

## Wstęp do programowania

### Laboratorium 4

1. Napisz program, obliczający średni dzienny opad z pewnego okresu czasu. Liczba dni, z których jest obliczana średnia nie jest znana. Liczba ujemna podana przez użytkownika wskazuje koniec serii danych.
2. Zmodyfikuj program, obliczający wskaźnik masy ciała (BMI) – laboratorium 7 zad. 5, aby poprawność danych wejściowych była weryfikowana (masa i wzrost powinny się mieścić w przyjętych zakresach).
3. Pobierz z repozytorium kodów plik: **14\_weryfikacja\_danych.py**. Zmodyfikuj funkcję **get\_grade()** (wykład 5, slajd 18), tak aby funkcja **input()** była wywoływana tylko na początku pętli (bez wywołania przed pętlą **while**).
4. Napisz program, który wypisuje na ekranie trójkąt z gwiazdek.  
\*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*
5. Napisz program, który wypisuje na ekranie choinkę z gwiazdek – liczba wierszy choinki powinna być podawana przez użytkownika. Przykładowo, dla  $n = 5$  choinka powinna wyglądać następująco:  
\*  
\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*
6. Napisz program, który znajduje numerycznie przybliżone rozwiązanie równania  $x^2 - 2 = 0$ . Program prosi użytkownika o podanie początkowego rozwiązania  $x_0$ , a następnie oblicza kolejne przybliżenia z zależności  $x_{i+1} = \frac{1}{2} \left( x_i + \frac{2}{x_i} \right)$ . Zastanów się nad możliwymi warunkami przerwania pętli.

7. Sprawdź działanie następujących poleceń w trybie interaktywnym interpretera:

```
>>> randint(1, 100)
>>> random.randint(1, 100)

>>> import random
>>> randint(1,100)
>>> random.randint(1,100)
```

8. Napisz program, który będzie generował zadania matematyczne z zakresu dodawania. Program powinien wyświetlać dwie liczby całkowite (od 0 do 500). Program powinien prosić użytkownika o podanie wyniku dodawania tych liczb. Jeśli użytkownik poda poprawny wynik, to program powinien wyświetlić komunikat z gratulacjami, w przeciwnym przypadku, komunikat z poprawnym wynikiem.

9. (\*) Przeanalizuj program wykonujący obliczenia pól i obwodów figur.

Dodaj do programu moduł `triangle` obsługujący trójkąty. W module zdefiniuj funkcje:

- `get_lengths()` – zwraca długości trzech boków podane przez użytkownika,
- `area()` – zwraca pole powierzchni trójkąta (skorzystaj ze wzoru Herona),
- `perimeter()` – zwraca obwód trójkąta o podanych bokach.

Zmodyfikuj program, tak aby dawał użytkownikowi możliwość przeprowadzenia obliczeń dla trójkątów.

Karol Tarnowski  
Wrocław, 2021