraní, které by bylo intuitivní a umožňovalo variabilitu výstupů podle potřeb konkrétního kartografického díla. Dalším nutným aspektem je důraz na výraznou úsporu času oproti manuální práci uživatele.

Součástí koncepčního návrhu byla i volba jazykové lokalizace nástroje. Bylo rozhodnuto, že nástroj bude vyvíjen v **anglickém jazyce** s cílem zajistit maximální univerzálnost a využitelnost pro co nejširší okruh uživatelů. Zatímco technické pozadí a systémové proměnné reflektují **americký standard (US)** kvůli nativní vazbě na jádro softwaru ArcGIS Pro a samotný jazyk Python, uživatelské rozhraní (User Interface), textové části a doprovodné nápovědy jsou navrženy v **britské angličtině (UK)**. Tento přístup byl zvolen z důvodu sjednocení s evropskou zvyklostí a standardy odborné komunikace, čímž je dosaženo přirozenějšího uživatelského rozhraní pro evropského uživatele. Pro zvýšení čitelnosti a estetickou jednotu s nativními nástroji ArcGIS Pro je u všech viditelných názvů parametrů využito formátování **Camel Case** (např. *Layer Selection*, *Use Colour Gradient*), což vizuálně odděluje uživatelské rozhraní od technických názvů parametrů v kódu.

### Volba vhodného názvu toolboxu

Výběr názvu nástroje prošel několika fázemi, přičemž cílem bylo najít označení, které bude krátké, srozumitelné a okamžitě definující hlavní funkci skriptu. Původní pracovní názvy jako *Proportional Numbers Label Generator* nebo *Graduated Number Labels Generator* byly však nakonec zamítnuty. Výsledný název „**Graduated Numbers Generator**“ byl zvolen z následujících důvodů:

- **Terminologická shoda:** Slovo *Graduated* (stupňovaný) přímo odkazuje na kartografickou metodu kartodiagramů (*Graduated Symbols Map*), která se v ArcGIS Pro standardně používá pro vizualizaci kvantitativních dat. Uživatel tak okamžitě pochopí analogii mezi nativní symbolologií a tímto generátorem popisků.
- **Zaměření na číselná data:** Doplnění o slovo *Numbers* jasně vymezuje, že nástroj je určen pro vizualizaci numerických atributů, nikoliv pro obecný styling textových popisků a vychází opět z názvu metody číselných kartodiagramů (*Proportional nebo Graduated Numbers Map*).
- **Funkční označení:** Výraz *Generator* podtrhuje automatizační charakter nástroje – uživatel pouze definuje parametry a nástroj na jejich základě „vygeneruje“ komplexní strukturu tříd popisků se zvoleným grafickým nastavením a typografií.

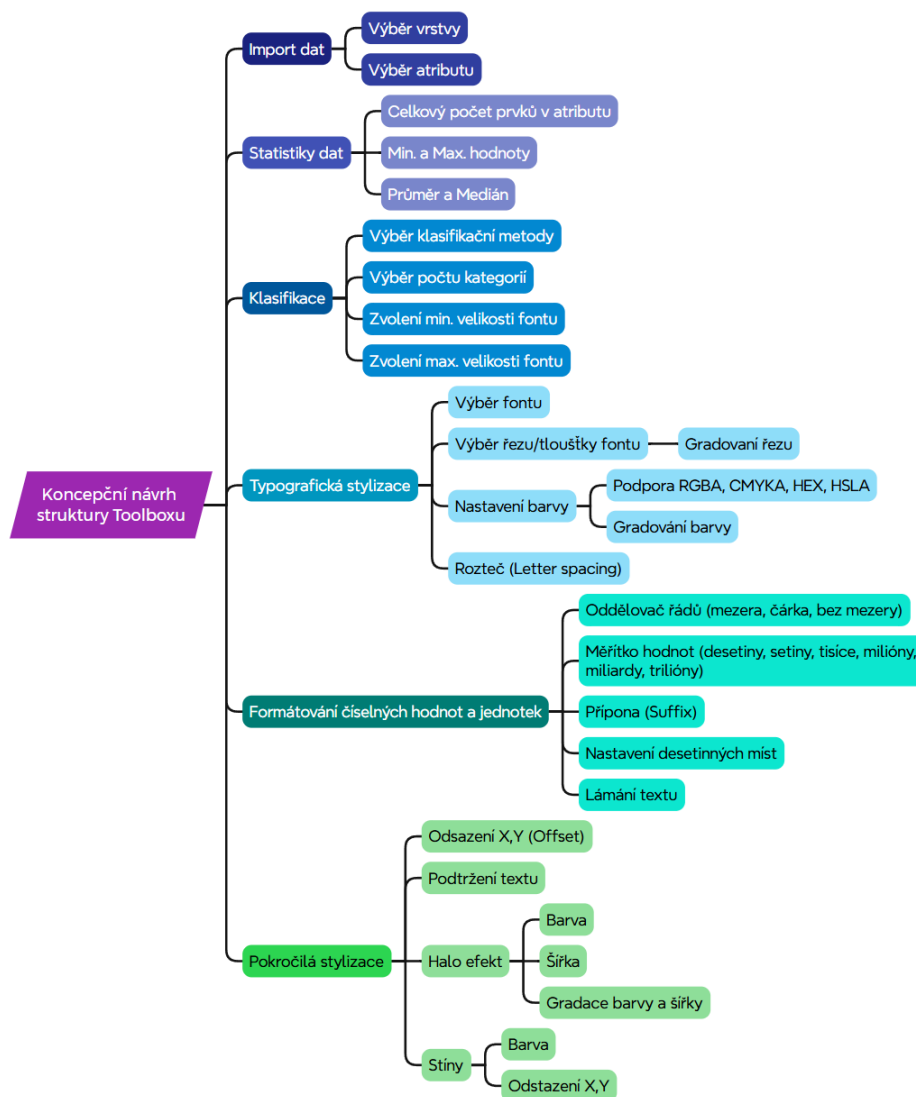
### Funkční požadavky a vymezení parametrů

Návrh nástroje (toolboxu) byl strukturován do několika logických bloků, které reflektují postup práce kartografa při použití metody číselných kartodiagramů v prostředí ArcGIS Pro. Ten zahrnuje jako první načtení a zvolení vstupních dat, jejich klasifikace a následná grafická úprava výsledné podoby alfanumerických znaků pro vizualizaci. Tvořený nástroj by tedy měl mít části, které z tohoto postupu přímo vycházejí:

- **Interaktivní výběr dat a kontrola rozsahu:** Po výběru vstupní vrstvy a atributového pole nástroj okamžitě poskytuje zpětnou vazbu v podobě celkového počtu prvků, minimální a maximální hodnoty atributu a základních statistických údajů (průměr a medián). Tato informace slouží k rychlé orientaci uživatele a správnému nastavení následné klasifikace.

- **Flexibilní výběr klasifikačních metod:** Nástroj podporuje standardní statistické metody používané v ArcGIS Pro – **stejněměrné intervaly** (*Equal Interval*), **kvantily** (*Quantile*), **přirozené zlomy** (*Jenks Natural Breaks*), **geometrické intervaly** (*Geometric Interval*), **manuální** definici tříd uživatelem a **proporcionální (neklasifikovanou)** metodu – *Proportional (Unclassed)*, u které se počet kategorií rovná počtu unikátních hodnot (případně počtu prvků) a velikost písma se mění rovnoměrně v zadaném rozsahu. Uživatel zde také nastavuje minimální a maximální velikost písma. Velikost textu je u číselných kartodiagramů jedním z hlavních vyjadřovacích prostředků kvantily vedle samotné číselné hodnoty, proto je její nastavení součástí klasifikační sekce.
- **Typografická stylizace:** Vzhledem k tomu, že číselný kartodiagram vyžaduje vysokou čitelnost alfanumerických znaků, byl kladen důraz na detailní nastavení písma. Nástroj se dynamicky napojuje na systémové i uživatelské fonty v operačním systému Windows a umožňuje volbu nejen rodiny fontu, ale i jeho řezu. Barvu textu lze definovat v několika barevných modelech (**HEX, RGBA, CMYKA, HSL**). Rozšířenou možností je **barevný gradient textu**, který umožňuje definovat počáteční, středovou a koncovou barvu. Nástroj následně pomocí lineární interpolace vypočítá hodnoty pro každou klasifikační třídu, čímž vzniká plynulý barevný přechod doplňující velikostní stupnici písma.
- **Lokalizace a formátování numerických údajů:** Pro české kartografické prostředí je důležitá možnost přizpůsobení formátu čísel. Nástroj proto umožňuje volbu **oddělovače tisíců** (čárka, mezera nebo žádný), z čehož vychází použití desetinné tečky, nebo čárky. Součástí je také nastavení **měřítko hodnot** (value scaling), která umožňuje převést velká čísla na přehlednější jednotky (např. tisíce nebo miliony) přímo při generování popisků. Uživatel dále určuje počet desetinných míst po škalování a může přidat **textovou příponu** (suffix). Veškeré formátování probíhá pouze v generovaných popiscích, aniž by docházelo k úpravě vstupních dat.
- **Grafická stylizace popisků:** Nad rámec základního nastavení písma nástroj podporuje další grafické efekty pro zvýšení čitelnosti na různých typech podkladu. Mezi hlavní prvky patří **halo efekt** (obrys textu), u kterého lze nastavit barvu a šířku. Při zapnutí gradientu halo je možné zadat počáteční, středovou a koncovou barvu a případně i odlišnou koncovou šířku. Pro výpočet hodnot mezi třídami je použita rovněž lineární interpolace. Dále je implementována podpora **stínu textu** (shadow) s nastavitelnou barvou a posunem v osách X a Y. Pro typografické doladění slouží parametr **letter spacing** (mezera mezi znaky) v procentech. Mezi další dostupné možnosti patří **posun popisků** (offset X a Y v bodech), jednotné podtržení textu a volitelná **prevence zalamování** popisků na více řádků (prevent wrapping). Všechny tyto parametry se aplikují na úrovni CIM (Cartographic Information Model) jednotlivých tříd popisků (label classes) a nemění vstupní data zvolené vrstvy.

Pro rozvržení uživatelského rozhraní byl vytvořen koncepční návrh struktury finálního toolboxu v softwaru XMind (obrázek 8):



Obr. 8 Koncepční návrh struktury a parametrů vyvíjeného toolboxu

## Struktura .atbx toolboxů

V prostředí ArcGIS Pro jsou geoprocessingové nástroje organizovány do toolboxů, které jsou uloženy ve formátu **.atbx** (ArcGIS Toolbox). Tento formát přináší moderní a standardizovaný způsob správy geoprocessingových nástrojů s přímou integrací s Pythonem a prostředím ArcGIS Pro. **Script tool** uložený v .atbx souboru funguje jako rozhraní mezi uživatelem a skriptem, přičemž jeho struktura je definována několika vzájemně provázanými částmi.

Při otevření dialogového okna vlastností **script toolu** v ArcGIS Pro se zobrazují tyto hlavní záložky: **General**, **Parameters**, **Execution**, **Validation** a **Environments**. Každá z těchto záložek má specifickou úlohu ve fungování nástroje a její obsah je uložen v rámci struktury .atbx toolboxu:

- **General** slouží k definici základních identifikačních a popisných informací nástroje. Obsahuje interní název nástroje používaný systémem, uživatelsky čitelný popis zobrazovaný v rozhraní ArcGIS Pro a textové informace určené pro nápovědu a dokumentaci. Tyto údaje nemají přímý vliv na výpočetní část nástroje, ale jsou důležité pro jeho srozumitelnou prezentaci a orientaci uživatele v rámci toolboxu.
- **Parameters** ve kterém jsou definovány vstupní a výstupní parametry nástroje. Parametry určují, jaká data nebo hodnoty nástroj očekává, jakého jsou datového typu, zda jsou povinné nebo volitelné a jakým způsobem jsou předávány skriptu. Každý parametr je zároveň svázán s konkrétním ovládacím prvkem v uživatelském rozhraní, což zajišťuje jednotný způsob zadávání vstupů a jejich základní kontrolu ještě před spuštěním nástroje (obrázek 9).

Label	Name	Data Type	Type	Direction	Description	Category	Filter	Dependency	Default	Environment	Symbology
*		String	Required	Input							

Obr. 9 Záložka Parameters formátu Script toolu v .atbx toolboxu

Při návrhu uživatelského rozhraní v prostředí ArcGIS Pro je nezbytné definovat parametry tak, aby tvořily robustní a logický celek. Každý parametr v toolboxu není pouze vstupním polem, ale komplexním objektem, jehož chování je řízeno skrze **vlastnosti parametrů** (Parameter Properties). Tyto vlastnosti zajišťují integritu dat dříve, než dojde k samotnému spuštění výpočetního skriptu. V rámci tvořeného nástroje byly pro parametry využity následující atributy:

- **Popisek (Label):** Název parametru, který se zobrazí uživateli přímo v panelu *Geoprocessing* po načtení uživatelského rozhraní nástroje.
- **Jméno (Name):** Unikátní identifikátor parametru používaný uvnitř Python skriptu. Zatímco *Label* může obsahovat mezery a speciální znaky, **Name** musí splňovat pravidla pro názvy proměnných (bez mezer, začínající písmenem), což zajišťuje bezchybné propojení rozhraní s kódem skrze metodu `arcpy.GetParameterAsText`.
- **Datový typ (Data Type):** Klíčová vlastnost definující povahu vstupu (např. *Feature Layer*, *Field*, *String* či *Double*). Správná volba datového typu v ArcGIS Pro automaticky vynucuje použití odpovídajících prvků.
- **Typ povinnosti (Type):** Rozlišení parametrů na povinné (*Required*), které jsou nezbytné pro základní chod algoritmu, volitelné (*Optional*), které slouží k pokročilemu ladění estetiky výstupu a Odvozené (*Derived*), které se v rozhraní uživateli nezobrazují. Slouží k výstupu dat, která vznikla až během běhu skriptu (např. aktualizovaná vrstva).
- **Směr (Direction):** Většina parametrů je vstupních (*Input*), avšak koncepce počítá i s výstupními parametry (*Output*), které slouží k předávání informací o výsledku procesu zpět do systému nebo uživateli.
- **Závislost (Dependency):** Tato vlastnost umožňuje logické řetězení parametrů. V praxi to znamená, že například výběr atributu je přímo závislý na vybrané vstupní vrstvě. Bez této vazby by nebylo možné zajistit dynamické filtrování obsahu rozbalovacích nabídek.

- **Kategorie (Category):** Vzhledem k vysokému počtu parametrů je nezbytné jejich vizuální seskupení do tematických bloků. Tento přístup zvyšuje přehlednost a vede uživatele procesem nastavení v logických krocích.
- **Filtr (Filter):** Slouží k dalšímu omezení povolených hodnot. Koncepce využívá filtry například pro omezení výběru polí pouze na číselné typy nebo pro definici pevného seznamu klasifikačních metod (*Value List*).
- **Výchozí hodnota (Default):** Definice počátečních hodnot je klíčová pro uživatelskou přívětivost. Umožňuje, aby byl nástroj po otevření přednastaven na ukázkové hodnoty (např. pět klasifikačních tříd, font Calibri a černou barvu písma), což uživateli dovoluje nástroj okamžitě otestovat bez nutnosti manuálního vyplňování všech parametrů.

Správná konfigurace těchto vlastností přímo v definici toolboxu tvoří první úroveň kontroly chyb (tzv. statickou validaci). Tím je zajištěno, že skript v exekeční fázi obdrží data v očekávaném formátu a rozsahu.

- **Execution** určuje, jaký zdrojový kód je s nástrojem spojen a jakým způsobem je při spuštění vykonán. V případě Python script toolu je zde definován Python skript, který může být buď uložen přímo v rámci toolboxu (**embedded**), nebo na něj může být odkazováno jako na externí soubor. ArcGIS Pro při spuštění nástroje automaticky předá skriptu hodnoty zadaných parametrů a zajistí jeho provedení v rámci geoprocessingového prostředí. Tato část tedy reprezentuje vlastní implementaci algoritmického řešení úlohy.
- **Validation** umožňuje definovat validační kód v jazyce Python, který je prováděn v průběhu nastavování parametrů, ještě před samotným spuštěním výpočtu. Prostřednictvím validační třídy lze kontrolovat správnost zadaných hodnot, upravovat výchozí nastavení parametrů nebo řídit jejich dostupnost na základě jiných vstupů.
- **Environments** souvisí s podporou geoprocessingových prostředí. Tato sekce určuje, jaká environmentální nastavení mohou ovlivnit běh nástroje, například pracovní prostor, výstupní souřadnicový systém nebo další globální vlastnosti zpracování. Omezením environmentů pouze na ty relevantní lze zajistit předvídatelné chování nástroje v různých projektech a datových sadách, aniž by bylo nutné tyto vlastnosti explicitně řešit v samotném skriptu.

Celkově struktura **.atbx** toolboxů poskytuje jasně definovaný rámec pro tvorbu toolboxů, který odděluje popisy nástrojů, definici parametrů, validační chování a vlastní výpočetní implementaci. Umožňuje vytvářet nástroje, které jsou srozumitelné pro uživatele, konzistentní v chování a relativně snadno udržovatelné z hlediska dalšího vývoje. Zároveň je však nutné upozornit, že formát **.atbx** má omezení na možnosti grafické úpravy uživatelského rozhraní nástroje. Vzhled toolboxu je do značné míry pevně daný prostředím ArcGIS Pro a vývojář má omezenou možnost grafické úpravy uživatelského rozhraní, u kterého může ovlivňovat pouze pořadí, seskupení a chování parametrů, nikoli však detailní rozvržení prvků nebo jejich vizuální stylizaci. Uživatelské rozhraní je tedy funkční a konzistentní, avšak méně flexibilní z hlediska individuálního grafického návrhu, což značně omezuje právě možnosti lepší uživatelské přívětivosti, která by nabízela lepší vizuální atraktivitu rozhraní v kombinaci s jeho srozumitelností.

## 4.4 Návrh klasifikačních metod

Jak již bylo vysvětleno v kapitole 4.2, labeling v prostředí ArcGIS Pro je postaven na principu tvorby tzv. **tříd popisků (Label Classes)**. Každá tato třída představuje samostatnou vrstvu informací s vlastními pravidly pro vzhled a umístění. V kontextu tvorby číselných kartodiagramů je však zásadním problémem absence nativní funkce, která by tyto třídy automaticky generovala na základě statistického rozdělení dat. Vlastní řešení proto tento proces plně automatizuje. Zatímco standardní postup v ArcGIS Pro vyžaduje, aby uživatel pro každou kategorii ručně vytvořil novou třídu, definoval její název a vložil SQL dotaz, vyvinutý skript tyto kroky provádí v jediném výpočetním cyklu. Návrh řešení vychází z potřeby zajistit, aby každá vytvořená Label Class byla vzájemně exkluzivní. Toho je dosaženo **dynamickým generováním SQL dotazů** na pozadí nástroje. Skript po výpočtu klasifikačních intervalů (např. pomocí metody Jenks Natural Breaks) sestaví pro každou třídu unikátní podmínku, která přesně vymezuje její působnost v rámci atributového pole. Tím je eliminováno riziko duplicitního zobrazení popisků u jednoho prvku. Návrh klasifikace v nástroji nepracuje pouze s prostým rozdělením, ale implementuje algoritmy, které jsou uživateli známé při standardní práci se symbolikou vrstev v prostředí ArcGIS. Ty reflektují distribuci dat prostřednictvím následujících matematických principů:

### Metoda stejných intervalů (Equal Interval):

Celý rozsah hodnot od nejmenší po největší se rovnoměrně rozdělí na zvolený počet tříd, každá třída má stejnou šířku. Pokud jsou data od 0 do 100 a uživatel chce 5 tříd, vzniknou třídy 0–20, 20–40, 40–60, 60–80 a 80–100 (Esri, 2026a).

Šířka jedné třídy:

$$d = \frac{x_{max} - x_{min}}{n}$$

Hranice tříd se pak počítají jako  $x_{min} + d$ ,  $x_{min} + 2d$ ,  $x_{min} + 3d$ , atd. Kde:  $x_{max}$  = nejvyšší hodnota v datech,  $x_{min}$  = nejnižší hodnota v datech,  $n$  = počet tříd zvolený uživatelem,  $d$  = šířka jedné třídy.

### Metoda kvantilů (Quantile):

Data se seřadí od nejmenší po největší a rozdělí se tak, aby každá třída obsahovala přibližně stejný počet prvků. Pokud je 100 prvků a 5 tříd, každá třída bude obsahovat přibližně 20 prvků – bez ohledu na to, jak jsou hodnoty rozmístěny na číselné ose (Esri, 2026b; Kladivo, 2013).

Protože metoda počítá s pořadím prvků, ne s hodnotami, mohou do stejné třídy spadnout i výrazně odlišná čísla, pokud jsou blízko sebe v pořadí.

Pořadové číslo (index) hodnoty, která tvoří hranici  $i$ -té třídy v seřazených datech:

$$Index_i = \left\lfloor \frac{i \cdot N}{n} \right\rfloor$$

Kde:  $N$  = celkový počet prvků v datech,  $n$  = počet tříd zvolený uživatelem,  $i$  = číslo třídy (1, 2, 3, ...),  $\lfloor \rfloor$  = zaokrouhlení dolů na celé číslo.

V implementaci nástroje jsou navíc odstraněny duplicitní hranice, které by jinak způsobily vznik prázdných tříd v případě, že se v datech opakují stejné hodnoty.

## Geometrický interval (Geometric Interval)

Třídy zde nemají stejnou šířku ani stejný počet prvků – každá následující třída je oproti předchozí přibližně stejněkrát větší. Metoda je vhodná pro data, kde se hodnoty pohybují v řádu tisíců nebo milionů a kde by lineární rozdělení nedávalo smysl (např. počty obyvatel měst, výše příjmů apod.).

