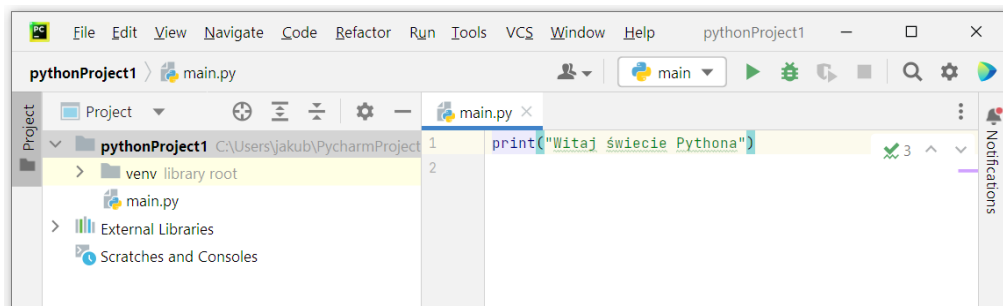


# Wstęp do programowania



## Lista 3

opracował dr inż. Jakub Długosz

**Funkcje i procedury. Rekursja. Domknięcia. Dekoratory. Wyrażenia lambda. Wybrane funkcje wbudowane języka Python. Paradygmaty programowania funkcyjnego. PEP 570.**

**Lista na ocenę.**

### Zadanie 1 – funkcja wyświetlająca skład grupy

Napisz funkcję `wyświetl_gрупę()`, która może przyjąć **пустą** bądź **dowolną liczbę parametrów z imionami członków grupy**. Funkcja zawiera również **parametr pozwalający na przypisanie opiekuna grupy, jeśli nie jest on podany, to przyjmuje domyślną wartość Cezary**. Działanie funkcji polega na wyświetleniu na ekranie składu grupy. Najpierw wymieniony jest opiekun. Jeśli w wywołaniu nie podano członków grupy, to wyświetlany jest komunikat „Grupa nie posiada członków”, w przeciwnym razie wyświetlani są wszyscy członkowie grupy.

Przykładowe działanie programu dla wywołania `wyświetl_gрупę()`

```
Oto skład grupy:  
Opiekun: Cezary  
Grupa nie posiada członków
```

Przykładowe działanie programu dla wywołania `wyświetl_gрупę("Anna")`

```
Oto skład grupy:  
Opiekun: Cezary  
Członkowie grupy:  
Anna
```

Przykładowe działanie programu dla wywołania `wyświetl_gрупę("Anna", "Cecylia", "Henryk")`

```
Oto skład grupy:  
Opiekun: Cezary  
Członkowie grupy:  
Anna  
Cecylia  
Henryk
```

Przykładowe działanie programu dla wywołania umożliwiającego również zmianę domyślnego opiekuna grupy

Oto skład grupy:

Opiekun: Bogdan

Członkowie grupy:

Anna

Cecylia

Henryk

## Zadanie 2 – dekorator @debuguj

Napisz dekorator `@debuguj` który przy wywołaniu dowolnej funkcji wyświetli na ekranie nagłówki z nazwą funkcji, argumenty pozycyjne funkcji oraz wynik działania funkcji.

Przykładowy wynik dla funkcji `dodaj()` (patrz **Dodatek 1**):

```
***** funkcja dodaj *****
argumenty pozycyjne: (5, 3))
wynik: 8
```

```
***** funkcja dodaj *****
argumenty pozycyjne: (3,))
wynik: 10
```

## Dodatek 1

`@debuguj`

`def dodaj(a, b=7):`

`"""Funkcja zwraca sumę dwóch parametrów a, b """`

`return(a + b)`

`dodaj(5, 3)`

`dodaj(3)`