



FAKULTA  
INFORMATIKY  
Masarykova univerzita

# IB111 Základy programovania

---

## Cvičenie 1 – organizácia & korytnačka

Matej Troják

(Poďakovanie: Valdemar Švábenský)

Uvítanie  
&  
Zoznámenie

# Organizácia cvičení

## Pravidlá:

- Začínáme presne, končíme :40 (100 minút)
- Čítajte materiály, e-maily a diskusné fórum
- Otázky: hlavne fórum v ISe, osobné/tajné veci e-mail/osobne
- Venujte pozornosť cvičiacemu (“v tichosti počúvam”)
- “Ste už veľkí”

## Študijné materiály

<https://is.muni.cz/auth/el/fi/podzim2019/IB111/um/trojak/>

# Prednášky a cvičenia, hodnotenie

- Prednáška: nepovinná, ale doporučovaná
- Cvičenia: povinné; max. 2 neospravedlnené absencie
- Hodnotenie: 200 bodov, z toho 100 počas semestra
  - Vnútrosemestrálka: 14 b (viac info dám vopred)
  - **Veľké domáce úlohy:** 75 b (5 úloh: 8, 8, 10, 12, 12)
    - automatická aj ručná kontrola
  - **Malé domáce úlohy:** 11 b (odpovedník v ISe každý týždeň)
    - automatická kontrola, zverejnený vždy v pondelok ráno, deadline je v utorok ďalší týždeň
- Podmienky: <http://www.fi.muni.cz/IB111/>
  - Splnená dochádzka
  - Aspoň 5 bodov za malé domáce úlohy
  - Aspoň 3 body z automatických testov z *každej* veľkej domácej úlohy
  - Aspoň 3 body z vnútrosemestrálky
  - Aspoň 60 bodov zo *semestru* celkom

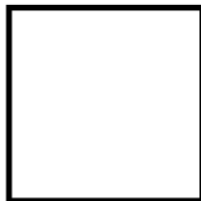
# Programovanie – motivácia

Úloha: nakreslite štvorec s obmedzenými operáciami.

Povolené operácie

- dopredu( $n$ )
- otoč\_doprava( $\alpha$ )
- otoč\_dol'ava( $\alpha$ )

$n$  – dĺžka;  $\alpha$  – uhol



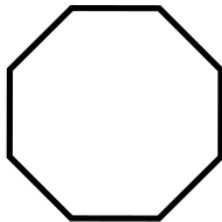
# Programovanie – motivácia

Úloha: nakreslite osemuholník s obmedzenými operáciami.

Povolené operácie

- dopredu( $n$ )
- otoč\_doprava( $\alpha$ )
- otoč\_dol'ava( $\alpha$ )

$n$  – dĺžka;  $\alpha$  – uhol



Programátor – tvor lenivý

# Programovací jazyk Python

- Jednoduchá syntax
- Minimum kódu (takmer pseudokód)
- Dobrá čitateľnosť kódu
- Rozsiahla štandardná knižnica
- Platformová nezávislosť
- Multiparadigmový: imperatívny, procedurálny, prvky OOP, funkcionálne prvky

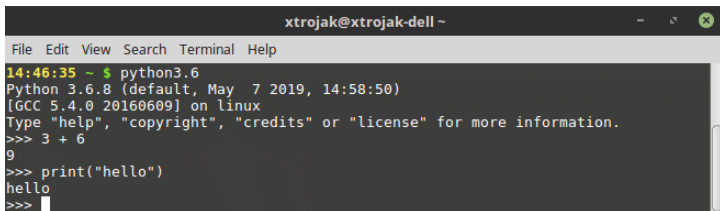


# Programovacie prostredie – PyCharm

- **PyCharm** – profesionálny Integrated DeveLopment Environment (IDE) pre Python
  - Interaktívny interpret príkazov (shell): jednorazové, pre krátke rýchle výpočty
  - Tvorba a spúšťanie skriptov (.py): uloženie práce, opakované spustenie programu
- používajte čo chcete... (napr. akýkoľvek textový editor + shell)

# Shell – interpreter

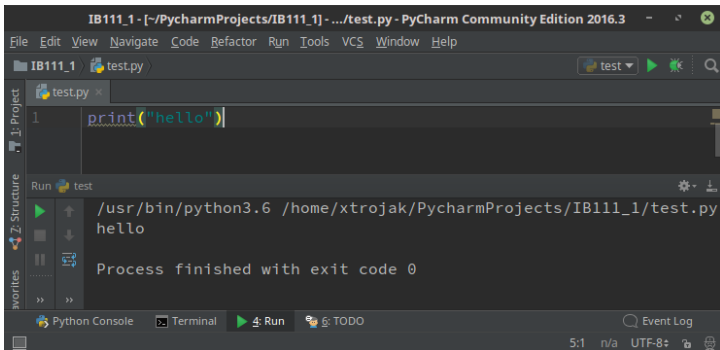
- Čakanie na príkaz: > > > („čo mám spraviť teraz?“)
- Vyhodnotenie jedného riadka príkazu: Enter
- Poslúži aj ako kalkulačka



```
xtrojak@xtrojak-dell ~  
File Edit View Search Terminal Help  
14:46:35 ~ $ python3.6  
Python 3.6.8 (default, May 7 2019, 14:58:50)  
[GCC 5.4.0 20160609] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> 3 + 6  
9  
>>> print("hello")  
hello  
>>> 
```

# PyCharm – spustenie

- zapneme PyCharm
- vytvoríme nový projekt (File → New Project... | Python 3.6+ !)
- vytvoríme nový súbor (File → New... → Python File)
- do textového editoru napíšeme kód
- spustíme kód (Run → Run... | alebo pomocou Shellu...)



