

**IB111**

# **Základy programování**

**František Lachman**

**[lachmanfrantisek@mail.muni.cz](mailto:lachmanfrantisek@mail.muni.cz)**

**cvičení 1**

**20. září 2017**

# Osnova

- organizace cvičení
- studijní materiály
- seznámení s vývojovým prostředím
- základy Pythonu
- želví grafika

# Organizace cvičení

- cvičení jsou povinná
  - na začátku zapisována účast
  - povolené maximálně 2 neomluvené absence
- přednášky nepovinné, **ale**:
  - předpokládá se znalost ([kontrolní otázky](#) na začátku cvičení)
  - videa nemusí být spolehlivá/včas

# Kontrolní otázky

- Jaké jsou podmínky zakončení předmětu?
- Je nezbytné vyřešit všechny domácí úlohy?
- Proč je dobré umět programovat?
- Co je to algoritmus?
- Je důležité rozlišovat Python verze 2 a verze 3? Kterou verzi používáme v tomto kurzu?
- Který z následujících pojmů se v tomto kurzu pravděpodobně nebude příliš vyskytovat: faktoriál, kombinační číslo, prvočíslo, jmenné číslo, logaritmus, binární soustava, logické operace?

# Přihlášení na počítače

- fakultní login ([seznam](#))
- fakultní heslo (defaultně sekundární)
  - nastavitelné [zde](#)
- vlastní notebook je možný  
(nainstalovat [Python 3.x](#) a IDLE)

# Hodnocení předmětu - vnitřní

<https://www.fi.muni.cz/IB111/?p=hodnoceni>

- “
- Dvě vnitrosemestrální zkoušky u počítače (140 bodů): programování v Pythonu v rámci dvou cvičení (40+100 bodů), zadané úlohy jsou variace na příklady ze cvičení, je povoleno používat podklady.
  - Alespoň 70 bodů za vnitrosemestrální zkoušky. „

# Hodnocení předmětu - úkoly

<https://www.fi.muni.cz/IB111/?p=hodnoceni>

- “
- Domácí úkoly (160 bodů): celkem 5 úloh v průběhu semestru (25, 25, 30, 40, 40 bodů). odevzdání všech 5 domácích úloh (alespoň 5 bodů za každou).
  - Alespoň 100 bodů za domácí úlohy.
- ”

# Studijní materiály

- Studijní materiály v ISe:  
[https://is.muni.cz/auth/el/1433/podzim2017/IB111/um/skupina\\_20\\_lachman/](https://is.muni.cz/auth/el/1433/podzim2017/IB111/um/skupina_20_lachman/)
- Harmonogram cvičení:  
<https://www.fi.muni.cz/IB111/?p=cviceni>
- Sbírka příkladů:  
<https://www.fi.muni.cz/IB111/sbirka/>
- Další zdroje:  
<https://www.fi.muni.cz/IB111/?p=zdroje>



**Základy programování ukázané na Pythonu**



**~~Základy programování v Pythonu~~**

# Python

- Používáme Python 3.x (nekompatibilní s 2.7)
- Test:

```
Python 2.7.13 (default, Aug 16 2017, 12:56:26)
[GCC 7.1.1 20170802 (Red Hat 7.1.1-7)] on linux2
Type "help", "copyright", "credits" or "license" for more
>>> 2/3
0
```

```
Python 3.6.2 (default, Sep 1 2017, 12:03:48)
[GCC 7.1.1 20170802 (Red Hat 7.1.1-7)] on linux
Type "help", "copyright", "credits" or "license" for more
>>> 2/3
0.6666666666666666
```

# IDLE

- jednoduché vývojové prostředí pro Python
- shell
  - restart shellu (`CTRL+F6`)
  - ukončení dlouhého/nekonečného běhu (`CTRL+C`)
- editor souborů (`CTRL+N`; spuštění pomocí `F5`)
  - nepojmenovávat soubory stejně jako názvy modulů (např. `turtle.py`) => chyby v importech

# Formátování

- bloky kódu odsazujeme mezerami (4)  
(editory mají většinou nastavené nahrazení tabulátoru patřičným množstvím mezer)

```
def hello():  
    for i in range(5):  
        print("Hello!")  
    print("Bye!")
```

# Dokumentace

- pojmenování proměnných je anglicky a smysluplné (ne jednopísmenné a matoucí názvy)
- kód je rozumně komentovaný

```
# comment
print("Hello") # comment

def hello(person):
    """
    Function description.

    :param person: description of parameter
    """
    print("Hello", person)
```

# Základní operace

```
print(2 + 3)           # scitani
print(2 * 5)          # nasobeni
print(10 / 3)         # desetinne deleni
print(10 // 3)       # celociselne deleni
print(10 % 3)        # modulo
print((5 - 3) * (7 - 4)) # uzavorkovani
```

```
5
10
3.333333333333333
3
1
6
```

# Proměnné

```
x = 2          # do promenne x prirad cislo 2
y = 5          # do promenne y prirad cislo 5
print(x + y)   # spocitej a vypis soucet hodnot promennych
print(type(x))
```

```
7
<class 'int'>
```

# Opakování

(podrobněji později)

```
for i in range(5):  
    print('Hello!', i)
```

```
Hello! 0  
Hello! 1  
Hello! 2  
Hello! 3  
Hello! 4
```



