

Kontynuujemy feature z poprzednich zajęć. Używając podejścia data driven napisz testy i zaimplementuj poniższy feature

Feature 13 Kredyt Firmowy:

Wprowadzamy usługę kredytów dla kont firmowych. Potrzebujemy zaimplementować metodę **zaciagnij_kredyt()** w klasie KontoFirmowe.

Warunki przyznania kredytu:

- saldo na rachunku musi być przynajmniej 2 razy większe niż kwota zaciąganego kredytu, **oraz**
- w historii konta musi być przynajmniej jeden przelew do ZUS (dla uproszczenia założmy tutaj, że warunek jest spełniony gdy w historii znajduje się przynajmniej jeden przelew wychodzący w kwocie dokładnie 1775 zł).

Pisząc testy używajmy parametryzacji.

Zastanów się czy możliwy jest dodatkowy refaktor.

Feature 14 Rejestr kont:

Nasz zespół został poproszony o zaimplementowanie Rejestru kont osobistych. Na chwile obecną będzie to MVP (Minimum Viable Product) więc staramy się osiągnąć cel w najprostszy możliwy sposób (bez użycia zewnętrznej bazy danych).

Zaimplementuj klasę **AccountsRegistry** która będzie:

- przechowywała konta osobiste w liście
- funkcję, która pozwoli na dodawanie konta do tej listy
- funkcję, która pozwoli na wyszukanie konta za pomocą peselu
- funkcję, która zwróci nam ilość kont w rejestrze

W późniejszym etapie możliwe będzie wymiana tej funkcjonalności na pełnoprawną bazę danych.

Wszystkie funkcje i lista powinny być dostępne z poziomu klasy (do funkcji używamy dekoratora `@classmethod`).

Napisz testy, a potem implementację. Zrób push do remote i sprawdź, czy testy przechodzą.

Refaktor:

Zastanów się czy możesz używając metod **setUp**, **setUpClass** oraz **tearDown** oraz **tearDownClass** (nie trzeba używać wszystkich) aby Twoja klasa testowa była bardziej czytelna