Nejprve se vypočte koeficient násobení, který říká, kolikrát se šířka třídy s každým krokem zvětší. Hranice tříd jsou pak řadou hodnot, kde každá je předchozí vynásobena tímto koeficientem (Esri, 2026a; Artmann, 1999).

Koeficient násobení:

$$r = \left( \frac{x_{max}}{x_{min}} \right)^{1/n}$$

Hranice tříd:  $x_{min}$ ,  $x_{min} \cdot r$ ,  $x_{min} \cdot r^2$ ,  $x_{min} \cdot r^3$ , atd. Kde  $x_{max}$  je nejvyšší hodnota v datech,  $x_{min}$  nejnižší hodnota v datech,  $n$  počet tříd a  $r$  koeficient geometrického násobení.

Protože výpočet koeficientu  $r$  vyžaduje dělení hodnotou  $x_{min}$ , je v kódu nulová nebo záporná minimální hodnota automaticky nahrazena konstantou 0,0001, aby výpočet nezkolaboval.

## Přirozené zlomy – Jenks (Natural Breaks)

Metoda automaticky hledá v datech přirozené mezery a seskupuje hodnoty, které jsou si navzájem podobné. Výsledné třídy mají co nejmenší vnitřní různorodost a co největší vzájemné rozdíly. Oproti metodě stejných intervalů tedy netvoří přesně symetrické třídy, ale přizpůsobuje jejich rozmístění skutečné struktuře dat.

Algoritmus funguje tak, že systematicky zkouší různá umístění hranic a pro každé porovnává, jak moc se hodnoty v každé třídě liší od průměru té třídy. Vybere se to rozdělení, kde je celková vnitřní různorodost nejmenší (Esri, 2026a; Jenks, 1967)).

Míra vnitřní různorodosti jedné třídy (součet čtvercových odchylek od průměru třídy):

$$V(j, i) = \sum_{t=j}^{i-1} x_t^2 - \frac{(\sum_{t=j}^{i-1} x_t)^2}{i-j}$$

Optimální rozdělení do  $n$  tříd se hledá tak, aby byl součet hodnot  $V$  přes všechny třídy co nejmenší, tedy aby byly třídy uvnitř co nejvíce homogenní. Kde  $x_t$  je konkrétní hodnota v daném úseku,  $j$  je index začátku úseku,  $i$  index konce úseku a  $V(j,i)$  je míra vnitřní různorodosti (součet čtvercových odchylek) jednoho úseku. Symbol  $t$  je pouze pomocný „běžící“ index ve sčítání, který postupně prochází prvky od  $j$  do  $i-1$ .

V implementaci nástroje je použita varianta s dynamickým programováním, která je výpočetně efektivnější. Pro soubory s více než 3 000 prvky se navíc pracuje s náhodným vzorkem 3 000 hodnot, přičemž skutečné minimum a maximum celého souboru jsou do výsledků vždy vráceny zpět.

## Proporcionální – bez tříd (Proportional / Unclassed)

Tato metoda nevytváří sdružené třídy. Každá unikátní hodnota v atributovém poli dostane vlastní kategorii. Vizualní proměnná (velikost písma, barva) se pak plynule mění podle toho, jakou hodnotu daný prvek má – podobně jako u spojité stupnice.

Hranice mezi sousedními hodnotami se stanovují jako jejich aritmetický průměr, takže každá hodnota leží uprostřed svého vlastního intervalu.

Hranice mezi dvěma sousedními unikátními hodnotami  $u_i$  a  $u_{i+1}$ :

$$b_i = \frac{u_i + u_{i+1}}{2}$$

Kde:  $u_i$  je  $i$ -tá unikátní hodnota v seřazeném seznamu,  $b_i$  hranice mezi  $i$ -tou a  $(i + 1)$ -tou kategorií a  $n_u$  celkový počet unikátních hodnot.

Protože každý prvek dostane vlastní třídu, je tato metoda kartograficky vhodná pouze pro datové soubory s malým počtem prvků. V kódu je proto implementováno upozornění pro soubory s více než 50 prvky.

### Manuální klasifikace (Manual)

Uživatel sám zadá hodnoty, kde mají být hranice tříd. Tato možnost je určena pro případy, kdy jsou třídy dány odborným požadavkem, normou nebo konkrétním analytickým záměrem, který statistické metody nemohou zohlednit.

Skript vstup automaticky upraví: dovoluje oddělovače v podobě mezer nebo středníků, převede desetinné čárky na tečky a zadané hodnoty seřadí. Poté doplní skutečné minimum a maximum celého datového souboru, takže výsledné třídy vždy pokryjí celý rozsah hodnot i tehdy, když uživatel krajní hodnoty nezadal (Esri, 2026a).

Výsledné hranice tříd mají tvar:

$$[x_{min}, u_1, u_2 \dots u_m, x_{max}]$$

Kde  $x_{min}$  a  $x_{max}$  jsou skutečné extrémy celého datového souboru a  $u_1, u_2, \dots, u_m$  jsou hodnoty hranic zadané uživatelem (seřazené vzestupně).

## 4.5 Algoritmus gradací a barevných přechodů

Tato kapitola popisuje řešení transformace statistických dat do vizuální podoby. Hlavním úkolem implementované logiky v toolboxu je zajistit plynulý přechod (gradient) mezi parametry první a poslední kategorie. Tento proces je v kódu toolboxu řešen pomocí obecného algoritmu **lineární interpolace (LERP)**, který je však specificky upraven pro různé gradace: tloušťky a velikosti písma, barvy a halo efektu (Burden a Faires, 2011).

### Gradace velikosti písma jako primární vizuální proměnná

Vzhledem k tomu, že je tato metoda určena pro tvorbu gradovaných alfanumerických znaků, představuje automatizovaná změna velikosti písma (*font size*) nejdůležitější prvek celého algoritmu. Zatímco v běžném prostředí ArcGIS Pro musí uživatel pro každou kategorii ručně definovat velikost, toolbox tento proces automatizuje na základě uživatelského nastavení minimální a maximální hodnoty velikosti písma.

Matematický výpočet velikosti pro každou  $i$ -tou kategorii probíhá lineárně:

$$V_i = V_{min} + (i - 1) \cdot \frac{V_{max} - V_{min}}{n - 1}$$

Kde  $n$  je počet kategorií,  $i$  pořadí kategorie,  $V_i$  velikost kategorie,  $V_{min}$  minimální velikost písma a  $V_{max}$  maximální velikost písma.

Algoritmus zajišťuje, že rozdíl ve velikosti mezi sousedními kategoriemi je konstantní, což usnadňuje uživateli odhadnout hodnotu prvků na mapě bez neustálého nahlížení do legendy (Burden a Faires, 2011).

## Algoritmus distribuce statických fontových řezů (Font Styles)

V případech, kdy zvolená rodina písma nevyužívá technologii variabilních os, je gradace popisků řešena skrze distribuci diskrétních řezů (např. *Thin, Light, Regular, Bold, Black*). Hlavní výzvou tohoto přístupu je nelinearita a omezený počet dostupných stylů v porovnání s počtem klasifikačních kategorií.

- **Logika mapování:** Návrh řešení počítá s vytvořením seřazeného pole dostupných stylů, které jsou do systému registrovány výrobcem písma.
- **Distribuce na klasifikační kategorie:** Aby byla zajištěna rovnoměrná distribuce, využívá algoritmus přepočtení indexu řezu pro každou  $i$ -tou kategorii z celkového počtu tříd. Výběr konkrétního stylu probíhá na základě normalizovaného indexu:

$$\text{Index} = \text{zaokrouhl.} \left( (i - 1) \cdot \frac{\text{počet řezů} - 1}{n - 1} \right)$$

Kde  $n$  je celkový počet kategorií a  $i$  pořadí aktuální kategorie.

## Gradace váhy u variabilních fontů (Font Weight)

Vedle absolutní velikosti písma nástroj umožňuje pracovat s **tloušťkou (weight)** jako s doplňkovou vizuální proměnnou. Ta na rozdíl od statických řezů umožňuje plynulou změnu tloušťky v libovolném kroku, čehož nástroj využívá pro vytvoření naprosto hladkého vizuálního gradientu.

- **Analýza binárních dat:** Skript využívá integrovanou knihovnu **fontTools** k přímému čtení binárních tabulek písma (konkrétně tabulky **fvar** – Font Variations), ze které extrahuje přesný rozsah osy wght. Tento rozsah je typicky definován standardem OpenType v rozmezí 100 až 950, ale může se u různých fontů lišit.
- **Kontinuální interpolace:** Na rozdíl od diskrétního výběru u statických řezů zde skript aplikuje přímou lineární interpolaci hodnoty tloušťky. Každá třída popisků tak získává unikátní číselnou hodnotu osy wght, která přesně odpovídá její pozici v klasifikaci dat.

## Interpolace barevných gradientů (RGB vs. CMYK)

Návrh barevných přechodů v rámci toolboxu plní funkci sekundární vizuální proměnné, která doplňuje velikost písma a posiluje čitelnost kvantitativních dat. Algoritmus provádí výpočet barevných složek individuálně pro každou vytvořenou třídu popisků, přičemž matematický přístup k interpolaci se liší podle zvoleného barevného modelu a počtu definovaných bodů gradientu. Tento mechanismus není omezen pouze na gradaci barvy samotného textu, ale je v nástroji využíván i pro barevnou gradaci „halo efektu“.

## Interpolace v aditivním modelu (RGB/HEX)

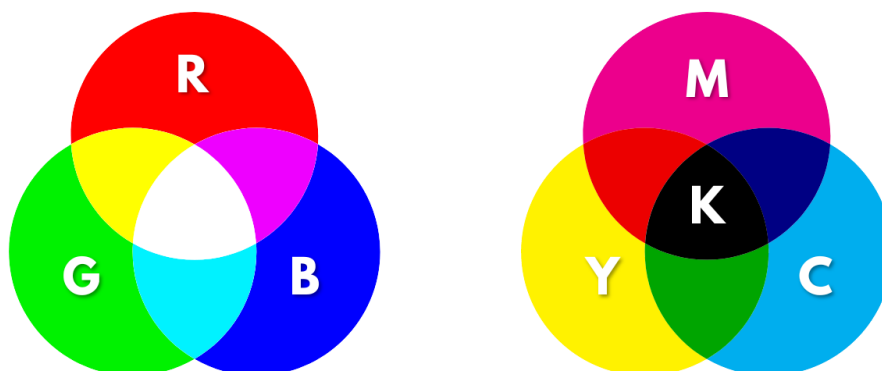
Režim RGB je v nástroji určen primárně pro digitální prezentaci dat. Algoritmus interpretuje barvu jako bod v třírozměrném euklidovském prostoru s osami **R** (červená), **G** (zelená) a **B** (modrá) (obrázek 10).

- **Lineární přepočtení složek:** Pro každou třídu je vypočítána hodnota každého barevného kanálu samostatně. Výpočet probíhá na základě lineárního nárůstu nebo poklesu intenzity daného kanálu mezi startovní a koncovou barvou, včetně průhlednosti **A** (alpha).
- **Zpracování vstupů:** Nástroj je navržen tak, aby dokázal pracovat s různými formáty zápisu (např. HEX kódy, RGBA), které vnitřně transformuje na normalizované číselné hodnoty v rozsahu **0–255** pro potřeby výpočtu kroků gradientu.

## Přímá interpolace v subtraktivním modelu (CMYK)

Pro účely tisku je implementována přímá interpolace v modelu CMYK. Tento přístup řeší častý problém "šednutí" nebo "špinění" barev v barevných přechodech (obrázek 10).

- **Zachování barevné věrnosti:** Pokud by se barvy CMYK během výpočtu konvertovaly do prostoru RGB a teprve poté interpolovaly, došlo by k nelineárnímu zkreslení barevných složek. Nástroj proto počítá změny přímo pro kanály **C** (azurová), **M** (purpurová), **Y** (žlutá) a **K** (černá).
- **Matematický postup:** Složky jsou interpolovány jako procentuální podíly (0–100 %). Výsledkem je exaktní matematický přechod, který zaručuje, že středové kategorie klasifikace budou mít čisté odstíny bez nežádoucích barevných posunů, které by vznikly nevhodnou konverzí mezi barevnými prostory.



Obr. 10 Princip míchání barev v aditivním modelu RGB a subtraktivním modelu CMYK  
(zdroj: [www.trbranddesign.com/blog/rbg-vs-cmyk](http://www.trbranddesign.com/blog/rbg-vs-cmyk))

## Tříbodový gradient (Mid-color)

Pro potřeby komplexnější barevné vizualizace (např. u tří barevného gradientu (typicky červená, žlutá, zelená), nebo u divergentních stupnic ukazujících odchylku od průměru) umožňuje nástroj definovat středový bod barevného přechodu.

- **Segmentace výpočtu:** V tomto režimu není gradient počítán jako jedna souvislá řada, ale je rozdělen na dva samostatné úseky: **Min – Mid** a **Mid – Max**. Pro každý segment je vypočítán vlastní krok změny barevných složek.
- **Logika rozdělení kategorií:** Algoritmus vnitřně analyzuje počet vytvořených klasifikačních tříd a středovou barvu přesně zacílí na prostřední kategorii. Zbývající barvy jsou pak rozděleny mezi spodní a horní polovinu intervalů. Tím je zajištěno, že barevný předěl ve znacích vždy odpovídá středu klasifikovaných hodnot.

## 5 TVORBA TOOLBOXU V ARCGIS PRO

Samotný proces vývoje a technická architektura nástroje **Graduated Numbers Generator** představují stěžejní část praktické realizace. Vývoj probíhal v několika fázích, od vytvoření .atbx toolboxu, vyplnění základních informací, definice parametrů a jejich logického rozřazení, přes samotné vytvoření hlavního funkčního spouštěcího (**Execution**) a validačního (**Validation**) skriptu, až po vyplnění nápověd a metadat.


Vzhledem k tomu, že vývoj probíhal v prostředí **ArcGIS Pro** (verze 3.4 a 3.6) v anglické lokalizaci (US) a v souladu s koncepčním návrhem (viz. kapitola 4.3), bude v následujícím textu používána anglická terminologie pro částí ArcGIS Pro, názvy parametrů a vnitřních objektů softwaru, aby byla zachována kontinuita s technickou realizací nástroje.

### 5.1 Vytvoření .atbx toolboxu

Prvním krokem při vývoji bylo rozhodnutí o formátu pro vyvíjený nástroj. Pro potřeby této práce byl zvolen formát **ArcGIS Toolbox (.atbx)**. Vytvoření samotného toolboxu předchází otevření si „Katalogového okna“ (Catalog Pane/ Catalog View) V tzv. Katalogu na stránce *Project* se krom ostatních nachází záložka *Toolboxes*, u které se po rozkliknutí objeví možnost vytvoření nového **.atbx Toolboxu** (obrázek 11).



Obr. 11 Založení nového .atbx toolboxu prostřednictvím kontextové nabídky v panelu Catalog

Po zadání názvu .atbx toolboxu se vytvoří nový soubor, u kterého je možnost vytvořit nový  **Script**. V tomto „Skriptu“ je potřeba vyplnit název, který je nutno vyplnit bez mezer a speciálních znaků. **Label** (popisek), který bude viditelný v rozhraní ArcGIS Pro je možno již napsat bez omezení. Po potvrzení se vytvoří Script, který je možné následně upravovat skrze kontextovou nabídku toolboxu po zvolení položky **Properties**.

### 5.2 Architektura toolboxu

Navržený nástroj je implementován jako tzv. **ScriptTool** integrovaný do proprietárního formátu ArcGIS Pro toolboxu (.atbx). Tento formát, zavedený v ArcGIS Pro 2.9, umožňuje ukládat skripty a jejich metadata v jediném binárním souboru, čímž odpadá nezbytnost distribuce externích .py souborů vedle toolboxu. Výsledný soubor je proto plně přenositelný – postačí zkopírovat jeden soubor **.atbx** a nástroj je okamžitě dostupný v prostředí ArcGIS Pro bez dalšího nastavení.

Celá logika nástroje je rozdělena do dvou částí, přičemž obě jsou embedovány přímo v toolboxu:

- **Validation** – Validační část: inicializace parametrů, dynamické filtry, validace vstupů
- **Execution** – Výkonná část: načítání parametrů, klasifikace, tvorba tříd popisků, aplikace CIM stylů

Soubor **Execution** je spuštěn teprve po kliknutí na tlačítko **Run**, jeho vstupním bodem je funkce *main*.

## Modulární závislosti

**Execution** závisí na následujících Python modulech a knihovnách:

```
import os
import random
import re
import statistics
import traceback
import winreg
from decimal import Decimal, ROUND_HALF_UP

import arcpy
from fontTools.ttLib import TTFont
```

**Validation** sdílí část těchto závislostí:

```
import arcpy
import os
import re
import winreg
import statistics
from fontTools.ttLib import TTFont
```

Klíčové jsou zejména dvě nestandardní závislosti:

- **arcpy** – balíček Esri dostupný výhradně v prostředí ArcGIS Pro; poskytuje přístup k vrstvám mapy, CIM API (Cartographic Information Model), Geoprocessing zprávám a správě parametrů toolboxu.
- **fontTools.ttLib.TTFont** – open-source knihovna pro čtení binárního obsahu fontových souborů (.ttf, .otf, .ttc). Umožňuje přístup k tabulkám name, fvar a OS/2, které jsou nezbytné pro detekci variable fontů a čtení os jako wght nebo ital.

Moduly **os** a **winreg** slouží ke čtení Windows Registry a prohledávání systémových adresářů s fonty (Fonts, %LOCALAPPDATA%\Microsoft\Windows\Fonts). Modul **decimal** s konstantou **ROUND\_HALF\_UP** zajišťuje přesné zaokrouhlování zlomkových hranic tříd. Modul **random** je využíván při optimalizaci algoritmu klasifikační metody přirozených zlomů (Jenks), viz. kapitola 5.5.

## Globální pomocné funkce v Execution

Část **Execution** obsahuje sadu globálních (modulárních) pomocných funkcí, které jsou volány jak z *main*, tak jsou logicky samostatné jednotky zodpovědné za jednu konkrétní operaci. Přehled těchto funkcí uvádí tabulka 2:

Tabulka 2 Přehled globálních pomocných funkcí implementovaných v části Execution

Funkce	Popis
<b>calculate_breaks(values, method, num_categories)</b>	Výpočet hranic tříd podle zvolené klasifikační metody
<b>natural_breaks(values, num_categories)</b>	Implementace Jenksova algoritmu přirozeného dělení
<b>cmyk_to_rgb(c, m, y, k)</b>	Převod CMYK (0–1) na RGB (0–255)
<b>parse_color_as_cmyk(color_text)</b>	Parsování textového vstupu jako CMYK hodnot
<b>detect_color_format(color_text)</b>	Identifikace formátu barevného řetězce
<b>parse_color(color_text, mode=None)</b>	Univerzální parsování barvy do formátu RGBA

<b>parse_alpha(a_raw)</b>	Normalizace hodnoty alfa kanálu na rozsah 0–255
<b>parse_cmyk_value(p)</b>	Parsování jedné CMYK složky (číslo nebo %)
<b>parse_hsl_angle(p)</b>	Parsování hodnoty odstínu v HSL
<b>parse_hsl_component(p)</b>	Parsování sytosti/světlosti v HSL
<b>looks_like_hsl(parts)</b>	Heuristická detekce HSL formátu ze složek
<b>hsl_to_rgb(h, s, l)</b>	Převod HSL na RGB (0–255)
<b>interpolate_colors(start_rgba, end_rgba, n)</b>	Lineární interpolace n barev (start → end)
<b>interpolate_colors_with_mid(start_rgba, mid_rgba, end_rgba, n)</b>	Interpolace n barev přes střed (start → mid → end)
<b>interpolate_colors_cmyk(start_cmyk, end_cmyk, n)</b>	Interpolace v CMYK prostoru, výstup jako RGB
<b>interpolate_colors_with_mid_cmyk(start_cmyk, mid_cmyk, end_cmyk, n)</b>	Interpolace v CMYK prostoru přes střed
<b>get_parsed_color(color_text, error_prefix)</b>	Obal parse_color s ArcGIS chybovou hláškou
<b>get_parsed_color_cmyk(color_text, error_prefix)</b>	Obal parse_color_as_cmyk s ArcGIS chybovou hláškou
<b>create_halo_symbol(color_rgba, width)</b>	Sestavení CIM objektu CIMPolygonSymbol pro halo text
<b>create_default_separators</b>	Vytvoření výchozích Maplex oddělovačů zalamování (CIMMaplexStackingSeparator)

Oddělení těchto funkcí na úroveň modulu (mimo *main*) má důsledek dvojitý: jednak usnadňuje testování jednotlivých logických celků bez nutnosti spouštět celý toolbox, jednak zpřehledňuje samotnou funkci *main*, která tak sdružuje pouze orchestraci celkového toku a volá tyto stavební bloky v definovaném pořadí.

Naproti tomu funkce silně závislé na konkrétním kontextu běhu (přístup k parametrům, registru nebo CIM symbolu) jsou definovány jako vnořené funkce (*nested functions*) uvnitř *main*, jak uvádí tabulka 3:

Tabulka 3 Přehled vnořených funkcí a jejich účel

Vnořená funkce	Účel
<b>_get_param_text(index, name)</b>	Načtení hodnoty parametru toolboxu (primárně dle jména, záložně dle indexu)
<b>_as_bool(raw_value, default)</b>	Konverze ArcGIS parametru na Python bool
<b>_as_float(raw_value, default)</b>	Bezpečná konverze na float
<b>_as_int(raw_value, default)</b>	Bezpečná konverze na int
<b>_get_registry_variable_family(family)</b>	Vyhledání registrového názvu variable fontu (např. "Oswald" → "Oswald variable")
<b>_apply_variable_axes_to_symbol(sym, axes)</b>	Aplikace hodnot os variable fontu (např. wght, ital) do CIM vlastnosti fontVariationSettings
<b>_normalize_style_name(raw_style)</b>	Normalizace názvu řezu fontu do ArcGIS Pro kompatibilního formátu
<b>_iter_font_file_paths</b>	Iterace přes unikátní cesty k fontovým souborům
<b>get_true_font_styles(fam)</b>	Detekce skutečně dostupných řezů pro danou rodinu fontu
<b>get_font_weight_range(font_fam)</b>	Čtení rozsahu osy wght z tabulky fvar variable fontu

Tato architektura je výsledkem pragmatické volby: vnořené funkce mohou přirozeně přistupovat k proměnným z obklopující funkce *main*, aniž by bylo nutné předávat velké množství argumentů. Zároveň jsou vizuálně odděleny od hlavní logiky komentáři a uspořádány na začátku funkce před prvním výpočetním krokem.

