

IB111

Základy programování

František Lachman lachmanfrantisek@mail.muni.cz

cvičení 3

2. říjen 2018

Osnova

- poznámka k domácí úloze
- kontrolní otázky
- pokročilejší výpočty, return

Docházka

Kontrolní otázky

```
import random  
random.randint(1,6)
```

**Jaký je rozdíl mezi číselnými
typy `int` a `float`?**

```
import random  
random.randint(1,6)
```

**Jak zapisujeme operace
„celočíselné dělení“ a „dělení
se zbytkem“?**


```
type(3)
```

```
type(3.0)
```

```
type(3 // 3)
```

```
type(3 / 3)
```

```
type(3.0 / 3)
```

```
type(3.0 // 3)
```

```
import random  
random.randint(1,6)
```

Co dělají funkce `round`,
`math.floor`, `math.ceil`?

```
from math import floor, ceil
```

```
help(round)
```

```
help(floor)
```

```
help(ceil)
```

```
import random  
random.randint(1,6)
```

**Jaká je základní myšlenka
algoritmu pro výpočet
ciferného součtu?**

```
import random  
random.randint(1,6)
```

**Jaká je základní myšlenka
algoritmu pro výpočet
odmocniny?**


```
import random  
random.randint(1,6)
```

**Jaká je základní myšlenka
algoritmu pro převod na
binární zápis?**

```
import random  
random.randint(1,6)
```

**Jaký je význam příkazu
return?**

**Jaký je rozdíl mezi tím, když
na konci funkce zavoláme
`return` a `print`?**

**Jaký je rozdíl mezi `return` a
`return x` ?**

```
def five():  
    return 5
```

```
a = five()  
b = five()
```

```
type(a)
```

```
print(a)  
print(a + b)
```

```
def double(a):  
    b = a * 2  
    return b # zkráceně return a * 2
```

```
b = double(5)  
print("something else")  
print(b)
```

```
print(double(five()))
```

```
print(double(double(double(double(five())))))
```



```
x = five()  
y = double(x)  
print(x + y)
```

```
def print_double(x):  
    print(x*2)
```

```
x = five()  
y = print_double(x)  
print(x + y)
```

```
def nothing():  
    return
```

```
nothing()  
a = nothing()  
type(a)
```

```
def my_function():  
    print("jsem tu")  
    return  
    print("sem se vypocet nikdy nedostane")
```

```
my_function()
```

Návrat z funkce pomocí `return`

```
def my_max(number1, number2):  
    if number1 == number2:  
        print("jsou stejna")  
        return number1  
    if number1 < number2:  
        print("prvni je vetsi")  
        return number1  
    print("druhe je vetsi")  
    return number2
```

```
a = my_max(2, 2)  
b = my_max(2, 3)  
c = my_max(3, 2)
```

```
print(a, b, c)
```

Jaký význam má `pass`?

Prázdné tělo - `pass`

```
def without_return():  
    pass
```

- spustitelný kód bez implementace
- explicitně vyjádřené "nedělání ničeho"

Příklady - 3.1. Pokročilé počítání

Napište funkci `series_sum(n)`, která vrátí součet čísel od `1` do `n`.

```
def series_sum(n):  
    pass
```


3.1.1. Faktoriál pomocí `for`

Napište funkci `factorial(n)`, která vrátí faktoriál čísla `n` a využívá cyklus `for`.

(Připomeňme, že $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$ a že $0! = 1$.)

```
def factorial(n):  
    pass
```

3.1.3. Ciferný součet

Napište funkci `digit_sum(n)`, která vrátí ciferný součet čísla `n`.

```
def digit_sum(n):  
    pass
```

Příklady - 3.2. Dělitelnost a prvočísla

3.2.1. Dělitelé

Napište funkci `divisors(n)`, která vypíše všechny dělitele čísla `n`.

```
def divisors(n):  
    pass
```

3.2.2. Počet dělitelů

Napište funkci `divisors_count(n)`, která vrátí počet dělitelů čísla `n`.

```
def divisors_count(n):  
    pass
```

3.2.3. Je prvočíslo

Napište funkci `is_prime(n)`, která vrátí `True` pokud je číslo `n` prvočíslo, jinak `False`.

```
def is_prime(n):  
    pass
```

3.2.4. Prvočísla menší než n

Napište funkci `primes_less_than(limit)`, která vypíše všechna prvočísla menší než `limit`.

```
def primes_less_than(limit):  
    pass
```

3.2.12. Euklidův algoritmus

Implementujte Euklidův algoritmus pomocí funkce

`euklid(a, b)`.

```
def euklid(a, b):  
    pass
```


Příklady - 3.3. Aproximace

- 3.3.1. Eulerovo číslo

Napište funkci `e()` pro přibližný výpočet Eulerova čísla pomocí nekonečného součtu s přesností na miliontiny.

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots$$

```
def e():  
    pass
```

Příklady - 3.4. Převody mezi číselnými soustavami

3.4.1. Převod z desítkové soustavy do dvojkové

Napište funkci `convert_10_to_2(n)`, která vrátí zadané číslo `n` ve dvojkové soustavě jako řetězec.

```
def convert_10_to_2(n):  
    pass
```

3.4.2. Převod z dvojkové soustavy do desítkové

Napište funkci `convert_2_to_10(n)`, která vrátí zadaný řetězec `n` reprezentující binární číslo v desítkové soustavě jako `int`.

```
def convert_2_to_10(n):  
    pass
```

Osnova

- poznámka k domácí úloze
- kontrolní otázky
- pokročilejší výpočty, return