

# Inlämningsuppgifter för vecka 7

## Anmärkningar

- Hur man lämnar in och vad som ska lämnas in förklaras i Blackboard. Denna dokument beskriver uppgifterna. Uppgifterna finns i Java filer som refereras i varje uppgift i detta dokument. I varje .java fil finns en kommentar som förklarar uppgifterna på engelska.
- När du löser veckans uppgifter behöver du bara använda det vi har presenterat under första, andra, tredje, fjärde, femte, sjätte och sjunde veckorna.
  - Många operationer och några metoder för att bygga upp uttryck i olika typer,
  - if- och switch kommandon,
  - for- och while kommandon,
  - tilldelning och `System.out.println`,
  - fält
  - standard input
  - omdirigering av standard input och standard output
  - statiska metoder,
  - klasser som programbibliotek
  - att rita från ett Java program
  - att använda fördefinierade abstrakta datatyper och objekt
  - datatyperna `Color`, `Picture` och `String`
  - att definiera abstrakta datatyper med klasser med instansvariabler, konstanter, konstruerare och icke statiska metoder.
  - enkla användargränssnitt programmerade enligt MVC mönstret.
- Inget nytt tillkommer som ingår i tentan.

# Inlämningsuppgifter - del 1

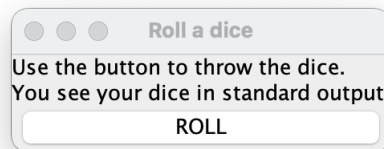
1. Filen *E1.java* innehåller två klasser:

- en klass *E1* som är *public* med en konstant *INSTR* av typen *String* och metoden *main*, samt
- en klass *Roller* som inte är *public* med metoden *actionPerformed*.

Du ska programmera definitionerna av metoden *main* i klassen *E1* och av metoden *actionPerformed* i klassen *Roller*. När du har programmerat klar ska man kunna köra programmet med

```
java E1
```

Då ska ett GUI fönster öppnas



Förnstret ska ha titeln *Roll a dice*. I fönstret ska en *JLabel* med texten från konstanten *INSTR* visas. Även ett *JButton* med texten *ROLL* ska visas. När man trycker på knappen ska programmet skriva till standard output en tärningskast (ett slump heltal mellan 1 och 6). Så här kan det se ut efter att ha tryckt på knappen tre gånger:

```
java E1
4
6
1
```

Programmet ska avslutas när användaren stänger GUI fönstret. Det du ska göra är:

- Programmera metoden *actionPerformed* i klassen *Roller*. Den ska bara beräkna ett slump heltal mellan 1 och 6 och skriva ut det på standard output.
- Programmera *main* för att skapa ett *JFrame*, ett *JLabel* och ett *JButton*, lägga etiketten och knappen i fönstret för att det ska se ut som i bilden ovan, samt koppla en *Roller* som händelsehanterare för knappen.

2. I filen *DiceModel.java* finns en klass *DiceModel* som ska vara en datatyp för tärningar. API:n för datatypen är

```
public class DiceModel
-----
    DiceModel()
    void roll()
    int  read()
```

Klassen innehåller en metod *main* för att kunna visa vad en *DiceModel* kan göra. När du kör programmet innan du har programmerat ser du följande:

```
java DiceModel 10
0 0 0 0 0 0 0 0 0 0
```

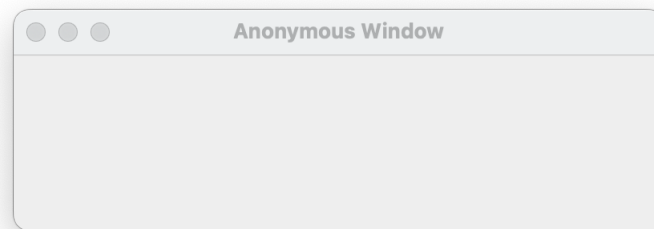
Det du ska göra är att komplettera implementationen av datatypen. Klassen innehåller redan den instansvariabel som behövs. Du behöver bara programmera konstrueraren och de två metoderna så att

- konstrueraren initierar värdet av instansvariabeln till ett slump heltal mellan 1 och 6.
- metoden `roll` tilldelar instansvariabeln ett slump heltal mellan 1 och 6.
- metoden `read` returnerar värdet av instansvariabeln.