## Logika validace v třídě ToolValidator

Část **Validation** definuje třídu **ToolValidator**, jejíž životní cyklus v prostředí ArcGIS Pro lze popsat čtyřmi metodami:

- Metoda `__init__` proběhne při každém otevření dialogu nástroje. Volá `arcpy.GetParameterInfo`, čímž získá seznam všech **36 parametrů** toolboxu a inicializuje dvě proměnné sloužící jako mezipaměť (`_font_registry_names`, `_font_catalog`). Tato mezipaměť je klíčová z výkonnostních důvodů: čtení Windows Registry a parsování binárních fontových souborů pomocí **fontTools** je operace relativně náročná, proto se provádí pouze jednou za session dialogu a výsledek se ukládá do instance.
- `initializeParameters` je zavolána ihned po `__init__` a nastavuje výchozí hodnoty, popisky (`displayName`, `description`), filtry (`filter.list`) a povolení (`enabled`) všech parametrů. Zároveň naplní seznam rodin fontů (parametr 10) dotazem do `_load_font_families`.
- `updateParameters` se volá po každé změně hodnoty libovolného parametru v dialogu. Řeší dynamické závislosti, například povolení nebo zakázání parametrů `Font Weight (min)` a `Font Weight (max)` v závislosti na výběru "Gradient Font Weight ..." v parametru `Font Weight/Style`, nebo výpočet statistik (`Total Element Count`, `Value Range`, `Average | Median`) při změně vstupní vrstvy nebo atributového pole.
- `updateMessages` provádí finální validaci a zobrazuje uživateli varovné nebo chybové zprávy pomocí `setErrorMassage` a `setWarningMessage`. Typicky zahrnuje kontrolu formátu barev, rozsahů hodnot fontů a limitu počtu prvků.

## Workflow funkce main

Funkce `main` implementuje lineární, sekvenční pracovní tok, jehož hlavní kroky jsou popsány v dokumentačním řetězci samotné funkce:

```
def main:
```

```
    Workflow:
```

1. Load and validate all parameters from the toolbox GUI.
2. Extract attribute values from input layer.
3. Generate class breaks using the chosen classification method.
4. Scale breaks based on user-selected value scale factor.
5. Create one label class per category with Arcade expression filters.
6. Interpolate font weights/styles and colors across categories.
7. Apply CIM styling (text symbol, halo, shadow, offset, etc.).
8. Configure Maplex label placement properties (wrapping, stacking, etc.).
9. Enable labels on the layer and report success.

Načítání parametrů probíhá v pořadí indexů 0–35, přičemž v komentářích uvnitř kódu je každý blok parametrů jasně označen:

```
# 0-4: Input and preview outputs
input_layer = _get_param_text(0, 'input_layer')
attribute_field = _get_param_text(1, 'attribute_field')
# 2 = total_element_count (output)
# 3 = value_range_info (output)
# 4 = average_and_median (output)
# 5-7: Classification
classification_method = _get_param_text(5, 'classification_method')
number_of_categories = _get_param_text(6, 'number_of_categories')
custom_breaks_text = _get_param_text(7, 'custom_breaks')
...
```

Celý výkonný kód *main* je obalený jedním blokem *try/except*, který zachytí jakoukoliv výjimku a zobrazí ji uživateli ve formě chybového hlášení v Geoprocessing okně ArcGIS Pro:

```
except Exception as exc: # pragma: no cover
    arcpy.AddError(f"An error occurred: {str(exc)}\n{traceback.format_exc}")
```

Použití *traceback.format\_exc* zajišťuje, že výstup obsahuje celý zásobník volání (stack trace), což výrazně usnadňuje diagnostiku problémů přímo v prostředí ArcGIS Pro.

## 5.3 Popis všech parametrů nástroje

Parametry toolboxu tvoří primární rozhraní mezi uživatelem a výkonným skriptem. Jejich přesná definice, datový typ, výchozí hodnota a vzájemné závislosti jsou deklarovány v metodě *initializeParameters* třídy *ToolValidator* a jsou čteny v *main* funkcemi *\_get\_param\_text*, *\_as\_bool*, *\_as\_float* a *\_as\_int*.

### Přehled a kategorizace parametrů nástroje

Nástroj disponuje **36 parametry** (obrázek 12). První dva (**Layer Selection** a **Attribute Selection**) jsou záměrně ponechány bez kategorie, protože představují základní vstupní volby.

Ostatní parametry jsou rozděleny do následujících pěti kategorií:

- **Data Statistics**
  - Parametry pro zobrazení základních statistických údajů o datech (např. celkový počet prvků, rozsah hodnot, průměr a medián).
  - Slouží pouze k informování uživatele, jsou automaticky vyplňovány a nelze je editovat.
- **Classification & Size**
  - Parametry určující způsob klasifikace hodnot, počet kategorií, případně ruční zadání hranic tříd.
  - Zahrnuje také nastavení minimální a maximální velikosti písma.
- **Font Styling**
  - Parametry pro výběr fontu, řezu/tloušťky, případně gradace mezi dvěma řezy (gradient font weight).
  - Dále obsahuje nastavení pro barvu textu, aktivaci barevného gradientu a jemné doladění vzhledu písma (např. letter spacing).
- **Format & Units**
  - Parametry ovlivňující formátování číselných hodnot, např. oddělovač tisíců, měřítko hodnot, počet desetinných míst a další.
- **Advanced Styling**
  - Rozšířená nastavení pro pokročilé úpravy vzhledu popisků, např. podtržení, posun popisku (offset), nastavení halo efektu a stínování.

Klíčovým designovým principem je **dynamická aktivace** – některé nepovinné parametry jsou v dialogu zakázány, dokud uživatel neprovede relevantní volbu (např. zapnutí gradientu barev aktivuje parametry *Text Color (mid)* a *Text Color (end)*; výběr "*Gradient Font Weight ...*" aktivuje *Font Weight (min)* a *Font Weight (max)*). Tato logika snižuje kognitivní zátěž uživatele a zabraňuje zadání nekonzistentních kombinací. Validace formátu barev, rozsahů hodnot a dostupnosti řezů fontů probíhá v *updateMessages* ještě před spuštěním výkonného kódu.

	Label	Name	Data Type	Type	Direction	Description	Category	Filter	Dependency	Default
0	Layer Selection	input_layer	Feature Layer	Required	Input	...				
1	Attribute Selection (numeric field only)	attribute_field	Field	Required	Input	...		Field	input_layer	
2	Total Element Count	total_element_count	String	Optional	Output	...	Data Statistics			Select Layer a...
3	Value Range Preview	value_range_info	String	Optional	Output	...	Data Statistics			Select Layer a...
4	Average and Median	average_median	String	Optional	Output	...	Data Statistics			Select Layer a...
5	Classification Method	classification_method	String	Required	Input	...	Classification & Size	Value List		Natural Breaks
6	Number of Categories	number_of_categories	Long	Required	Input	...	Classification & Size	Range		5
7	Manual Class Breaks (Space or ";" Separated)	custom_breaks	String	Optional	Input	...	Classification & Size			
8	Minimum Font Size (pt)	min_size	Double	Required	Input	...	Classification & Size	Value List		8
9	Maximum Font Size (pt)	max_size	Double	Required	Input	...	Classification & Size			24
10	Font Selection	font_family	String	Optional	Input	...	Font Styling	Value List		Calibri
11	Font Weight / Style	font_weight	String	Optional	Input	...	Font Styling	Value List		
12	Min Font Weight	font_weight_min	String	Optional	Input	...	Font Styling	Value List		
13	Max Font Weight	font_weight_max	String	Optional	Input	...	Font Styling			
14	Letter Spacing (%)	letter_spacing	Long	Optional	Input	...	Font Styling			0
15	Text Colour (RGBA, CMYKA, HEX, HSL)	text_color	String	Optional	Input	...	Font Styling			0,0,0,1
16	Use Colour Gradient	use_color_gradient	Boolean	Optional	Input	...	Font Styling			false
17	Mid Value Colour (optional) (RGBA, CMYKA, HEX, HSL)	text_color_mid	String	Optional	Input	...	Font Styling			
18	Max Value Colour (RGBA, CMYKA, HEX, HSL)	text_color_end	String	Optional	Input	...	Font Styling			0,0,0,1
19	Thousand Separator	thousand_separator	String	Optional	Input	...	Format & Units	Value List		Space
20	Value Scale	value_scale	String	Optional	Input	...	Format & Units	Value List		1 (no scaling)
21	Label Suffix	value_suffix	String	Optional	Input	...	Format & Units			
22	Decimal Places After Scale	decimal_places_after_scale	Long	Optional	Input	...	Format & Units			1
23	Prevent Label Wrapping	prevent_wrapping	Boolean	Optional	Input	...	Format & Units			true
24	Label Offset X (pt)	label_offset_x	Long	Optional	Input	...	Advanced Styling			0
25	Label Offset Y (pt)	label_offset_y	Long	Optional	Input	...	Advanced Styling			0
26	Underline Text	underline_text	Boolean	Optional	Input	...	Advanced Styling			false
27	Halo Colour (RGBA, CMYKA, HEX, HSL)	halo_color	String	Optional	Input	...	Advanced Styling			255,255,255,1
28	Halo Width (pt)	halo_width	Double	Optional	Input	...	Advanced Styling			0
29	Use Halo Gradient	use_halo_gradient	Boolean	Optional	Input	...	Advanced Styling			false
30	Mid Value Halo Colour (optional) (RGBA, CMYKA, HEX, HSL)	halo_color_mid	String	Optional	Input	...	Advanced Styling			
31	Max Value Halo Colour (RGBA, CMYKA, HEX, HSL)	halo_color_end	String	Optional	Input	...	Advanced Styling			255,255,255,1
32	Max Halo Width (pt)	halo_width_end	Double	Optional	Input	...	Advanced Styling			
33	Shadow Colour (RGBA, CMYKA, HEX, HSL)	shadow_color	String	Optional	Input	...	Advanced Styling			
34	Shadow Offset X (pt)	shadow_offset_x	Double	Optional	Input	...	Advanced Styling			0
35	Shadow Offset Y (pt)	shadow_offset_y	Double	Optional	Input	...	Advanced Styling			0

Obr. 12 Výpis všech parametrů nástroje ze záložky Parameters

## Vstupní data a náhled dat

První skupina parametrů identifikuje zdrojová data a zprostředkovává uživateli okamžitý náhled základních statistik ještě před spuštěním nástroje. Parametry 2–4 jsou odvozené (*read-only*) **výstupní parametry** (Output), jejichž hodnoty jsou automaticky aktualizovány v **updateParameters** při každé změně vstupní vrstvy nebo atributového pole.

**Parametr 0 – Input Layer** přijímá vrstvu z aktivní mapy projektu ArcGIS Pro. Výkonný skript ji vyhledává jménem nebo *longName* v *m.listLayers*.

**Parametr 1 – Attribute Field** je omezen filtrem na číselné datové typy:

```
self.params[1].filter.list = ['Short', 'Long', 'Float', 'Double']
```

Toto omezení zabraňuje uživateli vybrat atribut textového, nebo datumového typu, který by způsobil chybu při klasifikaci.

**Parametry 2–4** jsou plněny v **updateParameters** pomocí *arcpy.da.SearchCursor*, jež prochází atributové hodnoty a vypočítává počet prvků, rozsah (minimum – maximum) a dvojici průměr/medián ze standardní knihovny **statistics**.

## Klasifikace dat

Sekce popisuje nastavení parametrů pro klasifikaci dat a jejich vzájemné interakce. Cílem je zajistit, aby byl uživateli vždy přístupný pouze ten parametr, který je pro danou metodu relevantní, a aby zadané hodnoty u manuálních intervalů odpovídaly požadovanému formátu:

```

self.params[5].filter.list = [
    "Manual ...",
    "Proportional (Unclassed) ...",
    "Equal Interval",
    "Quantile",
    "Natural Breaks",
    "Geometric Interval",
]

```

**Parametr 5 – Classification Method** nabízí šest metod klasifikace. Hodnota "Manual ..." aktivuje **parametr 7** a deaktivuje **parametr 6**; hodnota "Proportional (Unclassed) ..." deaktivuje oba (počet tříd je odvozen z počtu unikátních hodnot). Pro všechny ostatní metody je aktivní **parametr 6** a deaktivován **parametr 7**. Tato dynamická logika je implementována v *updateParameters*:

```

classification_method = self.params[5].valueAsText or self.params[5].value
if classification_method in {"Manual", "Manual ..."}:
    self.params[6].enabled = False
    self.params[7].enabled = True
elif classification_method == "Proportional (Unclassed) ...":
    # Proportional unclassified has no fixed number of categories
    self.params[6].enabled = False
    self.params[7].enabled = False
else:
    self.params[6].enabled = True
    self.params[7].enabled = False

```

**Parametr 7 – Manual Class Breaks** přijímá hranice tříd jako textový řetězec oddělený mezerami nebo středníkem (např. "10; 500; 1000"). Středník byl zvolen z důvodu, aby výkonný skript mohl podporovat jak desetinné tečky, tak čárky (pro uživatele s evropským nastavením lokalizace). Validace v *updateMessages* kontroluje, zda jsou zadané hodnoty numerické a zda jsou v rostoucím pořadí.

## Velikost písma a výběr fontu

Nastavení typografie představuje klíčový prvek. Tato sekce popisuje mechanismus, jakým nástroj definuje gradaci velikostí písma pro jednotlivé klasifikační třídy a jak zajišťuje dostupnost systémových fontů v uživatelském rozhraní. Cílem je umožnit plynulou gradaci textových popisků v rámci operačního systému Windows.

**Parametry 8 a 9** definují rozsah velikosti písma v bodech (pt). Výkonný skript mezi nimi lineárně interpoluje pro každou třídu:

```

if num_classes > 1:
    step = (max_size - min_size) / (num_classes - 1)
    manual_sizes = [min_size + i * step for i in range(num_classes)]
else:
    manual_sizes = [min_size]

```

Validace v *updateMessages* ověřuje, že  $min\_size > 0$ ,  $max\_size > 0$  a  $max\_size \geq min\_size$ .

**Parametr 10 – Font Selection** je naplněn dynamicky při inicializaci metodou *load\_font\_families*, která čte Windows Registry a systémové adresáře fontů. Výchozí hodnota je font "Calibri".

**Parametr 11 – Font Weight/Style** nabízí seznam řezů dostupných pro vybraný font (dynamicky aktualizovaný při změně **parametru 10**) a navíc položku "Gradient Font Weight ...", která aktivuje parametry **12 a 13**.

Jsou povoleny pouze tehdy, pokud je v **parametru 11** vybráno "Gradient Font Weight ...". V tomto režimu jsou **parametry 12 a 13** naplněny stejným dynamickým seznamem řezů jako **parametr 11** (bez položky "Gradient Font Weight ..."), takže uživatel jednoduše vybere počáteční a koncový řez nebo weight gradientu.

**Parametr 14 – Letter Spacing (%)** nastavuje mezery mezi znaky v procentech. Nenulová hodnota způsobí, že výkonný skript obalí formátovaný text Arcade HTML značkami `<CHR spacing='...'>`:

```
chr_start = f"<CHR spacing='{letter_spacing}'>" if letter_spacing else ""
chr_end = "</CHR>" if letter_spacing else ""
```

## Barva textu a gradient

Tato část řeší barevné vstupy a jejich transformaci pro potřeby ArcGIS Pro. Klíčovým mechanismem je lineární interpolace, která zajišťuje automatický výpočet plynulých barevných přechodů mezi zvolenými extrémy pro jednotlivé třídy klasifikace.

**Parametr 15 – Text Color (start)** přijímá barvu v libovolném podporovaném formátu: *HEX* (#RRGGBB, RRGGBBAA), *RGB* (R,G,B), *RGBA* (R,G,B,A), *CMYK* (C,M,Y,K), *CMYKA* (C,M,Y,K,A), *HSL* (H,S,L) nebo *HSLA* (H,S,L,A). Výchozí hodnotou je "0,0,0,1" (černá, plně neprůhledná v RGBA formátu).

**Parametr 16 – Use color gradient** je přepínač, který aktivuje parametry 17 a 18. Pokud je False, parametry se uživateli nezobrazí a jejich hodnoty výkonný skript ignoruje. Pokud jsou zadány barvy ve formátu CMYK pro všechny aktivní barevné parametry (*start*, *end*, případně *mid*), nástroj automaticky přepne na interpolaci v CMYK prostoru (funkce [interpolate\\_colors\\_cmyk](#) nebo [interpolate\\_colors\\_with\\_mid\\_cmyk](#)) pro správný gradient.

**Parametr 17 – Text Color (mid)** je nepovinný středový bod gradientu. Je-li zadán, použije se trojbarevná interpolace (*start*→*mid*→*end*) namísto dvoubodové.

## Formátování číselných hodnot

Adaptace numerických výstupů na českou typografickou normu zajišťuje čitelnost a eliminaci interpretačních chyb. Implementovaná logika řeší zaokrouhlování na definovaný počet desetinných míst a lokalizaci oddělovačů řádů, což může být vhodné pro vizuální sjednocení řádově odlišných statistických dat.

**Parametr 19 – Thousands Separator** nabízí tři možnosti: "Comma" (formát 3,215,410.50), "Space" (formát 3 215 410,50) a "None" (3215410,50). Výchozí hodnotou je "Space", vhodná pro české kartografické standardy. Výkonný skript převádí tuto volbu na konkrétní Arcade výraz s funkcí [Replace](#).

**Parametr 20 – Value Scale** obsahuje předem definovaný seznam násobitelů od tisícín (0.001) po biliony (1,000,000,000,000). Hodnoty atributu jsou ve výsledném Arcade výrazu děleny zvoleným faktorem:

```

scale_map = {
    "0.000001 (millionths)": 0.000001,
    "0.001 (thousandths)": 0.001,
    "0.01 (hundredths)": 0.01,
    "0.1 (tenths)": 0.1,
    "1 (no scaling)": 1,
    "1,000 (thousands)": 1000,
    "1,000,000 (millions)": 1000000,
    "1,000,000,000 (billions)": 1000000000,
    "1,000,000,000,000 (trillions)": 1000000000000,
}
value_scale = scale_map.get(str(value_scale_raw), 1)

```

**Parametr 21 – Value Suffix** umožňuje připojit libovolný textový řetězec za zobrazené číslo (např. "mil.", "tis.", " km<sup>2</sup>"). Suffix je vložen přímo do Arcade výrazu.

**Parametr 22 – Decimal Places After Scaling** určuje počet desetinných míst po škálování. Zaokrouhlení hranic tříd probíhá pomocí modulu **decimal** s konstantou **ROUND\_HALF\_UP**, nikoli standardním plovoucí desetinnou čárkou:

```

breaks = [
    float(
        Decimal(str(b)).quantize(
            Decimal('1.' + '0' * max_decimals),
            rounding=ROUND_HALF_UP,
        )
    )
    for b in breaks
]

```

Validace v **updateMessages** ověřuje, že zadaná hodnota je nezáporné celé číslo.

## Pokročilé ovládání popisků

Konfigurace umožňuje doladění sazby v závislosti na geometrii prvků. Implementovaná logika se zaměřuje na řízení zalamování textu prostřednictvím *Maplex Label Engine* a definicí odsazení (offsetu) vygenerovaného popisku. Cílem je eliminovat vizuální konflikty a zajistit čitelnost popisků i v graficky exponovaných částech mapové kompozice.

**Parametr 23 – Prevent Label Wrapping** ovládá chování *Maplex Label Engine* při zalamování popisků. Je-li *True*, nástroj nastaví *mlyp.canStackLabel = False*, *stacking\_props.maximumNumberOfLines = 1*, *stacking\_props.stackingSeparators = []* a *mlyp.maximumCharactersPerLine = 500*. Je-li *False*, jsou obnoveny výchozí hodnoty *Maplex* (max. 3 řádky, oddělovače mezera a čárka). Výchozí hodnota *True* je vhodná pro metodu číselných kartodiagramů, kde je žádoucí zobrazit hodnotu na jednom řádku.

**Parametry 24 a 25 – Label Offset X/Y** nastavují posunutí popisku vůči referenčnímu bodu prvku v bodech (pt). Jsou aplikovány přes vlastnosti *sym.offsetX* a *sym.offsetY* CIM objektu *CIMTextSymbol*.

**Parametr 26 – Underline Text** přidá podtržení ke všem generovaným třídám popisků přes *sym.underline = underline\_text*.

## Halo a shadow efekty

Pokročilé grafické možnosti výstupů je dosaženo dynamickou konfigurací efektů halo a stínu. Implementace využívá objektů *CIMPolygonSymbol* pro tvorbu masek a umožňuje lineární gradaci parametrů napříč klasifikačními třídami.

**Parametr 27 – Halo Color (start)** definuje výchozí barvu halo efektu, přičemž výchozí hodnotou je bílá (255,255,255,1). Sdílí stejnou flexibilní syntaxi vstupu jako ostatní barevné parametry nástroje a je zpracován prostřednictvím funkce `get_parsed_color`, resp. `get_parsed_color_cmyk` v závislosti na režimu interpolace.

