



IB111 Základy programovania

Cvičenie 4 – moduly, náhodné čísla

Matej Troják

(Poďakovanie: Valdemar Švábenský)

1. Moduly

2. Náhodné čísla

3. Příklady

Modul – motivácia

- Chceme používať funkcie, ktoré nie sú vstavané

```
>>> sqrt(100)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'sqrt' is not defined
```

- Modul je súbor s definíciami funkcií a konštánt
- Najprv musíme modul načítať (*importovať*)

Načítanie a použitie modulu

- Načítanie celého modulu

```
>>> import math
>>> math.sqrt(100)
10
```

- Načítanie konkrétnej funkcie z modulu

```
>>> from math import sqrt
>>> sqrt(100)
10
```

- Načítanie celého obsahu modulu

```
>>> from math import *
>>> sqrt(100)
10
>>> gcd(18, 12)
6
```

Obsah a nápoveda modulu

```
>>> import math
>>> dir(math)
['_doc_', '__loader__', '__name__', '__package__', '__spec__', 'acos',
'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign',
'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs',
'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot',
'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log',
'log10', 'log1p', 'log2', 'modf', 'nan', 'pi', 'pow', 'radians', 'sin',
'sinh', 'sqrt', 'tan', 'tanh', 'trunc']
>>> help(math.gcd)
Help on built-in function gcd in module math:

gcd(...)
    gcd(x, y) -> int
    greatest common divisor of x and y
```

Vlastný modul

- Modul je (väčšinou) Pythonovský skripty (.py)
- Akýkoľvek napísaný skript sa dá použiť ako modul
- Importujeme pomocou názvu súboru (bez prípony .py)
- Pozor na umiestnenie súboru!
- Pri importe je všetok kód modulu spustený
 - vhodné ak obsahuje len definície funkcií a konštant
- Dostupné užitočné moduly:
<https://docs.python.org/3/py-modindex.html>

Balík

- Menný priestor, ktorý obsahuje moduly a ďalšie balíky
- Adresár, ktorý obsahuje súbor s názvom `__init__.py`
 - Indikuje, že adresár je Pythonovským balíkom

```
myPackage/  
|__ __init__.py  
|__ myModule.py
```

- Môže byť načítaný ako modul

```
>>> import myPackage.myModule  
>>> from myPackage import myModule
```

1. Moduly

2. Náhodné čísla

3. Příklady

Náhodné čísla

- Rovnomerne rozložené dáta z nejakého intervalu s požadovanými štatistickými vlastnosťami
 - čísla nevieme predpovedať na základe pozorovania predošlých hodnôt
- Príklad: hod kockou
 - 5, 1, 6, 6, 2, 3, ...

Pseudonáhodné čísla

- Generované zo semienka (*seed*), napr. zo systémového času / zvukovej karty / pohybu myši / ...
 - pre rovnaké semienko poskytne generátor vždy rovnaký výstup
- niekedy to stačí, ale inokedy nie (napr. kryptografia)
- V Pythone deterministický algoritmus Mersenne twister

(Pseudo)náhodné čísla v Pythone

- Modul random

```
from random import randint, random
```

- `randint(a, b)` - vráti náhodný `int` n taký, že $n \in [a, b]$
- `random()` - náhodný `float` n taký, že $n \in [0, 1)$

<https://docs.python.org/3.5/library/random.html>

1. Moduly

2. Náhodné čísla

3. Příklady

Úloha 1

- Napíšte funkciu `random_numbers(how_many)`, ktorá vypíše `how_many` náhodných čísel z intervalu $[0, 100]$, ich minimum, maximum a priemer.

```
>>> random_numbers(8)
86 81 59 74 28 24 58 50
Minimum: 24
Maximum: 86
Average: 48.453125
```

- Bonus: Pomocou parametrov funkcie s nastavenou východnou hodnotou vhodne obmedzte rozsah generovaných čísel
`random_numbers(how_many, lower_bound=0, upper_bound=100)`

```
>>> random_numbers(8, 1, 6)
3 6 1 5 3 6 2 5
Minimum: 1
Maximum: 6
Average: 4.16796875
```

Úloha 2

- Opilec je na pol ceste medzi domovom a krčmou. Napíšte funkciu `drunkman_simulator(size, num_steps)`, ktorá bude simulovať opilcov pohyb. Predtým, než opilec zaspí, dokáže spraviť `num_steps` krokov a vzdialenosť medzi domovom a krčmou určuje parameter `size`. Simulácia končí, ak opilec dôjde domov alebo do krčmy, prípadne ak zaspí.

```
>>> drunkman_simulator(11, 5)
home . . . . . * . . . . . pub
home . . . . . * . . . . . pub
home . . . * . . . . . . . pub
home . . . . * . . . . . . pub
home . . . . . * . . . . . pub
home . . . . * . . . . . . pub
Drunk guy fell asleep on the street.
```

```
>>> drunkman_simulator(11, 30)
home . . . . . * . . . . . pub
home . . . . . * . . . . . pub
home . . . . * . . . . . . pub
home . . * . . . . . . . pub
home . * . . . . . . . . pub
home * . . . . . . . . . pub
Drunk guy got home safely.
```

- Využite dekompozíciu na funkciu `print_row(size, position)` pre výpis jedného riadka stavu opilca.

Úloha 3

- Rozšírite riešenie z predchádzajúcej úlohy o dve veci:
 1. aby funkcia `drunkman_simulator` vracala `True` ak dôjde domov, inak vracala `False`,
 2. rožšírite funkciu `drunkman_simulator` o voliteľný parameter `output` (default je `True`), ktorý zabezpečí aby sa (ne)vypisoval stav opilca.

```
>>> drunkman_simulator(11, 5, False)
False
>>> drunkman_simulator(6, 5, True)
home . . * . . . pub
home . * . . . . pub
home * . . . . . pub
Drunk guy got home safely.
True
>>> drunkman_simulator(11, 100, False)
True
```

Úloha 4

- Napíšte funkciu `drunkman_analysis(size, steps, count)`, ktorá odsimuluje opilca `count` krát a vypíše percentuálnu úspešnosť toho, že opilec došiel domov.

```
>>> drunkman_analysis(11, 10, 10)
Arriving home in 20.00 % of cases
>>> drunkman_analysis(11, 100, 10)
Arriving home in 40.00 % of cases
>>> drunkman_analysis(11, 100, 1000)
Arriving home in 49.90 % of cases
```

Úloha 5a – rozšírenia opilca

- Iniciálna pozícia opilca zadaná ako parameter.

```
>>> drunkman_simulator(11, 10, 3)
home . . . * . . . . . . . pub
home . . * . . . . . . . pub
home . * . . . . . . . pub
home . . * . . . . . . . pub
home . * . . . . . . . pub
home * . . . . . . . . pub
Drunk guy got home safely.
True
```


Úloha 5b – rozšírenia opilca

- Zmena pravdepodobnosti toho, ktorým smerom sa opilec pohne.

```
>>> drunkman_simulator(11, 8, 0.7)
home . . . . . * . . . . . pub
home . . . . . * . . . . . pub
home . . . * . . . . . . . pub
home . . . . * . . . . . . pub
home . . . * . . . . . . . pub
home . . * . . . . . . . . pub
home . * . . . . . . . . . pub
home * . . . . . . . . . . pub
Drunk guy got home safely.
True
>>> drunkman_analysis(20, 50, 1000, 0.7)
Arriving home in 97.00 % of cases
```

- *Bonus:* Skontrolujte hodnotu parametra, ktorý nastavuje pravdepodobnosť (interval $[0, 1]$)