

Inlämning 1

Objektorienterad programmering med Java

Emil Björklund - embj3739
emilbjorklund@live.com

January 22, 2019

Inledning

- IDE: Eclipse IDE v4.5.0
- Java version: JDK 8 update 191
- OS: MacOS Mojave

Note. Källkoden bifogas i separat fil som ligger i samma .zip som detta dokument. All kod befinner sig i samma fil, dvs. alla classer i samma dokument.

Deluppgift

1

Programmet kompilerades genom Eclipse, ett litet fönster öppnades och en gul boll ”studsade” mot väggarna. När fönstret expanderades fortsatte bollen att studsas inom den föregående begränsingen.

2

För att ändra färg anropas funktionen ”setColor” efter att ett objekt av klassen Ball har skapats. Om objektet heter ”ball1” kan en färgändring göras genom att anropa ”ball1.setColor(Color.blue)” där objektet skapas i klassen BallPanel. Nu är färgen blå.

Här är funtionen som ändrar färg:

```
public void setColor( Color c ) {  
    color = c;  
}
```

3

Funtionen ”wasResized” anropas om ”getWidth()” och ”getHeight” inte stämmer överens med den givna storleken. Funktionen som anropar ”wasResized” triggar av klockan. I funktionen ”wasResized” anropas objektet balls funtion ”setBoundingBox” för att uppdatera boxen som bollarna studsar i.

Här är den nya implementationen:

```
public void wasResized( int newWidth, int newHeight ) {  
    width = newWidth;  
    height = newHeight;  
    // Uppdaterar boxen om wasResized körs, det vill säga if statsen i actionPerformed är uppfylld.  
    // Anropar funktionen för att ändra boxen som bollsen stutsar i.  
    ball1.setBoundingBox( new Rectangle( 0, 0, width, height ) );  
    ball2.setBoundingBox( new Rectangle( 0, 0, width, height ) );  
}
```

4

För att ändra storlek på bollen finns det en variabel för diameter i klassen ball. Sedan skapades det en funktion i klassen ball likt funktionen för färgsättning. Den nya funktionen tar en integer som argument för att sedan tilldela variabeln för diameter värdet på argumentet.

```
public void setDiameter( int d ) {  
    diameter = d;  
}  
...  
ball1.setDiameter(5);
```

5

För att lägga till en boll skapades det ett till objekt av klassen ball. Sedan gjordes det individuella anrop för färgsättning, storlek och hastighet.

Ändringar:

```
//Skapar två objekt av klassen Ball  
ball1 = new Ball( width / 10, height / 5, 10, 10 );  
ball2 = new Ball( width / 10, height / 5, 5, 5 );  
ball1.setBoundingBox( new Rectangle( 0, 0, width, height ) );  
ball2.setBoundingBox( new Rectangle( 0, 0, width, height ) );  
ball1.setColor(Color.blue);  
ball1.setDiameter(5);  
ball2.setColor(Color.red);  
ball2.setDiameter(30);  
  
...  
  
ball1.paint( g );  
ball2.paint( g );  
...  
  
ball1.action();  
ball2.action();
```

6

I kalssen ball skapades det en boolean med namnet "size" som default är false. Denna variabel används för att bestämma om bollen skall krympa eller växa i funtionen "constrain". Om diameter är mindre eller lika med 5 sätts "size" till true och om diameter är större eller lika med 40 sätts "size" till false. Beroende på om "size" är true eller false kommer diametern att öka med 1 respektive minska med 1 i funtionen "action" Ändringar:

```
private boolean    size = false;

...

//Ändringar i constrain
//Beroende på nuvarade diametern skall bollen öka eller minska i storlek.
if (diameter <= 5) size = true;
if (diameter >=40) size = false;

...

//Ändringar i action
// Ökar eller minskar diameter på bollen
if (size == false) diameter--;
if (size == true) diameter++;
```