**Parametr 28 – Halo Width (pt)** je stěžejním přepínačem halo efektu. Výchozí hodnota „0“ halo zcela deaktivuje – výkonný skript testuje `if halo_width > 0`: před jakoukoliv prací s halo barvami nebo šířkami. Halo je v CIM reprezentováno jako samostatný objekt `CIMPolygonSymbol` vytvořený funkcí `create_halo_symbol` a přiřazený přes `sym.haloSymbol`. Šířka halo je nastavena explicitně přes `sym.haloSize`, protože přiřazení objektu `haloSymbol` samo o sobě tloušťku neurčuje.

**Parametr 29 – Use Halo Gradient** zapíná gradientní halo efekt, které umožňuje plynulý přechod barvy haló i jeho šířky mezi prvním a posledním popiskem. Je-li True, jsou aktivovány **parametry 30, 31 a 32**. Gradient šířky haló je počítán lineárně:

```
if use_halo_gradient and halo_width_end is not None:
    if num_classes > 1:
        halo_widths_for_classes = [
            halo_width
            + (halo_width_end - halo_width) * i / (num_classes - 1)
            for i in range(num_classes)
        ]
    else:
        halo_widths_for_classes = [halo_width]
```

**Parametr 32 – Halo Width (end)** je nepovinný. Pokud není vyplněn, ale gradient halo efektu je aktivní, výkonný skript automaticky nastaví `halo_width_end = halo_width`, čímž zachová konstantní šířku haló při gradientu pouze barev.

**Parametry 33–35 – Shadow** ovládají stín textu. Stín je v ArcGIS Pro implementován jako přímé vlastnosti objektu `CIMTextSymbol` (`sym.shadowColor`, `sym.shadowOffsetX`, `sym.shadowOffsetY`). Stín je aktivní pouze tehdy, když je zadána barva (`shadow_color`) a alespoň jedno posunutí je nenulové. Pokud jsou podmínky nesplněny, jsou posunutí nastavena explicitně na nulu.

## 5.4 Validace vstupů – třída `ToolValidator`

Třída `ToolValidator` tvoří nezbytnou „obranou“ vrstvu mezi zamýšleným použitím nástroje a výkonným skriptem (**Execution**). Jejím účelem je maximálně zamezit situacím, kdy by `main` obdržel parametry v neplatném, nekonzistentním nebo kartograficky nevhodném stavu. Validace je rozložena do tří metod volaných v pevném pořadí: `initializeParameters`, `updateParameters` a `updateMessages`. Každá z metod je plně obalena bloky `try/except` tak, aby případná výjimka ve validačním kódu nikdy nezabránila otevření nebo použití dialogu nástroje.

### Inicializace parametrů

Metoda `initializeParameters` je volána jednou při otevření dialogu a má na starosti tři věci: nastavit `displayName` a `description` každého parametru, přiřadit výchozí hodnoty a nakonfigurovat statické filtry. Nejsložitější operací v rámci inicializace je naplnění seznamu fontů pro **parametr 10 – Font Selection**:

```
families = self._load_font_families
if families:
    self.params[10].filter.list = families
    if self.params[10].value is None:
        self.params[10].value = "Calibri" if "Calibri" in families else
        "Arial" if "Arial" in families else families[0]
```

Výsledný seznam pochází z metody `_load_font_families`, která interně volá `_build_font_catalog` – ta prochází Windows Registry a fontové adresáře a parsuje data fontových souborů. Proces stojí řádově stovky milisekund, proto je výsledek uložen do `self._font_catalog` a při opakovaném volání vrácen z cache bez nového čtení.

Ihned po naplnění seznamu rodin je pro výchozí rodinu sestaven seznam dostupných řezů a přiřazen **parametru 11**:

```
sel_family = (self.params[10].valueAsText or
              (self.params[10].filter.list[0] if self.params[10].filter.list else None))
styles = self._get_font_styles(sel_family)
if styles:
    full_list = ["Gradient Font Weight ..."] + styles
    self.params[11].filter.list = full_list
    cur_val = (self.params[11].valueAsText or '').strip
    if not cur_val or cur_val == "Gradient Font Weight ...":
        preferred = next((s for s in styles if s.lower == "regular"), styles[0])
        self.params[11].valueAsText = preferred
```

Tím je zajištěno, že uživatel nikdy nezahledne prázdný seznam řezů, a zároveň je vybráno výchozí "Regular" (nebo první dostupný řez, pokud Regular neexistuje).

Na konci metody je inicializována cache názvů fontů z registru:

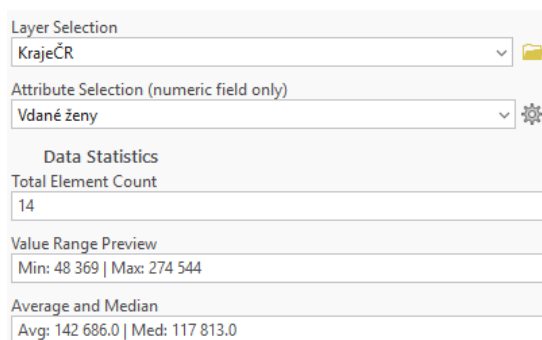
```
self._font_registry_names = dict(self._load_font_registry)
```

Celá metoda je rozdělena do dvou samostatných bloků `try/except` – první obsluhuje nastavení všech parametrů, druhý naplnění seznamu řezů. Toto oddělení zaručuje, že selhání při načítání fontů nezablokovuje inicializaci ostatních parametrů.

## Dynamická aktualizace parametrů

Metoda `updateParameters` se spouští po každé změně hodnoty libovolného parametru. Obsahuje pět logicky oddělených bloků, každý samostatně obalený `try/except`.

**Výpočet statistik náhledu** – je-li vybrána vrstva i pole, metoda spustí `arcpy.da.SearchCursor` a vypočítá počet prvků, rozsah hodnot a průměr/medián z modulu **statistics**. Výsledné hodnoty informují uživatele ještě před spuštěním nástroje, takže může snáze volit počet tříd nebo klasifikační metodu s ohledem na reálné rozložení dat (obrázek 13).



Obr. 13 Zobrazení automaticky vypočtených statistik zvoleného atributu v UI nástroje

Výsledky jsou formátovány s odsazením tisíců mezerami a zapsány do **param. 2, 3 a 4**:

```
values = [
    row[0]
    for row in arcpy.da.SearchCursor(input_layer, [attribute_field])
    if row[0] is not None
]
if values:
    total_count = len(values)
    self.params[2].value = f"{total_count}"
    min_val = min(values)
    max_val = max(values)
    formatted_min = f"{min_val:,"}.replace(",", " ")
    formatted_max = f"{max_val:,"}.replace(",", " ")
    self.params[3].value = f"Min: {formatted_min} | Max: {formatted_max}"
    avg_val = statistics.mean(values)
    median_val = statistics.median(values)
    formatted_avg = f"{avg_val:,.1f}".replace(",", " ")
    formatted_median = f"{median_val:,.1f}".replace(",", " ")
    self.params[4].value = f"Avg: {formatted_avg} | Med: {formatted_median}"
```

#### Aktivace parametrů klasifikace:

```
classification_method = self.params[5].valueAsText or self.params[5].value
if classification_method in {"Manual", "Manual ..."}:
    self.params[6].enabled = False
    self.params[7].enabled = True
elif classification_method == "Proportional (Unclassed) ...":
    # Proportional unclassified has no fixed number of categories
    self.params[6].enabled = False
    self.params[7].enabled = False
else:
    self.params[6].enabled = True
    self.params[7].enabled = False
```

**Aktualizace seznamu řezů fontu** při změně rodiny (**parametr 10**). Nový seznam je vytvořen voláním `get_font_styles(sel_family)` a přiřazen do `self.params[11].filter.list`. Aby nedocházelo k zbytečnému blikání dialogu, seznam se aktualizuje pouze tehdy, je-li skutečně jiný než stávající.

Pokud vybraná rodina není v dostupném seznamu (uživatel ji ručně přepsal), je zobrazeno varování:

```
self.params[10].setWarningMessage("Selected font family is not available on this system.
Choose a family from the list or the tool will fall back during execution.")
```

**Aktivace parametrů gradientu váhy fontu** na základě hodnoty **parametru 11**. Je-li vybráno "Gradient Font Weight ...", jsou povoleny **parametry 12 a 13** a jejich filtry jsou naplněny aktuálním seznamem řezů (bez položky gradient).

**Aktivace barevných gradientů** – **parametry 17 a 18** jsou povoleny nebo zakázány podle hodnoty `bool(self.params[16].value)`, **parametry 30, 31 a 32** podle `bool(self.params[29].value)`.

## Finální validace zpráv

Metoda `updateMessages` je posledním krokem validace a je volána teprve po `updateParameters`. Zobrazuje uživateli chybové (`setErrorMessages`) nebo varovné (`setWarningMessage`) zprávy pro konkrétní parametry, čímž blokuje nebo varuje před spuštěním.

**Validace atributového pole (parametr 1):** Nástroj ověří, zda je zvolené pole skutečně numerické porovnáním jeho `field.type` s povoleným seznamem typů:

```
numeric_types = ['Integer', 'Single', 'Double', 'OID', 'SmallInteger', 'Long']
if field_obj.type not in numeric_types:
    self.params[1].setWarningMessage(
        f"Field '{attribute_field}' is of type '{field_obj.type}'. "
        "A numeric field (Integer, Double, etc.) is required for classification."
    )
```

Jde o varování (nikoli chybu), protože `filter.list` na **parametru 1** je již nastaven na numerické typy v `initializeParameters` – dvojí ochrana snižuje pravděpodobnost průchodu neplatného pole.

**Validace počtu tříd (param 6):** Číslo musí být kladné celé číslo. Při neplatném vstupu je zobrazena chyba.

**Validace velikosti písma (parametry 8 a 9):** Kontrolují se tři podmínky a doporučuje se, aby byl rozdíl alespoň 3 body:

```
if min_val <= 0:
    self.params[8].setErrorMessage("Font size must be greater than 0.")
elif min_val >= max_val:
    self.params[9].setErrorMessage("Maximum Font Size must be strictly greater than
    Minimum Font Size.")
elif (max_val - min_val) < 3:
    self.params[9].setWarningMessage("The difference between Min and Max font size is
    very small.")
```

**Validace gradientu váhy fontu (parametry 11–13):** Je-li aktivní "Gradient Font Weight ...", zkontroluje se, zda má vybraný font alespoň dva řezy (jinak gradient nebude viditelný). Pak se ověří, zda jsou zadané hodnoty **parametrů 12 a 13** buď platnými názvy řezu ze seznamu, nebo numerickými hodnotami váhy pomocí `is_numeric_weight(val)`.

**Validace desetinných míst (param 22):** Hodnota musí být celé číslo  $\geq 0$ .

**Validace formátů barev:** Pro **parametry 15, 17, 18, 27, 30 a 31** je volaná lokální funkce `valid_color_text(s)`, která implementuje plnou sadu heuristik pro všechny podporované formáty. Funkce je definována přímo uvnitř `updateMessages` a testuje postupně `HEX`, `RGB/HSL` (3 složky), `RGBA/HSLA/CMYK` (4 složky) a `CMYKA` (5 složek).

Hodnoty **parametrů 17 a 18** (*mid/end barva textu*) a **30–31** (*mid/end barva halo efektu*) jsou validovány pouze tehdy, jsou-li příslušné gradienty aktivní.

**Varování před přetížením ArcGIS Pro:** Přítomnost velkého počtu prvků může způsobit vážné problémy s výkonností Maplex Label Engine. Proto jsou implementována dvě varování s různými prahy *if count is not None and count > 200*:

a *if classification\_method == "Proportional (Unclassed) ..." and count is not None and count > 50*.

Práh 200 prvků pro obecné varování a 50 prvků pro metodu `Proportional` vychází z experimentálního testování chování ArcGIS Pro při větším počtu label classes, viz. kapitola 5.11.

**Validate Value Scale vůči datovému rozsahu (parametry 3 a 20):** Jedná se o křížovou kontrolu mezi dvěma parametry. Z náhledového textu **parametru 3** (*Value Range*) je extrahována maximální hodnota a porovnána se zvoleným měřítkem. Jsou testovány dva scénáře:

1. Zvolené měřítko je větší než maximální hodnota v datech → popisky by zobrazovaly nuly nebo extrémně malá čísla skrze *if max\_value < scale and scale > 1*.
2. Zvolené měřítko je menší než 1 → potenciální výsledná hodnota by překročila 1 miliardu, čímž by se popisky staly nečitelnými pomocí *elif scale < 1 and max\_value > 0*.

## Načítání a analýza systémových fontů

Schopnost dynamicky načítat seznam nainstalovaných fontů, detekovat jejich dostupné řezy a rozlišit variable fonty od statických je jednou z nejdůležitějších technických možností tohoto nástroje. ArcGIS Pro do této doby nenabízí žádné nativní API, které by umožnilo dotaz na seznam systémových fontů nebo jejich metadat. Celá tato funkcionality proto musela být implementována kombinací čtení Windows Registry prostřednictvím modulu **winreg** a binární analýzy fontových souborů pomocí knihovny **fontTools**.

Logika načítání fontů je částečně zdvojená: ve **Validation** je realizována formou metod třídy **ToolValidator** (volaných při inicializaci dialogu), v **Execution** jako vnořené funkce uvnitř **main** (volané při spuštění). Obě implementace sdílejí totožnou logiku, jsou však nezávislé na sobě – dialogová třída nesdílí instanci s výkonným skriptem.

## Sběr cest k fontovým souborům

Výchozím bodem celého procesu je metoda ***\_iter\_font\_file\_paths*** v **Validation** (resp. stejnojmenná vnořená funkce v **Execution**), která vrací seřazený seznam absolutních cest ke všem dostupným fontovým souborům (.ttf, .otf, .ttc). Data jsou sbírána ze dvou zdrojů souběžně.

**Zdroj 1 – Windows Registry:** Jsou prohledány dva *hive* klíče:

***HKEY\_LOCAL\_MACHINE*** obsahuje systémové fonty (instalované pro všechny uživatele), ***HKEY\_CURRENT\_USER*** obsahuje fonty instalované pouze pro aktuálního uživatele. Registry hodnota může obsahovat buď absolutní cestu (pak je ověřena přímá existence souboru) nebo pouze název souboru bez cesty. V takovém případě je soubor hledán ve dvou systémových adresářích:

```
search_dirs = (  
    os.path.join(os.environ.get("WINDIR", r"C:\Windows"), "Fonts"),  
    os.path.join(local_appdata, "Microsoft", "Windows", "Fonts"),  
)
```

Druhý adresář (***%LOCALAPPDATA%\Microsoft\Windows\Fonts***) je umístění per-user fontů ve Windows 10 a novějších.

**Zdroj 2 – Přímé prohledání adresářů:** Nad rámec Registry jsou oba adresáře prohledány také přímým výpisem souborů s příponou .ttf, .otf nebo .ttc. Tím jsou zachyceny fonty instalované bez záznamu v Registry (např. přepokopované ručně):

```

for base_dir in search_dirs:
    if not os.path.isdir(base_dir):
        continue
    try:
        for file_name in os.listdir(base_dir):
            if file_name.lower().endswith((".ttf", ".otf", ".ttc")):
                font_paths.add(os.path.join(base_dir, file_name))
    except Exception:
        pass

```

Výsledný set `font_paths` eliminuje duplicitu a je vrácen jako seřazený seznam pro deterministické chování.

## Sestavení katalogu fontů

Jádrem celého systému je metoda `_build_font_catalog`, která iteruje přes všechny nalezené cesty a pro každý fontový soubor extrahuje sadu metadat pomocí `fontTools.ttLib.TTFont`. Výsledkem je slovník `catalog`, jehož klíče jsou normalizované názvy rodin, hodnotami pak záznamy se sadou dostupných řezů, pravdivostními příznaky variable fontu a rozsahem osy váhy (font weight).

Parametr `recalcBBoxes=False` zakazuje přepočítání bounding boxů při načtení, čímž se výrazně zkracuje doba otevření souboru – pro účely analýzy metadat nejsou bounding boxy potřeba. Čtení tabulky `name`: OpenType/TrueType fonty obsahují standardizovanou tabulku `name` s řetězovými záznamy (**NameRecords**).

Záznamy relevantní pro identifikaci rodiny a řezu jsou:

Tabulka 4 Přehled nameID záznamů využívaných pro identifikaci rodiny a řezu písma

nameID	Obsah
1	Font Family Name (tradiční, max. 4 řezy)
2	Font Subfamily Name (tradiční styl)
16	Typographic Family Name (rozšířená rodina)
17	Typographic Subfamily Name (rozšířený řez)

Záznamy `nameID` 16 a 17 (*Typographic Family/Subfamily*) jsou preferovány, protože správně reflektují celou rodinu fontů i u rodin s více než čtyřmi řezy. Pro fontové soubory, kde jsou přítomné záznamy obou typů, jsou sbírány všechny a pak normalizovány. Dekódování záznamu zajišťuje metoda `_decode_name_record`, která se nejprve pokusí o `toUnicode` (pokryje UTF-16 BE záznamy pro Windows platformu) a při selhání zkusí UTF-8 s `errors="ignore"`

**Normalizace názvů rodin – `_normalize_family_name`:** Registry ani fontové soubory nejsou konzistentní v pojmenování. Například variable font Oswald může být v Registry uložen jako Oswald Variable, v souboru jako Oswald-VariableFont-wght a v Typographic Family Name jako Oswald. Normalizace odstraňuje tři kategorie přebytečných suffixů.

**Normalizace názvů řezů – `_normalize_style_name`:** Fontové soubory používají různá schémata pro pojmenování řezů. Zkratky jako "bi", "sb", "b", nebo plné názvy jako "Bold Italic", "SemiBold", případně i "Regular Italic" (což je sémanticky ekvivalentní "Italic"). Metoda `_normalize_style_name` mapuje tyto variace na sadu kanonických ArcGIS-kompatibilních názvů prostřednictvím dvou slovníků `style_name_map` a `canonical_style_names`.

Složené názvy jako "Bold Italic", "Light Italic" nebo "ExtraBold Italic" jsou rozebrány na „*tokens*“, každý token je normalizován samostatně a redundantní "Regular" je odstraněn pokud je přítomen jiný token:

```
if "Regular" in normalized_tokens and len(normalized_tokens) > 1:
    normalized_tokens = [token for token in normalized_tokens if token != "Regular"]

style_value = " ".join(dict.fromkeys(normalized_tokens))
if style_value.lower() == "regular italic":
    style_value = "Italic"
return style_value or None
```

**Detekce variable fontu a os** – tabulka *fvar*: Pokud fontový soubor obsahuje tabulku *fvar* (Font Variations), jedná se o variable font. Metoda iteruje přes osy a hledá **wght** (*váha*) a *ital/slnt* (*kurzíva/sklon*)

Výsledné hodnoty *real\_min* a *real\_max* (např. 200–700 pro Oswald) jsou uloženy do katalogu a slouží v *get\_font\_styles* ke generování realistického seznamu dostupných vah.

**Struktura záznamu v katalogu:**

```
entry = catalog.setdefault(
    family_name,
    {
        "styles": set(),
        "files": set(),
        "is_variable": False,
        "has_wght_axis": False,
        "has_italic_axis": False,
        "real_min": 400,
        "real_max": 400,
    },
),
```

## Generování seznamu řezů

Metoda *get\_font\_styles(family\_name)* je přímo volaná z *initializeParameters* a *updateParameters* a vrací seřazený seznam řezů pro zadanou rodinu, vhodný k přímému přiřazení do *filter.list* parametru toolboxu.

Chování metody se větví podle toho, zda je rodina variable font nebo statický font.

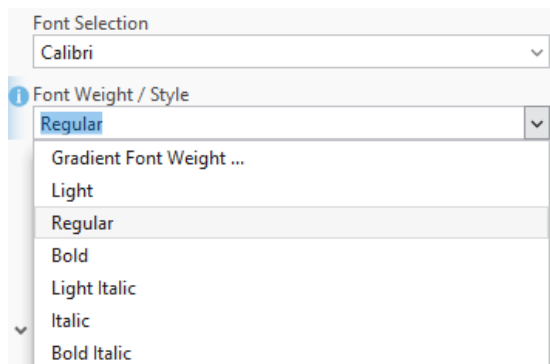
**Pro variable fonty** (*entry["is\_variable"] or entry["has\_wght\_axis"]*): Pokud soubor explicitně deklaruje alespoň dva ne-kurzívní řezy v tabulce *name*, jsou použity tyto řezy. Pokud nikoliv, jsou vygenerovány syntetické řezy na základě reálného rozsahu osy *wght* – ze slovníku *weight\_to\_preferred* jsou vybrány všechny váhy, které leží v intervalu [*real\_min*, *real\_max*]:

Kde *weight\_to\_preferred* mapuje číselné váhy na preferované názvy:

```
weight_to_preferred = {
    100: "Thin", 200: "ExtraLight", 300: "Light", 400: "Regular",
    500: "Medium", 600: "SemiBold", 700: "Bold", 800: "ExtraBold",
    900: "Black", 950: "UltraBlack"
}
```

Kurzívní řezy jsou pak připojeny za ne-kurzívní. Pokud má font osu *ital* (ale nemá explicitní kurzívní řez v tabulce *name*), je synteticky přidáno "Italic".

