

Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych

PROS PYTHON LAB2

Autorka:
NATALIA ŚLEPOWROŃSKA 318847

Warszawa 2024Z

Spis treści

1. Wstęp	2
2. Opis działania programu	2
2.1. Funkcja inicjująca	2
2.2. Funkcja wczytująca dane	2
2.3. Funkcje przetwarzające dane	3
2.4. Funkcja wyświetlająca dane	4
3. Schemat blokowy	5
4. Wynik działania programu	6

1. Wstęp

W ramach drugiego laboratorium dotyczącego języka python, opracowałam program do analizy finansowej. Jest on rozwinięciem kodu z laboratorium pierwszego. Wykorzystałam słownik wielopoziomowy, by mój program był typu bazodanowego. Program ten pobiera od użytkownika liczbę analiz do wykonania, a następnie dochody dla danego miesiąca. Dla zadanaego okresu przez użytkownika, oblicza on średnią oraz medianę zarobków dla każdej analizy.

2. Opis działania programu

Na początku działania programu deklaruję zmienne **incomes_all**, **months** oraz **num**, są to słownik, tablica, oraz zmienna typu int - zmienne globalne.

Listing 1: Deklaracja zmiennych na początku programu

```
1 incomes_all = {}
2 months = []
3 num = 0
```

2.1. Funkcja inicjująca

W funkcji **initialize** odwołuję się do zmiennych globalnych. Następnie słownik **incomes_all** pozostawiam pusty, 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_all
3     global months
4     incomes_all = {}
5     months = ["styczen", "luty", "marzec", "kwiecień", "maj", "czerwiec",
6              "lipiec", "sierpień", "wrzesień", "październik", "listopad", "grudzień"]
```

2.2. Funkcja wczytująca dane

W funkcji **input_incomes** tworzę odwołania do zmiennych globalnych, następnie pobieram od użytkownika liczbę analiz do wykonania i zapełniam słownik **incomes_all** pustymi słownikami.

Listing 3: Funkcja input_incomes, część I

```
1 def input_incomes():
2     global incomes_all
3     global months
4     global num
5     num = int(input("Podaj liczbę analiz finansowych do wykonania: "))
6     for n in range(num):
7         incomes_all[n+1] = {}
8     ...
```

W kolejnym kroku pobieram liczbę miesięcy dla kolejnej 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 4: Funkcja input_incomes, część II

```
1     ...
2     for n in range(num):
3         months_num = int(input("Podaj liczbę miesięcy (od początku roku)
```

```

4             dla ktorych chcesz wykonac analize finansowa
5             numer {}: ".format(n+1)))
6     dict = incomes_all[n+1]
7
8     if months_num < 0 or months_num > 12:
9         months_num = int(input("Podaj poprawna liczbe miesiecy: "))
10    ...

```

Następnie w kolejnej pętli for pobieram dla każdego miesiąca dochód i umieszczam go w słowniku **dict**. Kluczami w słowniku są nazwy miesięcy. Program sprawdza, czy dochód jest liczbą dodatnią. Jeżeli nie, prosi o ponowne podanie dochodu.

Listing 5: Funkcja input_incomes, część III

```

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

```

Funkcja ta nie zawiera argumentów i nie zwraca wartości.

2.3. Funkcje przetwarzające dane

Do analizy finansowej wykorzystuję funkcje obliczającą średnią oraz funkcję obliczającą medianę. Argumentem wejściowym obu tych funkcji jest zmienna **num**. Jest to numer kolejnej analizy i jednocześnie klucz w słowniku **incomes_all**. Funkcje te odwołują się do zmiennych globalnych, zwracają one wyniki swoich obliczeń.

Listing 6: Funkcja average_income

```

1 def average_income(num):
2     global incomes_all
3     global months
4     dict = incomes_all[num]
5
6     if not dict:
7         print("Brak danych dla analizy {}".format(num))
8         return "Brak danych"
9
10    l = len(dict)
11    su = sum(dict.values())
12    average = su / l
13    return average

```

Listing 7: Funkcja median_income

```

1 def median_income(num):
2     global incomes_all
3     global months
4     dict = incomes_all[num]
5
6     if not dict:

```

