Laboratorium 2 — Ćwiczenia w programowaniu proceduralnym.

Krotki (tuple) i listy

Krotka (tupla) pozwala na przechowywanie kilku wartości (mogą mieć one różne typy) w jednej zmiennej, przy czym po utworzeniu tupli nie można już tych wartości zmodyfikować. Stosuje się je najczęściej jako argumenty funkcji, lub do zwrócenia kilku wartości z funkcji. W przypadku drugiej opcji pozwala to na proste rozdzielenie (**rozpakowanie**) zwróconych wartości do kilku zmiennych.

```
>>> b = (5, 'a', 1.0)
>>> type(b)
<class 'tuple'>
>>> print(b[0])
5
```

Przykład rozpakowania (po prawej stronie „=" może znajdować się funkcja zwracająca kilka wartości):

```
>>> c = (2, 4)
>>> a, b = c
```

Listy w przypadku języka Python są podstawową strukturą tablicową. W przeciwieństwie do tablic znanych z języka c++ pozwalają one jednak na przechowywanie wartości różnych typów. Umożliwiają one także dynamiczne dodawanie nowych wartości. Różnice są widoczne także w samej ich obsłudze, możliwe jest np. wybranie danych z podanego zakresu indeksów (początek:koniec:krok).

```
>>> c = [1, 2, 3]
>>> c.append('a')
>>> print(c)
[1, 2, 3, 'a']
>>> print(c[0])
1
>>> print(c[-1])
a
>>> print(c[1:3])
[2, 3]
>>> a = [1, 2, 5, 2, 9, 3]
>>> print(a[0:6:2])
[1, 5, 9]
```

Zadanie 1 Utwórz listę zawierającą 6 elementów — 0, −2, 1, 7, 3, 4 a następnie wypisz jej zawartość w odwrotnej kolejności, **bez stosowania funkcji, czy pętli**.

Moduły oraz pakiety

Moduły w Pythonie są zbiorem funkcji zapisanych w pliku „nazwa_modulu.py". W celu załadowania funkcji z utworzonego modułu stosuje się polecenie **import**.

```
>>> import math
>>> print(math.pi)
```

Pakiety są natomiast przestrzeniami nazw zawierającymi wiele modułów. Aby zaimportować moduł z pakietu możemy skorzystać z jednej z dwóch metod:

```
>>> import pakiet.modul # metoda 1
>>> from pakiet import modul # metoda 2
```

Jeśli nie chcemy poprzedzać nazwy funkcji pełnymi nazwami modułów, możemy użyć komendy **as** i podać własną nazwę pod którą dostępne będą zaimportowane obiekty:

```
>>> import math as m
>>> print(m.pi)
```

Aby zaimportować zawartość modułu do aktualnej przestrzeni nazw możemy zastosować poniższą komendę:

```
>>> from math import *
>>> print(pi)
```

Zadanie 2 Zaimportuj moduł matematyczny, a następnie napisz skrypt którzy pobierze wartości dla dwóch zmiennych x, y i wyznaczy oraz wyświetli wartość, dla podanej funkcji:

$$f(x,y) = e + log_{10}(x^2+1)\frac{x-1}{\cos(y^3-1)+6}$$

w komentarzu podaj wartość dla x = 2, y = 4 oraz x = -9.5, y = 6.4.

Instrukcje warunkowe

Przy pisaniu instrukcji warunkowych w języku Python **kluczową role pełnią wcięcia**. Szczególne uwagę należy zwrócić, na to czy zostały one wykonane tą samą metodą. Pomieszanie spacji i tabulatorów sprawi, że kod się nie wykona. Należy również pamiętać, że Python nie używa nawiasów do grupowania instrukcji, zamiast tego grupuje je na podstawie poziomu zagnieżdżenia tekstu. Przykładowa instrukcja warunkowa może wyglądać następująco:

```
if warunek:
    instrukcja
    instrukcja
    if warunek: # wewnetrzny if
        instrukcja
        instrukcja
elif warunek:
    instrukcja
else:
    instrukcja
instrukcja poza if-em
```

