

Seznam zápočtových úkolů (ZS 2025/2026)

Zásady používání nástrojů AI:

Student může použít AI pro inspiraci, vysvětlení nebo dílčí pomoc, ale musí umět obhájit vlastní kód, popsat rozhodnutí a příp. reagovat na operativní požadavky k úpravám či rozšířením.

Úkol 1: Síť kartografického zobrazení

Cíl úkolu

Student vytvoří program, který pomocí knihovny *turtle* vykreslí kartografickou síť (poledníky a rovnoběžky) pro zvolené nekonformní zobrazení. Cílem je procvičit algoritmizaci, matematické transformace a vizualizaci.

Specifikace výstupu

- 1) Výstupem je vykreslení kartografické sítě pomocí turtle.
- 2) Musí být zobrazeny poledníky a rovnoběžky v daném rozsahu.
- 3) Výstup musí být čitelný i při vyšší hustotě čar.
- 4) Uživatel může změnit měřítko, barvy, posun a parametry vzorkování.

Vstupní parametry programu

Program umožní nastavit:

- `u_min`, `u_max` – rozsah zeměpisné délky
- `v_min`, `v_max` – rozsah zeměpisné šířky
- krok poledníků a rovnoběžek
- `sampling step` – hustotu vzorkování
- výběr kartografického zobrazení

Povolená kartografická zobrazení

Student musí implementovat minimálně dvě z následujících:

- Azimutální ekvidistantní
- Azimutální Lambertovo (rovnoploché)
- Kónické zobrazení (např. Lambertovo konformní)
- Azimutální stereografické

Válcová zobrazení nejsou povolena.

Požadavky na implementaci

- Min. dvě funkce: výpočet transformace + vykreslení čar.
- Pouze standardní knihovny (math, turtle).
- Povinná parametrizace kroků.
- Student musí být schopen vysvětlit použity algoritmus.

Minimální rozsah projektu

- 1) Min. 2 kartografická zobrazení.
- 2) Kód rozdělen do funkcí.
- 3) Každé zobrazení se vykreslí zvlášť.
- 4) Odchyceny výjimky

Výstup pro obhajobu

Student:

- 1) ukáže funkční render,
- 2) vysvětlí klíčové výpočetní funkce,
- 3) zodpoví otázky a příp. program upraví

Úkol 2: Kreslení v Pythonu

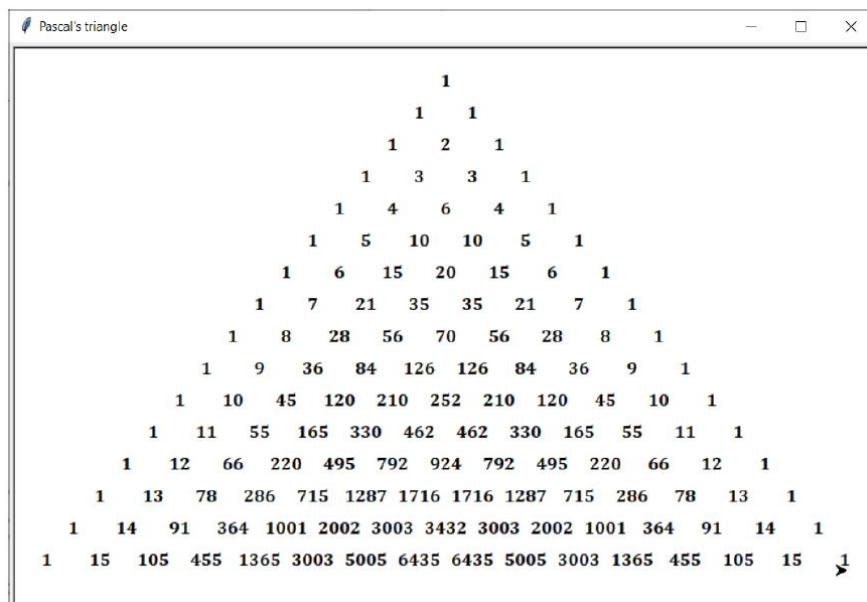
Cíl úkolu

Procvičení rekurze, algoritmického kreslení a grafické reprezentace složitých struktur.