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python file named 'test.py' with the following code:

```
1 print("hello")
```

Below the editor, the Run console is open, showing the execution command and output:

```
Run test
/usr/bin/python3.6 /home/xtrojak/PycharmProjects/IB111_1/test.py
hello
Process finished with exit code 0
```

The status bar at the bottom indicates the current file is 'test.py', the encoding is 'UTF-8', and the cursor is at line 5, column 1.

# Korytnačia grafika

# Korytnačia grafika

- Korytnačka má:
  - Pozíciu (súradnice)
  - Otočenie
  - Vlastnosti pera (farba, hrúbka čiary, . . . )
- Ovládanie
  - `from turtle import Turtle`
    - Sprístupní príkazy pre ovládanie korytnačky
  - `fred = Turtle()` - vytvorí novú korytnačku, pripravenú na príkazy
  - `fred.reset()` - nastaví korytnačku na počiatočnú pozíciu
  - `fred.forward(x)` - pohyb o x pixelov v danom smere
  - `fred.right(x)` - rotácia o x stupňov v smere hodinových ručičiek
  - `fred.left(x)` - rotácia o x stupňov proti smeru hodinových ručičiek

# Korytnačia grafika

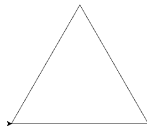
## Pokročilé ovládanie

- `speed(x)` – rýchlosť `x` vykresľovania
- `tracer(False)` – vypne postupné vykresľovanie
- `clear()` – vymaže kresliacu plochu
- `reset()` – vymaže plochu a vráti korytnačku doprostred
- `bye()` – zavrie okno
- `exitonclick()` – zavrie okno pri kliknutí
- `done()` – ukončí vykresľovanie a okno stačí zavrieť
- `pendown()` – režim aktívneho kreslenia
- `penup()` – režim neaktívneho kreslenia

## Trojuholník – sekvencia príkazov (level 1/5)

```
>>> from turtle import Turtle
>>> fred = Turtle()

>>> fred.forward(100)
>>> fred.left(120)
>>> fred.forward(100)
>>> fred.left(120)
>>> fred.forward(100)
>>> fred.left(120)
```



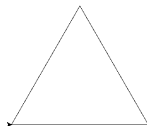
## Trojuholník – príkazy v skripte (level 2/5)

- Shell je jednorazový: keď ho vypneme, program sa stratí
- Príkazy ale nechceme zakaždým písať znova
- Ak si vytvoríme skript, dá sa uložiť a spustiť neskôr

```
from turtle import Turtle, exitonclick
fred = Turtle()

fred.forward(100)
fred.left(120)
fred.forward(100)
fred.left(120)
fred.forward(100)
fred.left(120)

exitonclick()
```

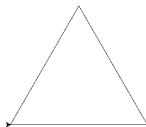




## Trojuholník – definícia vlastnej procedúry (level 3/5)

- Chceme skryť postupnosť 6 príkazov pod 1 nový príkaz
- Píšeme do nového skriptu
- Dvojbodka a zanorenie (odsadenie!)

```
def triangle():  
    fred.forward(100)  
    fred.left(120)  
    fred.forward(100)  
    fred.left(120)  
    fred.forward(100)  
    fred.left(120)
```



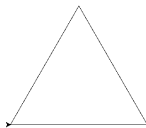
- Skript spustíme a potom napíšeme v shelli:

```
>>> triangle()
```

## Trojuholník – opakované príkazy v cykle (level 4/5)

- Príkaz `for i in range(3):` spôsobí 3 opakovania
- Pozor na odsadenie

```
def triangle():  
    for i in range(3):  
        fred.forward(100)  
        fred.left(120)
```

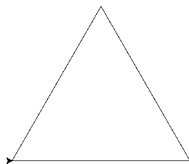


## Trojuholník – nastaviteľná dĺžka (level 5/5)

- Parameter *funkcie*

```
def triangle(length):  
    for i in range(3):  
        fred.forward(length)  
        fred.left(120)
```

```
>>> triangle(200)
```



## Zhrnutie príkazov

- `reset()` nastaví korytnačku na počiatočnú pozíciu
- `forward(x)` pohyb o x pixelov v danom smere
- `left(x)` rotácia o x stupňov proti smeru hodinových ručičiek
- `right(x)` rotácia o x stupňov v smere hodinových ručičiek
- `speed(x)` rýchlosť x vykresľovania
- `tracer(False)` vypne postupné vykresľovanie
- `clear()` vymaže kresliacu plochu
- `reset()` vymaže plochu a vráti korytnačku doprostred
- `bye()` zavrie okno
- `exitonclick()` zavrie okno pri kliknutí
- `done()` ukončí vykresľovanie a okno stačí zavrieť
- `pendown()` režim aktívneho kreslenia
- `penup()` režim neaktívneho kreslenia

# Precvičovanie na doma

- <http://www.fi.muni.cz/IB111/sbirka/>
- <https://www.umimeprogramovat.cz/python-zelva>
- <https://www.umimeprogramovat.cz/programovani-v-pythonu>