Switch–case

Wraz z wersją 3.10 języka Python został wprowadzony odpowiednik instrukcji switch. **Match–case** umożliwia dopasowanie nie tylko konkretnych wartości, ale także bardziej ogólnych wzorców (jak np. listy o danej liczbie elementów).

```
x = 5
match x:
    case 1:
        print("x = 1")
    case _:
        print("x != 1")

lista = [2, 3]
match lista:
    case [x, y]:
        print(f"lista dwuelementowa: {x}, {y}")
    case [x, y, z]:
        print(f"lista trzyelementowa: {x}, {y}, {z}")
    case _:
        print("inna lista")
```

Zadanie 3 Napisz skrypt, który pobierze od użytkownika trzy wartości x, y, z oraz numer opcji c. Jeśli $c == 1$ zwróć ich sumę, jeżeli $c == 2$ różnicę, dla $c == 3$ iloczyn, a w przeciwnym razie iloraz. Upewnij się, że nie dojdzie do dzielenia przez 0, jeśli taka sytuacja miała by miejsce, wyświetl odpowiedni komunikat.

Zadanie 4 Rozszerz zadanie z poprzednich laboratoriów, które dotyczyło wyznaczania współczynnika BMI o wyświetlanie dodatkowej informacji o kategorii wagowej w zależności od wartości obliczonego BMI:

- wygłodzenie, gdy $BMI < 16$;
- wychudzenie, gdy $BMI \in [16, 17)$;
- niedowaga, gdy $BMI \in [17, 18.5)$;
- norma, gdy $BMI \in [18.5, 25)$;
- nadwaga, gdy $BMI \in [25, 30)$;
- otyłość stopnia I, gdy $BMI \in [30, 35)$;
- otyłość stopnia II, gdy $BMI \in [35, 40)$;
- otyłość stopnia III, gdy $BMI \geqslant 40$;

Pętle

Przy tworzeniu pętli należy kierować się takimi samymi zasadami jak w przypadku instrukcji warunkowych. Możemy wyróżnić tu dwie podstawowe pętlę: while oraz for. W odróżnieniu od pętli znanych z C, pętla for iteruje po sekwencjach np. listach, napisach, czy słownikach. Poniżej przykłady:

```
while warunek:
    instrukcja
    instrukcja

lista = [1, 5, 9]
for x in lista: # pod x podstawiane sa kolejne wartosci z lista
    instrukcja
    instrukcja

for x in range(max):
    instrukcja
for x in range(min, max):
    instrukcja
for x in range(min, max, krok):
    instrukcja
```

Funkcje typu range można zaimplementować samodzielnie wykorzystując słowo kluczowe **yield** lub tworząc klasę iteratora.

Pętle for można także wykorzystać do uzupełniania list.

```
>>> lista = [i for i in range(0, 10)]
>>> print(lista)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Zadanie 5 Napisz program drukujący na ekranie trójkąt o wysokości podanej przez użytkownika. Przykładowy trójkąt o wysokości 4:

$$X$$
$$XX$$
$$XXX$$
$$XXXX$$

Zadanie 6 Utwórz poniższą macierz, wykorzystując **jedną** linijkę kodu (dwie pętlę for inicjalizujące listę):

$$\begin{matrix} 1 & 4 & 7 \\ 1 & 4 & 7. \\ 1 & 4 & 7 \end{matrix}$$

Zadanie 7 Napisz program, który znajdzie i wyświetli n pierwszych liczb pierwszych (zakładając, że pierwszą liczbą pierwszą jest liczba 2).

Zadanie 8 Napisz program, który wyznaczy i wyświetli na ekranie sumę liczb naturalnych mniejszych od n (liczba n podawana jest przez użytkownika), które są zakończone liczbą 3 lub 14.

Zadanie 9 Napisz program, który wyświetli na ekranie n pierwszych wierszy trójkąta Pascala (liczba n podawana jest przez użytkownika).