

Sprawozdanie 1

Języki programowania obiektowego

Paweł Jamro
292510
IMT, WIMIR

KOD NAPISANYCH KLAS + KOMENTARZE:

1. Klasa Punkt:

```
public class Punkt
{
    protected float masa;

    //konstruktor domyślny
    public Punkt()
    {
        //przypisanie wartości domyślnej do masy
        this.masa = 1;
    }

    //konstruktor z parametrami
    public Punkt(float masa)
    {
        // kontrola wartości
        if (masa > 0) {
            this.masa = masa;
        }
        else {
            this.masa = 1;
        }
    }

    //Główny moment bezwładności
    public float GłównyMomentBezwładności()
    {
        return 0;
    }

    //Moment bezwładności ze Steinera
    public float MomentBezwładnościLiczonySteinerem(float odl)
    {
        return GłównyMomentBezwładności() + masa*odl*odl;
    }

    //metoda zwracająca masę punktu
    public String Opis() { return "Punkt materialny"; }

    //metoda, która pozwala zmienić masę punktu
    public void zmienianieMasy(float masa) { this.masa = masa; }

    //metoda pozwalająca zwrócić masę punktu
    public float zwracanieMasy() { return this.masa; }
}
```

2. Klasa Main:

```
public class Main {

    public static void main(String[] args) {

        //1

        Punkt punkt1 = new Punkt();

        //2

        Punkt punkt2 = new Punkt( mass: 3);

        //3

        System.out.println("Masa punktu 1 = "+punkt1.zwracanieMasy());
        System.out.println("Główny moment dla punktu 1 = "+punkt1.GlownyMomentBezwladnosci());
        System.out.println("Moment bezwładności liczony względem osi dla punktu1 = "+punkt1.MomentBezwladnosciLiczonySteinerem( odt: 3));
        System.out.println("Opis punktu 1 : " +punkt1.Opis());
        System.out.println("Masa punktu 2 = "+punkt2.zwracanieMasy());
        System.out.println("Główny moment dla punktu 2 = "+punkt2.GlownyMomentBezwladnosci());
        System.out.println("Główny moment dla punktu 2 = "+punkt2.MomentBezwladnosciLiczonySteinerem( odt: 3));
        System.out.println("Opis punktu 2 : "+punkt2.Opis());

        //4

        punkt1.zmienianieMasy( mass: 3.5F);

        //5

        System.out.println("Masa punktu 1 = "+punkt1.zwracanieMasy());
        System.out.println("Główny moment dla punktu 1 = "+punkt1.GlownyMomentBezwladnosci());
        System.out.println("Moment bezwładności liczony względem osi dla punktu 1 = "+punkt1.MomentBezwladnosciLiczonySteinerem( odt: 2));
        System.out.println("Opis punktu 1 : " +punkt1.Opis());

        //6

        int NumOfObj = 3; //liczba obiektów w tablicy

        Punkt[] punkt = new Punkt(NumOfObj); // stworzona tablica obiektów

        Punkt[] punkt = new Punkt(NumOfObj); // stworzona tablica obiektów

        for ( int i = 0; i<punkt.length ; i++)
        {
            punkt[i] = new Punkt( mass: i+5); //zmiana masy
        }

        //7

        int licznik = 0; // iterowanie obiektów

        for (Punkt obj : punkt)
        {
            System.out.println("Masa obiektu o numerze = "+ licznik + " to: " + obj.zwracanieMasy());
            System.out.println("Opis dla obiektu numer : " + licznik + " to: "+obj.Opis());
            System.out.println("Główny moment bezwładności dla obiektu numer = " + licznik + " to: " + obj.GlownyMomentBezwladnosci());
            System.out.println("Moment bezwładności liczony względem osi dla obiektu numer = " + licznik + " to: " + obj.MomentBezwladnosciLiczonySteinerem( odt: 2));

            licznik++;
        }
    }
}
```

WYNIK DZIAŁANIA PROGRAMU TESTOWEGO + OMÓWIENIE:

```
Masa punktu 1 = 1.0
Główny moment dla punktu 1 = 0.0
Moment bezwładności liczony względem osi dla punktu1 = 9.0
Opis punktu 1 : Punkt materialny
Masa punktu 2 = 3.0
Główny moment dla punktu 2 = 0.0
Główny moment dla punktu 2 = 27.0
Opis punktu 2 : Punkt materialny
Masa punktu 1 = 3.5
Główny moment dla punktu 1 = 0.0
Moment bezwładności liczony względem osi dla punktu 1 = 14.0
Opis punktu 1 : Punkt materialny
Masa obiektu o numerze = 0 to: 5.0
Opis dla obiektu numer : 0 to: Punkt materialny
Główny moment bezwładności dla obiektu numer = 0 to: 0.0
Moment bezwładności liczony względem osi dla obiektu numer = 0 to: 20.0
Masa obiektu o numerze = 1 to: 6.0
Opis dla obiektu numer : 1 to: Punkt materialny
Główny moment bezwładności dla obiektu numer = 1 to: 0.0
Moment bezwładności liczony względem osi dla obiektu numer = 1 to: 24.0
Masa obiektu o numerze = 2 to: 7.0
Opis dla obiektu numer : 2 to: Punkt materialny
Główny moment bezwładności dla obiektu numer = 2 to: 0.0
Moment bezwładności liczony względem osi dla obiektu numer = 2 to: 28.0

Process finished with exit code 0
```

a)

```
Masa punktu 1 = 1.0
Główny moment dla punktu 1 = 0.0
Moment bezwładności liczony względem osi dla punktu1 = 9.0
Opis punktu 1 : Punkt materialny
Masa punktu 2 = 3.0
Główny moment dla punktu 2 = 0.0
Główny moment dla punktu 2 = 27.0
Opis punktu 2 : Punkt materialny
```

