

IV122 Matematika a programování

Úvod kurzu

Radek Pelánek

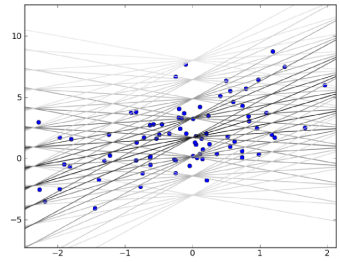
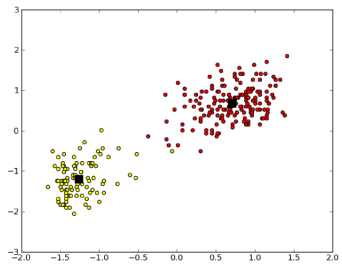
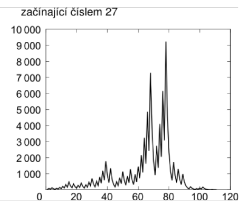
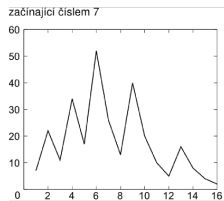
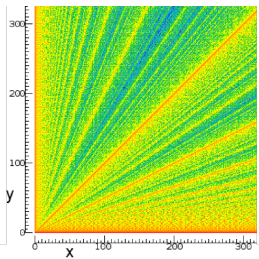
Cíle předmětu

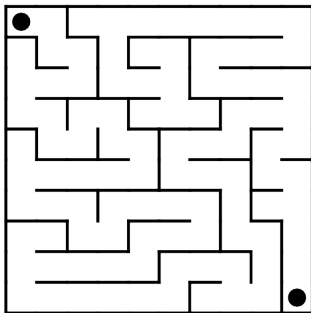
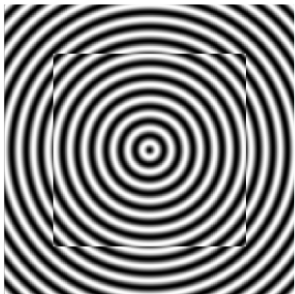
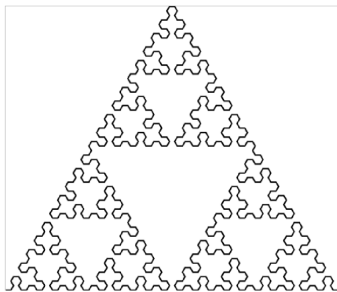
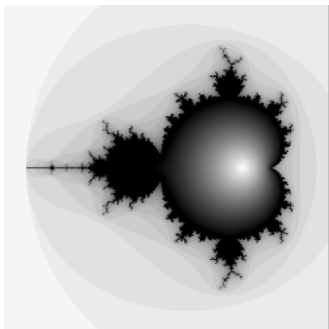
- Lepší pochopení matematických pojmů a metod skrze praktické programování.
- Procvičení programátorských schopností; trénink přechodu od teoretického algoritmu k funkční implementaci.
- Porozumění souvislostem mezi pojmy z různých oblastí.
- Radost z objevování, ocenění elegance, experimentování, inženýrská kreativita.

O čem předmět není

- Úvodní výuka matematických pojmů.
- Úvodní výuka programování.
- Výuka syntaxe konkrétního programovacího jazyka, technické triky, detaily vývojového prostředí, ...

- teorie čísel, kombinatorika
- geometrie, lineární algebra
- fraktály, chaos
- pravděpodobnost, statistika
- grafy, bludiště





- volně provázaná „přednáška“ a „cvičení“
 - připomenutí matematických pojmů
 - vysvětlení principů použitých v příkladech
 - rady ke struktuře programu, k vhodnému pohledu na problém
 - upozornění na typické chyby
- samostatná práce

důraz na řešení příkladů

Hodnocení předmětu

- domácí úlohy
- účast a aktivita v hodinách
- zkouška
 - „tužka & papír“ – příklady na matematické pojmy
 - programování – variace na úlohy z průběhu semestru
 - diskuze o domácích úlohách

především „slovní“ hodnocení, bez explicitního bodování

Domácí úlohy

- řešeny částečně již v průběhu semináře
- každý týden jedno zadání (většinou obsahuje několik podčástí)
- hodnocení na konci, průběžná orientační kontrola v průběhu semestru

Portfolio domácích úloh

- obsah:
 - všechny výtvary relevantní k předmětu – programy z hodin, domácí úkoly
 - zdrojové kódy, výsledky experimentů, vytvořené obrázky, komentáře, ...
- realizace:
 - vlastní webová stránka – IS / Můj web / IV122, tj. <http://is.muni.cz/www/UCO/IV122>
 - může být kdekoliv jinde (např. blog, autorizovaný web, github) ⇒ na uvedenou adresu dát odkaz

Kritéria hodnocení – domácí úlohy

- nezbytné:
 - z každého zadání alespoň jedna podúloha
 - celkově vypracováno alespoň 75 %
 - alespoň jednou výraznější bonus: vlastní rozšíření úlohy, netypické pojetí, ...
- hodnocení A: \pm vše a kreativita