```

7         print("Brak danych dla analizy {}".format(num))
8         return "Brak danych"
9
10        sorted_data = sorted(dict.values())
11        len_data = len(dict)
12
13        if len_data % 2 == 1:
14            median = sorted_data[len_data // 2]
15        else:
16            mid1 = sorted_data[(len_data // 2) - 1]
17            mid2 = sorted_data[len_data // 2]
18            median = (mid1 + mid2) / 2
19
20        return median

```

Funkcje te sprawdzają, czy zmienna `dict` zawiera dane. Są one wywoływane wewnątrz funkcji **`display_results`**.

2.4. Funkcja wyświetlająca dane

Do wyświetlenia analizy finansowej wykorzystuję funkcję **`display_results`**.

Listing 8: Funkcja `display_results`

```

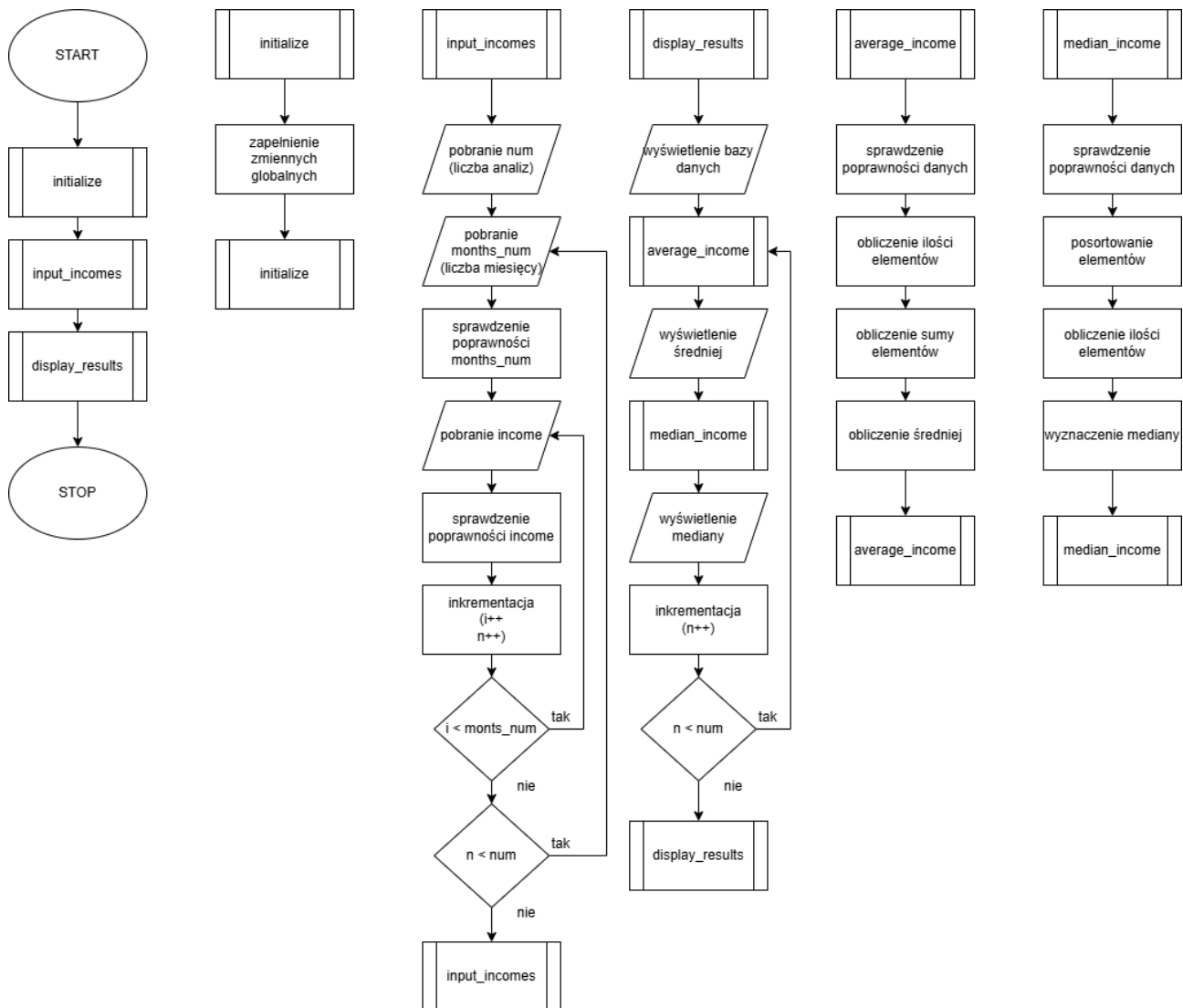
1 def display_results(num):
2     global incomes_all
3     print("-----PODSUMOWANIE-----")
4     print("Baza danych do analizy: {}".format(incomes_all))
5     for n in range(num):
6         average = average_income(n+1)
7         median = median_income(n+1)
8         print("Srednia zarobkow dla analizy {} dla podanego
9               okresu: {}".format(n+1,average))
10        print("Mediana zarobkow dla analizy {} dla podanego
11              okresu: {}".format(n+1,median))

```

Funkcja ta posiada argument `num` - jest to liczba analiz zadeklarowana przez użytkownika. **`display_results`** nie zwraca argumentów.

3. Schemat blokowy

Poniżej przedstawiam schemat blokowy mojego programu:



Rysunek 1: Schemat

4. Wynik działania programu

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

Scenariusz pierwszy: wprowadzenie zbyt dużej liczby miesięcy i ujemnego dochodu.

