

ZAL – 2. cvičení

2016

Algoritmus

- Algoritmus je posloupnost kroků, které jsou potřeba k vyřešení problému.
- Algoritmus na vstupu přijímá instanci problému a na výstupu dává řešení zadané instance.
- U algoritmu ověřujeme správnost a konečnost.
- Nekonečný algoritmus by neměl smysl - nezískali bychom řešení instance problémů.



Ukázka

- Je algoritmus správný?

funkce isSkolniDen(den)

kdyz(den == 'sobota' | | den == 'nedele')

NE

jinak

ANO

Ukázka - rozuzlení

funkce `isSkolniDen(den)`

`kdyz(den == 'sobota' || den == 'nedele')`

`NE`

`jinak`

`ANO`

ANO, pokud má funkce vrátit den kdy obvykle chodíme do školy.

NE, pokud má funkce vrátit zdali musím dnes do školy – šel snad někdo z vás 1. září do školy?

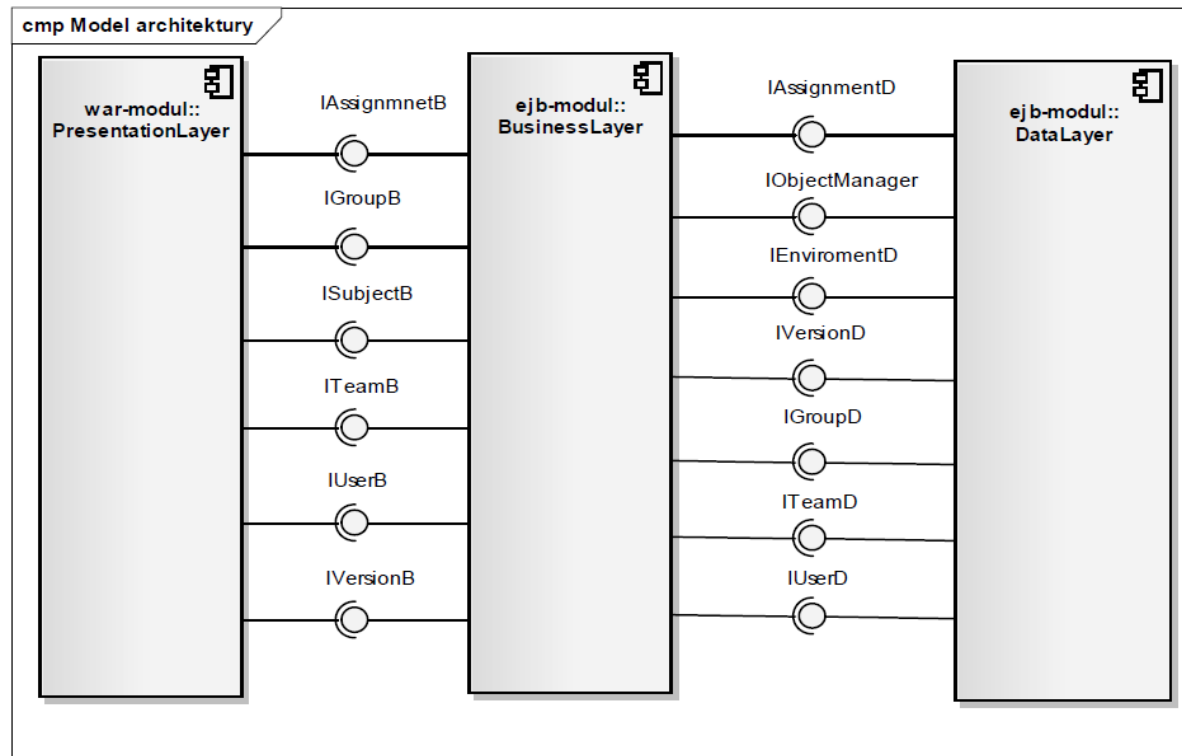


Software

- Software jsou logicky spolupracující funkční bloky, které slouží k vykonání požadované funkcionality.
- Obvykle rozdělen do tříd a balíčků. Balíček je kontejner pro několik tříd či balíčků, které jsou zodpovědné za jednu část funkcionality SW.
- Balíčky by měli být dobře pojmenovány. Obvykle se doporučují malá písmena – code conventions.
- Třídy jsou součástí balíčků a implementují nebo definují funkcionality, kterou poskytují.