Zadání:

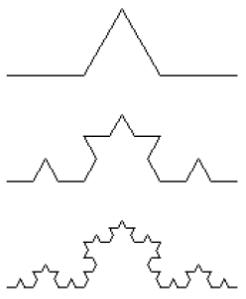
Pomocí turtle implementujte následující části:

a/ Pascalův trojúhelník



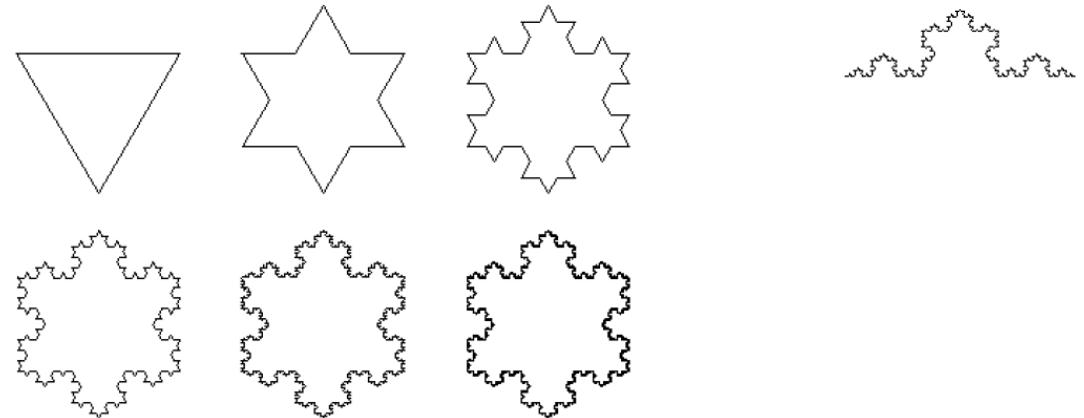
b/ Von Kochovy posloupnosti

Uživatel zadá číslo n . Program vypíše n -tý člen Von Kochovy posloupnosti křivek



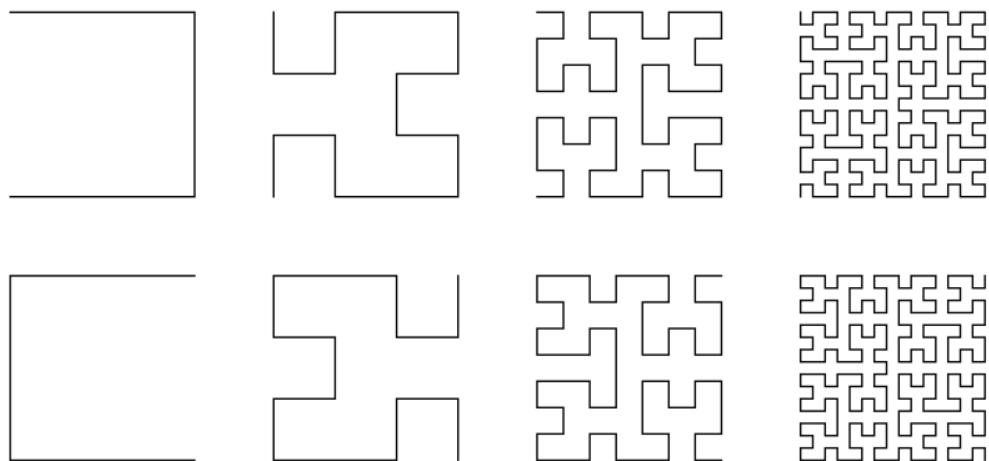
c/ Von Kochovy vločky

Uživatel zadá číslo n . Program vypíše n -tý člen Von Kochovy vločky.



d/ Hilbertovy křivky

Uživatel zadá číslo n . Program vypíše n -tý člen Hilbertovy křivky typu P či L



Požadavky

- 1) Rekurze musí být použita alespoň u Kochových a Hilbertových křivek.
- 2) Definujte vlastní funkce pro jednotlivé fraktály.
- 3) Omezte maximální řád (např. $n \leq 7$), aby se předešlo extrémnímu času vykreslení.
- 4) Program musí být parametrisovatelný (barvy, rychlosť, měřítka).