Opisování

- zcela samostatně – žádné přebírání kódu od kolegů
- inspirace na webu
 - obecně nedělat, cíl je psát zcela samostatně
 - výjimečně může být smysluplné část kódu převzít a rozšířit ⇒ jasně označit převzatý kód

očekávám jednotnost zpracování (např. volba jazyka) – výjimky osvětlit

Matematika – předpoklady a poznámky

- znalosti na úrovni bakalářské matematiky na FI
- bude stručné připomenutí pojmů
- dílčí neznalosti je reálné doučit se za běhu
- zdroje např: knihovna, Wikipedia, Khan Academy

Matematika – příklady potřebných pojmů

- prvočísla, dělitelnost, mod
- log, exp, sin, cos
- vektor, matice, afinní transformace
- pravděpodobnost, normální distribuce
- graf, cesta, kostra

Programovací jazyk

- můžete pracovat v libovolném jazyce a prostředí
- problémy se syntaxí musíte zvládnout samostatně
- žádné sofistikované knihovny – budeme sami implementovat „základy“
- doporučuji Python (reálné i pokud ho zatím neznáte)

Kultura programování

... se bere v potaz, např:

- volba reprezentace dat
- názvy funkcí a proměnných
- smysluplné komentáře
- dekompozice problému na funkce
- nepoužívání „copy & paste kódu“

Samostatnost, prezentace

- kreativita a samostatnost vítána (a alespoň trochu očekávána)
 - rozšíření zadaných úloh
 - **vlastní drobné variace** (ne ulehčení úlohy, ale kreativní prozkoumání)
 - hrátky s barvami, experimenty s parametry, pokusy „co se stane když“, alternativní algoritmy, ...
- prezentace výtvorů
 - přehlednost
 - výstižné popisky
 - **dobré vyjádření hlavní myšlenky**

Náročnost předmětu

- předmět není těžký, ale je časově náročný
- **velmi** vhodné řešit úlohy průběžně

- **vysoko-úrovňový** – velká míra abstrakce, „spustitelný pseudokód“
- **interpretovaný** – pomalejší než kompilovaný, ale větší volnost
- **pedagogický** – byl tak navržen
- **moderní a široce používaný**
- volně a snadno **dostupný** na všech platformách

Python: velmi rychlé uvedení

- dynamicky typovaný
- bílé znaky důležité – vyznačení bloků
- Python 2.7 vs 3
- zdroje pro naučení např.:
 - <http://python.cz/>
 - <http://howto.py.cz/index.htm>

Mnoho příkladů vede na grafické znázornění, budou nám stačit základní operace:

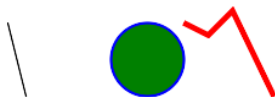
- vektorová grafika
 - `line(x1, y1, x2, y2)`, `circle(x, y, r)` + barvy, výplň
 - doporučeno SVG
- bitmapová grafika
 - `put_pixel(x, y, color)`
 - Python: knihovna Image
 - ostatní jazyky: canvas, vhodná knihovna
- vykreslování základních grafů: liniový, bodový (scatter), histogram

Scalable Vector Graphics (SVG)

- vektorový formát založený na XML
- snadný způsob vytváření obrázků v jakémkoliv jazyce (generujeme prostý text)
- prohlížení: např. webový prohlížeč
- ruční editování: např. Inkscape
- převod na bitmapu: např. convert (ImageMagick)

SVG příklad

```
<svg xmlns="http://www.w3.org/2000/svg">  
<line x1="15" y1="20" x2="30" y2="80"  
      stroke="black" stroke-width="1"/>  
<circle cx="130" cy="50" r="30" stroke="blue"  
        stroke-width="2" fill="green" />  
<polyline fill="none" stroke="red" stroke-width="4"  
          points="160,20 180,30 200,10 234,80"/>  
</svg>
```

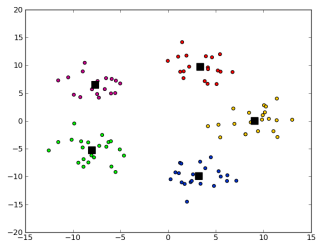
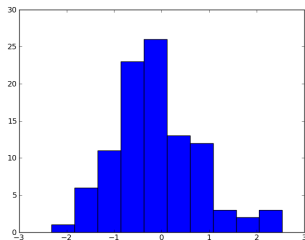
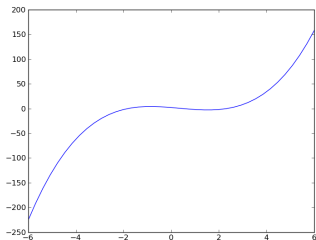


Bitmapová grafika: příklad Python

```
from PIL import Image

def disc(size=150, r=50):
    im = Image.new("RGB", (size, size))
    for x in range(size):
        for y in range(size):
            if (x-size/2)**2 + (y-size/2)**2 < r**2:
                im.putpixel((x, y), (0, 0, 0))
            else:
                im.putpixel((x, y), (255, 255, 255))
    im.show()
```


Vykreslování základních grafů



Python, NumPy, matplotlib

```
import pylab as plt
import numpy as np

def demo_hist(n):
    values = np.random.randn(n)
    plt.hist(values)
    plt.show()

def demo_lines():
    x = np.linspace(-6, 6, 40)
    plt.plot(x, x**3 - x**2 - 4*x + 2)
    plt.show()
```