# Abstracte klasse en overerven

### Voorbeeld

### Te gebruiken bij:

-

### Eigenschappen:

-

### Info

Overerving en abstracte klassen zijn concepten in objectgeoriënteerd programmeren die helpen bij het bouwen van hiërarchie en het delen van gedrag en eigenschappen tussen klassen. Hier is een voorbeeld van code om het concept van overerving en abstracte klassen uit te leggen:

java

// Abstracte klasse 'Shape' met een abstracte methode en een gewone methode

abstract class Shape {

abstract void draw(); // Abstracte methode

void printInfo() { // Gewone methode

System.out.println("Dit is een vorm.");

}

}

// Concrete klasse 'Circle' die overerft van 'Shape'

class Circle extends Shape {

void draw() {

System.out.println("Een cirkel tekenen...");

}

}

// Concrete klasse 'Rectangle' die overerft van 'Shape'

class Rectangle extends Shape {

void draw() {

System.out.println("Een rechthoek tekenen...");

}

}

public class Main {

public static void main(String[] args) {

Shape circle = new Circle(); // Instantie van 'Circle' gemaakt als een 'Shape'

Shape rectangle = new Rectangle(); // Instantie van 'Rectangle' gemaakt als een 'Shape'

circle.draw(); // Roep de 'draw' methode van de cirkel aan

rectangle.draw(); // Roep de 'draw' methode van het rechthoek aan

circle.printInfo(); // Roep de 'printInfo' methode van de cirkel aan

rectangle.printInfo(); // Roep de 'printInfo' methode van het rechthoek aan

}

}

In dit voorbeeld hebben we een abstracte klasse genaamd 'Shape' met een abstracte methode draw() en een gewone methode printInfo(). De klasse 'Shape' kan niet direct worden geïnstantieerd, maar kan worden uitgebreid door concrete klassen.

We hebben twee concrete klassen, 'Circle' en 'Rectangle', die overerven van de abstracte klasse 'Shape'. Deze klassen implementeren de abstracte methode draw() en kunnen ook de gewone methode printInfo() van de abstracte klasse gebruiken.

In de main-methode maken we instanties van 'Circle' en 'Rectangle', maar we slaan ze op als 'Shape'-objecten. Dit wordt mogelijk gemaakt door polymorfisme, waarbij een object van de afgeleide klasse kan worden toegewezen aan een variabele van het basistype.

Binnen de main-methode roepen we de draw()- en printInfo()-methoden aan op de 'Shape'-objecten. Dit toont het overerven van gedrag van de abstracte klasse naar de concrete klassen. De 'Circle'-klasse gebruikt de implementatie van de abstracte methode 'draw' om een cirkel te tekenen, terwijl de 'Rectangle'-klasse de abstracte methode 'draw' implementeert om een rechthoek te tekenen.

Het gebruik van een abstracte klasse stelt ons in staat om gemeenschappelijke eigenschappen en gedragingen te definiëren in de abstracte klasse en deze te delen met de afgeleide klassen. Hierdoor kunnen we code hergebruiken en een hiërarchie van klassen creëren die onderling verbonden zijn door over