

# Zadania w Pythonie – Kurs Programowania

---

## Zadanie 1: Programowanie funkcyjne

Napisz program, który analizuje listę transakcji zapisanych jako słowniki, np.:

```
transactions = [  
    {"amount": 100, "type": "income"},  
    {"amount": 50, "type": "expense"},  
    {"amount": 200, "type": "income"},  
    {"amount": 80, "type": "expense"}  
]
```

Używając wyłącznie funkcji wbudowanych i funkcji wyższego rzędu (map, filter, reduce, lambda), wykonaj następujące operacje:

- Wyfiltruj wszystkie przychody (income) i policz ich łączną wartość.
- Wyfiltruj wszystkie wydatki (expense) i policz ich łączną wartość.
- Oblicz wynik końcowy (saldo) jako różnicę przychodów i wydatków.

Nie używaj pętli ani instrukcji warunkowych if-else.

## Zadanie 2: Klasy, obiekty, property

Stwórz klasę `Book`, która będzie reprezentować książkę w bibliotece. Klasa powinna zawierać:

- atrybuty: tytuł, autor, rok wydania, liczba stron
- metodę wyświetlającą opis książki w formacie: "Tytuł (rok), autor: Autor, liczba stron: X"
- property `age`, które zwraca wiek książki na podstawie aktualnego roku
- zabezpieczenie liczby stron (musi być większa niż 0, w przeciwnym razie rzuć wyjątek)

Dodatkowo, stwórz 3 obiekty klasy Book i przetestuj ich działanie.

## Zadanie 3: Obsługa błędów, pliki, dane

Napisz program, który:

- Wczytuje dane z pliku `dane.txt`, gdzie każda linia zawiera parę: imię, wiek (oddzielone przecinkiem).

- Sprawdza poprawność danych: imię to ciąg znaków, wiek to liczba całkowita większa od 0.
- Dla niepoprawnych danych zapisz linię do osobnego pliku `bledy.log`, a poprawne dane przechowuj w strukturze danych (np. lista słowników).
- Na końcu wypisz średni wiek poprawnych danych.

Zastosuj odpowiednią obsługę błędów (try-except), zarządzanie plikami (with open) oraz walidację danych.

## Zadanie 4: numpy, pandas, pytest

Napisz program do analizy danych pomiarowych temperatury zapisanych w pliku CSV w formacie:

```
data,temperatura
2024-01-01,3.5
2024-01-02,2.7
...
```

Zadania:

1. Załaduj dane do ramki danych Pandas i wyświetl podstawowe statystyki (średnia, min, max, odchylenie std).
2. Używając numpy oblicz trend (różnica średnich temperatur tygodniowych).
3. Dodaj kolumnę "zmiana" pokazującą różnicę temperatury względem poprzedniego dnia.
4. Napisz testy jednostkowe w pytest sprawdzające poprawność działania funkcji analitycznych.

Uwaga: nie zakładaj poprawności danych wejściowych – obsłuż brakujące wartości, błędny format daty i inne potencjalne problemy.