

Druhá domácí úloha

- Deadline: 16. 10. 2018 23:59
- Odevzdání do odevzdáárny v ISe:
Skupina 03 [Lachman]/Domácí úkol 2
- ~~soft-deadline~~

Jeden soubor `UČO.py`

- Prohlášení a sebehodnocení ([Pokyny k domácím úlohám](#))

```
"""
Autor: Jméno Příjmení, UČO

Prohlašuji, že celý zdrojový kód jsem zpracoval(a) zcela s
Jsem si vědom(a), že nepravdivost tohoto tvrzení může být
"""

def my_hw_function():
    """
    Description.

    Známé nedostatky: Rámcově funguje, ale nevykresluje př
    Styl: Příliš kryptická jména funkcí a proměnných.
    """
```

Druhá domácí úloha

Úkol 1:

Vytvořte simulátor zjednodušené hry "Člověče nezlob se".

- Hraje se na hracím plánu o n polích.
- Figurka začíná na jedné straně hracího plánu.
- Háže se kostkou (1-6), když padne 6, hází se znovu. Pokud padne lichý počet šestek, figurka zůstává stát.
- Figurka se posunuje o součet hodnot z kostek, zůstává stát ocitla-li by se za cílem.
- Hra končí když figurka dorazí na poslední pole.

Druhá domácí úloha

Poznámky:

- Při délce hracího plánu < 1 skončete funkci a vypište nějakou smysluplnou hlášku.
- Figurka se posunuje buď o celkový součet hodů na kostkách, nebo vůbec (v situacích, kdy by přešla domeček, nebo padl lichý počet šestek).

Druhá domácí úloha

- Ukázkový výpis programu (mezi závorkami je mezera):

```
>>> board_game(16)
```

```
()(X)()()()()()()()()()()()()() [1]  
()(X)()()()()()()()()()()()()() [6][6][6][2]  
()()()()()()()()()()()()()()()X() [6][6][1]  
()()()()()()()()()()()()()()()X() [5]  
()()()()()()()()()()()()()()()X() [1]
```

The game ended in round 5.

Druhá domácí úloha

- Vytvětlení předchozího příkladu:
 - Začínáme na prvním poli.
 1. Padlo **1**, posun o jedno místo.
 2. Padl lichý počet **6** a **2**, stojím na místě.
 3. Padl sudý počet **6** a **1**, posun o 13 polí.
 4. Padlo **5**, přeskočili bychom cíl (poslední pole), zůstáváme stát.
 5. Padlo **1**, skočíme na poslední pole

Druhá domácí úloha

Úkol 2:

Vytvořte funkci `game_analysis`, který analyzuje jaká je průměrná délka (vrátí typ `float`) hry pro zadaný počet polí a počet her.

Druhá domácí úloha

Úkol 3:

Vytvořte funkci `game_average_length`, která pomocí funkce `game_analysis` z úkolu 2 zjistí průměrné délky her pro plány o velikostech 1-50 (do řešení přiložte/zkopírujte výpis).

Druhá domácí úloha

Kostrá

```
from random import randint, random

def board_game(length, output = True):
    pass

def game_analysis(length, count):
    pass

def game_average_length(count):
    pass
```

Druhá domácí úloha

- rozumné množství rozumných komentářů
- vhodné členění kódu na funkce
- bez zbytečných výpisů
- správné názvy funkcí
- úprava kódu:
 - `pycodestyle` (PEP8)
 - `pylint` (přísnější než PEP8, ale lehce přehlednější výstup)