**Pro statické fonty** je vrácen pouze seznam řezů z tabulky *name*, seřazený funkcí *sort\_key*, která třídí primárně podle váhy (*Thin* → *Black*) a sekundárně kurzíva za ne-kurzívou při stejné váze. Výsledný seznam pro typický statický font (např. Calibri) vypadá jako na obrázku 14:



Obr. 14 Dynamicky generovaný seznam dostupných řezů písma pro zvolenou rodinu fontu

### Detekce rozsahu váhy

Vnořená funkce *get\_font\_weight\_range(font\_fam)* v **Execution** je specializovanou funkcí pro výkonný skript: nalezne fontový soubor odpovídající zadané rodině, přečte z tabulky fvar rozsah osy wght a vrátí trojici (*min\_weight*, *max\_weight*, *has\_wght\_axis*). Tyto hodnoty jsou pak použity k ořezání interpolovaných vah na reálný rozsah:

```
weight_start = max(min_weight, min(max_weight, weight_start))
weight_end = max(min_weight, min(max_weight, weight_end))
```

Pokud font nemá osu wght (tj. není variable font), vrací výchozí hodnoty (100, 950, False).

### Vyhledání registrového názvu variable fontu

Nástroj v **parametru 10** zobrazuje normalizované názvy rodin (tj. bez suffixu "Variable"), ale ArcGIS Pro CIM API při nastavování *fontStyleName* = "Regular" a *fontVariationSettings* potřebuje přesný registrový název. Funkce *\_get\_registry\_variable\_family(family)* tento překlad zajišťuje.

## 5.5 Klasifikace dat a výpočet tříd

Klasifikace dat představuje první skutečně výpočetní krok celého pracovního toku: na základě hodnot atributového pole a uživatelské volby metody jsou vygenerovány hranice tříd, které pak určují, které kategorie popisků budou vytvořeny, jaké hodnoty do nich patří a jak budou zobrazeny. Všechny klasifikační algoritmy jsou implementovány v modulárních funkcích v **Execution** a volány z *main*.

### Načítání hodnot atributu a příprava dat

Po ověření existence vrstvy a oprávnění zobrazovat popisky (*lyr.supports("SHOWLABELS")*) jsou všechny nenulové hodnoty zvoleného atributu načteny prostřednictvím *arcpy.da.SearchCursor*.

Podmínka *if row[0] is not None* ve **Validatoru** explicitně filtruje NULL hodnoty. Nominální minimum a maximum jsou vypočítány přímo z načtených hodnot – tyto hodnoty slouží jako absolutní hranice pro všechny klasifikační metody i jako záchranná síť, která zajistí, že žádná hranice třídy nepřesáhne skutečný datový rozsah.

## Normalizace metody klasifikace

Hodnota parametru **Classification Method** může obsahovat suffixové "..." (zobrazené uživateli v dialogu, což zvyšuje srozumitelnost v uživatelském rozhraní – uživatel okamžitě vidí, že volba s „...“ znamená, že jde o speciální režim). Před předáním do klasifikačních funkcí je proto normalizována:

```
method_text = str(classification_method or "").strip()
method_norm = method_text.lower()
normalized_method = classification_method
if "manual" in method_norm:
    normalized_method = "Manual"
elif "proportional" in method_norm:
    normalized_method = "Proportional (Unclassed)"
```

## Vstupní bod – funkce calculate\_breaks

Pro všechny metody s pevným počtem tříd je volána globální funkce `calculate_breaks(values, method, num_categories)`, která podle hodnoty parametru `method` deleguje výpočet na příslušný algoritmus klasifikační metody, viz. kapitola 4.4.

Funkce vrací seznam  $n + 1$  hranic pro  $n$  tříd. **Equal Interval** rozdělí datový rozsah na  $n$  stejně širokých intervalů pomocí konstantního kroku, **Quantile** rozdělí data tak, aby každá třída obsahovala přibližně stejný počet hodnot. Hranice jsou vypočítány indexováním do seřazeného pole. Výsledný seznam je deduplikován voláním `sorted(list(set(breaks)))`. Tento krok je nezbytný: při datasetech s mnoha identickými hodnotami může vzniknout více hranic na stejné hodnotě, což by způsobilo prázdné nebo duplicitní label classes a vedlo k chybě v ArcGIS CIM. **Geometric Interval** generuje hranice s geometricky rostoucím krokem vhodným pro data s exponenciálním rozložením. Poměr sousedních hranic je konstantní. Pokud je `min_val <= 0`, je nahrazena hodnotou 0.0001, aby se předešlo dělení nulou nebo záporné základně. **Proportional (Unclassed)** je speciální metoda, která nevytváří třídy o pevném počtu, ale jednu třídu pro každou unikátní hodnotu. Hranice jsou midpointy mezi sousedními unikátními hodnotami. Tato metoda pro  $k$  unikátních hodnot vytvoří  $k$  label classes – proto jsou v updateMessages implementována výkonnostní varování (> 50 prvků), viz. kapitola 5.4. Metoda **Natural Breaks (Jenks)** je implementována přímo v Pythonu jako funkce `natural_breaks(values, num_categories)`, protože arcpy nenabízí veřejné Python API pro výpočet Jenksových hranic mimo kontext vrstevní klasifikace.

Časová složitost tohoto algoritmu je  $O(n^2 \cdot k)$ , kde  $n$  je počet hodnot a  $k$  počet tříd. Pro velká  $n$  je tato složitost nepraktická, proto je implementována **optimalizace vzorkováním**: při  $n > 3000$  je algoritmem zpracován náhodný vzorek 3000 hodnot, přičemž globální minimum a maximum jsou zachovány z celého datasetu:

```
if len(values) > 3000:
    full_min = min(values)
    full_max = max(values)
    sampled = random.sample(values, 3000)
    sampled_breaks = natural_breaks(sampled, num_categories)
    if not sampled_breaks:
        return []
    sampled_breaks[0] = full_min
    sampled_breaks[-1] = full_max
    return sampled_breaks
```

Použití `random.sample` bez pevného seed zajišťuje nestranné vzorkování. Přepsání krajních hranic (`sampled_breaks[0] = full_min`, `sampled_breaks[-1] = full_max`) je klíčové pro správnost Arcade podmínek – zajistí, že žádná hodnota z datasetu nespadne mimo definovaný rozsah tříd.

Je-li zvolena metoda **Manuální hranice tříd**, jsou hranice parsovány z textového vstupu **parametru 7**. Parser podporuje oddělení středníkem ("10; 500; 1000") i mezerou ("10 500 1000") a akceptuje evropský formát desetinné čárky:

```
normalized_breaks = custom_breaks_text.replace(";", " ").replace(",", ".")
inner_breaks = [float(b.strip()) for b in normalized_breaks.split() if b.strip()]
breaks = [min_val] + sorted(inner_breaks) + [max_val]
number_of_categories = len(breaks) - 1
```

Automatické doplnění `min_val` na začátek a `max_val` na konec zajistí, že všechny hodnoty datasetu jsou zahrnuty do některé třídy, i pokud uživatel nezadal krajní hodnoty. Vnitřní hranice jsou seřazeny, aby se předešlo chybě při nelogickém pořadí vstupu.

## Škálování a zaokrouhlování hranic

Po vygenerování hranic jsou provedeny dvě transformace. Nejprve je zachována kopie původních hranic (pro použití v Arcade podmínkách, kde se porovnávají surové hodnoty atributu), poté jsou zobrazovací hranice vyděleny parametrem `value_scale`.

Následuje zaokrouhlení na zadaný počet desetinných míst. Místo standardního plovoucí desetinné čárky je použit modul **decimal** s explicitním módem **ROUND\_HALF\_UP**, který odpovídá kartografickým i statistickým konvencím.

Konverze přes `str(b)` před předáním do `Decimal` zabraňuje chybám způsobeným binární reprezentací čísla plovoucí desetinné čárky (např.  $0.1 + 0.2 = 0.30000000000000004$ ). Zaokrouhlení se aplikuje i při `value_scale == 1`, protože i neškálovaná data mohou mít hranice s nadbytečnými desetinnými místy po operacích Equal Interval nebo Geometric Interval.

Tabulka 5 Srovnání vypočtených vrchních hranic klasifikačních tříd pro statistické metody (vrstva: Kraje (ArcČR 2024) se zvoleným atributem „Bez vzdělání“ s min. hodnotou 2 045 a max. 7 794)

Třída	Equal Interval (Rovnoměrné)	Quantile (Kvantily)	Natural Breaks (Jenks)	Geometric Interval (Geometrický-interval)
1.	3194	2655	3632	2672
2.	4344	2816	4946	3492
3.	5494	3632	6681	4563
4.	6644	6681	7506	5964
5.	7794	7794	7794	7794

## Formátování číselných hodnot

ArcGIS Pro Arcade engine používá pro funkci `Text` interně anglosaský formát (oddělovač tisíců čárkou, desetinná tečka), zatímco česká kartografická praxe vyžaduje naopak mezeru jako oddělovač tisíců a čárku jako oddělovač desetin. Zároveň musí každá label class zobrazovat pouze hodnoty ze svého intervalu a hodnotu přitom přepočítat podle zvoleného hodnotového měřítká. Celá tato logika je vložena do Arcade výrazu generovaného dynamicky pro každou třídu.

## Struktura Arcade výrazu

Pro každou ze  $n$  tříd je programově vygenerován Arcade výraz v jazyce JavaScript-like syntaxe. Výraz má jednotnou strukturu: nejprve vydělí hodnotu atributu hodnotovým měřítkem, pak zkontroluje, zda hodnota patří do intervalu dané třídy, naformátuje číslo a vrátí řetězec. Pokud hodnota do intervalu nespadá, vrátí *null* – to ArcGIS Maplex Label Engine interpretuje jako „nezobrazovat popisek pro tento prvek v této třídě“.

Výsledná podoba Arcade výrazu pro separátor „**Space**“ (české lokalizační nastavení) (obrázek 15):

```
var val = $feature.Poc_obyv_2021 / 1000;
if ($feature.Poc_obyv_2021 >= 279100 && $feature.Poc_obyv_2021 < 497517) {
  var formatted = Text(val, "#,##0.#");
  formatted = Replace(formatted, ",", "");
  formatted = Replace(formatted, ".", ",");
  return "" + formatted + "tis.";
} else {
  return null;
}
```

Obr. 15 Ukázka vygenerovaného Arcade výrazu zajišťujícího formátování čísel a filtraci intervalů

Struktura výrazu je vždy totožná. Mění se pouze: název atributového pole, hodnota *value\_scale*, podmínkové hranice (*orig\_lower*, *orig\_upper*), formátovací vzor, řetězce nahrazení a suffix.

## Generování formátovacího vzoru

Arcade funkce *Text* (*value*, *format*) přijímá formátovací vzor v syntaxi podobné Excel/Java NumberFormat. Pro každou třídu je vzor *format\_pattern\_base* generován dynamicky na základě parametru *max\_decimals*:

```
if max_decimals > 0:
    decimal_part = "#" * max_decimals
    format_pattern_base = f"#,##0.{decimal_part}"
else:
    format_pattern_base = "#,##0"
```

Znaky # v části za desetinnou tečkou označují nepovinná číslíce (pokud je hodnota např. 1.5 a vzor #,##0.##, zobrazí se 1.5, nikoli 1.50). Vzor #,##0 před desetinnou tečkou zajistí oddělení tisíců čárkou v Arcade a zobrazení alespoň jedné číslice (i pro hodnotu 0). Přitom ale samotná čárka v tomto vzoru není finální – viz. následující kapitola.

## Konverze oddělovačů – tři varianty výrazu

Výsledný Arcade výraz je generován ve třech variantách podle hodnoty *thousand\_separator*:

- **Varianta „Comma“ (anglosaský formát):** Nevyžaduje žádné náhrady – výstup Arcade Text je přímo použitelný.
- **Varianta „Space“ (mezera):** Odpovídá české typografické normě. Čárka (oddělovač tisíců Arcade) je nahrazena nezlomitelnou mezerou Unicode `\u00A0` a tečka (desetinná) je nahrazena čárkou. Pořadí náhrad je záměrně přesné: nejprve čárka `\u00A0` (jinak by druhá náhrada tečky čárky zkombinovala s první a znovu nahradila právě vloženou čárku).
- **Varianta „None“ (bez oddělovače):** Odstraní čárku Arcade (oddělovač tisíců) a nahradí tečku čárkou.

## Pojmenování tříd popisků

Název každé label class (v ArcGIS Pro zobrazovaný v tabulce **Contents** v záložce **List by Labeling**) je odvozen ze zobrazovaných hranic intervalu po škálování a zaokrouhlení a z přípony *value\_suffix*. U nezávěrečných tříd je horní hranice záměrně snížena, aby se zobrazené intervaly nepřekrývaly se začátkem následující třídy. Pro desetinný režim se používá krok  $10^{-(max\_decimals)}$ , zatímco pro celočíselný režim ( $max\_decimals = 0$ ) se odečítá hodnota 1:

```
upper_display = int(round(upper)) - 1
name = (
    f"Values {int(round(lower))}{value_suffix} "
    f"- {upper_display}{value_suffix}"
)
```

Příklad pro  $max\_decimals=0$ ,  $value\_suffix="k"$ , 5 tříd s hranicemi [0, 20, 40, 60, 80, 100]:

Tabulka 6 Ukázka automaticky generovaných názvů tříd popisků

Třída	Název label class
<b>i=0</b>	Values 0 k – 19 k
<b>i=1</b>	Values 20 k – 39 k
<b>i=2</b>	Values 40 k – 59 k
<b>i=3</b>	Values 60 k – 79 k
<b>i=4</b>	Values 80 k – 100 k

Prefix "Values" je záměrný a slouží jako identifikátor – při opakovaném spuštění nástroje jsou všechny stávající label classes začínající tímto prefixem odstraněny před vygenerováním nových. Tím je zaručena stálost výsledku nástroje: opakované spuštění nepřidává label classes, ale nahrazuje předchozí.

## 5.6 Správa a vytváření label classes

Jakmile jsou k dispozici klasifikační zlomy a validované parametry, přistoupí nástroj k tvorbě popisných tříd. Celá správa popisků probíhá prostřednictvím CIM API metodou `lyr.getDefinition("V3")`, která vrátí plnou reprezentaci vrstvy v paměti; aby se změny projevil v projektu, musí být objekt zpětně zapsán voláním `lyr.setDefinition(cim_def)`.

Nástroj je **idempotentní**. To v tomto kontextu znamená, že opakované spuštění nástroje se stejnými vstupními parametry nad stejnou vrstvou vede ke stejnému výsledku. Toho je dosaženo tím, že před každým spuštěním jsou odstraněny všechny třídy (Label Classes) s prefixem "Values", zatímco ručně vytvořené třídy s jiným názvem zůstávají nedotčeny. Poté jsou popisky dočasně skryty nastavením `lyr.showLabels = False`, aby se předešlo vizuálním artefaktům při operacích.

Nové třídy jsou vytvářeny ve smyčce `for i in range(len(breaks) - 1)`. Pro každý interval jsou paralelně vedeny dvě sady hranic: *lower/upper* v zobrazovaných jednotkách po škálování a *orig\_lower/orig\_upper* v jednotkách zdrojového atributu, protože podmínka Arcade výrazu musí porovnávat surovou hodnotu pole, nikoli hodnotu upravenou pro zobrazení. Poslední interval je uzavřený z obou stran ( $\leq$ ), všechny předcházející jsou polootevřené ( $<$ ), aby žádná hodnota nebyla opomenuta (viz. proměnná *comparison*):

```

if i == len(breaks) - 2:
    # Last category includes upper boundary
    comparison = (
        f"$feature.{attribute_field} >= {orig_lower} "
        f"&& $feature.{attribute_field} <= {orig_upper}"
    )
else:
    # Other categories: exclude upper boundary
    comparison = (
        f"$feature.{attribute_field} >= {orig_lower} "
        f"&& $feature.{attribute_field} < {orig_upper}"
    )

```

Název každé třídy nese prefix "*Values*" následovaný zobrazeným rozsahem; u mezilehlých kategorií je horní hranice snížena o  $10^{(-max\_decimals)}$ , aby se sousední třídy v legendě vizuálně nepřekrývaly (viz proměnná *upper\_display*). Třída je registrována voláním *lyr.createLabelClass(name, expression)* s explicitně nastaveným *expressionEngine = "Arcade"*.

Po dokončení smyčky nástroj provede druhé volání *lyr.getDefinition("V3")*, které zachytí aktualizovaný stav včetně nově vzniklých tříd; z kolekce jsou opět odfiltrovány pouze třídy s prefixem "*Values*" do seznamu *label\_classes*. Tento seznam je referenčním vstupem pro celou fázi CIM stylování a proměnná *num\_classes* z něj odvozená určuje délky interpolačních řad fontů, barev a šířek hala. Dva oddělené CIM cykly jsou záměrnou architektonickou volbou: *createLabelClass* operuje přímo s vrstvou v ArcPy vrstvou, nikoli nad CIM objektem, a musí proto proběhnout mezi oběma *getDefinition/ setDefinition* páry. V závěrečném kroku stylování jsou změny zapsány zpět do vrstvy a popisky aktivovány (*lyr.showLabels = True*). Nástroj záměrně nevyvolává uložení projektu – veškeré změny zůstávají v paměti otevřeného dokumentu a je na uživateli, kdy projekt uloží.

## 5.7 Pokročilé stylování pomocí CIM API

Poté, co jsou interpolační řady barev, fontových stylů a šířek halo efektu předpočítány a label classes vytvořeny, prochází nástroj v cyklu *for idx, lc in enumerate(label\_classes)* každou třídu a aplikuje na ni vizuální nastavení prostřednictvím CIM API. Jde o nejrozsáhlejší část funkce *main* – zahrnuje přiřazení rodiny a stylu písma, velikost a barvy textu, mezer, podtržení, stínu atd. Všechny změny se provádějí na objektu *sym*, který je přímou referencí na CIM textový symbol dané třídy:

```
sym = lc.textSymbol.symbol
```

Modifikace *sym* tedy ihned ovlivní odpovídající CIM záznam v *lay\_cim*, aniž by bylo nutné provádět kopírování nebo samostatné přiřazení.

### Přiřazení rodiny a stylu písma

Rodina písma se nastavuje vlastností *sym.fontFamilyName*. V prostředí s gradienty byly jednotlivé rodiny pro každou třídu předem uloženy do *font\_families\_for\_classes*, a to například z důvodu, že lehčí řezy jako "*Calibri Light*" jsou ve Windows registrovány jako samostatná rodina, nikoli jako styl "*Light*" uvnitř rodiny "*Calibri*". Přiřazení proto bere hodnotu z tohoto seznamu:

```

assigned_family = font_families_for_classes[idx] if idx < len(
    font_families_for_classes) else font_families_for_classes[-1]
sym.fontFamilyName = assigned_family

```

Styl písma se poté nastavuje odlišně podle toho, zda jde o **variabilní** nebo **statický** font. U variabilních fontů se vždy nastavuje "Regular" bez ohledu na interpolovanou váhu – skutečná tloušťka řezu je pak řízena výhradně prostřednictvím os [fontVariationSettings](#).

## Velikost textu a základní typografická nastavení

Velikost alfanumerických znaků v bodech odpovídá lineárně interpolované hodnotě z pole [manual\\_sizes](#), jehož hodnoty leží v rozsahu [\[min\\_size, max\\_size\]](#):

```
sym.height = float(manual_sizes[idx])
sym.allowOverrun = True
sym.lineSpacing = 0
sym.maximumLines = 1
sym.wordWrap = False
```

Vlastnosti [lineSpacing](#), [maximumLines](#) a [wordWrap](#) jsou nastaveny na fixní hodnoty přizpůsobené kartografickému použití: nulové řádkování zamezuje nevyžádaným mezerám, omezení na jeden řádek a vypnuté zalamování zabraňují spontánnímu dělení hodnot u tříd, kde není aktivní *Maplex stacking*. Vliv param. **prevent\_wrapping** je popsán v kapitole 5.10.

## Odsazení a podtržení popisku

Popisky lze posunout od polohy prvku o zadaný počet bodů ve vodorovném či svislém směru. Nastavení probíhá v bloku [try/except](#), protože ne všechny verze CIM schématu garantují přítomnost těchto vlastností:

```
try:
    sym.offsetX = offset_x
    sym.offsetY = offset_y
except Exception:
    pass
```

Uniformní podtržení platí pro všechny label classes a nastavuje se vlastností [sym.underline](#), která přijímá hodnotu boolean.

## Stín textu

Stín je volitelný efekt dostupný tehdy, kdy uživatel zadal barvu stínu a alespoň jeden nenulový posun. Barva je parsována pomocí [parse\\_color](#) (podrobněji v kapitole 5.9) a předána do CIM objektu [CIMRGBColor](#), který je poté přiřazen přímo na [sym.shadowColor](#). Pokud podmínka selže nebo je stín zakázán, jsou posuny [shadowOffsetX/Y](#) explicitně vynulovány – tím se zabrání přenosu reziduálního stínu z předchozího spuštění, které CIM definici mohlo zanechat s nenulovými hodnotami.

## Barva textu

Barva textu je uložena jako *RGBA* v prvním symbolovém vrstvě objektu [sym.symbol](#). Nástroj přistupuje k výplni přes [symbolLayers\[0\]](#) a přiřadí jí předpočítanou *RGBA* hodnotu z [colors\\_for\\_classes\[idx\]](#).

Explicitní přepsání celého seznamu [symbolLayers](#) jedním prvkem je nezbytné: pouhá modifikace [fill.color.values](#) by nemusela v některých verzích CIM způsobit skutečné přenačtení symbolu. Přepsáním seznamu je zaručeno, že změna barvy se projeví.

## Halo efekt

ArcGIS Pro Halo efekt je aplikován, pokud je `halo_width > 0`. Barevný objekt pro každou třídu je vytvořen funkcí `create_halo_symbol` popsanou v kapitole 5.2, přičemž jako parametry slouží aktuální barva z `halo_colors_for_classes` a odpovídající šířka z `halo_widths_for_classes`. Výsledný CIM symbol je přiřazen na `sym.haloSymbol` a šířka je nastavena zvlášť na `sym.haloSize`.

