CHALMERS TEKNISKA HÖGSKOLA
Datavetenskap

TDA547

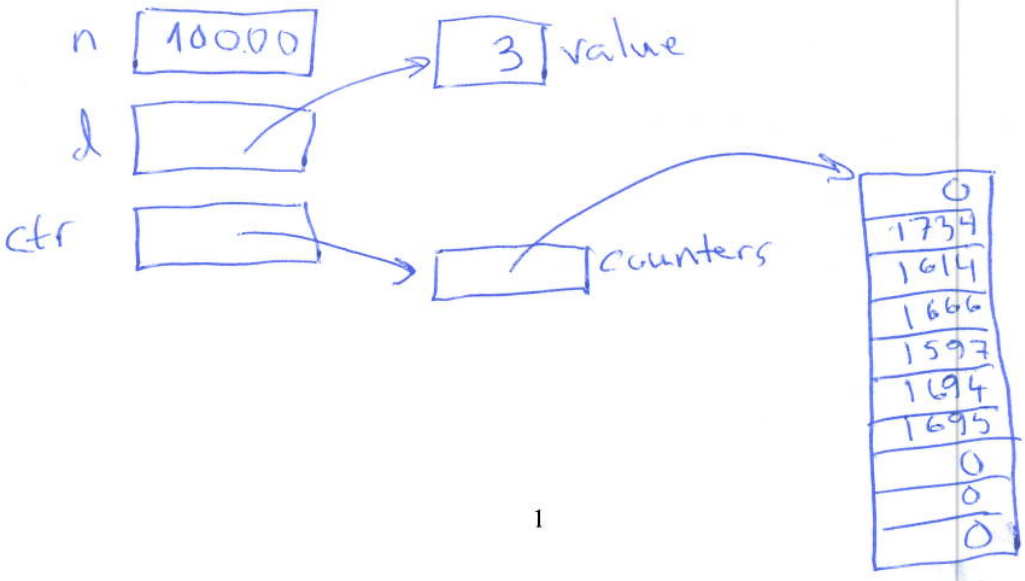
Tentamen i Objektorienterad programmering

Fredagen 9 mars 2012, 8.30 – 12.30.
Förslag till lösningar.

1. (a)
```
public class Dice {
    int value;

    public Dice() {roll();}

    public void roll() {
       value = (int)(6*Math.random()) + 1;
    }

    public int getValue() {return value;}
}
```

(b)
```
public class Counters {

    private int[] counters;

    public Counters(int maxValue) {
        counters = new int[maxValue + 1];
    }

    public void count(int i) {counters[i]++;}

    public int getCounter(int i) {return counters[i];}

    public int getMaxValue() {return counters.length - 1;}
}
```

(c)



2.
```
public class Uppgift2 {

    public static double average(double[] a) {
        double sum = 0;
        for (int i=0; i< a.length; i++)
            sum = sum + a[i];
        return sum/a.length;
    }

    public static void main(String[] args) {
        double[] v = new double[args.length];
        for (int i=0; i<v.length; i++)
            v[i] = Double.parseDouble(args[i]);
        System.out.println("Medelvärdet är " + average(v));
    }

    public static double average(Scanner in) {
        double sum = 0;
        int count = 0;
        while (in.hasNextDouble()) {
            sum = sum + in.nextDouble();
            count++;
        }
        return sum/count;
    }
}
```

3.
```
private static boolean hasWhiteNeighbor(int x, int y, boolean[][] image) {
    if (x<0 || x>= image.length || y<0 || y>=image[0].length)
        return true;
    return !image[x][y-1] || !image[x][y+1] || !image[x-1][y] || !image[x+1][y];
}

public static boolean[][] contour(boolean[][] image) {
    boolean[][] res = new boolean[image.length][image[0].length];
    for (int x=0; x<res.length; x++)
        for (int y=0; y<res[0].length; y++)
            res[x][y] = image[x][y] && hasWhiteNeighbor(x,y,image);
    return res;
}
```

4. (a)
```
public class BitmapModel {

  private boolean[][] pixels;

  public BitmapModel(int w, int h) {
    pixels = new boolean[w][h];
  }

  public void flip(int x, int y) {
    pixels[x][y] = !pixels[x][y];
  }
```

```
  public int getWidth() {
    return pixels.length;
  }

  public int getHeight() {
    return pixels[0].length;
  }

  public boolean getPixel(int x, int y) {
    return pixels[x][y];
  }
}
```

(b) Till modellklassen läggs metoden

```
public void contour() {
   pixels = Uppgift3.contour(pixels);
}
```

Till `main`-rutinen läggs

```
JButton but = new JButton("Contour");
but.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e) {
            m.contour();
            v.repaint();
        }
    });
f.add(but,BorderLayout.SOUTH);
```

Dessutom måste `m` och `v` deklareras att vara `final`.

5.
```
private static void fill(int x, int y, boolean[][] pixels) {
    if (!pixels[x][y]) {
        pixels[x][y] = true;
        if (x>0) fill(x-1,y,pixels);
        if (y>0) fill(x,y-1,pixels);
        if (x<pixels.length-1) fill(x+1,y,pixels);
        if (y<pixels[0].length-1) fill(x,y+1,pixels);
    }
}
```