

Lista 6

Każde zadanie należy umieścić w osobnym pliku `imie.nazwisko.XX.YY.py` gdzie `XX` to numer listy a `YY` numer zadania.

Każda funkcja musi zawierać dokumentację docstring - za brak -1 pkt za zadanie.

Uwaga: docstring powinien opisywać działanie funkcji, np.:

```
def delta(a, b, c):  
    """Zwraca wyróżnik trójmianu kwadratowego o współczynnikach a, b, c."""  
    return b**2 - 4*a*c
```

jest dobrym opisem, podczas gdy poniższy opis czyni dokumentację bezużyteczną:

```
def delta(a, b, c):  
    """Zwraca  $b*b - 4*a*c$ ."""  
    return b**2 - 4*a*c
```

Zadanie 1 (5 pkt)

Napisz program do nauki matematyki. Na starcie program powinien drukować ekran powitalny oraz menu, np:

```
#####  
# KURS MATEMATYKI #  
#####
```

1. Dodawanie
2. Odejmowanie
3. Mnożenie
4. Dzielenie
5. Wyjście

Wybierz działanie:

Po wyborze działania program powinien wylosować dwie liczby oraz zapytać o wynik, np

2 + 2 =

Po podaniu wyniku przez użytkownika program sprawdza, czy wynik jest poprawny oraz drukuje odpowiedni komunikat. Program działa w pętli aż zostanie wybrana opcja "Wyjście".

Zadanie 2 (5 pkt)

Przeanalizuj poniższy kod:

```
import random

imiona = ("Kasia", "Basia", "Marek", "Darek")
nazwiska = ("Nowak", "Burak", "Smith", "Doe")
przedmioty = ("Matematyka", "Fizyka", "Chemia")

def generuj_studenta():
    """Funkcja zwraca losowe imię i nazwisko"""
    return "{} {}".format(random.choice(imiona), random.choice(nazwiska))

def generuj_dziennik(n):
    """Funkcja generuje n studentów i przypisuje im losowe oceny"""
    studenci = []

    for i in range(n):
        # dodaje losowo wygenerowanego studenta z unikalnym id
        student = {
            "id": i,
            "student": generuj_studenta()
        }

        # generuje losowe oceny
        for przedmiot in przedmioty:
            student[przedmiot] = random.randint(2, 5)

        # dodaj studenta z ocenami do dziennika
        studenci.append(student)

    return studenci

def drukuj_dziennik(studenci):
    """Drukuje listę studentów wraz z ocenami"""
    for student in studenci:
        print("{}- {}".format(student["id"] + 1, student["student"]))

        for przedmiot in przedmioty:
            print("\t- {}: {}".format(przedmiot, student[przedmiot]))
```

- Dodaj funkcję, która liczy średnią ocen dla podanego studenta (na wejściu przyjmuje słownik z ocenami i zwraca nowy słownik ze średnią), np.

```
{'id': 0, 'student': 'Marek Smith', 'Matematyka': 5, 'Fizyka': 2, 'Chemia': 2}
-> {'id': 0, 'student': 'Marek Smith', 'Średnia': 3}
```

Wsk. `help(dict.pop)`

- Dodaj funkcję, która liczy średnią ocen dla każdego studenta z podanej listy (wykorzystując funkcję z pierwszego punktu).
- Dodaj funkcję, która drukuje na ekranie ranking studentów (zaczynając od tych z najwyższą średnią).

Zadanie 3 (5 pkt)

Napisz program do kompresji i dekompresji ciągu znaków, który zawiera poniższe funkcje:

- `compress(string)`, która
 - zwraca `None`, gdy podany ciąg zawiera znaki inne niż litery (*wsk.* `help(str.isalpha)`)
 - lub zwraca skompresowany *string* wg reguły
AAABBBBCAAAaaDD -> A3B4CA3a2D2
- `decompress(string)`, która odtwarza oryginalny tekst ze skompresowanego
- `test(n)`, która
 - generuje *n* losowych ciągów znaków (*wsk.* `string.ascii_letters`)
 - sprawdza, czy dekompresja skompresowanego ciągu znaków zwraca oryginał