När du kör programmet efter att ha programmerat klart kan utskriften bli:

```
java DiceModel 10
5 6 6 2 6 5 5 5 4 4
```

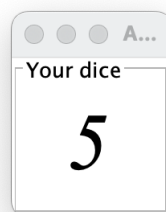
3. I filen *DiceView.java* finns en klass *DiceView* som ska implementera ett sätt att visa ett tärningsvärde i en GUI. Klassen innehåller en metod `main` för att visa vad en *DiceView* kan göra genom att lägga en *DiceView* i ett GUI fönster. Innan du programmerar kan programmet köras och då bör du se ett litet fönster som, om du förstorar det kan du se att det har titeln *Anonymous Window*.



Det du ska göra är att programmera färdig konstrueraren och metoden `paintComponent`. Utseendet ska vara enligt följande

- bakgrundsfärgen ska vara vit.
- på kanten (Border) ska det stå *Your dice*
- fonten ska vara SERIF och ITALIC
- JPanel ska vara kvadratisk (100 x 100).
- tärningsvärdet från instansvariabeln ska visas centrerat i panelen.

När du har programmerat klart och kör programmet kan det som visas vara



Du kan ta hjälp av koden för *CounterView* som vi programmerade i föreläsningen:

```

import javax.swing.JPanel;
import javax.swing.border.Border;
import javax.swing.border.EtchedBorder;
import javax.swing.BorderFactory;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.Color;
import java.awt.Graphics;

public class CounterView extends JPanel{

    private CounterModel c;

    public CounterView(CounterModel cm){
        c = cm;
        setPreferredSize(new Dimension(150,100));
        setFont(new Font(Font.SANS_SERIF,Font.BOLD,50));
        setBackground(Color.MAGENTA);
        setOpaque(true);
        Border b = BorderFactory.createEtchedBorder(EtchedBorder.LOWERED);
        Border tb = BorderFactory.createTitledBorder(b, "The counter");
        setBorder(tb);
    }

    public void paintComponent(Graphics g){
        super.paintComponent(g);
        g.setColor(Color.BLACK);
        g.drawString("" + c.read(),40,70);
    }
}

```

4. I filen *DiceController.java* finns en klass *DiceController* som du ska programmera färdig genom att modifiera klassen *CounterController* som vi programmerade i föreläsningen för att styra en räknare. För denna uppgift ska klassen styra en tärning istället. Här är koden du ska modifiera:

```

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class CounterController implements ActionListener{

    private CounterModel c;
    private CounterView cv;

    public CounterController(CounterModel cm, CounterView cv){
        c = cm;
        this.cv = cv;
    }

    public void actionPerformed(ActionEvent e){
        c.inc();
        cv.repaint();
    }
}

```

5. I filen *E5.java* finns en klass *E5* med bara en metod *main*. Programmet ska använda en GUI för att simulera tärningskast. Du ska programmera färdig *main* för att

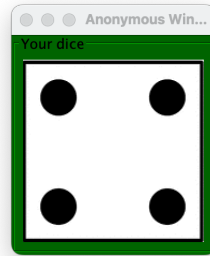
- skapa en *DiceModel*, en *DiceView* och en *DiceController*.
- skapa ett GUI fönster med en *DiceView* och en *JButton* med text *ROLL*.
- koppla programmets *DiceController* till knappen.

