

Sprawozdanie 3

Języki Programowanie Obiektowego

Paweł Jamro, 292510, IMT

1. Klasa SimTask

```
package com.company;

import java.util.TimerTask;

public class SimTask extends TimerTask{

    @Override
    public void run() {

    }

}
```

2. Klasa SimEngine

```
package com.company;

public class SimEngine {

}
```

3. Klasa Vector2D

```
package com.company;

import static java.lang.Math.sqrt;
public class Vector2D {
    //1
    //vector współrzędne
    public float posX;
    public float posY;
    //2
    //default konstruktor
    public Vector2D()
    {
        this.posX = 0f;
        this.posY = 0f;
    }
    //konstruktor z parametrami
    public Vector2D(float posX, float posY)
    {
        this.posX = posX;
        this.posY = posY;
    }
    //3
    public Vector2D VectorSumowanie(Vector2D v)
    {
        Vector2D wektor = new Vector2D( posX: this.posX + v.posX, posY: this.posY+v.posY);
        return wektor;
    }
    //4
    public Vector2D VectorOdejmowanie(Vector2D v)
    {
        Vector2D wektor = new Vector2D( posX: this.posX-v.posX, posY: this.posY-v.posY);
        return wektor;
    }

    //5
    public Vector2D VectorMnozenie (float x)
    {
        Vector2D wektor = new Vector2D( posX: this.posX * x, posY: this.posY * x);
        return wektor;
    }
    //6
    public float VectorDlugosc() { return (float)sqrt(this.posX*this.posX + this.posY*this.posY); }
    //7
    public Vector2D VectorZnormalizownay()
    {
        Vector2D wektor = new Vector2D( posX: this.posX/VectorDlugosc(), posY: this.posY/VectorDlugosc());
        return wektor;
    }
}
```

4. Klasa SpringApplet

a) Wersja testowa

```
public class SpringApplet extends JApplet{

    public void init() {

    }

    public void paint() {

    }

    public static void main(String[] args) {
        //1
        Vector2D vector1 = new Vector2D( posX: 5f, posY: 3f);
        Vector2D vector2 = new Vector2D( posX: 4f, posY: 6f);
        //2
        Vector2D suma = vector1.VectorSumowanie(vector2);
        Vector2D odjecie= vector1.VectorOdejmnowanie(vector2);
        //3
        Vector2D znormalizowany1 = vector1.VectorZnormalizownay();
        Vector2D znormalizowany2 = vector2.VectorZnormalizownay();
        //4
        float length1 = vector1.VectorDlugosc();
        float length2 = vector2.VectorDlugosc();
        //5
        Vector2D przemnozenie = vector1.VectorMnozenie(5f);
        //6
        System.out.println("vector1 wspolrzedne (X: "+vector1.posX+", Y: "+vector1.posY+"");
        System.out.println("vector2 wspolrzedne (X: "+vector2.posX+", Y: "+vector2.posY+"");
        System.out.println("sumowanie wspolrzednych (X: "+suma.posX+", Y: "+suma.posY+"");
        System.out.println("odejmowanie wspolrzednych (X: "+odjecie.posX+", Y: "+odjecie.posY+"");
        System.out.println("znormalizowany wektor1 (X: "+znormalizowany1.posX+", Y: "+znormalizowany1.posY+"");
        System.out.println("znormalizowany wektor2 (X: "+znormalizowany2.posX+", Y: "+znormalizowany2.posY+"");
        System.out.println("wymnozenie wektora1 przez liczbe (X: "+przemnozenie.posX+", Y: "+przemnozenie.posY+"");
        System.out.println("dlugosc1: "+vector1.VectorDlugosc());
        System.out.println("dlugosc: "+vector2.VectorDlugosc());
    }
}
```

b) Z polecenia

```
package com.company;
import javax.swing.*;

//main class
public class SpringApplet extends JApplet{

    public void init() {

        //1
        Vector2D vector1 = new Vector2D( posX: 5f, posY: 3f);
        Vector2D vector2 = new Vector2D( posX: 4f, posY: 6f);
        //2
        Vector2D suma = vector1.VectorSumowanie(vector2);
        Vector2D odjecie= vector1.VectorOdejmnowanie(vector2);
        //3
        Vector2D znormalizowany1 = vector1.VectorZnormalizownay();
        Vector2D znormalizowany2 = vector2.VectorZnormalizownay();
        //4
        float length1 = vector1.VectorDlugosc();
        float length2 = vector2.VectorDlugosc();
        //5
        Vector2D przemnozenie = vector1.VectorMnozenie(5f);
        //6
        System.out.println("vector1 wspolrzedne (X: "+vector1.posX+", Y: "+vector1.posY+")");
        System.out.println("vector2 wspolrzedne (X: "+vector2.posX+", Y: "+vector2.posY+")");
        System.out.println("sumowanie wspolrzednych (X: "+suma.posX+", Y: "+suma.posY+")");
        System.out.println("odejmowanie wspolrzednych (X: "+odjecie.posX+", Y: "+odjecie.posY+")");
        System.out.println("znormalizowany wektor1 (X: "+znormalizowany1.posX+", Y: "+znormalizowany1.posY+")");
        System.out.println("znormalizowany wektor2 (X: "+znormalizowany2.posX+", Y: "+znormalizowany2.posY+")");
        System.out.println("wymnozenie wektora1 przez liczbe (X: "+przemnozenie.posX+", Y: "+przemnozenie.posY+")");
        System.out.println("dlugosc1: "+vector1.VectorDlugosc());
        System.out.println("dlugosc: "+vector2.VectorDlugosc());

    }

    public void paint() {

    }

}
```

Wyniki programu testowego:

```
vector1 wspolrzedne (X: 5.0, Y: 3.0)
vector2 wspolrzedne (X: 4.0, Y: 6.0)
sumowanie wspolrzednych (X: 9.0, Y: 9.0)
odejmowanie wspolrzednych (X: 1.0, Y: -3.0)
znormalizowany wektor1 (X: 0.857493, Y: 0.5144958)
znormalizowany wektor2 (X: 0.5547002, Y: 0.8320503)
wymnozenie wektora1 przez liczbe (X: 25.0, Y: 15.0)
dlugosc1: 5.8309517
dlugosc: 7.2111025
```

Wyświetlone informacje kolejno:

- współrzędne wektora1,
- współrzędne wektora2,
- dodanie dwóch wektorów,
- odjęcie dwóch wektorów,
- znormalizowanie wektora1,
- znormalizowanie wektora2,
- wymnożenie wektora1 przez liczbę,
- wyświetlenie długości wektora1,
- wyświetlenie długości wektora2.

Odpowiedzi na pytania:

1. Klasa Timer umożliwia odmierzenie upływu czasu podczas pracy programu. Korzystanie z tej klasy polega na stworzeniu obiektu Timer i wywoływanie na jego rzecz odpowiednich metod, w zależności od tego czego potrzebujemy.
2. Zwracanie z metod obiektów różni się od zwracania typów prostych tym, że zwracając obiekt zwracamy jego adres. Natomiast zwracając typ prosty zwracamy jego wartość.
3. Rysowanie grafiki na powierzchni appletu odbywa się za pomocą wywołania metody draw() na rzecz obiektu typu Graphics. Są gotowe metody pozwalające rysować okręgi, linie, wielokąty.