Výstup pro obhajobu

Student

- 1) ukáže vykreslené fraktály v samostatných kreslících oknech,
 - 2) zodpoví otázky a příp. program upraví
-

Úkol 3: Analýza textových dat (CSV/TXT)

Cíl úkolu

Procvičit práci se soubory, výjimkami, funkcemi a statistickým zpracováním dat.

Zadání:

Vytvořte program, který načte textový soubor (CSV, TXT) obsahující informace o studentech ve tvaru:

```
jméno;příjmení;body z testu
```

Program umožní uživateli vybrat operaci:

- Vypsat 5 nejlepších studentů
- Vypsat studenty pod průměrem
- Uložit statistiku třídění dle bodů do nového souboru

Požadavky

- 1) Ošetřete všechny chyby, např. `FileNotFoundException`, špatný formát, prázdné řádky.
- 2) Min. 3 funkce: načtení, výpočet statistiky, výpis.
- 3) Data lze zpracovat pomocí seznamů nebo slovníků.
- 4) Využijte konstrukci `with open()`.

Výstup

Program vypíše statistiky, uloží zpracovaný soubor a umožní pokračovat v práci.

Student

- 1) zodpoví otázky a příp. program upraví

Úkol 4: Geometrie v Pythonu pro 2D grafický systém

Cíl úkolu:

Prosvítejte objektový návrh, dědičnost, abstrakci, kompozici a polymorfismus.

Zadání:

Navrhněte a implementujte sadu tříd v Pythonu pro reprezentaci a manipulaci s geometrickými objekty v 2D grafickém systému. Využijte principů OOP, tj. dědičnost, kompozice, abstraktní třída a metody.

1. Abstraktní Třída `AGO`:

- Vytvořte abstraktní třídu `AGO` (Abstract Graphical Object), která bude sloužit jako základ pro všechny grafické objekty.
- Třída by měla obsahovat základní atributy jako `color` (barva), `width` (šířka) a `layer` (vrstva).
- Implementujte `property` metody pro tyto atributy a definujte abstraktní metody `print` a `length`.

2. Třída `Point2D`:

- Implementujte třídu `Point2D`, která reprezentuje bod ve 2D prostoru s x a y souřadnicemi.
- Třída by měla mít metody pro nastavení a získání souřadnic, stejně jako metodu pro výpis informací o bodu.

3. Třída `Point`:

- Vytvořte třídu `Point`, která dědí od `AGO` a `Point2D`. Třída `Point` bude reprezentovat bod s grafickými atributy.
- Přetěžte (override) metody `print` a `length`.

4. Třídy `PolyLine` a `Polygon`:

- Implementujte třídy `PolyLine` a `Polygon`, obě dědící od `AGO`. Tyto třídy budou reprezentovat mnohoúhelníkové čáry a polygon s určenými body.
- Každá třída by měla mít metodu pro výpis bodů a výpočet celkové délky.

5. Demonstrace funkcionalit:

- Vytvořte několik instancí třídy `Point` a `Point2D`.
- Sestavte seznam obsahující různé geometrické objekty, včetně instancí `PolyLine` a `Polygon`.
- Pro každý objekt v seznamu vypočítejte a vypište jeho délku.

Vaše řešení bylo mělo vycházet z klíčových OOP principů, tj. abstrakce, zapouzdření, dědičnost, polymorfismus. Implementace by měla být čistá, modularní a snadno rozšiřitelná pro další geometrické objekty.

Výstup pro obhajobu

Student

- 1) ukáže vytvořené objekty, vč. výpisu délek všech čar a polygonů,
- 2) vysvětlí návrh, princip dědičnosti a polymorfismu,
- 3) zodpoví otázky a příp. program upraví