# Funkce

```
# definice funkce pro vypis obvodu ctverce,  
# jehoz delka strany je 'side'  
def print_square_perimeter(side):  
    perimeter = 4 * side  
    print(perimeter)  
  
# volani funkce pro delky strany 50 a 100  
print_square_perimeter(50)  
  
# urceni parametru pres jmeno  
print_square_perimeter(side=100)
```

200

400

# Knihovny

```
# z knihovny math importujeme funkci sqrt (odmocnina) a konstantu pi  
from math import sqrt, pi  
  
print(sqrt(4))  
print(pi)
```

Případně import modulu:

```
# importujeme modul math  
import math  
  
# k členům se odkazujeme pomocí "tečkove notace"  
print(math.sqrt(4))  
print(math.pi)
```

# Dobré praktiky

- Psát čitelný, přehledný kód.
- Používání `CTRL+C` / `CTRL+V` zavání špatným návrhem.
- Nevytvářet "dlouhé" metody.
- Podrobněji bude zmíněno na některé z přednášek.

# Želví grafika

```
# import tridy Turtle pro zelvi grafiku
from turtle import Turtle, done

# vytvoreni nove zelvy jmenem julie
julie = Turtle()

# kresleni
julie.forward(100) # popojdi dopredu o 100 px
julie.left(90) # zatoc doleva o 90 stupnu
julie.forward(50) # popojdi dopredu o 50 px
julie.right(135) # zatoc doprava o 135 stupnu
julie.forward(25) # popojdi dopredu o 25 px

# ukonci kresleni, lze zavrit okno
done()
```

# Želví grafika - doplňující metody

```
speed(10)      # urychli vykreslovani
tracer(False)  # vypne postupne vykreslovani
clear()        # vymaze kreslici plochu
reset()        # vymaze plochu a vrati zelvu doprostred
bye()          # zavre okno
exitonclick() # zavre okno pri kliknuti na nej
done()        # ukonci vykreslovani a okno staci zavrit
```

Dokumentace:

<http://docs.python.org/library/turtle.html>

# Příklady:

Předlohy obrázků:

[cviko01-zelvi-grafika.pdf](#)

# Příklady (1):

## Sbírka 1.1 Rozcvička

- 1.1.2. Čtverec:

Využijte for cyklus pro nakreslení čtverce o délce strany 100 pixelů.

- 1.1.3. Obecný čtverec

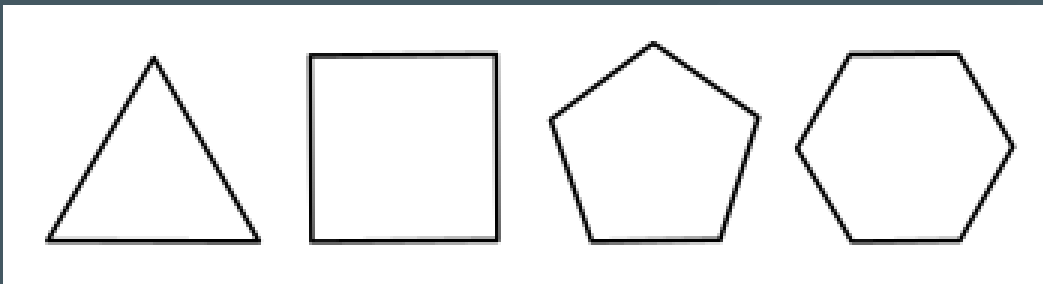
Napište funkci pro vykreslení čtverce s danou délkou strany.

# Příklady (2):

## 1.2. Pokročilé kreslení

- 1.2.1. Mnohoúhelníky

Napište obecnou funkci pro vykreslení libovolného pravidelného  $n$ -úhelníku.



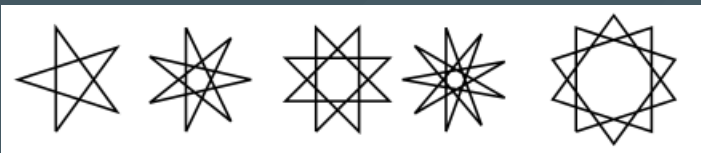


# Příklady (3):

## 1.2. Pokročilé kreslení

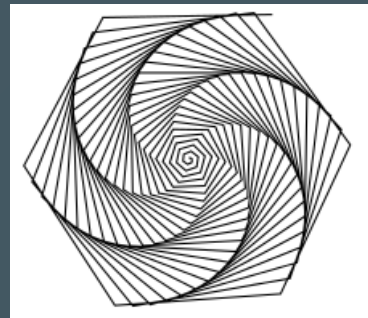
- 1.2.2. Hvězdy

Napište obecnou funkci pro vykreslení hvězdy. Hvězda je zobecněním pravidelného  $n$ -úhelníka, kde želva nenavštívuje bezprostředně sousední vrcholy, ale “přeskakuje”. Délka skoku je daná parametrem  $step$ , ten je např. pro první, pěticípou hvězdu roven 2 a pro druhou, sedmicípou hvězdu roven 3. Při  $step = 1$  půjde o  $n$ -úhelník.



# Příklady (procvičení):

## 1.2.4. Spirála



## 1.2.6. Kytky

