

## Wyłudzanie pożyczek

Nasz bank ma problem z ludźmi którzy zakładają konta i próbują wielokrotnie zaciągnąć pożyczki. Musimy wprowadzić funkcjonalność pozwalającą prowadzić rejestr zablokowanych kont ("black list"). Skupiamy się tylko na kontach osobistych.

Szczegóły:

- rejestr będziemy przetrzymywać w bazie danych (np. mongo lub postgres)
- w rejestrze będziemy przechowywać obiekty w postaci `{'pesel': $pesel, 'reason': "reason for blocking account"}`
- potrzebujemy nowej klasy (np. **BlackList**)
- powinniśmy mieć możliwość dodania peselu do listy oraz sprawdzenia czy pesel już znajduje się na naszej liście. W nowej klasie potrzebujemy dwóch metod:
  - `add_account_to_black_list(self, pesel, reason)`
  - `is_account_on_black_list(self, pesel)`
- jeżeli pesel jest na naszej liście - nie udzielamy takiej osobie kredytu
- jeżeli pesel nie ma na liście - sprawdzamy pozostałe warunki (tak jak do tej pory)

Instrukcja:

- odpal dowolną bazę danych
- stwórz klasę **BlackList**
- zaktualizuj testy na zaciąganie pożyczki (dopisz test na obecność peselu na naszej liście) oraz wymagane metody
- "zamockuj" testy tak aby nie wysyłały requestów do bazy danych

Do odpalenia Mongo możesz użyć docker compose:

```
services:
  mongo:
    image: mongo:4.1
    container_name: mongo
    ports:
      - "27017:27017"
```

Odpalamy go za pomocą polecenia: ***docker compose -f mongo.yml up***

Do stworzenia instancji obiektu możesz użyć:

```
def __init__(self, db_name='test_db', collection_name='test_collection'):
    self.client = MongoClient('localhost', 27017)
    self.db = self.client[db_name]
    self.black_list_collection = self.db["black_list"]
    self.collection = self.db[collection_name]
```

