Lista 10

Uwaga. W obu zadaniach należy napisać testy jednostkowe z użyciem modułu `unittest`.

Zadanie 1 (1 punkt + 1 punkt za testy jednostkowe). Napisz **własną**[1] klasę `Matrix` służącą do reprezentowania macierzy prostokątnych. Zaimplementuj metody:

- `__init__(self, values)` – parametr `values` to lista wierszy macierzy (każdy wiersz to lista liczb[2]). Jeśli wiersze są różnej długości, inicjalizator ma rzucić stosowny wyjątek.
- `dim(self)` – zwraca parę liczb reprezentującą wymiary macierzy `self`.
- `transpose(self)` – zwraca macierz będącą transpozycją `self`.
- `__str__(self)` – zwraca napis reprezentujący macierz `self`.
- `__neg__(self)` – zwraca macierz odpowiadającą $-$`self`.
- `__add__(self, other_matrix)` – zwraca macierz odpowiadającą `self` + `other_matrix`. Jeśli macierze są różnych rozmiarów, metoda ma rzucić wyjątek.
- `__sub__(self, other_matrix)` – zwraca macierz odpowiadającą `self` $-$ `other_matrix`. Jeśli macierze są różnych rozmiarów, metoda ma rzucić wyjątek.
- `__mul__(self, other_matrix)` – zwraca macierz odpowiadającą `self` $\cdot$ `other_matrix`. Jeśli macierze nie są kompatybilnych wymiarów, metoda ma rzucić wyjątek.
- `__eq__(self, other_matrix)` – zwraca `True`, gdy `self` i `other_matrix` są równe, `False` w przeciwnym wypadku.

Następnie napisz testy jednostkowe **każdej** z tych metod.

Zadanie 2 (1 punkt + 1 za testy jednostkowe). Uzupełnij klasę `Bank` z wykładu[3] dodając metody realizujące następujące działania (nadaj im odpowiednie nazwy i parametry):

- Zasilenie każdego konta w banku kwotą podaną jako argument metody.
- Zwrócenie listy kont (instancji `BankAccount`), których saldo nie przekracza wartości podanej jako argument metody.
- Usunięcie wszystkich kont o zerowym saldzie.
- Zsumowanie stanów wszystkich kont w banku i zwrócenie tej wartości.
- Przelanie wszystkich pieniędzy z wszystkich kont banku na podane w argumencie konto (instancję `BankAccount`, potencjalnie z innego banku).
- Przeniesienie wszystkich kont z banku do nowego banku podanego jako argument. Przeniesione konta mogą przyjąć w nowym banku nowe numery. Po operacji, bank oryginalnie zawierający konta powinien być pusty (nie posiadać kont). Konta które istniały w nowym banku przed operacją nie powinny ulec zmianie.

Następnie napisz testy jednostkowe każdej z metod klasy `Bank` (dotyczy również metod z wykładu).

[1] W implementacji tej klasy nie możesz używać analogicznej klasy z bibliotek Pythona.
[2] Ogólniej: lista elementów pewnego pierścienia.
[3] `https://www.math.uni.wroc.pl/~jagiella/files/p1python/bank.zip`