

Politechnika Warszawska  
Wydział Elektroniki i Technik Informacyjnych

---

# PROS PYTHON LAB1

---

*Autorka:*  
NATALIA ŚLEPOWROŃSKA 318847

Warszawa 2024Z

## Spis treści

|                                    |   |
|------------------------------------|---|
| <b>1. Wstęp</b>                    | 2 |
| <b>2. Opis działania programu</b>  | 2 |
| 2.1. Funkcja inicjująca            | 2 |
| 2.2. Funkcja wczytująca dane       | 2 |
| 2.3. Funkcje przetwarzające dane   | 3 |
| 2.4. Funkcja wyświetlająca dane    | 3 |
| <b>3. Schemat blokowy</b>          | 3 |
| <b>4. Wynik działania programu</b> | 5 |

## 1. Wstęp

W ramach pierwszego laboratorium dotyczącego języka python, opracowałam program do prostej analizy finansowej. Program ten pobiera od użytkownika dochody dla danego miesiąca, a następnie dla zadanaego okresu przez użytkownika, oblicza on średnią oraz medianę zarobków.

## 2. Opis działania programu

Na początku działania programu deklaruję zmienne **incomes** oraz **months**, są to tablice, zmienne globalne.

Listing 1: Deklaracja zmiennych na początku programu

```
1 incomes = []
2 months = []
```

### 2.1. Funkcja inicjująca

W funkcji **initialize** odwołuję się do zmiennych globalnych. Następnie tablice **incomes** pozostawiam pustą, zaś tablicę **months** wypełniam nazwami miesięcy. Funkcja ta nie posiada argumentów oraz nie zwraca wartości.

Listing 2: Funkcja initialize

```
1 def initialize():
2     global incomes
3     global months
4     incomes = []
5     months = ["styczen", "luty", "marzec", "kwiecien", "maj", "czerwiec",
6              "lipiec", "sierpnien", "wrzesien", "pazdziernik", "listopad", "grudzien"]
```

### 2.2. Funkcja wczytująca dane

W funkcji **input\_incomes** tworzę odwołania do zmiennych globalnych, następnie pobieram od użytkownika liczbę miesięcy do analizy finansowej. Program sprawdza, czy liczba znajduje się w zakresie od 0 do 12. Jeżeli liczba znajduje się poza zakresem, program prosi o ponowne podanie liczby.

Listing 3: Funkcja input\_incomes, część I

```
1 def input_incomes():
2     global incomes
3     global months
4     months_num = int(input("Podaj liczbe miesiecy (od poczatku roku)
5                          dla ktorych chcesz wykonac analize finansowa: "))
6     if months_num < 0 or months_num > 12:
7         months_num = int(input("Podaj poprawna liczbe miesiecy: "))
8     ...
```

Następnie w pętli for pobieram dla każdego miesiąca dochód i umieszczam go w tablicy **incomes**. Program sprawdza, czy dochód jest liczbą dodatnią. Jeżeli nie, prosi o ponowne podanie dochodu.

Listing 4: Funkcja input\_incomes, część II

```
1     for i in range(months_num):
2         user_input = input("Podaj dochod w zlotowkach
3                             dla miesiaca {}: ".format(months[i]))
4         income = float(user_input)
5         if 0 > income:
6             user_input = input("Dochod musi byc rowny
7                                 badz wiekszy niz 0 zl, podaj dochod: ")
8         income = float(user_input)
```

```

9         incomes.append(income)
10    else:
11        incomes.append(income)

```

Funkcja ta nie zawiera argumentów.

### 2.3. Funkcje przetwarzające dane

Do analizy finansowej wykorzystuję funkcje obliczającą średnią oraz funkcję obliczającą medianę. Argumentem wejściowym obu tych funkcji jest zmienna **data**. Funkcje te nie odwołują się do zmiennych globalnych.

Listing 5: Funkcja `average_income`

```

1 def average_income(data):
2     if not data:
3         print("Brak danych do analizy")
4         return "Brak danych"
5     average = sum(data) / len(data)
6     return average

```

Listing 6: Funkcja `median_income`

```

1 def median_income(data):
2     if not data:
3         print("Brak danych do analizy")
4         return "Brak danych"
5     sorted_data = sorted(data)
6     len_data = len(sorted_data)
7     if len_data % 2 == 1:
8         median = sorted_data[len_data // 2]
9     else:
10        mid1 = sorted_data[(len_data // 2) - 1]
11        mid2 = sorted_data[len_data // 2]
12        median = (mid1 + mid2) / 2
13    return median

```

Funkcje te sprawdzają, czy zmienna `data` zawiera dane. Są one wyłowywane wewnątrz funkcji `display_results`.

### 2.4. Funkcja wyświetlająca dane

Do wyświetlenia analizy finansowej wykorzystuję funkcję `display_results`.

