

Opgave 1.

```
public class Vector{

    private double[] elementen;

    public Vector(int n){
        elementen = new double[n];
    }

    public double get(int i){
        if (0 <= i && i < elementen.length)
            return elementen[i];
        else
            return Double.MIN_VALUE;
    }

    public void set(int i, double e){
        if (0 <= i && i < elementen.length)
            elementen[i] = e;
    }

    public Vector som(Vector that){
        int n = elementen.length;
        if (n == that.elementen.length){
            Vector res = new Vector(n);
            for(int i = 0; i < n;i++)
                res.elementen[i] = this.elementen[i] + that.elementen[i];
            return res;
        }else
            return new Vector(0);
    }

    public double lengte(){
        double res = 0;
        int n = elementen.length;
        for(int i = 0;i < n;i++)
            res = res + elementen[i]*elementen[i];
        return Math.sqrt(res);
    }
}
```

Opgave 2.

```
public static boolean isCijfer(char c){
    return '0' <= c && c <= '9';
}

public static String[] verdeel(String s){
    String s1 = "";
    String s2 = "";
    String[] res = new String[2];
    for(int i = 0;i < s.length();i++){
        char c = s.charAt(i);
        if (isCijfer(c))
            s1 = s1 + c;
        else
            s2 = s2 + c;
    }
    res[0] = s1;
    res[1] = s2;
    return res;
}
```

Opgave 3.

```
public double omtrek(){
    return 2 * (hoogte + breedte);
}

public boolean bevat(Punt p){
    double xlb = linksBoven.getX();
    double ylb = linksBoven.getY();
    double x = p.getX();
    double y = p.getY();
    return (xlb <= x && x <= xlb + breedte) &&
           (ylb <= y && y <= ylb + hoogte);
}

public Rechthoek roteer(){
    double xlb = linksBoven.getX();
    xlb = xlb-breedte;
    double ylb = linksBoven.getY();
    Punt p = new Punt(xlb,ylb);
    return new Rechthoek(p,hoogte,breedte);
}
```

