

## Oefening p 62

```
package testoefeningen;
```

```
/** Gegeven de volgende klassen: Trapezium, Rechthoek, Vierkant
```

1. Maak een object aan van elke klasse. Kies je eigen waarden voor de parameters.
2. Duidt aan welke constructors worden opgeroepen en welke superconstructors.
3. Geef voor elk object de waarde van de attributen.
4. Bereken de oppervlakte van de figuren en print deze af.
5. Voeg een methode toe aan Rechthoek die de oppervlakte berekent.
6. Maak van de rechthoek een subklasse van het trapezium. Dan zijn de attributen 'breedte' en 'lengte' in feite niet meer nodig, die mag je dus schrappen. De oppervlaktmethode ook.
7. Vierkant erft alle attributen over, maar heeft er in feite slechts 1 nodig. Schrap de 'extends Rechthoek', voeg het nodige attribuut toe en de methode voor oppervlakteberekning. \*/

```
public class KlasseOefening1 {  
    /** PROGRAMMA */  
    public static void main(String[] args) {  
  
    }  
}  
class Trapezium {  
    double lengte1, lengte2, breedte;  
    Trapezium(double lengte1, double lengte2, double breedte){  
        this.lengte1 = lengte1;  
        this.lengte2= lengte2;  
        this.breedte = breedte;  
    }  
    double oppervlakte(){ return breedte * (lengte1 + lengte2)/2; }  
}  
class Rechthoek {  
    double breedte, lengte;  
    Rechthoek(double lengte, double breedte){  
  
        this.breedte = breedte;  
        this.lengte = lengte;  
    }  
}  
class Vierkant extends Rechthoek {  
  
    Vierkant(double zijde){  
        super(zijde, zijde);  
    }  
}  
}
```

# Oplossing

1. Maak een object aan van elke klasse. Kies je eigen waarden voor de parameters.

Dit doe je in de main:

```
public static void main(String[] args) {  
    Trapezium trap = new Trapezium(10, 20, 15); // (1)  
    Rechthoek recht = new Rechthoek(60, 32); // (2)  
    Vierkant vierk = new Vierkant(23); // (3)  
}
```

2. Duidt aan welke constructors worden opgeroepen en welke superconstructors.

(1) bij het aanmaken van de Trapezium wordt de volgende constructor-methode opgeroepen:  
Trapezium(double lengte1, double lengte2, double breedte)

(2) bij het aanmaken van de Rechthoek wordt de volgende constructor-methode opgeroepen:  
Rechthoek(double lengte, double breedte)

(3) Nu wordt Vierkant(double zijde) opgeroepen, maar omdat vierkant een subklasse van Rechthoek is wordt ook een rechthoek aangemaakt. Met super(zijde, zijde); wordt de constructor van rechthoek opgeroepen met de juiste 2 parameters (Rechthoek(double lengte, double breedte)).

3. Geef voor elk object de waarde van de attributen.

Object trap: lengte1 = 10, lengte2 = 20, breedte = 15

Object recht: breedte = 60, lengte = 32

Object vierk: breedte = 23, lengte = 23

We weten dat vierkant in feite maar 1 parameter heeft (zijde), maar hier is hij gedefinieerd als een rechthoek waarvan breedte en lengte gelijk zijn.

4. Bereken de oppervlakte van de trapezium en print deze af.

```
double opp = trap.oppervlakte();
```

```
System.out.println("De oppervlakte van trapezium "+ trap +" = "+ opp);
```

5. Voeg een methode toe aan Rechthoek die de oppervlakte berekent.

We voegen de volgende methode toe aan de klasse Rechthoek (tussen de accolades van de klasse)

```
double oppervlakte(){ return breedte * lengte; }
```

6. Maak van de rechthoek een subklasse van het trapezium. Dan zijn de attributen 'breedte' en 'lengte' in feite niet meer nodig, die mag je dus schrappen. De oppervlaktmethode ook.

De klasse Rechthoek wordt dan

```
class Rechthoek {  
    Rechthoek(double lengte, double breedte){  
        super(lengte, lengte, breedte);  
    }  
}
```

Met super() roepen we de constructor van Trapezium op. Deze heeft 3 parameters.

7. Vierkant erft alle attributen over, maar heeft er in feite slechts 1 nodig. Schrap de 'extends Rechthoek', voeg het nodige attribuut toe en de methode voor oppervlakteberekening.

```
class Vierkant {  
    double zijde;  
    Vierkant(double zijde){  
        this.zijde = zijde;  
    }  
    double oppervlakte(){ return zijde * zijde; }  
}
```