Listing 7: Funkcja `display_results`

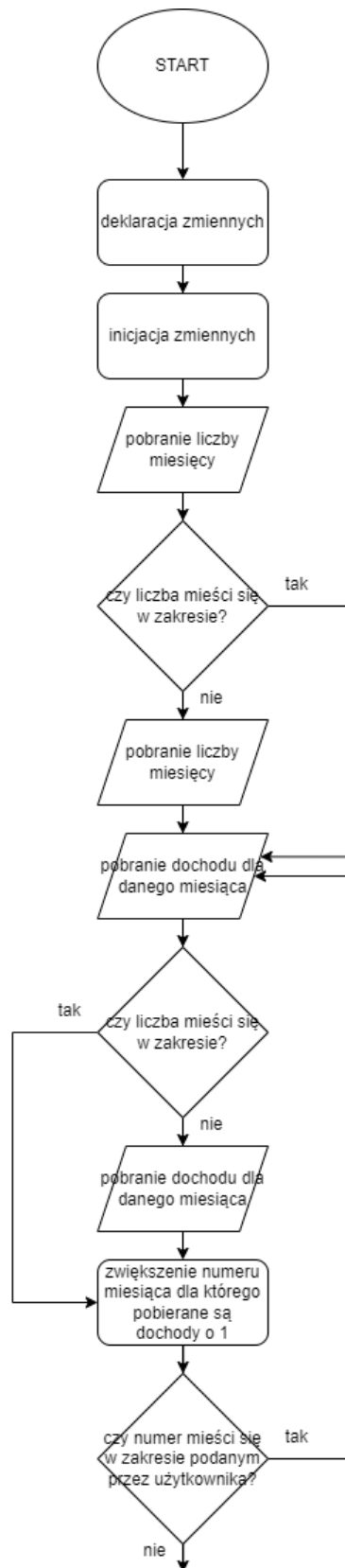
```

1 def display_results():
2     global incomes
3     average = average_income(incomes)
4     median = median_income(incomes)
5     print("Średnia zarobków dla podanego okresu: {}".format(average))
6     print("Mediana zarobków dla podanego okresu: {}".format(median))

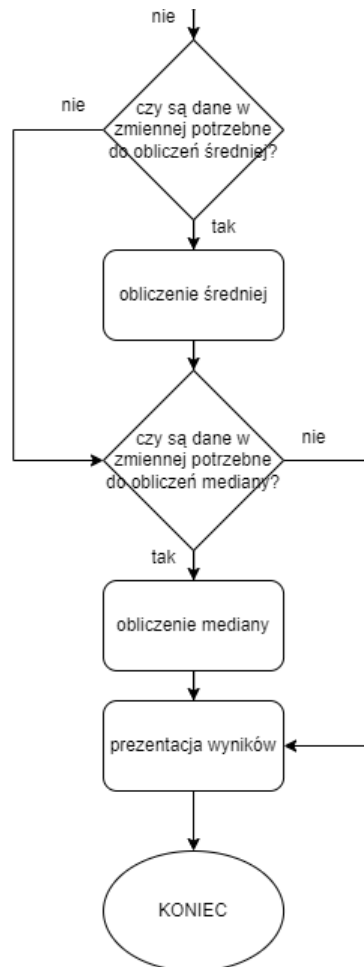
```

## 3. Schemat blokowy

Poniżej przedstawiam schemat blokowy mojego programu:



Rysunek 1: Część I



Rysunek 2: Część II

#### 4. Wynik działania programu

Poniżej przedstawiam wynik działania mojego programu. Przetestowałam różne scenariusze wprowadzenia danych wejściowych.

```

root@natalia:~# python3 lab1.py
Podaj liczbę miesięcy (od początku roku) dla których chcesz wykonać analizę finansową: -1
Podaj poprawną liczbę miesięcy: 6
Podaj dochód w złotych dla miesiąca styczeń: -3
Dochód musi być równy bądź większy niż 0 zł, podaj dochód: 1000
Podaj dochód w złotych dla miesiąca luty: 2000
Podaj dochód w złotych dla miesiąca marzec: 3000
Podaj dochód w złotych dla miesiąca kwiecień: 2000
Podaj dochód w złotych dla miesiąca maj: 1500
Podaj dochód w złotych dla miesiąca czerwiec: 2000
Średnia zarobków dla podanego okresu: 1916.6666666666667
Mediana zarobków dla podanego okresu: 2000.0
root@natalia:~# |
  
```

Rysunek 3: Wynik działania

```
root@natalia:~# python3 lab1.py
Podaj liczbe miesiecy (od poczatku roku) dla ktorych chcesz wykonac analize finansowa: 15
Podaj poprawna liczbe miesiecy: |
```

Rysunek 4: Przypadek podania zbyt dużej liczby miesięcy

Program działa zgodnie z moimi założeniami.