

Klasy – wprowadzenie

Klasa to definicja formatu danych według którego tworzone są obiekty.

Klasa opisuje jakie dane może zawierać obiekt oraz jakim operacjom może zostać poddany lub jakie operacje może sam wykonać.

Obiekt jest po prostu zmienną (rozbudowany typ danych). Obiekt jest definiowany przez użytkownika i tworzony za pomocą klas. Zawiera dane i operacje jakie może wykonać

Tworzenie klas i obiektów

Definicja klasy

```
class nazwa_klasy
```

Utworzenie obiektu danej klasy

```
tworzony_obiekt = nazwa_klasy()
```

<pre>class Dom:</pre>	Definicja klasy Słowo kluczowe class i nazwa klasy Dom:
<pre> Powierzchnia = 100 Cena = 1000000 IloscPokoi = 6 IloscPieter = 2 Garaz = True Piwnica = False Ogrod = True</pre>	Parametry klasy Dom wraz z wartościami inicjalizującymi. Są to jednocześnie wartości domyślne, które będą zapisane w powstałym obiekcie jeśli nie zostaną wprowadzone inne.

Atrybuty i metody

Atrybuty (pola) to cechy, które będzie posiadał obiekt danej klasy.

W praktyce to zmienne mające jakąś wartość i przechowywane w danej klasie.

Przykład

```
class Dom:  
    Powierzchnia = 100  
    Cena = 1000000
```

Metody to funkcje zdefiniowane wewnątrz klasy.

Definiują zachowanie, umożliwiając im wykonywanie konkretnych zadań,

Przykład

```
class Dom:
```

```
pokoje = 4

def __init__(self, powierzchnia, cena):
    self.powierzchnia = powierzchnia
    self.cena = cena

def WartoscPokoju(self):
    avercost = self.cena/self.pokoje
    return (avercost)
```

```

'''
Klasy
'''
import math

class Punkt:
    x = 0
    y = 0

    def __init__(self, x, y):
        # instance attributes
        self.x = x
        self.y = y

    def przesun (self, a, b):
        self.x = self.x + a
        self.y = self.y + b

    def odleglosc_0_0 (self):
        d = math.sqrt (self.x * self.x + self.y * self.y)
        return d

    def odleglosc (self, a, b):
        d = math.sqrt (a * a + b * b)
        return d

    def trojkat (self, ax, ay, bx, by, cx, cy) :
        pole=1/2*abs(ax*by + bx*cy + ay*cx - by*cx - ax*cy - ay*bx)
        return pole

    def srodekX (self, ax, bx, cx) :

```

```
        srodekX = (ax + bx + cx) / 3
        return srodekX

    def srodekY (self, ay, by, cy)    :
        srodekY = (ay + by + cy) / 3
        return srodekY

print (Punkt.x)
print (Punkt.y)

A = Punkt (5, 12)
print (A.x)
print (A.y)
print (A.odleglosc_0_0 ())
A.przesun (11, 11)
print (A.x)
print (A.y)
print (A.odleglosc_0_0 ())

Punkt.x = 10
Punkt.y = 10
B= Punkt (Punkt.x, Punkt.y)
print (B.x)
print (B.y)
print (B.odleglosc (3,4))

print (A.trojkat(8,1,2,4,3,9))
print ("Srodek ", A.srodekX(1,9,2), "\t",A.srodekY(4, -10,9))
```

Zadanie

Utwórz program opisujący liczby zespolone.

Stwórz klasę `liczba zespolona`.

Utwórz 4 obiekty tej klasy.

Atrybuty:

- Część rzeczywista
- Część urojona

Metody:

- Obliczenie modułu liczby zespolonej
- Dodawanie liczb zespolonych
- Odejmowanie liczb zespolonych
- Mnożenie liczb zespolonych
- Liczba sprzężona do danej liczby
- Wyświetlenie wartości liczby zespolonej w postaci `a+ib`