Software - pokračování

Rychlá ukázka – jednotlivých vrstev



Proměnné 1

- Program si musí v paměti uchovávat informace (mezi výpočty a pod).
- Programátor potřebuje k těmto datům přístup a musí s nimi operovat.
- Programátor proto ukládá tyto data to proměnných, s kterými pak pracuje
- Jaké by mohli být operace s proměnnými?

Proměnné 1 - operace

- Definice proměnné – alokace v paměti
- Přiřazení (úprava či vložení dat)
- Získání hodnoty, adresy v paměti, typu apod
- Přetypování
- Smazání
- Odstranění s paměti
-

Proměnné 1 – práce s nimi

- Nejprve je potřeba proměnou vytvořit a nastavit ji počáteční hodnotu. V Pythonu velice jednoduché:

```
name = 'Karel Gott'
```

- Nyní lze s proměnou pracovat. Například vytisknout její obsah:

```
print(name) # vytiskne Karel Gott
```

- Můžeme ji přiřadit i jinou hodnotu:

```
name = 'Mistr Gott'
```

- A vytisknout novou hodnotu:

```
print(name) #vytiskne mistr
```



Standardní vstup / výstup

- Standardní vstup je místo, kde program očekává vstupní data (instanci problémů).
- Standardní výstup je místo, kam program tiskne výstupy (řešení problému).
- Jsou jimi: Vstup z klávesnice, senzorů, souborů, funkcí a dalších systémů.
- Otázka: Jaké jsou výstupy?



Jak načíst vstup?

- Existuje velké množství způsobů. Například:
`name = raw_input("Napiste vase jmeno: ")` – funkce `raw_input` (Python 2.7)
- `name = input("Napiste vase jmeno: ")` – funkce `input` (Python 3.x)
- Předání pomocí pipe – puštění z `cmd / shell`:
OS Windows: `type vstup.txt | print.py`
OS Linux: `cat vstup.txt | print.py`
Vstup je v souboru `vstup.txt`, `print.py` musí obsahovat blog kódu, který tento předaný vstup zpracuje.

`data = sys.stdin.readlines()` #Otázka – co znamená `sys`?

Kdo jsem?

- Vytvořte si algoritmus, který načte ze standardního vstupu (klávesnice) svoje jméno.
- Toto jméno vytiskne na standardní výstup (konzole/obrazovka)

Kdo jsem - řešení



whoAml.py

Kdo jsem - vysvětlení

- `#!/usr/bin/env python`
 - Umístění s pythonem - doporučený zápis
- `whoAml = input("Kdo jsem? ")`
 - Načtení vstupu a přiřazení do proměné `whoAml` – na obrazovku vytiskne text `Kdo jsem?`
- `print("Jsem: " +whoAml)`
 - Vytisknutí informace a zřetězení vstupu.

Třídy, struktury, moduly- základ

- Třídy a struktury poskytují operace, které může uživatel použít.
- Klíčové slovo `import SomeClass`
- Volání funkce z importované třídy:
`SomeClass.someFunction()`
- Co je to funkce uvidíte na příštím cvičení.



Práce se soubory - úvod

- Soubory můžeme, číst, vytvářet, přepisovat a mazat. Python poskytuje vestavěné funkce. Stačí mít nainportovaný základní modul `sys`.
- Se soubory se obvykle pracuje jako se streamem a ten je vždy potřeba zavírat – proto to za nás Python umí dělat automaticky 😊
- Předtím, než se začne se souborem pracovat tak je potřeba mu říci v jakém režimu chceme soubor používat.