Dvoji přiřazení šířky (jednou jako parametr `create_halo_symbol`, jednou explicitně jako `sym.haloSize`) je záměrné: `create_halo_symbol` přijímá šířku pouze informačně a sám o sobě ji nenastavuje, protože ArcGIS CIM uchovává šířku hala mimo strukturu symbolového objektu. Bez explicitního `sym.haloSize = ...` by halo bylo sice barevné, ale o nulové tloušťce a tedy neviditelné.

## Pojmenování label classes v režimu gradace fontu

Pokud je aktivní režim gradace fontu volbou `Gradient Font Weight ... (is_gradient == True)`, je ke jménu každé třídy přidáno označení aktuálního stylu. U variabilních fontů jde o numerickou hodnotu váhové osy (**weight=700**), u statických fontů o název řezu (**Bold**):

```
if is_gradient:
    try:
        if is_variable and numerical_weight is not None:
            suffix = f"wght={int(round(numerical_weight))}"
        else:
            suffix = raw_style or "Regular"
        lc.name = f"{lc.name} ({suffix})"
    except Exception:
        pass
```

Toto pojmenování nemá funkční dopad na Arcade výrazy ani filtrování, ale výrazně zlepšuje čitelnost tříd popisků (label class) v záložce **Contents** v ArcGIS Pro, kde uživatel okamžitě vidí, jaký styl konkrétní třída využívá.

## 5.8 Podpora variabilních fontů a interpolace váhy

**Variabilní fonty** (*variable fonts*) umožňují plynulé přechody mezi typografickými variantami. Místo sady statických fontů obsahují jednu binární datovou strukturu s osami (*fvar*), podél nichž lze plynule pohybovat. Nejběžnější osa je *wght* (váha tloušťky, typicky 100–950). Nástroj tyto možnosti plně využívá a přináší tak interpolaci tloušťky písma jako nativní vizuální proměnnou – každá klasifikační třída nese jiný stupeň tloušťky, což přidává do vizualizace metody číselných kartodiagramů další dimenzi.

### Detekce variabilního fontu

Rozpoznání variabilního fontu probíhá ve třech vrstvách. Primárním zdrojem je výsledek funkce `get_font_weight_range`, která prochází fontové soubory odpovídající zvolené rodině a hledá tabulku *fvar* s osou *wght*:

```

all_styles = get_true_font_styles(font_family_cim)
if not all_styles:
    all_styles = ["Regular"]

family_lower = font_family_cim.lower()
min_weight, max_weight, has_wght_axis = get_font_weight_range(font_family_cim)
is_variable = has_wght_axis or family_lower.endswith("variable") or " variable"
in family_lower

```

Doplňkové podmínky `endswith("variable")` a `"variable" in family_lower` zachycují případy, kdy font sice není správně popsán tabulkou `fvar`, ale jeho registrový název ho jednoznačně identifikuje jako variabilní (například "Oswald Variable").

Funkce `get_font_weight_range` vrací trojici (`min_weight`, `max_weight`, `has_wght_axis`). Skutečný rozsah osy je čten přímo z `fvar` tabulky pomocí `fontTools`. Návrátová hodnota (`100`, `950`, `False`) v případě nenalezení je defenzivní fallback: hodnoty pokrývají celý konvenční rozsah font weights, takže i bez ověřeného `fvar` záznamu nedojde k oříznutí uživatelem zadaných vah.

## Mapování názvů registru

ArcGIS Pro zobrazuje ve výběrovém seznamu fontů jméno rodiny v normalizované podobě, například "Oswald". Skutečný registrový záznam pro variabilní verzi téže rodiny však může nést název "Oswald Variable". Funkce `get_registry_variable_family` tento nesoulad řeší prohledáním registru (`HKLM + HKCU`) a preferencí záznamu obsahujícího klíčové slovo `"variable"` nebo `"var"`. Na proměnnou `font_family_cim` se pak odkazují všechna navazující volání (`get_true_font_styles`, `get_font_weight_range`, `sym.fontFamilyName`). Tím je docíleno konzistentního propojení uživatelsky zobrazeného názvu s technickým identifikátorem potřebným pro CIM.

## Interpolace váhy v gradientním režimu

Je-li aktivní gradientový režim (`is_gradient == True`) a font je variabilní, dochází k numerické interpolaci váhové hodnoty podél osy `wght`. Počáteční a koncová váha jsou získány buď přímo z číselné hodnoty parametru (například "400") nebo přes slovník `name_to_weight`, který převádí názvy stylů na standardní numerické hodnoty:

```

weight_start = parse_weight(font_weight_min)
weight_end = parse_weight(font_weight_max)
if weight_start is None:
    weight_start = name_to_weight.get(str(font_weight_min).split()[0], 400)
if weight_end is None:
    weight_end = name_to_weight.get(str(font_weight_max).split()[0], 700)
weight_start = max(min_weight, min(max_weight, weight_start))
weight_end = max(min_weight, min(max_weight, weight_end))
for i in range(num_classes):
    if num_classes > 1:
        t = i / (num_classes - 1.0)
        weight = weight_start + t * (weight_end - weight_start)
    else:
        weight = weight_start
weights_for_classes[i] = weight
styles_for_classes[i] = "Regular"

```

Výsledná váha pro každou třídu je lineárně interpolovaná hodnota parametru  $t \in [0, 1]$ . Oříznutí voláním `max(min_weight, min(max_weight, ...))` zabraňuje zadání hodnot mimo fyzicky realizovatelný rozsah konkrétního fontu.

Funkce `parse_weight` zpracovává různé formáty vstupu – čistá čísla ("400"), číselné hodnoty s popisem ("400 (Regular)") i rozsahy ("100-900"). Vždy extrahuje první číselný token pomocí regulárního výrazu `parse_weight(val)`.

## Aplikace os na CIM symbol

Konkrétní váhová a kurzívní hodnota jsou na každou label class aplikovány prostřednictvím funkce `_apply_variable_axes_to_symbol`. Tato funkce vytváří pro každou osu objekt `CIMFontVariation` a přiřadí celý seznam na `sym.fontVariationSettings`. Celý předchozí obsah `fontVariationSettings` je přepsán novým seznamem – tím se zabrání situaci, kdy by z minulého spuštění zůstala reziduální osa s jiným tagem nebo hodnotou. Sestavení slovníku `axes` probíhá před každým voláním v rámci stylového cyklu. Váha je určena buď klíčovými slovy z názvu stylu (pro případ, kdy `raw_style` nese textový popis), nebo přímo interpolovanou hodnotou `numerical_weight`.

Detekce kurzívy (`italic_requested`) probíhá ze zdrojového parametru uživatele (`font_weight_val`), nikoli z interpolovaného `raw_style`, aby se v gradientním režimu správně přeneslo uživatelské přání o kurzívní variantě i při přechodu mezi váhami.

## Zachování kompatibility se statickými fonty

Celá větev variabilních fontů je striktně podmíněna hodnotou `use_variable`. U statických fontů probíhá přiřazení stylu konvenční cestou přes `sym.fontStyleName = raw_style`, přičemž `_apply_variable_axes_to_symbol` není volána. Tím jsou statické i variabilní fonty zpracovávány jedním kódem bez nutnosti paralelních implementací.

## 5.9 Parsování barev a interpolace barevných gradientů

Podpora barevných přechodů je jednou z vizuálně nejzajímavějších schopností nástroje: každé klasifikační třídě lze přiřadit odlišnou barvu textu i halo efektu, přičemž přechod mezi kategoriemi je plynulý. Celý subsystém tvoří skupina globálních funkcí, které zajišťují tři separátní odpovědnosti: rozpoznání a parsování vstupu uživatele v libovolném barevném formátu, výběr interpolačního prostoru (RGB nebo CMYK) a samotný výpočet mezihodnot.

### Podporované barevné formáty

Nástroj přijímá osm vstupních formátů. Každý je platný v parametrech **Text Colour** (`start/mid/end`), **Halo Colour** (`start/mid/end`) a **Shadow Colour**:

Tabulka 7 Typy akceptovaných barevných modelů a příklady jejich textového zápisu

Formát	Příklad vstupu	Poznámka
<b>RGB</b>	255,136,0	R, G, B v rozsahu 0–255; výchozí alpha = 255
<b>RGBA</b>	255,136,0,0.5	Alpha kanál v rozsahu 0–1 nebo 0–255
<b>CMYK</b>	0,0.47,1,0	Složky v rozsahu 0–1, 0–100 nebo se znakem %
<b>CMYKA</b>	0,0.47,1,0,0.9	Pátá volitelná složka definuje alpha kanál
<b>HEX 6-místný</b>	#FF8800 nebo FF8800	# je volitelný
<b>HEX 8-místný</b>	#FF8800CC	Poslední dvě cifry (AA) = alpha 0–255
<b>HSL</b>	30,100%,50%	Odstín 0–360°, sytost (S) a světllost (L) procentuálně
<b>HSLA</b>	30deg,100%,50%,0.8	Přijímá volitelnou příponu <code>deg</code> u hodnoty odstínu

## Detekce formátu

Funkce `detect_color_format` slouží výhradně ke klasifikaci vstupu – nevytváří žádnou barevnou hodnotu, pouze vrátí řetězcový identifikátor formátu. Logika detekce probíhá rozhodovacím stromem. Čtyřprvkový vstup (s kanálem průhlednosti – alpha) je nejprve testován jako **RGBA** (priorita, protože R,G,B v rozsahu 0–255 je velmi běžné), poté jako **HSLA** a teprve nakonec jako **CMYK**. Pomocná funkce `looks_like_hsl` hledá procentuální zápis složek S/L nebo příponu *deg* u odstínu.

## Parsování barevné hodnoty

Klíčová funkce `parse_color` přijímá řetězec a vrací normalizovaný seznam [R, G, B, A], kde všechny složky leží v rozsahu 0–255, pokrývající všechny formáty v jediném vstupním bodě. **HEX** je rozpoznán jako první na základě shody prvního tokenu se vzorem `^#?[0-9A-Fa-f]{6}([0-9A-Fa-f]{2})?$. Šestimístný hex dostane alpha 0xFF, osmimístný si nese vlastní:`

```
if re.match(r"^#?[0-9A-Fa-f]{6}([0-9A-Fa-f]{2})?$", first_token):
    hex_s = first_token
    if hex_s.startswith("#"):
        hex_s = hex_s[1:]
    rgb_hex = hex_s[0:6]
    alpha_hex = hex_s[6:8] if len(hex_s) == 8 else "FF"
    r = int(rgb_hex[0:2], 16)
    g = int(rgb_hex[2:4], 16)
    b = int(rgb_hex[4:6], 16)
    a = int(alpha_hex, 16)
    return [r, g, b, a]
```

**HSL / HSLA** využívá dvojici pomocných funkcí. `parse_hsl_angle` přijímá odstín v stupních a normalizuje příponu *deg*. `parse_hsl_component` zpracuje sytost a světlost v jakémkoli z podporovaných zápisů (*přímý float, 0–100 nebo procento*). Konverzi **HSL** → **RGB** zajišťuje funkce `hsl_to_rgb`, implementující standardní algoritmus přes pomocný koeficientu **C** ( $C=(1-|2L-1|) \times S$ ) a následnou lineární transformací v rámci šesti sektorů barevného kruhu.

**CMYK / CMYKA** používá funkci `parse_cmyk_value`, která přijímá floating-point hodnotu, celé číslo 0–100 i procentuální zápis. Hodnoty větší než 1 jsou automaticky převedeny na rozsah 0–1. Samotný převod **CMYK** → **RGB** je realizován standardním vzorcem v `cmyk_to_rgb`. Zpracování hodnoty alpha je sjednoceno funkcí `parse_alpha`, která rozlišuje zlomkový zápis (0–1) od přímého zápisu (0–255).

## Výběr interpolačního prostoru

Před spuštěním stylovacího cyklu nástroj detekuje, zda jsou **všechny** zadané barvy v CMYK/CMYKA formátu. Pokud ano, provede se interpolace v CMYK prostoru a teprve výstup je převeden do RGB pro CIM. Tím se eliminují nežádoucí přechody přes šedé tóny, které vznikají, když jsou interpolovány CMYK barvy po předchozím převodu do RGB:

```
formats = [detect_color_format(ct) for ct in color_texts_for_interpolation if ct]
use_cmyk_interpolation = all(f in ("CMYK", "CMYKA") for f in formats) and
len(set(formats)) == 1
```

Podmínka `len(set(formats)) == 1` zaručuje, že CMYK interpolace proběhne pouze tehdy, jsou-li **všechny** barvy ve stejném, čistě CMYK formátu – smíšení CMYK s jiným formátem by narušilo konzistenci výsledných barev.

## Interpolační funkce

Nástroj implementuje celkem čtyři interpolační funkce pokrývající kombinace dvou os (RGB/CMYK) a dvoubodové (min → max) či třibodové (min → mid → max) definice barevného rozsahu:

- *interpolate\_colors(start\_rgba, end\_rgba, n)* – lineární přechod v RGB prostoru mezi dvěma barvami.
- *interpolate\_colors\_with\_mid(start\_rgba, mid\_rgba, end\_rgba, n)* – třístupňový přechod v RGB přes střední bod. Cyklus je rozdělen na dva segmenty; střední bod je součástí prvního segmentu a v druhém je přeskočen, aby nedošlo k duplikaci.
- *interpolate\_colors\_cmyk(start\_cmyk, end\_cmyk, n)* – lineární přechod v CMYK prostoru s následným převodem každého kroku do RGB.
- *interpolate\_colors\_with\_mid\_cmyk(start\_cmyk, mid\_cmyk, end\_cmyk, n)* – třístupňový přechod v CMYK prostoru, strukturálně totožný s RGB variantou, ale pracující se čtyřprvkovými CMYK vektory.

## 5.10 Konfigurace Maplex Label Engine

ArcGIS Pro nabízí dva enginy pro umístění popisků: **standardní** a tzv. **Maplex**, který umožňuje pokročilé řízení zalamování, skládání řádků (*label stacking*) a umístění popisků mimo polygon. Nástroj explicitně nastavuje vlastnosti Maplex label enginu pro každou třídu popisků (label class) prostřednictvím objektu *maplexLabelPlacementProperties*, který je dostupný přes **CIM**.

Před jakýmkoli zápisem nástroj ověří, zda vrstva vůbec Maplex vlastnosti nese – jde o opatření pro případ, kdy by vrstva používala standardní engine nebo měl CIM objekt neúplnou strukturu. Objekt *labelStackingProperties* je dále ověřen pomocí *hasattr*, protože tato vnořená struktura nemusí být přítomna u všech verzí CIM schématu.

Klíčový parametr **Prevent Label Wrapping** ovládá celou skupinu Maplex vlastností. Pokud je aktivní parametr **Prevent Label Wrapping**, je zalamování potlačeno na všech dostupných úrovních:

```
if prevent_wrapping:
    mlpp.canStackLabel = False

    if stacking_props:
        stacking_props.maximumNumberOfLines = 1
        stacking_props.stackingSeparators = []
    mlpp.allowTextToWrap = False
    mlpp.maximumCharactersPerLine = 500
    arcpy.AddMessage(
        "Prevent wrapping enabled: canStackLabel=False, "
        "max lines=1, no separators"
    )
```

Nastavení pracuje na třech nezávislých vrstvách ochrany:

- *canStackLabel = False* – hlavní přepínač Maplex stacking mechanismu
- *stacking\_props.maximumNumberOfLines = 1* a *stackingSeparators = []* – limitem jednoho řádku a prázdným seznamem oddělovačů se zamezí i fallback dělení uvnitř stacking engine
- *allowTextToWrap = False* a *maximumCharactersPerLine = 500* – vypnutí automatického textového zalamování s vysokým limitem jako dodatečná pojistka

Funkce `create_default_separators` vrací dvojici `CIMMaplexStackingSeparator` objektů, které Maplex engine používá jako dovolené body dělení při skládání řádků. Ty jsou nově generovány, nikoli recyklovány, aby se předešlo sdílení referencí mezi třídami popisků.

Vlastnost `forcedSplit = False` znamená, že Maplex engine může label na daném znaku rozdělit, ale není k tomu nucen – konečné rozhodnutí závisí na optimalizaci umístění. Vlastnost `visible = True` zajistí, že oddělovač bude po zalamování viditelný (mezera nebo čárka zůstane na konci / začátku řádku).

Poslední Maplex vlastnost nastavovaná nástrojem je `canPlaceOutsidePolygon`, a to bezpodmínečně pro každou třídu. U číselných kartodiagramů nad polygonovou vrstvou může být popisek příliš velký na to, aby se vešel do plochy prvku – zejména u malých obcí nebo krajů při maximálních velikostech písma. Nastavení `True` umožní Maplex enginu umístit popisek vně plochy, čímž se předejde situaci, kdy by label nebyl zobrazen vůbec.

## 5.11 Optimalizace výkonu, ošetření chyb a diagnostika

Praktická použitelnost nástroje v prostředí ArcGIS Pro závisí nejen na správnosti výstupů, ale také na stabilitě při nestandardních vstupech, rychlosti zpracování rozsáhlých datových sad a srozumitelnosti zpětné vazby vůči uživateli. Tato kapitola shrnuje konkrétní techniky použité v toolboxu pro zajištění těchto vlastností.

### Vzorkování pro výpočet přirozených zlomů

Jenksova metoda přirozených zlomů (**Natural Breaks**) má časovou složitost  $O(n^2 \cdot k)$ , kde  $n$  je počet hodnot a  $k$  počet tříd. Pro plynulou odezvu nástroje bylo zavedeno adaptivní vzorkování: datové sady přesahující 3 000 hodnot jsou před výpočtem redukovány na náhodný výběr 3 000 prvků pomocí `random.sample`. Globální minimum a maximum celé datové sady jsou přitom vždy zachovány a přepsány do výsledných zlomů.

Přirozené rozložení vzorku zajišťuje, že tvarové charakteristiky distribuce jsou zachovány, přičemž výpočetní čas je omezen na přijatelnou mez bez ohledu na velikost vstupní vrstvy. Ovšem při valné většině použití tohoto nástroje se s takovýmto problémem uživatel nesetká, protože nástroj se při takovémto počtu prvků používat nedoporučuje.

### Výkonnostní optimalizace načítání fontů

Proces analýzy fontových souborů pomocí `fontTools.ttLib.TTFont` je limitován četností přístupů k diskovému úložišti a výpočetně náročným procesem deserializace binárních tabulek. Nástroj proto implementuje dvě protiopatření:

- **Parametr** `recalcBBoxes=False` zakazuje přepočítání bounding boxů glyfů při otevírání souboru. Protože nástroj čte výhradně metadata z tabulek `name` a `fvar`, je geometrická přesnost bounding boxů irelevantní a jejich výpočet by byl zbytečným výkonostním nákladem:

```
font = TTFont(font_path, recalcBBoxes=False)
```

- **„Kešování“ katalogu** v třídě `ToolValidator` zajišťuje, že seznam fontů a jejich stylů je sestavován nejvýše jednou za životnost dialogu. Výsledek je uchováván v `self._font_catalog` a `self._font_registry_names`; každé následující volání `_build_font_catalog` vrátí existující referenci bez nového prohledávání souborového systému.

## Hierarchické ošetření chyb

Nástroj rozlišuje tři úrovně závažnosti chybových stavů a pro každou volí odpovídající způsob reakce:

- **Fatální chyby** – situace, kdy nelze pokračovat ve vykonávání (chybí vstupní vrstva, pole neobsahuje hodnoty, neplatný barevný formát startovní barvy). V těchto případech je zavolána `arcpy.AddError` a funkce `main` je ukončena explicitním `return`.

Obálková funkce `get_parsed_color` zachytí výjimku z `parse_color`, přidá chybovou zprávu do Geoprocessing okna a vrátí `None`, aniž by výjimku šířila dál. Symetrická verze `get_parsed_color_cmyk` existuje pro CMYK vstup.

- **Napravitelné problémy** – stav, kdy je možné pokračovat, ale s jiným nastavením, než uživatel zamýšlel (neznámý styl písma, neúspěšné nastavení os variabilního fontu). Tyto stavy jsou reportovány přes `arcpy.AddWarning` a kód přechází na náhradní hodnotu:

```
arcpy.AddWarning(  
    f"Requested font style '{requested_style}' is not available for  
    '{font_family_cim}'. Using '{selected_style}' instead."  
)
```

Záložní logika pro výběr stylu sleduje přesně definovanou prioritu: přesná shoda → přibližná shoda podřetězcem → "Regular" → první dostupný styl → pevné "Regular". Tím je zaručeno, že nástroj vždy produkuje výstup i při nestandardních kombinacích parametrů.

- **Dílčí selhání v cyklu** – operace, které mohou selhat pouze pro část label classes (nastavení `offsetX/Y`, modifikace `lc.name`, aplikace variabilních os). Jsou obaleny izolovanými bloky `try/except`, které zachytí výjimku a pokračují s dalším prvkem.
- **Globální záchytná síť** celé funkce `main` je tvořena jediným blokem na nejvyšší úrovni:

```
except Exception as exc:  
    arcpy.AddError(f"An error occurred: {str(exc)}\n{traceback.format_exc()}")
```

Modul `traceback` zajistí, že do Geoprocessing okna je zapsán kompletní výpis zásobníku volání, nikoliv jen finální chybová zpráva. To výrazně zjednodušuje právě diagnostiku chyb.

## Správa prostředků fontových souborů

Každý otevřený fontový soubor musí být korektně uzavřen bez ohledu na to, zda zpracování proběhlo úspěšně. Vzor `try/finally` je použit konzistentně v každé funkci, která pracuje s `TTFont`. Vnitřní `try/except` v bloku `finally` chrání před hypotetickou výjimkou při samotném zavírání – například na poškozeném souboru – která by jinak maskovala původní chybu ze zpracovacího bloku.

