



IB111 Základy programovania

Cvičenie 3 – funkcie, chyby, debugger

Matej Troják

(Poďakovanie: Valdemar Švábenský)

1. Funkcie s návratovou hodnotou

2. Chyby v kóde

3. Príklady

Funkcia

- Z minulej hodiny: funkcia je pomenovaná sekvencia kódu
- Nový pohľad: krabica, ktorá vezme vstup, niečo s ním spraví a vráti výstup



```
def decrease_by_two(number):  
    return number - 2
```

```
>>> decrease_by_two(5)  
3
```

Návratová hodnota

- Funkcia vráti výstup, tzv. *návratovú hodnotu*, volajúcemu
- pomocou príkazu `return`
- okamžite ukončí beh funkcie

```
def divide_two_numbers(a, b):  
    if b == 0:  
        return -1  
    return a/b
```

print() vs. return

- return vracia hodnotu, print() vypisuje do konzole!
- Ak vo funkcii nie je return, vráti sa prázdna hodnota None

```
def decrease_by_two_return(number):  
    return number - 2  
  
def decrease_by_two_print(number):  
    print(number - 2)
```

```
>>> result = decrease_by_two_return(5)  
>>> print(result)  
3  
>>> result = decrease_by_two_print(5)  
3  
>>> print(result)  
None
```

1. Funkcie s návratovou hodnotou

2. Chyby v kóde

3. Príklady

Chybový výpis

```
>>> 5 -  
File "<stdin>", line 1  
  5 -  
    ^  
SyntaxError: invalid syntax
```

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

```
>>> 1 + '5'  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

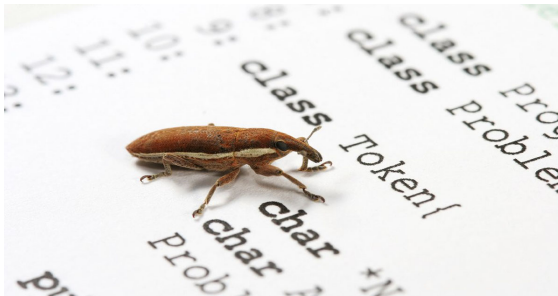
Chybové hlášky

- `SyntaxError`: “Nerozumiem, čo píšeš”
- `ZeroDivisionError`: “Nesmieš deliť nulou”
- `NameError`: “Nepoznám toto meno (premennej, funkcie)”
- `TypeError`: “Nefunguje to s týmto typom hodnoty”
- `ValueError`: “Nenašiel som túto hodnotu”

<https://docs.python.org/3/tutorial/errors.html>

Logické chyby

- kód, ktorý dáva nesprávne výsledky (tzv. *bug*)
- Aj v krátkom a skontrolovanom kóde môžu byť chyby



Logické chyby – ako na ne

- Metóda *pokus-omyl* (a.k.a. tipuj)
- Náhodne zmeň niečo v programe a čakaj, či bude fungovať
- Obchádzaj chyby
- **určite nie!**

```
x = double(n)
# Suppose double(n) contains a bug and does not work for n = 10
if n == 10:
    x = 20
```

Logické chyby – detekcia

- Čítať (a chápať) chybové hlášky
- Trasovať kód (v hlave alebo na papieri)
- Ladiace výpisy (print())
- Debugger – SW nástroj pre hľadanie chýb

Ladiace výpisy

- Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

```
def gcd(a, b):  
    while b != 0:  
        t = b  
        b = a % b  
        a = t  
    return a
```

Ladiace výpisy

- V každej iterácii cyklu vypisujeme stav premenných:

```
def gcd(a, b):  
    cycle = 0  
    while b != 0:  
        cycle += 1  
        print("Cycle", cycle)  
        t = b  
        print("t =", t)  
        b = a % b  
        print("b =", b)  
        a = t  
        print("a =", a)  
    return a
```

Ladiace výpisy

```
>>> gcd(18, 24)
```

```
Cycle 1
```

```
t = 24
```

```
b = 18
```

```
a = 24
```

```
Cycle 2
```

```
t = 18
```

```
b = 6
```

```
a = 18
```

```
Cycle 3
```

```
t = 6
```

```
b = 0
```

```
a = 6
```

```
6
```

Debugger

- Nástroj priamo v PyCharm
- Run → Debug

```
12     area = 0   area: 0
13     for i in range(1, n):   i: 1
14     area += math.pi * pow(i, 2)
15     return area
```

Debug debug_demo

Debugger Console

Frames

- MainThread
- circles_area, debug_demo.py:14
- <module>, debug_demo.py:17
- execfile, _pydev_execfile.py:18
- run, pydevd.py:974
- <module>, pydevd.py:1596

Variables

- 14 area = {int} 0
- 14 i = {int} 1
- 14 n = {int} 5

- Video tutoriál: <https://www.youtube.com/watch?v=QJtWxm12Eo0>

1. Funkcie s návratovou hodnotou

2. Chyby v kóde

3. Príklady

Úloha 1

- Napíšte funkciu `factorial(n)`, ktorá spočíta a vráti faktoriál čísla n pomocou cyklu `for`.

$$n! = 1 \times 2 \times 3 \times \dots \times n$$

$$0! = 1$$

```
>>> factorial(5)
120
>>> factorial(10)
3628800
```

- Bonus: Vyriešte úlohu pomocou cyklu `while`.

Úloha 2

- Napíšte funkciu `lcm(a, b)`, ktorá spočíta a vráti najmenší spoločný násobok čísel `a` a `b`. Využite pritom funkciu `gcd(a, b)` z predchádzajúcej časti cvičenia.