Práce se soubory - ukázka

- **with** open('example.txt', 'wt') **as** outputFile:
outputFile.write(textTofile)
- **with** open('example.txt', 'r') **as** inputFile:
data = inputFile.readlines()
- Důležité postřehy:
 - w->Write
 - r->Read
 - t->text mode

Funkce

- Funkce přijímá na vstupu 0-X parametrů a na výstupu poskytuje 0-X hodnot. Algoritmus se dělí do funkčních bloků z nichž každý poskytuje určitou část a do hromady tvoří funkční celek – to usnadňuje vývoj a ladění programu.
- Funkce má vždy definovanou svojí hlavičky

```
def functionname( parameters ):  
    "function_docstring"  
    function_suite  
    return [expression]
```



Funkce ukázka – kdo jsem

```
def printNameAndAge(name):  
    age = input("How old are you? ")  
    print("I am: "+name+"I am:"+str(age)+ " years old")  
    nameAndAge = name + str(age)  
    return nameAndAge
```

```
result = printNameAndAge('John')  
print('Returned from function:'+result)  
result = printNameAndAge('Jaina')  
print('Returned from function:'+result)
```

Důležité: V Pythonu jsou funkční bloky odděleny mezerami či tabulátory.

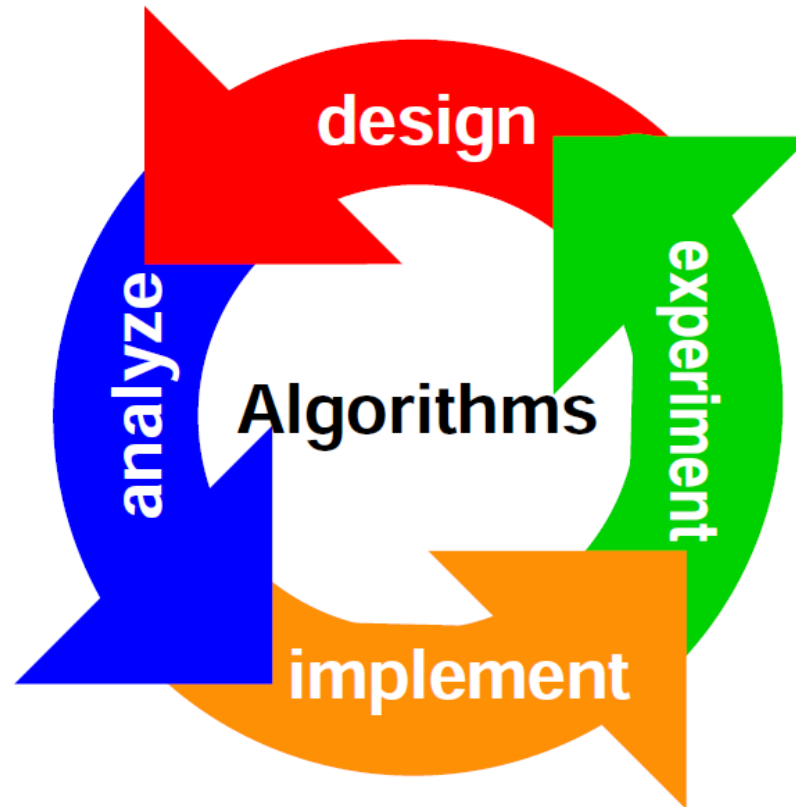


2. domácí úloha

- Zadání úlohy je zde:
https://cw.fel.cvut.cz/wiki/courses/b6b36zal/zadani/2_python_in_action
- Vypracujte ji a nahrajte do odevzdávacího systému.
- Systém vám poskytne zpětnou vazbu a bodově ji ohodnotí.
- Pozn: Otestuje, že správně spojujete čísla a texty.



Končíme otázky?



Obrázek převzat z: <http://shankhdhar143.blogspot.cz/>