## Diagnostické zprávy

Nástroj vydatně využívá `arcpy.AddMessage` pro průběžný reporting do Geoprocessing okna. Zprávy tvoří auditní stopu celého procesu (obrázek 16).

```

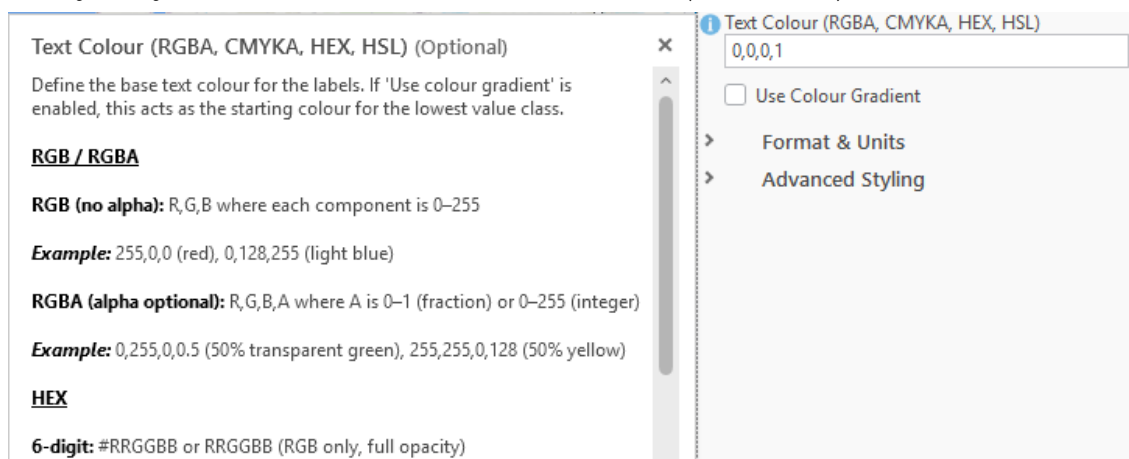
Start Time: Tuesday, April 14, 2026 12:56:45 PM
Loading parameters...
Color formats: RGB -> R,G,B (R,G,B 0-255); RGBA -> R,G,B,A (R,G,B 0-255, A 0-1 or 0-255); CMYK -> C,M,Y,K (C,M,Y,K 0-1, 0-100, or 0%-100%);
CMYKA -> C,M,Y,K,A (C,M,Y,K 0-1, 0-100, or 0%-100%, A 0-1 or 0-255); HSL/HSLA -> H,S,L or H,S,L,A (H 0-360, S/L 0-1/0-100/0%-100%).
Generated class breaks:
- 41.4 to 42.4
- 42.4 to 42.9
- 42.9 to 43.2
- 43.2 to 43.6
- 43.6 to 43.8
Removed 5 old Label classes.
Category 0: $feature.Prumerny_vek >= 41.400001525878906 && $feature.Prumerny_vek < 42.400001525878906
Category 1: $feature.Prumerny_vek >= 42.400001525878906 && $feature.Prumerny_vek < 42.900001525878906
Category 2: $feature.Prumerny_vek >= 42.900001525878906 && $feature.Prumerny_vek < 43.20000076293945
Category 3: $feature.Prumerny_vek >= 43.20000076293945 && $feature.Prumerny_vek < 43.599998474121094
Category 4: $feature.Prumerny_vek >= 43.599998474121094 && $feature.Prumerny_vek <= 43.79999923706055
Shadow disabled (no color or zero offsets)
Prevent wrapping enabled: canStackLabel=False, max lines=1, no separators
Shadow disabled (no color or zero offsets)
Prevent wrapping enabled: canStackLabel=False, max lines=1, no separators
Shadow disabled (no color or zero offsets)
Prevent wrapping enabled: canStackLabel=False, max lines=1, no separators
Shadow disabled (no color or zero offsets)
Prevent wrapping enabled: canStackLabel=False, max lines=1, no separators
Shadow disabled (no color or zero offsets)
Prevent wrapping enabled: canStackLabel=False, max lines=1, no separators
Labels were created and styled successfully.
Succeeded at Tuesday, April 14, 2026 12:56:49 PM (Elapsed Time: 3.53 seconds)

```

Obr. 16 Geoprocessing okno ArcGIS Pro s výpisem diagnostických zpráv z jednoho spuštění nástroje – seřazené fáze od načítání parametrů po závěrečné potvrzení.

## 5.12 Textové nápovědy, metadata a formální náležitosti

Důležitou součástí vývoje každého nástroje v prostředí ArcGIS Pro je nejen samotná funkčnost, ale také pečlivé vyplnění nápovědy, metadat a dalších formálních náležitostí. Tyto prvky významně přispívají k uživatelské přívětivosti a srozumitelnosti práce s toolboxem. Každý parametr v .atbx toolboxu je opatřen polem **Description**, které slouží k vysvětlení jeho účelu, možných hodnot a případných omezení. Tyto popisy jsou zobrazovány přímo v dialogovém okně nástroje a pomáhají uživateli správně nastavit všechny volby bez nutnosti nahlížet do dokumentace (obrázek 17).



Obr. 17 Příkladová ukázka nápovědy k parametrům nástroje

Součástí toolboxu jsou také metadata na úrovni celého nástroje, která zahrnují jméno autora, kontaktní údaje, instituci, datum vytvoření a stručný popis účelu nástroje a funkčnosti jeho parametrů. Formální náležitosti zahrnují také kontrolu konzistence názvů parametrů, jejich pořadí, správné nastavení výchozích hodnot a kategorií. Všechny tyto aspekty byly průběžně ověřovány a testovány, aby byl výsledný toolbox nejen funkční, ale i připravený pro další sdílení a následné budoucí použití.

## 5.13 Uživatelské testování nástroje

Před závěrečnou fází vývoje bylo provedeno uživatelské testování toolboxu, jehož cílem bylo ověřit srozumitelnost ovládání, přehlednost parametrů a praktickou funkčnost nástroje v prostředí ArcGIS Pro pro jeho cílovou skupinu, kterou tvoří především kartografové připravující mapové vizualizace v prostředí GIS.

Testování proběhlo 5. března 2026 v rámci předmětu **VIZUL** a zúčastnilo se jej **14 studentů** 1. ročníku navazujícího magisterského studia **katedry geoinformatiky PřF UPOL**. Účastníci pracovali s nástrojem na základě zadaných úloh a následně poskytli zpětnou vazbu. Ta vedla jak k identifikaci některých technických problémů, tak k finalizaci uživatelského rozhraní:

- Byla odhalena nefunkčnost nástroje ve verzích ArcGIS Pro starších než **3.3** (obrázek 18). Následnou analýzou bylo zjištěno, že příčinou byla absence funkce vytvoření třídy popisků (*CreateLabelClass*) v knihovně **ArcPy**, která byla doplněna až v ArcGIS Pro verzi 3.3 (ArcGIS Ideas Blog, 2024).
- Ze dvou testovaných verzí UI toolboxu (s pěti sekcemi (Category) a bez nich) byla jako přehlednější vybrána varianta s pěti sekcemi, viz. kapitola 5.3.
- Byly odhaleny problémy se zalamováním textu v generovaných popisích (**parametr Prevent label Wrapping**).
- Byly zjištěny nedostatky v nastavení omezení počtu desetinných míst pro data s více než deseti desetinnými místy (**parametr Decimal Places After Scale**).

```
Datum 05.03.2026
Verze ArcGIS Pro: 3.1.1
VÝPIS KONZOLE:
Proportional Numbers Label Generator
=====
Messages

Start Time: čtvrtek 5. března 2026 9:53:27
Loading parameters...
Color formats: RGBA -> R,G,B,A (R,G,B 0-255, A 0-1 or 0-255); CMYKA -> C,M,Y,K,A (C,M,Y,K 0-1 or
0-100, A 0-1 or 0-255); HEX -> RRGGBB or RRGGBBAA (optionally with leading '#').
Generated class breaks:
- 1392.0 to 2952.3333333333333
- 2952.3333333333333 to 4512.6666666666666
- 4512.6666666666666 to 6073.0
Removed 0 old label classes.
Category 0: $feature.SNATKY >= 1392.0 && $feature.SNATKY < 2952.3333333333333
An error occurred: 'Layer' object has no attribute 'createLabelClass'
Traceback (most recent call last):
  File
"C:\Users\████████████████████\ProportionalNumbersGenerator.
atbx#generatorlabelu4v3v26_ProportionalNumbersGenerator.py", line 467, in main
AttributeError: 'Layer' object has no attribute 'createLabelClass'

Failed script Proportional Numbers Label Generator ...
Failed to execute (generatorlabelu4v3v26).
Failed at čtvrtek 5. března 2026 9:53:29 (Elapsed Time: 1,75 seconds)
```

Obr. 18 Výstup z testování nástroje studentem: konzolový log s chybou createLabelClass

## 6 VÝSLEDKY

Hlavním výsledkem diplomové práce je funkční komplexní nástroj ve formátu .atbx pro automatizovanou tvorbu metody číselných kartodiagramů v prostředí ArcGIS Pro. V souladu se stanovenými cíli práce došlo k algoritmizaci procesu, který byl dříve závislý na značném množství manuálních úkonů, čímž byla výrazně zvýšena efektivita tvorby této metody vizualizace tematických map.

Výsledky práce jsou dále strukturovány do dvou částí:

1. **Technický výstup:** implementovaný .atbx toolbox pro tvorbu metody číselných kartodiagramů v ArcGIS Pro skrze generaci popisků (labelů).
2. **Ukázkové mapové výstupy:** soubor mapových děl vytvořených na základě různých datových sad a pro různá zájmová území, jejichž cílem bylo demonstrovat univerzálnost a praktickou užitečnost nástroje při tvorbě metody číselných kartodiagramů v prostředí ArcGIS Pro.

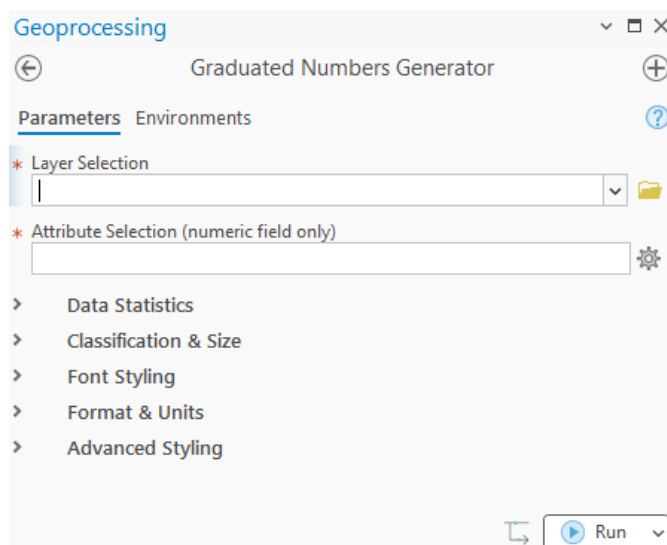
### 6.1 Toolbox pro tvorbu číselných kartodiagramů

Výsledný nástroj integruje klasifikační, typografické, barevné a poziční parametry do jednoho pracovního postupu. Uživatel tak může v jediném rozhraní nastavovat způsob třídění hodnot, rozsah velikostí písma, práci s desetinnými místy, formát čísel, barevnost popisků i další prvky ovlivňující čitelnost výsledné mapy.

Přínos nástroje spočívá především v těchto bodech:

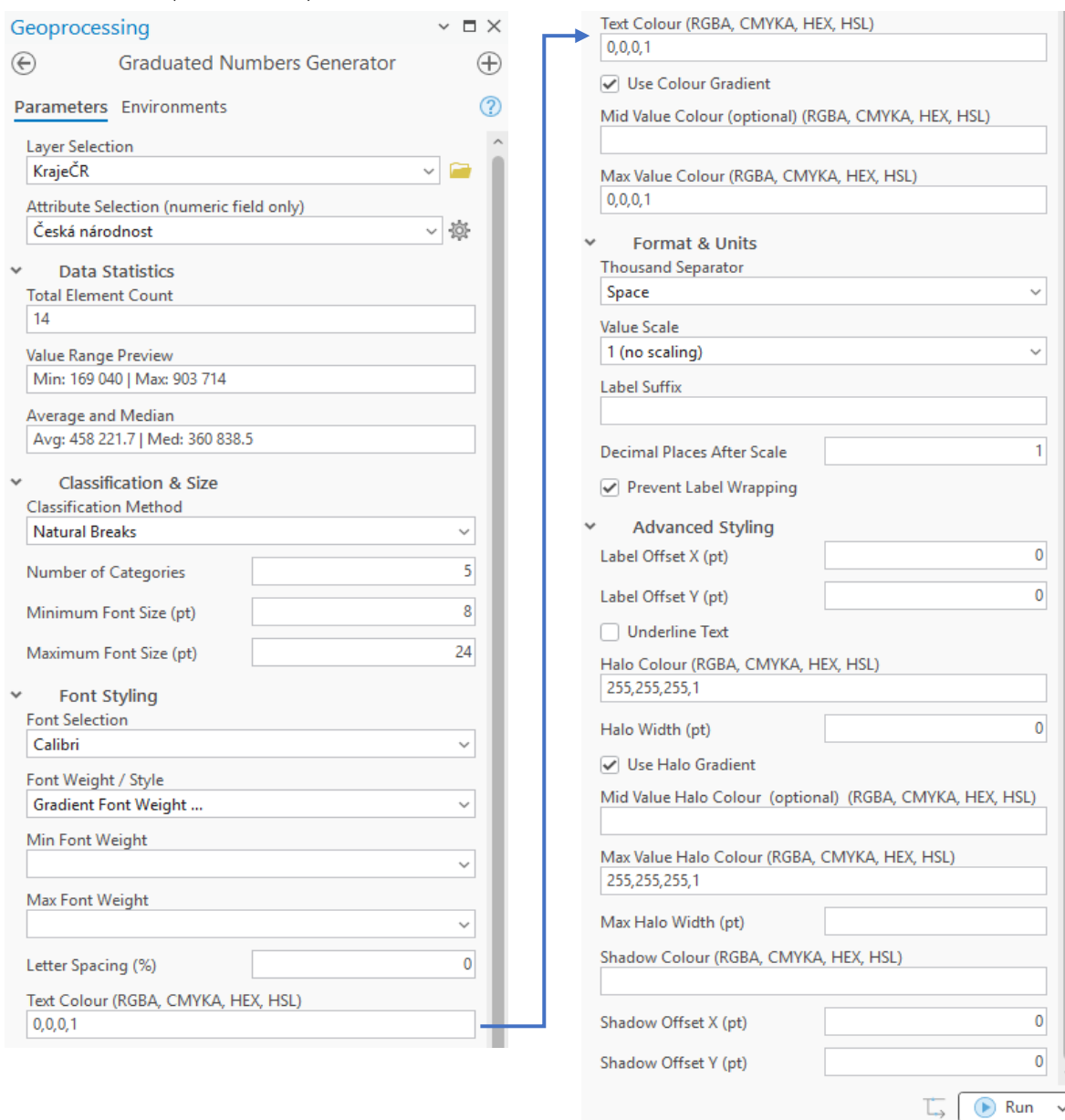
1. Redukce manuálních kroků při nastavování popisků a jejich grafiky.
2. Vyšší konzistence mapových výstupů díky jednotné logice parametrů.
3. Rychlejší iterace návrhu mapy při testování různých variant vizualizace.
4. Lepší kontrola nad kvalitou výsledku u dat s různým rozsahem hodnot.

Vytvořený nástroj byl implementován ve formátu ArcGIS Toolbox (**.atbx**) pojmenovaný "**Graduated Numbers Generator**". Po otevření toolboxu je viditelný jediný Script Tool se stejným názvem, připravený ke spuštění. Veškeré parametry s výjimkou prvních dvou vstupních polí jsou členěny do pěti tematických sekcí: *Data Statistics*, *Classification & Size*, *Font Styling*, *Format & Units* a *Advanced Styling*. Tato struktura umožňuje přehlednou orientaci v nástroji a skrytí parametrů, které uživatel aktuálně nepotřebuje (obrázek 19).



Obr. 19 Finální podoba zabaleného uživatelského rozhraní nástroje po inicializaci

Po rozbalení všech sekcí je přístupná kompletní sada 36 parametrů. Sekce *Data Statistics* zobrazuje statistický náhled vstupní vrstvy – celkový počet prvků, rozsah hodnot a průměr s mediánem – generovaný dynamicky na základě zvolené vrstvy a atributu. Sekce *Classification & Size* nabízí volbu klasifikační metody, počtu kategorií a rozsahu velikosti písma. *Font Styling* zahrnuje výběr rodiny písma, tloušťky/stylu (včetně možnosti gradientu), mezery mezi znaky a barvy textu. *Format & Units* obsahuje nastavení oddělovače tisíců, škálování hodnot, textové přípony a počtu desetinných míst. Sekce *Advanced Styling* zpřístupňuje odsazení popisku, podtržení, nastavení halo efektu a stínu textu (obrázek 20).



Obr. 20 Ukázka kompletně rozbaleného nástroje s výchozími hodnotami parametrů, aktivovanými gradienty a zvolenou vstupní vrstvou a atributem; z důvodu rozsahu je ukázka rozdělena do dvou navazujících částí téhož dialogového okna.

## 6.2 Ukázkové mapové výstupy

Funkčnost toolboxu byla ověřena na **sérii ukázkových mapových výstupů**. Tyto výstupy byly zvoleny tak, aby pokryly různé scénáře v ArcGIS Pro, zahrnující rozdílné velikosti a tvary území, charakter a řád číselných hodnot zvolených kvantitativních dat pro ukázkou využití parametrů a možností vytvořeného nástroje. Přesto byl u celé série map zachován jednotný vizuální styl, čímž bylo také demonstrováno, že nástroj umožňuje vytvářet vizuálně konzistentní mapové díla i při využití odlišných funkcí toolboxu.

Níže jsou uvedeny jednotlivé ukázkové mapy, pro které je stručně popsána použitá konfigurace nástroje. Samotné mapové výstupy jsou zařazeny v přílohách práce. U každé mapy jsou shrnuty klíčové parametry nastavení, aby bylo zřejmé, jaké volby vedly k výsledné podobě výstupu:

### **Počet hasičských sborů v Německých spolkových zemích – (příloha 4)**

Na této mapě byl předvedený výsledek téměř výchozího nastavení toolboxu, jediná úprava spočívala ve změně velikosti písma úměrná měřítku, dále změna rodiny fontu a barvy alfanumerických znaků pro grafickou konzistenci napříč celou sérií ukázek.

### **Average monthly wages in polish voivodeships in 2020 (PLN) – (příloha 5)**

Druhá mapa série zobrazuje průměrnou měsíční mzdu v jednotlivých polských vojvodstvích za rok 2020. Vzhledem k tomu, že uživatelské rozhraní i veškerá dokumentace toolboxu jsou vytvořeny v angličtině, byl tento výstup záměrně zpracován v anglickém jazyce. Cílem bylo demonstrovat využití nástroje pro mezinárodní publikaci, což vyžaduje anglosaský standard zápisu číselných údajů (čárky mezi řády a desetinná tečka). Dále bylo užito klasifikační metody „*Equal Interval*“ s pěti kategoriemi popisků, barevné gradace textu od bílé po žlutou barvu, užití Halo efektu spolu se stínem textu pro zvýraznění alfanumerických hodnot na mapovém poli pro ukázkou použití těchto parametrů.

### **Počet letišť s ICAO kódem v zemích evropského regionu – (příloha 6)**

Třetí výstup vizualizuje absolutní počty letišť s přiděleným ICAO kódem v rámci evropských států. Geometrie evropského kontinentu je specifická velkými rozdíly v rozloze států, což v oblastech s vysokou koncentrací menších celků (například na Balkánském poloostrově či ve střední Evropě) může vést k překrývání popisků. Toolbox byl v tomto případě využit k rychlému a matematicky přesnému vytvoření gradované velikosti písma pomocí metody „*Proportional (Unclassed) ...*“ a to včetně finální grafického podoby (např. užití Halo efektu a reverzní gradace tloušťky fontu (od Bold po Regular) pro zlepšení čitelnosti menších hodnot) Takto vygenerované popisky byly následně převedeny na grafické objekty (nativní ArcGIS Pro funkce **Convert Labels to Graphics**). Tento postup umožnil po konverzi z popisků na grafické objekty ruční posun vybraných čísel mimo jejich původní těžiště, čímž byla zajištěna lepší čitelnost bez překryvů, aniž by došlo ke ztrátě přesných velikostní alfanumerických znaků definovaných nástrojem.

## Počet žen a mužů v krajích ČR v roce 2021 (v tisících) – (příloha 7)

Poslední mapa série se věnuje demografické struktuře krajů České republiky, konkrétně srovnání počtu mužů a žen v roce 2021. Tento výstup byl zvolen k demonstraci schopnosti nástroje vizualizovat více číselných hodnot pro jeden územní prvek současně. Toho bylo dosaženo dvojitým spuštěním toolboxu pro duplikované vrstvy krajů. Aby byla mapa okamžitě čitelná bez nutnosti legendy, bylo využito barevného odlišení hodnot – růžová barva pro ženy a modrá barva pro muže korelující s grafikou názvu mapy. Tato konfigurace ukazuje, že nástroj umožňuje efektivně pracovat s více vrstvami nebo třídami popisků v rámci jedné mapové kompozice. Pro klasifikační metodu byla využita možnost „*Manual ...*“, kde byly hraniční hodnoty tříd ručně zadány postupně po sto tisících. Aby byla zachována čistota a přehlednost mapové kompozice, byla data uváděna s hodnotovým měřítkem (**parametr Value Scale**) nastavený v tisících a zároveň počet desetinných míst (**parametr Decimal Places After Scale**) nastaven na nulovou hodnotu, aby se zamezilo nevhodnému použití desetinných míst u počtu osob. Mapa tak uzavírá celou sérii jako ukázka pokročilejšího využití toolboxu.