När du har programmerat klart ska programmet öppna ett GUI fönster så som det visas här (vi visar start fönstret, fönstret efter en knapptryckning och efter två knapptryckningar):

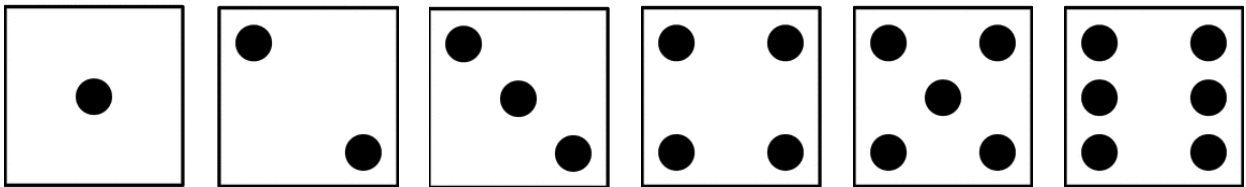
java E5



6. I filen *DiceView2.java* finns en klass *DiceView2* som ska implementera ett nytt sätt att visa en tärningsvärde i en GUI. När du har programmerat färdig ska man kunna köra programmet för att öppna ett fönster så som detta:



I zip-filen finns filerna *dice1.gif*, *dice2.gif*, *dice3.gif*, *dice4.gif*, *dice5.gif* och *dice6.gif* med bilderna



som du ska använda i programmet. Bilder kan visas i en GUI genom att lägga de i en *JLabel*:

- För att skapa en *JLabel* som visar en bild använder man konstrueraren med en bild istället för en text:

```
new JLabel(new ImageIcon(filename))
```

Bilden ska finnas i filen med namnet *filename* (som är en *String*).

- För att ändra bilden i en *JLabel* använder man metoden *setIcon* som i

```
.setIcon(new ImageIcon( filename ))
```

Det du ska göra är att komplettera konstrueraren och metoden *paintComponent*.

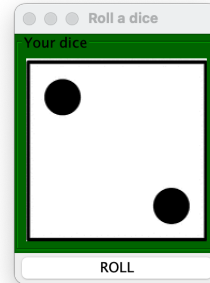
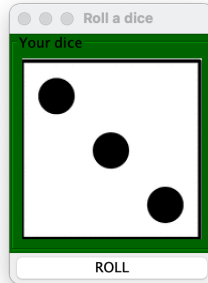
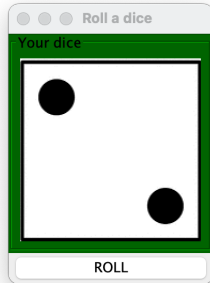
- I konstrueraren ska du bestämma bakgrundsfärgen till 0, 100, 0. Kanten ska vara som i *DiceView*. Du ska skapa en *JLabel* med bilden för tärningen med värde 1 och initiera instansvariabeln (som redan finns i klassen) med detta. Du ska även använda *add* så att det finns i panelen (istället för den text som fanns i *DiceView*).
- I *paintComponent* måste du uppdatera bilden i instansvariabeln med den bild som motsvarar tärningens värde.

Vi föreslår att du utgår från koden för *DiceView*.

7. I filen *DiceController2.java* finns klassen *DiceController2*. Du ska komplettera koden så att den är som *DiceController* men använder en *DiceView2* istället för en *DiceView*.

8. I filen *E8.java* finns en klass E8 med bara en metod `main`. Du ska komplettera programmet så att den liknar E5 men där du använder `DiceView2` och `DiceController2` istället för `DiceView` och `DiceController`. När du har programmerat färdig ska programmet öppna ett GUI fönster så som det visas här (vi visar start fönstret och efter två knapptryckningar):

java E8



## Inlämningsuppgifter - del 2

Programmera en applikation Bonus som använder en GUI för att simulera att man kastar fem tärningar. Programmet ska styras med bara ett JButton. Alla fem tärningar ska visas med bilder bredvid varandra. Programmet ska även visa en JPanel med ett meddelande som indikerar om alla fem tärningar är jämna tal eller alla är udda tal eller inget av dessa. Meddelandet ska vara ett av dessa beroende på utfallet: *No Luck!*, *Evens*, *Odds*.

Här ser du hur det kan se ut när man kör programmet:

```
java Bonus
```

