

Ćwiczenie 3

Funkcje. Klasy

1 Wprowadzenie

1.1 Definicja funkcji

```
def nazwaFunkcji(argumenty):  
    #... treść (ciało) funkcji ...
```

W definicji funkcji można stosować argumenty z wartościami domyślnymi. Wartości mogą być zwracane z funkcji za pomocą instrukcji **return**. Jeśli funkcja nie posiada instrukcji **return**, zwracany jest obiekt **none**.

1.2 Przekazywanie argumentów do funkcji

Sposoby przekazywania argumentów do funkcji:

- przez wartości:

```
nazwaFunkcji(wartość1, wartość2, ...)
```

- przez nazwy:

```
nazwaFunkcji(argumentX=wartośćX, argumentY=wartośćY, ...)
```

- mieszane:

```
nazwaFunkcji(wartość1, argumentY=wartośćY, ...)
```

Przy przekazywaniu argumentów przez nazwy ich kolejność na liście wywołania nie ma znaczenia.

1.3 Definicja klasy

```
class nazwaKlasy:  
    # ... pola ...  
    # ... definicja konstruktorów ...  
    # ... definicja metod ...
```

1.4 Słowo kluczowe self

Słowo kluczowe **self** służy do odwoływania się do pól i metod w klasie. Jego stosowanie jest obowiązkowe.

```
self.pole  
self.metoda(argumenty)
```

1.5 Tworzenie pól

Instrukcje przypisania wewnątrz klasy tworzą pola klasy.

1.6 Definicja konstruktora

Konstruktor klasy to zagnieżdżona w klasie funkcja o nazwie `__init__` z pierwszym argumentem `self` i opcjonalnymi pozostałymi argumentami.

```
def __init__(self, argumenty):
    #... treść (ciało) konstruktora ...
```

Przy wywołaniu konstruktora, argument `self` nie musi być jawnie specyfikowany.

1.7 Definicja metody

Metoda klasy to zagnieżdżona w klasie funkcja z pierwszym argumentem `self` i opcjonalnymi pozostałymi argumentami.

```
def nazwaMetody(self, argumenty):
    #... treść (ciało) metody ...
```

Przy wywołaniu metody, argument `self` nie musi być jawnie specyfikowany. Wszystkie metody są metodami wirtualnymi.

1.8 Tworzenie obiektów

```
obiekt=nazwaKlasy(argumenty)
```

1.9 Dziedziczenie

```
class nazwaKlasyPochodnej(nazwaKlasyBazowej):
    # ... deklaracja pól ...
    # ... definicja konstruktorów ...
    # ... definicja metod ...
```

Dopuszczalne jest dziedziczenie wielokrotne. W klasie pochodnej można wywołać konstruktor klasy bazowej:

```
nazwaKlasyBazowej.__init__(self, argumenty)
```

W klasie pochodnej można również wywołać metodę klasy bazowej:

```
nazwaKlasyBazowej.nazwaMetody(self, argumenty)
```

2 Zadania

2.1

Napisz skrypt, w którym zdefiniowana zostanie funkcja obliczania podatku dochodowego przy danej podstawie obliczenia podatku.

Podstawa		Podatek
ponad	do	
	85 528	18% podstawy minus kwota 556 zł 02 gr
85 528		14 839 zł 02 gr plus 32% nadwyżki ponad 85 528 zł.

Przetestuj w skrypcie utworzoną funkcję.

2.2

Napisz skrypt, w którym zdefiniowana zostanie funkcja **signum** dla argumentu rzeczywistego x . Funkcja **signum** określona jest wzorem:

$$\text{signum}(x) = \begin{cases} 1 & \text{dla } x > 0 \\ 0 & \text{dla } x = 0 \\ -1 & \text{dla } x < 0 \end{cases}$$

Przetestuj w skrypcie utworzoną funkcję.

2.3

Napisz skrypt, w którym zdefiniowane zostaną funkcje rekurencyjne wyznaczające n -te wyrazy ciągów:

$$\begin{aligned} \bullet a(x) &= \begin{cases} 2 & \text{dla } x = 0 \\ a(x-1) + 3x & \text{dla } x > 0 \end{cases} \\ \bullet b(x) &= \begin{cases} 1 & \text{dla } x = 0 \\ 3 & \text{dla } x = 1 \\ 2b(x-1) + 3b(x-2) & \text{dla } x > 1 \end{cases} \end{aligned}$$

Przetestuj w skrypcie utworzone funkcje.

2.4

Zdefiniuj w skrypcie klasę opisującą czytelnika biblioteki (nazwisko, imię, numer karty bibliotecznej, liczba wypożyczonych książek). Metody klasy powinny umożliwiać:

- wyświetlenie informacji o czytelniku,
- zmianę liczby wypożyczonych książek przy wypożyczeniu kolejnych książek (czytelnik nie może wypożyczyć więcej jak 10 książek),
- zmianę liczby wypożyczonych książek przy oddaniu podanej ilości książek.

Przetestuj w skrypcie utworzoną klasę.

2.5

Zdefiniuj w skrypcie klasę opisującą wektor w przestrzeni dwuwymiarowej (nazwa wektora, współrzędne x oraz y). Metody klasy powinny umożliwiać:

- wyświetlenie informacji o wektorze,
- zmianę nazwy wektora,
- obliczenie długości wektora.

Przetestuj w skrypcie utworzoną klasę.

2.6

Rozszerz możliwości klasy z zadania 2.4 o przechowywanie informacji o liczbie wypożyczonych czasopism. Nowe metody klasy pochodnej powinny umożliwiać:

- wyświetlenie informacji o czytelniku z dodatkową informacją,
- zmianę liczby wypożyczonych czasopism przy wypożyczeniu kolejnych czasopism (czytelnik nie może wypożyczyć więcej jak 5 czasopism),
- zmianę liczby wypożyczonych czasopism przy oddaniu podanej ilości czasopism.

Przetestuj w skrypcie nową utworzoną klasę.

2.7

Rozszerz możliwości klasy z zadania 2.5 o przechowywanie informacji o punkcie zaczepienia. Nowe metody klasy pochodnej powinny umożliwiać:

- wyświetlenie informacji o wektorze z dodatkową informacją,
- zmianę punktu zaczepienia.

Przetestuj w skrypcie nową utworzoną klasę.