`Programování` 1 pro matematiky

13. cvičení, 5/6-1-2022

tags: `Programovani 1 2021`, `středa` `čtvrtek`

Obsah:

Farní oznamy

1. **Materiály k přednáškám** najdete v GitHub repozitáři https://github.com/PKvasnick/Programovani-1. Najdete tam také kód ke cvičením a pdf soubory textů cvičením.
2. **Poslední cvičení** Přehled standardní knihovny je předmětem našeho posledního cvičení.

   Zápočty

   - Zápočet dostane automaticky každý, kdo má v ReCodExu víc jako 70% povinných bodů, tedy řekněme **105** bodů.
   - Kdo má 50-70% povinných bodů, tedy **70 - 104** bodů, napíše mi a domluvíme se, jak může dodatečné body získat. Typickou možností je vyřešení dodatečných příkladů.
   - Kdo má ještě méně bodů a důveryhodné vysvětlení, co dělal celý semestr, napíše mi a můžeme se pokusit najít nějaké řešení.

Dotazy?

Standardní knihovna

Python sice poskytuje některé hezké prostředky pro vyjádření algoritmů, ale jeho opravdová užitečnosti spočívá v existujících knihovnách, které máme k dispozici.

Moduly a import

```
1 import module
```

způsobí, že systém vyhledá soubor `module.py`, a to v aktuálním adresáři anebo v 4adresářích obsažených v proměnné prostředí `PYTHONPATH`. Ze souboru pak načte všechny objekty, které jsou v něm definovány. Příkaz ve skutečnosti volá zabudovanou funkci **import**, a dělá něco takovéhoto:

```
1 module = __import__("module")
```

Objekty modulu `module` pak voláme `module.objekt`.

Při importu můžeme moduly také přejmenovat, aby se nám snáze používal. Některé přejmenování jsou dokonce standardní:

```
import numpy as np
from matplotlib import pyplot as plt
```

Děje se toto:

```
np = __import__("numpy")
```

a objekty modulu pak můžeme volat stručněji `np.objekt`, resp. `plt.objekt`.

```
from module import func
```

zkopíruje definici funkce `func` z modulu `module` do aktuálního modulu, takže namísto `module.func` můžeme psát jednoduše `func`. Používat s mírou, nechcete si zaplevelit prostor jmen jmény z jiných modulů.

Standardní knihovny

Potkali jsme

- `math` - matematické funkce a konstanty
- `operator` - funkce, implementující operátory
- `collections` - `defaultdict`, ale také třeba `deque`, `Counter` a další užitečné datové kontejnery.
- `itertools` - permutace a kombinace.

Knihovna random

```
>>> import random
>>> random.random()
0.28947857702914326 (z intervalu [0; 1])
>>> random.uniform(0, 1000)
50.64122748168531 (z intervalu [a; b])
>>> random.randrange(0, 1000)
524 (celé číslo od a do b-1)
>>> random.randrange(1000)
451 (stejně jako u range jde dolní mez vynechat)
>>> random.seed(12345)
>>> random.random()
0.41661987254534116 (vyjde vždy stejně)
>>> random.choice(['pátek','sobota','neděle'])
'neděle' (náhodný prvek seznamu)
>>> random.choices("0123456789", k=10)
['3','4','9','8','1','3','2','1','8','6']
(výběr s vracením)
>>> random.sample("0123456789", k=10)
['3','0','6','1','2','7','4','9','5','8']
(výběr bez vracení)
>>> random.sample(range(1000000), k=5)
[301599, 128323, 695182, 627325, 967424]
(vzorkuje přímo rozsah, nepřevádí na seznam)
```

Zlomky

Velice užitečné tam, kde chceme počítat beze ztráty přesnosti.

```
>>> from fractions import Fraction
>>> Fraction(1, 2) + Fraction(1, 3)
Fraction(5, 6)
>>> Fraction(1/3)
Fraction(6004799503160661, 18014398509481984)
>>> Fraction(1/3).limit_denominator(100000)
Fraction(1, 3)
>>> Fraction("1/3")
Fraction(1, 3)
>>> print(Fraction(1, 3))
'1/3'
```

Další zajímavé knihovny

- `datetime`, `calendar` – práce s datem a časem
- `copy` – "hluboké", neboli rekurzivní kopírování
- `cmath` – komplexní čísla
- `statistics` – statistické funkce
- `re` – prohledávání a nahrazování textu pomocí regulárních výrazů

Příklady

- Simulujte 1000 hodů kostkou. Spočítejte výskyty každého čísla. Kolik jich je nejméně a kolik nejvíce?
- Generujte náhodně body ve čtverci [-1,1] × [-1,1]. Počítejte, kolik z nich padlo do jednotkového kruhu, a tím aproximujte π.
- Generujte náhodně permutace slov věty "Nemám rád zbytečně použité permutace." Kolik z nich dávalo smysl? :)
- Vygenerujte náhodně permutaci na množině 1 až N a spočítejte, kolik má pevných bodů. Opakováním experimentu odhadněte, jaká je pravděpodobnost, že náhodná permutace nemá pevný bod.
- Simulujte náhodnou procházku po celých číslech od 0 do N. Začínáme v 0, v každém kroku náhodně buď zvýšíme nebo snížíme o 1 (v 0 jen zvyšujeme). Po kolika krocích se dostaneme do N? Jak dopadne dvojrozměrná verze (ze středu čtverce se chceme dostat k okraji)?