Lösningsförslag till övning 3.

Uppgift 1

```
public class Boss  extends Employee {
    private double bonus;
    private boolean maid;
    public Boss() { }
    public Boss(String name, String address,  double payRate,  double bonus, boolean maid) {
        super(name, address, payRate);
        this.bonus = bonus;
        this.maid = maid;
    }
    public void setBonus( double moreBonus) {
        bonus = moreBonus;
    }
    public double getBonus() {
        return bonus;
    }
    public void setMaid(boolean maid) {
        this.maid = maid;
    }
    public boolean getMaid() {
        return maid;
    }
    public double getPayRate() {
        return  super.getPayRate () + bonus;
    }
}//Boss
```

Uppgift 2 *Lösningen reviderad 090917*

```
import java.util.*;

public class Die {
   private int nrOfSides;
   private int nrOfDots;
   private static Random dieRandom = new Random();

   public Die(int nrOfSides) {
       this.nrOfSides = nrOfSides;
      roll();
   }

    public int getNrOfSides() {
      return nrOfSides;
   }

   public int getNrOfDots() {
      return nrOfDots;
   }

   public void roll() {
      nrOfDots = dieRandom.nextInt(nrOfSides) + 1;
   }
}

public class Die6 extends Die {
   public Die6() {
      super(6);
   }
}

public class DoubleDie extends Die6 {
   public int getNoOfDots() {
      return (int)Math.pow(2,super.getNrOfDots());
   }
}
```

```
public class TestDice {

   public static void main(String[] arg) {
      Die[] theDice = {new Die6(), new Die(11), new DoubleDie()};
      for (int i = 0; i < theDice.length; i = i + 1) {
         theDice[i].roll();
         System.out.println(theDice[i].getNrOfDots());
      }
      System.out.println("Värdet av tärningarna är: " +
                         getValue(theDice));
   }

   public static int getValue(Die[] theDice) {
      int sum = 0;
      int prod = 1;
      for (int i = 0; i < theDice.length; i = i + 1)  {
         if (theDice[i] instanceof DoubleDie)
            prod = prod * theDice[i].getDots();
         else
            sum = sum + theDice[i].getDots();
      }
      return sum*prod;
   }
}
```

Uppgift 3

```
import java.awt.*;
public class GeometricObject {
    private Color color;
    private Point position;

    protected GeometricObject(){
        position = new Point(0,0);  //placeras i punkten  (0,0)
        color = Color.WHITE;       //ges färgen vit
    }
    protected GeometricObject(Point position, Color color){
        this.position = position;
        this.color = color;
    }
    public void setColor(Color color) {
        this.color = color;
    }
    public Color getColor() {
        return color;
    }

    public void setPosition(Point position) {
        this.position = position;
    }
    public Point getPosition() {
        return position;
    }
    public void move(Point position) {
        this.position = position;
    }
    public String toString() {
        return "Color: " + color + "\nPosition: " + position;
    }
}// GeometricObject
```

```
import java.awt.*;
public class Rectangle extends GeometricObject {
    private double width;
    private double length;
    public Rectangle(){
        this(1.0, 1.0);       //ges bredden 1 och längden 1
    }
    public Rectangle(double width, double length){
        this.width = width;
        this.length = length;
    }
    public Rectangle(double width, double length, Point position, Color color){
        super(position, color);
        this.width = width;
        this.length = length;
    }

    public void setWidth(double width) {
        this.width = width;
    }

    public double getWidth() {
        return width;
    }

    public void setLength(double length) {
        this.length = length;
    }

    public double getLength() {
        return length;
    }

    public double findArea() {
        return width*length;
    }
    public double findPerimeter() {
        return 2*(width + length);
    }

    public String toString() {
        return super.toString() + "\nWidth: " + width + "\nLength: " + length;
    }
}// Rectangle
```

```
import java.awt.*;
public class Circle extends GeometricObject {
    private double radius;
    public Circle(){
        this.radius = 1.0;
    }
    public Circle(double radius){
        this.radius = radius;
    }
    public Circle(double radius, Point position, Color color){
        super(position, color);
        this.radius = radius;
    }
    public void setRadius(double radius) {
        this.radius = radius;
    }

    public double getRadius() {
        return radius;
    }
    public double findArea() {
        return Math.PI*radius*radius;
    }
    public double findPerimeter() {
        return 2*radius*Math.PI;
    }
    public String toString() {
        return super.toString() + "\nRadius: " + radius;
    }
}// Circle
```

```
import java.awt.*;
public class Cylinder extends Circle {
    private double height;
    public Cylinder(){
        this.height = 1.0;
    }
    public Cylinder(double radius, double height){
        super(radius);
        this.height = height;
    }
    public Cylinder(double radius, double height, Point position, Color color){
        super(radius, position, color);
        this.height = height;
    }
    public void setHeight(double height) {
        this. height = height;
    }

    public double getHeight () {
        return height;
    }
    public double findArea() {
        return 2* super.findArea() +  height*findPerimeter();
    }
    public double findVolym() {
        return super.findArea() * height;
    }
    public String toString() {
        return super.toString() + "\nHeight: " + height;
    }
}// Cylinder
```

TESTPROGRAM:

```
import java.awt.*;
public class TestGeo {
    public static void main(String[] arg) {
        GeometricObject[] a = {new Rectangle(10, 10, new Point(5, 5),Color.BLACK),
                               new Circle(20, new Point(15, 15),Color.RED),
                               new Cylinder(20, 100, new Point(25, 25),Color.GREEN)};
        for (int i = 0; i < a.length; i = i + 1) {
            System.out.println(a[i] + "\n");
        }
    }
}
```

Ovanstående testprogram ger utskriften

```
Color: java.awt.Color[r=0,g=0,b=0]
Position: java.awt.Point[x=5,y=5]
Width: 10.0
Length: 10.0

Color: java.awt.Color[r=255,g=0,b=0]
Position: java.awt.Point[x=15,y=15]
Radius: 20.0

Color: java.awt.Color[r=0,g=255,b=0]
Position: java.awt.Point[x=25,y=25]
Radius: 20.0
Height: 100.0
```

Förklara varför det blir ett kompileringsfel i exemplet nedan:

```
    import java.awt.*;
    public class TestGeo {
        public static void main(String[] arg) {
            GeometricObject[] a = {new Rectangle(10, 10, new Point(5, 5),Color.BLACK),
                                   new Circle(20, new Point(15, 15),Color.RED),
                                   new Cylinder(20, 100, new Point(25, 25),Color.GREEN)};
            for (int i = 0; i < a.length; i = i + 1) {
                System.out.println(a[i] + "\n");
                System.out.println("Arean = " + a[i].getArea()); // KOMPILERINGSFEL!!!
            }
        }
}
```

Inför en getArea-metod i GeometricObject, och påpeka att abstrakta klasser kommer att gås igenom på nästa föreläsning.