Wyświetlona jest tu informacja zawierająca opis, masę, wartość głównego momentu bezwładności oraz wartość momentu bezwładności względem nowej osi dla dwóch powstałych obiektów tj. punkt 1 i punkt 2. Dla odległości 3.

b)

```
Masa punktu 1 = 3.5
Główny moment dla punktu 1 = 0.0
Moment bezwładności liczony względem osi dla punktu 1 = 14.0
Opis punktu 1 : Punkt materialny
```

Wyliczony główny moment bezwładności oraz moment bezwładności oraz wyświetlony opis punktu 1. Dla odległości 2.

c)

```
Masa punktu 2 = 3.0
Główny moment dla punktu 2 = 0.0
Główny moment dla punktu 2 = 27.0
Opis punktu 2 : Punkt materialny
```

Wyliczony główny moment bezwładności oraz moment bezwładności oraz wyświetlony opis punktu 2. Dla odległości 3.

d)

```
Masa obiektu o numerze = 0 to: 5.0
Opis dla obiektu numer : 0 to: Punkt materialny
Główny moment bezwładności dla obiektu numer = 0 to: 0.0
Moment bezwładności liczony względem osi dla obiektu numer = 0 to: 20.0
Masa obiektu o numerze = 1 to: 6.0
Opis dla obiektu numer : 1 to: Punkt materialny
Główny moment bezwładności dla obiektu numer = 1 to: 0.0
Moment bezwładności liczony względem osi dla obiektu numer = 1 to: 24.0
Masa obiektu o numerze = 2 to: 7.0
Opis dla obiektu numer : 2 to: Punkt materialny
Główny moment bezwładności dla obiektu numer = 2 to: 0.0
Moment bezwładności liczony względem osi dla obiektu numer = 2 to: 28.0
```

Wyliczenie i wyświetlenie tych samych informacji co dla punktów powyżej, ale dla nowo powstałych obiektów w tablicy. Dla odległości 2.

Odpowiedzi na pytania:

1. *Modyfikatory dostępu* w języku Java

- ✓ **public** – pozwala na dostęp do danego elementu ze wszystkich klas
- ✓ **protected** – pozwala na dostęp do danego elementu tylko dla klas dziedziczących oraz klas z tego samego pakietu
- ✓ **default** – pozwala na dostęp do danego elementu tylko klasom z danego pakietu (nie istnieje słowo w Javie określające ten rodzaj dostępu, jeżeli chcemy go użyć to po prostu nie podajemy żadnego modyfikatora)
- ✓ **private** – dostęp do danego elementu ograniczony tylko do klasy w którym jest zdefiniowany

2. **Akcesory** („getter” i „setter”) – zawarte są wewnątrz obiektu, pozwalają na odczyt lub modyfikację jego atrybutów. Akcesory są metodami poprzedzonymi „public”, ponieważ chcemy z nich korzystać poza klasą. Zwracają lub pozwalają modyfikować atrybuty z modyfikatorem „private”.

3. **Konstruktor** jest to specjalna metoda danej klasy, wywoływana podczas tworzenia jej instancji. Ma on za zadanie zainicjowanie obiektu. Możemy za jego pomocą nadać nowemu obiektowi wartości początkowe.

Wyróżniamy:

a) **konstruktor domyślny**, czyli nie przyjmujący argumentów

```
public nazwa_klasy()
{
    //kod
}
```

b) **konstruktor z parametrami**

```
public nazwa_klasy(typ_zmienna1, typ_zmienna2)
{
    this.x = zmienna1;
    this.y = zmienna2;
    //kod
}
```

4. Metody są to podprogramy działające na rzecz danej klasy lub klas pokrewnych (w przypadku dziedziczenia). Zadaniem metod jest skrócenie i uproszczenie kodu oraz zapobieganie tworzeniu funkcji globalnych mających wykonywać daną czynność. Metody mogą (lecz nie muszą) przyjmować argumenty oraz mogą je zwracać.

a) **Metoda nie przyjmująca argumentów i nie zwracająca wartości**

```
void nazwa_metody()
{
    //kod
}
```

b) **Metoda przyjmująca argumenty i nie zwracająca wartości**

```
void nazwa_metody(typ_zmienna1, typ_zmienna2)
{
    //kod
}
```

c) **Metoda nie przyjmująca argumentów i zwracająca wartości**

```
typ_metody nazwa_metody()
{
    //kod
    return wartosc;
}
```

d) **Metoda przyjmująca argumenty i zwracająca wartości**

```
typ_metody nazwa_metody(typ_zmienna1, typ_zmienna2)
{
    //kod
    return wartosc;
}
```

