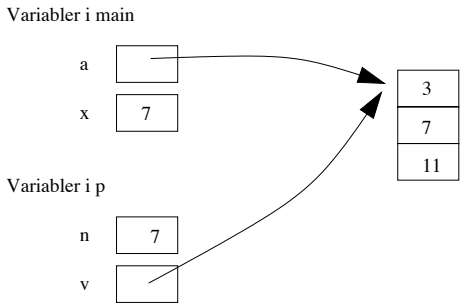
Objektorienterad programmering E
Lösningar till tentamen 20 augusti 2010.

1. Situationen när `p`'s kod ska börja exekvera är följande:



   Därefter uppdateras `n` till värdet 2 och `v[0]`, dvs element 0 i det fält som nås genom att följa pekaren från `v` i `p`, sätts till 8. När vi kommer tillbaka till `main` har `x` inte ändrats, medan det fält som nås via pekaren i `a` är detsamma som det som `p` uppdaterar. Utskriften blir därför

```
x=7
a[0]=8
```

2.
```
public class Uppgift2 {

    public static void reverse(int[] a) {
        for (int i=0; i<a.length/2; i++) {
            tmp = a[i];
            a[i] = a[a.length-1-i];
            a[a.length-1-i] = tmp;
        }
    }

    public static void main(String[] args) {
        int[] a = new int[args.length];
        for(int i=0; i<a.length; i++)
            a[i] = Integer.parseInt(args[i]);
        reverse(a);
        for (int i=0; i<a.length; i++)
            System.out.print(a[i] + " ");
        System.out.println();
    }
}
```

3.
```
public class Uppgift3 {

    public static int[] reverse1(int[] a) {
        int[] res = new int[a.length];
        for (int i=0; i<a.length; i++)
            res[i] = a[a.length-1-i];
        return res;
```

```
    }

    public static void main(String[] args) {
        int[] a = new int[args.length];
        for(int i=0; i<a.length; i++)
            a[i] = Integer.parseInt(args[i]);
        int[] b = reverse(a);
        for (int i=0; i<b.length; i++)
            System.out.print(b[i] + " ");
        System.out.println();
    }
}
```

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)
```
import java.awt.*;
import javax.swing.*;

public class BitmapMain {

  public static void main(String [] args) {
    JFrame f = new JFrame();
    BitmapModel m = new BitmapModel(15,15);
    BitmapView v = new BitmapView(m);
    f.add(v);
    f.pack();
    f.setVisible(true);
  }
}
```

(c)
```
public void mouseClicked(MouseEvent e) {
```

```
        model.flip(e.getX()/SIZE,e.getY()/SIZE);
        repaint();
    }
```

5. (a)
```
public class SmallIntSet  {

    private boolean[] isElem;

    public SmallIntSet(int maxElem) {
        isElem = new boolean[maxElem+1];
    }

    public void insert(int n) {
        if (n>=isElem.length)
            throw new IllegalArgumentException("SmallIntSet.insert: argument out o
        isElem[n] = true;
    }

    public boolean isMember(int n) {
        if (n>isElem.length)
            throw new IllegalArgumentException("SmallIntSet.isMember: argument out
        return isElem[n];
    }
}
```

Anm: För full poäng på uppgiften krävs inte att man kastar undantag som ovan; det är OK att lita på att argumenten är inom intervallet.

(b) Vi lägger till nedanstående metod och klass samt i klasshuvudet påpekandet att klassen `implements Iterable<Integer>` (OBS: I uppgiften står att metoden ska returnera en `Iterator`, inte `Iterator<Integer>`; detta är ett minne från Java 1.4; borde i dag uppdateras till nedanstående).

```
    public Iterator<Integer> iterator() {
        return new MyIterator();
    }

    private class MyIterator implements Iterator<Integer> {

        private int nextElem;

        public MyIterator() {
            nextElem=0;
            advance();
        }

        private void advance() {
            while (nextElem < isElem.length && !isElem[nextElem])
               nextElem++;
        }

        public boolean hasNext() {return nextElem < isElem.length;}

        public Integer next() {
```

```
            Integer res = nextElem;
            nextElem++;
            advance();
            return res;
        }

        public void remove() {
            throw new UnsupportedOperationException("SmallIntSet.remove");
        }
    }
```