## 7 DISKUZE

Předkládaná práce úspěšně automatizovala tvorbu číselných kartodiagramů v ArcGIS Pro prostřednictvím nástroje *Graduated Numbers Generator*. Zatímco open-source software QGIS disponuje pro tvorbu gradovaných číselných kartodiagramů nativní a relativně snadno, uživatelé platformy ArcGIS Pro byli dosud odkázáni na manuální a vysoce neefektivní postupy. Vyvinutý toolbox tento nedostatek odstraňuje a zároveň respektuje kartografické zásady konstrukce stupnic (Kaňok, 1999 a Kaňok, Voženílek 2008). Práce tak logicky rozšiřuje dosavadní snahy o automatizaci kartografických postupů na UPOL (Pospíšil, 2024; Šramo, 2023). Hodnotné metodické rady a připomínky vedoucího práce Mgr. Radka Barvíře, Ph.D. a doc. Ing. Zdeny Dobešové, Ph.D. pomohly upřesnit koncepční rámec vyvíjeného nástroje a ukázkových map, a tím přispěly k lepší formální i strukturní integritě celé práce.

### **Obhajoba zvolených metod a technologická rozhodnutí**

Při koncepčním návrhu se jako alternativa nabízel vývoj plnohodnotného doplňku (Add-In) (platforma .NET). Tato varianta by sice poskytla mnohem širší možnosti tvorby interaktivního uživatelského rozhraní, avšak byla zamítnuta z důvodu zachování jednoduchosti a kompatibility. Zvolený formát geoprocessingového toolboxu (.atbx) sice vývojáře silně limituje v možnostech grafického přizpůsobení dialogového okna, ale zajišťuje maximální přenositelnost a jednoduchost instalace.

Zásadním a úmyslným rozhodnutím byl rovněž nedestruktivní přístup k datům. Nástroj záměrně nepřepisuje žádné hodnoty ve zdrojových vrstvách, ani negeneruje nové pomocné atributové sloupce. Všechny operace, včetně aplikace statistických metod a formátování (např. oddělovače tisíců), probíhají výhradně dynamicky v paměti pomocí výrazů jazyka Arcade a objektů CIM. Tento postup výrazně šetří výpočetní čas, eliminuje riziko porušení datové integrity a udržuje databázi bez zásahů, což umožňuje využití nástroje také na vzdálená data publikovaná prostřednictvím webových služeb (např. z ArcGIS Online), u kterých uživatel zpravidla nedisponuje oprávněním k zápisu nebo k modifikaci dat.

### **Technologické problémy a limity řešení**

Vývoj v prostředí ArcGIS Pro přinesl specifické překážky. Největším technickým problémem, byla absence nativního API pro čtení systémových fontů. ArcGIS Pro vývojářům do této doby neposkytuje seznam nainstalovaných písem a jejich řezů. Skript proto musel být vybaven velmi nestandardní logikou, která načítá data přímo z registrů Windows. Rozdíly mezi klasickými a variabilními fonty navíc vyžadovaly komplexní algoritmy pro interpolaci tloušťky písma. Toto řešení je sice funkční, ale činí nástroj závislým na struktuře operačního systému Windows. Dalším objektivním limitem je výkonost překreslovacího enginu Maplex; při požadavku na vygenerování extrémního počtu tříd může dojít k pádům softwaru, čemuž se nástroj snaží předcházet preventivním varováním uživatele.

## Nevyřešené problémy a náměty pro další rozvoj

Ačkoliv je nástroj plně provozuschopný, praxe odhalila prostor pro jeho další vylepšení, která se v rámci této práce již nerealizovala. Jedním z nedostatků aktuální verze je absence ošetření nulových (0) hodnot. V současnosti nástroj tyto záznamy automaticky zpracuje a vykreslí do mapy, což může vytvářet nežádoucí vizuální zátěž. V budoucím vývoji by mělo být implementováno rozšíření o volitelnou podmínku, která zápis těchto nulových hodnot zcela potlačí. Tento problém však není u prvků s prázdnou, nevyplněnou hodnotou, které se do počtu prvků a celého fungování nástroje automaticky nezařazují. Pozornost by měla být v budoucnu věnována také vývoji robustnějšího vyhledávání nainstalovaných fontů a jejich dostupných řezů. Současný algoritmus závislý na registrech Windows by bylo vhodné optimalizovat tak, aby dokázal stabilněji zpracovávat jejich vyhledávání.

Konečné zhodnocení silných a slabých stránek toolboxu, včetně identifikace příležitostí pro jeho rozvoj a možných rizik, nabízí uvedená SWOT analýza:

- **Silné stránky:** Zásadní úspora času, nezasahuje do vstupních dat, podpora gradace řezů/tloušťky fontů a přímý výpočet barevných gradientů v CMYK/RGB.
- **Slabé stránky:** Nepřizpůsobitelné grafické rozhraní .atbx toolboxu, závislost na čtení registrů operačního systému Windows.
- **Příležitosti:** Nasazení nástroje ve výuce vizualizačních metod tematické kartografie, diseminace pro širší mezinárodní komunitu uživatelů ArcGIS Pro.
- **Hrozby:** Změny v budoucích aktualizacích ArcGIS Pro, které mohou narušit funkčnost nástroje, nebo přidáním nových funkcí, které by nástroj mohly nahradit.

Závěrem diskuze lze konstatovat, že ačkoliv vývoj v uzavřeném ekosystému softwaru ArcGIS Pro nevyhnutelně přinesl řadu technologických kompromisů, přínosy vytvořeného řešení tyto limity prokazatelně převyšují. Nástroj *Graduated Numbers Generator* úspěšně převedl teoretické principy metody tematické kartografie do podoby uživatelsky přívětivé aplikace. Oproti neefektivním manuálním postupům, které kartografy od využití číselných kartodiagramů v tomto softwaru často odrazovaly, nyní existuje rychlejší automatizovaná alternativa. Nástroj má tak reálný potenciál přispět k rozšíření použití metody číselných kartodiagramů a její širší i metodicky korektnější aplikaci v praxi.

## 8 ZÁVĚR

Cílem této diplomové práce bylo navrhnout a implementovat nástroj umožňující **automatizaci tvorby číselných kartodiagramů** v prostředí ArcGIS Pro. Práce reagovala na skutečnost, že tvorba této metody tematických map je v současné době prováděna skrze zdlouhavý manuální proces, vyžadující opakované nastavování tříd popisků (label classes) a jejich grafických vlastností. Hlavním záměrem práce proto bylo navrhnout postup, který by tento proces zjednodušil, zefektivnil a umožnil jeho částečnou automatizaci s využitím skriptovacích možností prostředí ArcGIS Pro.

V první části práce byla pozornost věnována teoretickému vymezení problematiky tematické kartografie se zaměřením na kartodiagramy a jejich využití při vizualizaci kvantitativních dat. Byly popsány základní principy tvorby kartodiagramů a jejich klasifikace jak v českém, tak i zahraničním prostředí. Dále byla provedena rešerše dostupných metod a postupů pro **tvorbu metody číselných kartodiagramů** v prostředí současných GIS softwarů. Tato analýza ukázala, že i přes pokročilé možnosti moderních GIS systémů zůstává tvorba číselných kartodiagramů v prostředí ArcGIS Pro do značné míry závislá na značném počtu manuálních kroků, což představuje prostor pro automatizaci.

Na základě získaných poznatků byl v další části práce navržen postup automatizace tvorby číselných kartodiagramů. Důležitým krokem bylo definování algoritmů pro tvorbu intervalů hodnot a jejich následnou vizuální reprezentaci prostřednictvím tříd popisků. Návrh řešení vycházel z možností programovacího jazyka **Python**, knihovny **ArcPy** a **CIM API**, které umožňují automatizovanou práci s atributovými daty, mapovými vrstvami a jejich vlastnostmi. Součástí návrhu bylo rovněž stanovení způsobu nastavování vizuálních charakteristik popisků, jako je gradace velikosti, barvy a typografické vlastnosti.

Výsledkem praktické části práce je vytvoření funkčního nástroje implementovaného ve formátu **geoprocessingového toolboxu formátu .atbx** pro prostředí ArcGIS Pro. Tento nástroj umožňuje automatizovanou tvorbu číselných kartodiagramů na základě vstupních atributových dat a uživatelem definovaných parametrů. Uživatel má možnost zadat potřebné vstupy, jako jsou hodnoty atributů, způsob intervalového třídění nebo parametry grafické vizualizace, přičemž nástroj následně automaticky vytvoří odpovídající třídy popisků a nastaví jejich vizuální vlastnosti. Tím je výrazně omezen podíl manuální práce a zjednodušen celý proces tvorby mapového výstupu. Funkčnost navrženého řešení byla ověřena prostřednictvím testování, která umožnila posoudit jeho praktickou použitelnost. Testování prokázalo, že využití vytvořeného nástroje vede k výraznému snížení časové náročnosti tvorby číselných kartodiagramů ve srovnání s tradičním manuálním postupem. Zároveň bylo ověřeno, že výsledné mapové výstupy splňují požadavky na přehlednost a kartografickou správnost. Automatizace procesu navíc přispívá k vyšší konzistenci výsledků a omezuje možnost vzniku chyb způsobených ručním nastavováním jednotlivých parametrů.

Z výsledků vyplývá, že stanovený cíl diplomové práce byl splněn. Byl navržen a implementován nástroj umožňující automatizovanou tvorbu číselných kartodiagramů v prostředí ArcGIS Pro, který představuje prakticky využitelné řešení pro práci s tematickými mapami. Práce tak přispívá k rozvoji automatizace kartografických postupů a rozšiřuje možnosti využití skriptování v prostředí GIS.

Hlavním přínosem práce je zejména vytvoření nástroje, který **zjednodušuje a zpřístupňuje proces tvorby metody číselných kartodiagramů** a umožňuje jejich efektivnější využití v kartografické praxi. Přínosem je rovněž systematické zpracování problematiky tvorby této metody tematické kartografie a návrh metodického postupu, který může sloužit jako základ pro další vývoj podobných nástrojů. Vytvořené řešení může být využito nejen v odborné praxi, ale také ve výuce kartografie a geoinformatiky, kde může sloužit jako podpora při ukázce tvorby vizualizačních metod tematické kartografie. Navržený nástroj představuje otevřený základ, který je možné dále rozšiřovat o další funkcionality. Perspektivním směrem dalšího vývoje je například přidání dalších volitelných parametrů, klasifikačních metod dat nebo rozšíření možností grafického nastavení výstupů.

Diplomová práce přináší funkční řešení konkrétního kartografického problému a současně ukázala možnosti využití automatizace v prostředí ArcGIS Pro při tvorbě tematických map. Výsledky práce potvrzují význam skriptování jako nástroje pro zvyšování efektivity postupů a tvorby kartografických výstupů, a tím představují praktický příspěvek k rozvoji moderních metod vizualizace prostorových dat.

## POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

ADOBE, 2025. PostScript Type 1 fonts end of support. Online. Adobe. Dostupné z: <https://helpx.adobe.com/fonts/kb/postscript-type-1-fonts-end-of-support.html>. [cit. 2026-04-09].

ARTMANN, Benno, 1999. Euclid Book VIII: Numbers in Continued Proportion, the Geometry of Numbers. Online. In: *Euclid—The Creation of Mathematics*. New York, NY: Springer New York, s. 193-201. ISBN 9781461271345. Dostupné z: [https://doi.org/10.1007/978-1-4612-1412-0\\_19](https://doi.org/10.1007/978-1-4612-1412-0_19). [cit. 2026-04-23].

BARVÍŘ, Radek; HOLUB, Martin a VONDRÁKOVÁ, Alena, 2025. Proportional Symbol Maps: Value-Scale Types, Online Value-Scale Generator and User Perspectives: Value-Scale Types, Online Value-Scale Generator and User Perspectives. Online. *ISPRS International Journal of Geo-Information*. Roč. 14, č. 9, s. 340. ISSN 2220-9964. Dostupné z: <https://doi.org/10.3390/ijgi14090340>. [cit. 2026-02-05].

BREWER, Cynthia a CAMPBELL, Andrew J., 1998. Beyond Graduated Circles: Varied Point Symbols for Representing Quantitative Data on Maps. Online. *Cartographic Perspectives*. Č. 29, s. 6-25. ISSN 1048-9053. Dostupné z: <https://doi.org/10.14714/CP29.672>. [cit. 2025-06-20].

BURDEN, Richard L. a FAIRES, J. Douglas, 2011. *Numerical analysis*. 9th ed. Boston: Brooks/Cole. ISBN 978-0-538-73351-9. [cit. 2026-03-23].

BURKE, John, 2023. Why and how to use Google Colab. Online. In: *TECHTARGET*. TechTarget. Dostupné z: <https://www.techtarget.com/searchenterpriseai/tutorial/Why-and-how-to-use-Google-Colab>. [cit. 2025-10-05].

DENT, Borden D.; TORGUSON, Jeffrey a HODLER, T. W., 2009. *Cartography: thematic map design*. 6th ed. Boston, Mass.: McGraw-Hill Higher Education. ISBN 978-0-07-294382-5. [cit. 2025-06-18].

ELKINS, Rob, 2010. What is ArcPy? Online. ESRI. ArcGIS Blog. Dostupné z: <https://pro.arcgis.com/en/pro-app/3.4/arcpy/get-started/what-is-arcpy-.htm>. [cit. 2025-08-20].

ELKINS, Rob, 2015. ArcGIS Pro – Reinventing Desktop GIS. Online. ESRI. Jan 27, 2015. Dostupné z: <https://www.esri.com/arcgis-blog/products/3d-gis/3d-gis/arcgis-pro-reinventing-desktop-gis>. [cit. 2025-08-20].

ESRI, 2025a. ArcGIS Pro: A quick tour of ArcPy. Online. ESRI. ArcGIS Pro. Dostupné z: <https://pro.arcgis.com/en/pro-app/latest/arcpy/get-started/a-quick-tour-of-arcpy.htm>. [cit. 2025-10-07].

ESRI, 2025b. ArcGIS Pro: license levels. Online. ESRI. ArcGIS Pro. Dostupné z: <https://pro.arcgis.com/en/pro-app/latest/get-started/license-levels.htm>. [cit. 2025-08-20].

ESRI, 2025c. CIM symbol. Online. ESRI. Esri Developer. Dostupné z: <https://developers.arcgis.com/documentation/glossary/cim-symbol/>. [cit. 2025-09-26].

ESRI, 2026a. Data classification methods. Online. ESRI. ArcGIS Pro. Dostupné z: <https://pro.arcgis.com/en/pro-app/latest/help/mapping/layer-properties/data-classification-methods.htm>. [cit. 2026-04-22].

ESRI, 2026b. Quantile. Online. ESRI. GIS Dictionary. Dostupné z: <https://support.esri.com/en-us/gis-dictionary/quantile>. [cit. 2026-04-23].

GENERAL PYTHON FAQ, 2001. Online. PYTHON SOFTWARE FOUNDATION. Python documentation. 2026. Dostupné z: <https://docs.python.org/3/faq/general.html>. [cit. 2026-04-24].

GENEVA: INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2019. ISO/IEC 14496-22. Online. Dostupné z: <https://www.iso.org/standard/74461.html>. [cit. 2026-04-09].

GOOGLE OPEN SOURCE BLOG, 2016. Introducing OpenType Font Variations. Online. Dostupné z: <https://opensource.googleblog.com/2016/09/introducing-opentype-font-variations.html>. [cit. 2026-04-09].

JENKS, George F. The Data Model Concept in Statistical Mapping. International Yearbook of Cartography 7. 1967, 186–190 [cit. 2026-03-23].

KAŇOK, Jaromír, 1999. Tematická kartografie. Ostrava: Ostravská univerzita. ISBN 80-7042-781-7. [cit. 2025-07-05].

KAŇOK, Jaromír a VOŽENÍLEK, Vít, 2008. Chyby v mapách - Stupnice. Online. GeoBusiness. Roč. 2008, č. 6+7/2008, s. 28-32. Dostupné z: <https://issuu.com/springwinter/docs/namece9294>. [cit. 2025-09-15].

KLADIVO, Petr, 2013. Základy statistiky. Studijní opory. Olomouc: Univerzita Palackého v Olomouci. ISBN 978-80-244-3841-2. Dostupné také z: <http://geography.upol.cz/soubory/studium/e-ucebnice/978-80-244-3842-9.pdf>. [cit. 2026-3-23].

KORYCKA-SKORUPA, Jolanta a GOŁĘBIEWSKA, Izabela, 2020. Numbers on Thematic Maps: Helpful Simplicity or Too Raw to Be Useful for Map Reading? Online. ISPRS International Journal of Geo-Information. Roč. 9, č. 7. ISSN 2220-9964. Dostupné z: <https://doi.org/10.3390/ijgi9070415>. [cit. 2025-06-19].

MICROSOFT, 2020. TrueType overviewOnline. In: MICROSOFT. Learn. Dostupné z: <https://learn.microsoft.com/en-us/typography/truetype/>. [cit. 2026-04-26].

MICROSOFT, 2024. OpenType® Specification Version 1.9.1. Online. In: MICROSOFT. Dostupné z: <https://learn.microsoft.com/en-us/typography/opentype/spec/>. [cit. 2026-04-09].

MIKLÍN, Jan; DUŠEK, Radek; KRTIČKA, Luděk a KALÁB, Oto, 2018. Tvorba map. Ostrava: Ostravská univerzita. ISBN 978-80-7599-017-4. [cit. 2025-06-15].

NĚTEK, Rostislav; BURIAN, Jaroslav; PASZTO, Vít a BARVÍŘ, Radek, 2020. Fenomén brain drain města Olomouce – Odliv lidského kapitálu z Olomouce do českých obcí v letech 2001-2020: Specializovaná mapa. Online. In: UNIVERZITA PALACKÉHO V OLOMOUCI. Dostupné z: <https://radiat.cz/bd.pdf>. [cit. 2026-04-22].

PEEBLES, Ian, 2020. Exprodat: Four Reasons to Jump into ArcGIS Pro. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation. Dostupné z: <https://www.exprodat.com/blog/four-reasons-to-jump-into-arcgis-pro/>. [cit. 2025-08-20].

POSPÍŠIL, Jakub, 2024. Automatizace zobrazení turistických tras na mapách. Diplomová práce (Mgr.). Olomouc: UNIVERZITA PALACKÉHO V OLOMOUCI. Přírodovědecká fakulta. [cit. 2025-07-01].

ROBINSON, Arthur Howard; SALE, Randall D. a MORRISON, Joel L., 1978. Elements of Cartography. III. Title. John Wiley. ISBN 0-471-01781-7. [cit. 2025-06-18].

SHORTRIDGE, Barbara G. a WELCH, Robert B., 2013. The Effect of Stimulus Redundancy on the Discrimination of Town Size on Maps. Online. The American Cartographer. Roč. 9, č. 1, s. 69-78. ISSN 0094-1689. Dostupné z: <https://doi.org/10.1559/152304082783948277>. [cit. 2025-06-20].

SLOCUM, Terry A.; MCMASTER, Robert B; KESSLER, Fritz C. a HOWARD, Hugh.H, 2022. Thematic Cartography and Geovisualization. Online. ISBN 9781003150527. Dostupné z: <https://doi.org/10.1201/9781003150527>. [cit. 2025-09-09].

SŁOMSKA, Katarzyna, 2018. Types of maps used as a stimuli in cartographical empirical research. Online. Miscellanea Geographica. Roč. 22, č. 3, s. 157-171. ISSN 2084-6118. Dostupné z: <https://doi.org/10.2478/mgrsd-2018-0014>. [cit. 2025-06-20].

ŠRAMO, Benjamín, 2023. Automation of Processing GNSS Track Records for Designing Intensity Maps. Diplomová práce (Mgr.). Olomouc: UNIVERZITA PALACKÉHO V OLOMOUCI. Přírodovědecká fakulta. [cit. 2025-07-01].

THE PYTHON WIKI, 2025. Online. 2025-05-14. Dostupné z:  
<https://wiki.python.org/moin/FrontPage>. [cit. 2025-06-19].

VISUAL STUDIO CODE, 2015. The open source AI code editor. Online. MICROSOFT.  
Visual Studio Code. 2026. Dostupné z: <https://code.visualstudio.com/>.  
[cit. 2026-05-03].

VOŽENÍLEK, Vít a KAŇOK, Jaromír, 2011. Metody tematické kartografie: vizualizace  
prostorových jevů. Olomouc: Univerzita Palackého v Olomouci pro katedru  
geoinformatiky. ISBN 978-80-244-2790-4. [cit. 2025-06-18].

# **PŘÍLOHY**

# SEZNAM PŘÍLOH

## Volné přílohy

- Příloha 1 ArcGIS Pro toolbox Graduated Numbers Generator ve formátu .atbx
- Příloha 2 Extrahovaný programový kód části Execution v souboru Execution.py
- Příloha 2 Extrahovaný programový kód části Validation v souboru Validation.py
- Příloha 4 Mapa „Počet hasičských sborů v Německých spolkových zemích“
- Příloha 5 Mapa „Average monthly wages in polish voivodeships in 2020 (PLN)“
- Příloha 6 Mapa „Počet letišť s ICAO kódem v zemích evropského regionu“
- Příloha 7 Mapa „Počet žen a mužů v krajích ČR v roce 2021 (v tisících)“
- Příloha 8 Poster

## Popis struktury odevzdávaných digitálních dat na datové úložiště katedry

### Text\_Prace

- Sedlacek\_DP.pdf
- Sedlacek\_DP.docx

### Prilohy

- GraduatedNumbersGenerator.atbx
- Execution.py
- Validation.py
- Mapa\_priloha4.pdf
- Mapa\_priloha5.pdf
- Mapa\_priloha6.pdf
- Mapa\_priloha7.pdf
- Poster.pdf

### WEB