IB111/03 Vnitrosemestrání písemka 4. 12. 2018

- 5 příkladů po cca 20 bodech, **100 minut** čistého času
- odevzdání nahrát v jednom souboru do odevzdávárny

1. Počet nenulových prvků v matici

- Naprogramujte funkci `zero_count(matrix)`, která vrátí počet nenulových prvků v matici.
- Matice je reprezentována seznamem seznamů.

2. Délka řetězce (rekurze)

- Naprogramujte funkci `recursive_len(text)`, která vypočte délku řetězce.
- Funkce musí být zapsána **rekurzivně** (bez použití `for` / `while` cyklů).

3. Kámen-nůžky-papír (simulace)

- Napište funkci `jan_ken_pon(count)` simulující `count`-her kámen-nůžky-papír s následujícími pravidly:
  - Jedna hra probíhá následovně:
    - Oběma hráčům vygenerujeme volbu (`rock` / `paper` / `scissors`).
    - Každá volba (`rock` / `paper` / `scissors`) má stejnou pravděpodobnost.
    - U každé hry nás zajímá počet zvolení kemene (`rock`). (ne kdo vyhrál)
- Finálně vypíšeme průměrný, nejmenší a největší počet kamenů na hru.
- Kód je třeba **dekomponovat** na dílčí funkce!

```
>>> jan_ken_pon(6)
average: 3.2
min: 0
max: 7
```

4. Frekvence knih (řazení, datové struktury)

Naprogramujte funkci `book_frequency(books)`.

- Na vstup dostane řetězec následujícího tvaru (Pobočky oddělené středníkem, knihy čárkou. Bloky dvojtečkou.):

```
>>> books = "Brno Střed:Hobbit,Name of the Rose,1984,Robin Hood;Obřany:;"
+ "Řečkovice:Mystery of the teaser,Hobbit;Martin:1984,Hobbit"
>>> print(book_frequency(books))
['Hobbit', '1984', 'Name of the Rose', 'Robin Hood', 'Mystery of the teaser']
```

- **Výstupem** jsou knihy seřazené dle množství výskytů (sestupně).

5. Studenti a Předměty (Objekty)

Mějme následující definici tříd `Student` a `Course`:

```
class Student:

    def __init__(self, name, uco):
        self.name = name
```

```
        self.uco = uco

class Course:

    def __init__(self, code, room):
        self.code = code
        self.students = []
        self.room = room

    def add_student(self, student):
        self.students.append(student)
```

1. Změňte třídu student, aby uchovávala zvlášť jméno a příjmení. (`name` / `surname`)
2. Zadefinujte třídu `Room`, která má atribut `code` a `capacity`. (Jsou třeba k vytvoření instance.)
3. Napište funkci `capacity_checker(courses)`, která pro seznam kurzů vypíše `Ano` / `Ne`, zda se studenti vejdou do učebny.

```
d1 = Room(code="D1", capacity=100)
d3 = Room(code="D3", capacity=80)
a218 = Room(code="A218", capacity=2)

ib111 = Course(code="ib111", room=d1)
pb150 = Course(code="pb150", room=d3)
mb201 = Course(code="mb201", room=a218)

paul = Student("Paul","McCartney", "123456")
george = Student("George", "Harrison", "4321")
john = Student("John","Lennon", "43671")

ib111.add_student(paul)
ib111.add_student(george)

mb201.add_student(paul)
mb201.add_student(george)
mb201.add_student(john)

>>> capacity_checker([ib111, pb150, mb201])
ib111: Ano
pb150: Ano
mb201: Ne
```