

## Laboratorium 5 – Fixtures i parametryzacja

### Wprowadzenie

Na dzisiejszych zajęciach zaimplementujemy kolejną funkcjonalność oraz wprowadzimy narzędzia ułatwiające **organizację grup testów**.

---

### Feature 12 – Zaciąganie kredytu

Nasz bank wchodzi na rynek kredytów osobistych.

Naszym **zadaniem** jest stworzenie **funkcjonalności** odpowiedzialnej za zaciąganie kredytów.

Na chwilę obecną bank będzie udzielał kredytów wyłącznie **osobom prywatnym**, dlatego funkcjonalność wprowadzamy tylko w **kontach osobistych**.

Nasze konto powinno posiadać funkcję `submit_for_loan(amount)`.

#### Warunki udzielenia kredytu

Kredyt może zostać udzielony lub nie. Warunki przyznania są następujące:

1. Ostatnie trzy zaksięgowane transakcje powinny być **transakcjami wpłaty**, lub
2. Suma ostatnich pięciu transakcji (konto musi mieć **co najmniej pięć** transakcji) powinna być **większa** niż kwota wnioskowanego kredytu.

Opłaty za przelew ekspresowy również traktujemy jako transakcję.

#### Zachowanie funkcji

Funkcja `submit_for_loan()` powinna:

- powiększyć saldo konta o wnioskowaną kwotę (jeżeli kredyt został udzielony),
  - zwrócić `True`, gdy kredyt został przyznany,
  - zwrócić `False`, gdy kredyt nie został przyznany.
- 

### Dalsze kroki

Gdy kod będzie działał, zrób **push** i wystaw **PR**.

Sprawdź status **CI**, ale **jeszcze nie merguj** zmian.

Założmy, że w trakcie **code review** kolega z zespołu (wredny senior developer) poprosił Cię o **refaktoryzację** zanim dodamy zmiany do main brancha.

Głównym zarzutem jest **powtarzanie kodu** w wielu miejscach testów.

---

## Refaktoryzacja

**Krok 1:** Wynieś powtarzający się kod testów do **fikstury** `@pytest.fixture`

(→ [dokumentacja pytest fixtures](#))

**Krok 2:** Użyj `@pytest.mark.parametrize`, aby połączyć podobne testy w jedną funkcję z parametrami.

(→ [dokumentacja parametrize](#))

**Krok 3:** Jeżeli metoda `submit_for_loan()` zawiera więcej niż dwa warunki `if` (lub ma złożoną logikę), wydziel część kodu do metod pomocniczych.

---

## Na zakończenie

Gdy refaktoryzacja będzie skończona:

- zrób **push**,
- sprawdź status **CI**,
- jeśli jest zielono — **zmerguj** swoje zmiany.

Dodatkowo, wykonaj **refaktoryzację testów** w pozostałych klasach:

tam, gdzie to możliwe, użyj **parametryzacji oraz fikstur**.