```
root@natalia:~# python3 lab2.py
Podaj liczbe analiz finansowych do wykonania: 1
Podaj liczbe miesiecy (od poczatku roku) dla ktorych chcesz wykonac analize finansowa numer 1: 14
Podaj poprawna liczbe miesiecy: 2
Podaj dochod w zlotowkach dla miesiaca styczen: -10
Dochod musi byc rowny badz wiekszy niz 0 zl, podaj dochod: 2000
Podaj dochod w zlotowkach dla miesiaca luty: 3000
-----PODSUMOWANIE-----
Baza danych do analiz: {1: {'styczen': 2000.0, 'luty': 3000.0}}
Srednia zarobkow dla analizy 1 dla podanego okresu: 2500.0
Mediana zarobkow dla analizy 1 dla podanego okresu: 2500.0
```

Rysunek 2: Scenariusz pierwszy

Scenariusz drugi: wprowadzenie ujemnej, a następnie zerowej liczby miesięcy.

```
root@natalia:~# python3 lab2.py
Podaj liczbe analiz finansowych do wykonania: 1
Podaj liczbe miesiecy (od poczatku roku) dla ktorych chcesz wykonac analize finansowa numer 1: -3
Podaj poprawna liczbe miesiecy: 0
-----PODSUMOWANIE-----
Baza danych do analiz: {1: {}}
Brak danych dla analizy 1
Brak danych dla analizy 1
Srednia zarobkow dla analizy 1 dla podanego okresu: Brak danych
Mediana zarobkow dla analizy 1 dla podanego okresu: Brak danych
root@natalia:~#
```

Rysunek 3: Scenariusz drugi

Scenariusz trzeci: poprawne wykonanie kilku analiz.

```
root@natalia:~# python3 lab2.py
Podaj liczbe analiz finansowych do wykonania: 3
Podaj liczbe miesiecy (od poczatku roku) dla ktorych chcesz wykonac analize finansowa numer 1: 2
Podaj dochod w zlotowkach dla miesiaca styczen: 5500
Podaj dochod w zlotowkach dla miesiaca luty: 4700
Podaj liczbe miesiecy (od poczatku roku) dla ktorych chcesz wykonac analize finansowa numer 2: 5
Podaj dochod w zlotowkach dla miesiaca styczen: 1000
Podaj dochod w zlotowkach dla miesiaca luty: 7900
Podaj dochod w zlotowkach dla miesiaca marzec: 4200
Podaj dochod w zlotowkach dla miesiaca kwiecień: 5600
Podaj dochod w zlotowkach dla miesiaca maj: 2300
Podaj liczbe miesiecy (od poczatku roku) dla ktorych chcesz wykonac analize finansowa numer 3: 3
Podaj dochod w zlotowkach dla miesiaca styczen: 500
Podaj dochod w zlotowkach dla miesiaca luty: 400
Podaj dochod w zlotowkach dla miesiaca marzec: 760
-----PODSUMOWANIE-----
Baza danych do analiz: {1: {'styczen': 5500.0, 'luty': 4700.0}, 2: {'styczen': 1000.0, 'luty': 7900.0, 'marzec': 4200.0, 'kwiecień': 5600.0, 'maj': 2300.0}, 3: {'styczen': 500.0, 'luty': 400.0, 'marzec': 760.0}}
Srednia zarobkow dla analizy 1 dla podanego okresu: 5100.0
Mediana zarobkow dla analizy 1 dla podanego okresu: 5100.0
Srednia zarobkow dla analizy 2 dla podanego okresu: 4200.0
Mediana zarobkow dla analizy 2 dla podanego okresu: 4200.0
Srednia zarobkow dla analizy 3 dla podanego okresu: 552.3333333333333
Mediana zarobkow dla analizy 3 dla podanego okresu: 500.0
root@natalia:~#
```

Rysunek 4: Scenariusz trzeci

Program działa zgodnie z moimi założeniami.