```
>>> lcm(6, 9)
18
>>> lcm(23, 42)
966
```

Úloha 2

- Napíšte funkciu $\text{lcm}(a, b)$, ktorá spočíta a vráti najmenší spoločný násobok čísel a a b . Využite pritom funkciu $\text{gcd}(a, b)$ z predchádzajúcej časti cvičenia.

```
>>> lcm(6, 9)
18
>>> lcm(23, 42)
966
```

- Hint:

$$\text{lcm}(a, b) = \frac{|a \cdot b|}{\text{gcd}(a, b)}$$

Úloha 3

- Napíšte funkciu `divisors_count(n)`, ktorá vráti počet (kladných) deliteľov celého čísla `n`.

```
>>> divisors_count(18)
6
>>> divisors_count(100)
9
```

- Bonus: Šlo by výpočet nejako urýchliť bez overovania všetkých `n` čísel?

Úloha 4

- Napíšte funkciu `is_prime(n)`, ktorá pre dané prirodzené číslo n rozhodne, či je alebo nie je prvočíslo. Využite pritom funkciu `divisor_count(n)` z predchádzajúcej úlohy.

```
>>> is_prime(23)
True
>>> is_prime(42)
False
```

- *Hint:* Číslo $n \in \mathbb{N}$ je prvočíslo práve vtedy, keď má dva rôzne delitele: číslo 1 a číslo n (samé seba).

Úloha 5

- Napíšte funkciu `primes_less_than(n)`, ktorá vypíše všetky prvočísla menšie ako `n`. Využite pritom funkciu `is_prime(n)` z predchádzajúcej úlohy.

```
>>> primes_less_than(5)
2 3
>>> primes_less_than(30)
2 3 5 7 11 13 17 19 23 29
```

- Bonus: Funkcia `primes(n)` vypíše prvých `n` prvočísel.

```
>>> primes(10)
2 3 5 7 11 13 17 19 23 29
```

Úloha 6

- Napíšte funkciu `pi()`, ktorá vráti približnú hodnotu čísla π s presnosťou na tisíciný. Použite pri tom Leibnizovu radu.

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \frac{\pi}{4}$$

```
>>> pi()  
3.141
```

- Bonus: ... presnosťou na desatinné miesta zadanou ako parameter `n`.

```
>>> pi(5)  
3.14157  
>>> pi(6)  
3.141591
```

- Riešenie pomocou náhodných čísel.

<http://www.stealthcopter.com/blog/2009/09/python-calculating-pi-using-random-numbers/>

Úloha 7

- Napíšte funkciu `area_of_circle(r)`, ktorá spočíta a vráti obsah kruhu s polomerom `r`. Využite pritom funkciu `pi()` z predchádzajúcej úlohy.

$$A = \pi \times r^2$$

```
>>> area_of_circle(5)
78.525
>>> area_of_circle(10)
314.1
```


Úloha 7

- Napíšte funkciu `area_of_circle(r)`, ktorá spočíta a vráti obsah kruhu s polomerom `r`. Využite pritom funkciu `pi()` z predchádzajúcej úlohy.

$$A = \pi \times r^2$$

```
>>> area_of_circle(5)
78.525
>>> area_of_circle(10)
314.1
```

- S väčším `r` sa chyba približného výpočtu zväčšuje. Je možné výpočet nejako zpresniť?

```
>>> area_of_circle(1000)
3141000.0
>>> import math
>>> math.pi*1000**2
3141592.653589793
```

Úloha 8

- a) Napíšte funkciu `convert_10_to_2(n)`, ktorá vráti zadané číslo `n` v dvojkovej sústave ako reťazec.

```
>>> convert_10_to_2(999)
1111100111
>>> convert_10_to_2(123)
1111011
```

- b) Napíšte funkciu `convert_2_to_10(n)`, ktorá vráti zadaný reťazec `n` reprezentujúci binárne číslo v desiatkovej sústave ako `int`.

```
>>> convert_2_to_10('1111100111')
999
>>> convert_2_to_10('1111011')
123
```

- Bonus: Napíšte funkciu `convert_10_to_x(n, base)`, ktorá vráti zadané číslo `n` v sústave so základom `base`.