# Klasse en objecten

### Voorbeeld

### Te gebruiken bij:

-

### Eigenschappen:

-

### Info

#### overzicht

// Superklasse (hoofdklasse)

public class Superklasse {

// attribuut1 en attribuut2 behoren tot de Superklasse

private String attribuut1; // behoort tot Superklasse

private int attribuut2; // behoort tot Superklasse

// Constructor van de Superklasse

public Superklasse(String attribuut1, int attribuut2) {

this.attribuut1 = attribuut1;

this.attribuut2 = attribuut2;

}

// Getter en Setter voor attribuut1

public String getAttribuut1() {

return attribuut1;

}

public void setAttribuut1(String attribuut1) {

this.attribuut1 = attribuut1;

}

// Getter en Setter voor attribuut2

public int getAttribuut2() {

return attribuut2;

}

public void setAttribuut2(int attribuut2) {

this.attribuut2 = attribuut2;

}

}

// Subklasse 1 (afgeleide klasse van Superklasse)

public class Subklasse1 extends Superklasse {

// attribuut3 behoort tot Subklasse1

private boolean attribuut3; // behoort tot Subklasse1

// Constructor van Subklasse1

public Subklasse1(String attribuut1, int attribuut2, boolean attribuut3) {

// Roep de constructor van de Superklasse aan

super(attribuut1, attribuut2);

this.attribuut3 = attribuut3;

}

// Getter en Setter voor attribuut3

public boolean isAttribuut3() {

return attribuut3;

}

public void setAttribuut3(boolean attribuut3) {

this.attribuut3 = attribuut3;

}

}

// Subklasse 2 (afgeleide klasse van Superklasse)

public class Subklasse2 extends Superklasse {

// attribuut3 behoort tot Subklasse2

private double attribuut3; // behoort tot Subklasse2

// Constructor van Subklasse2

public Subklasse2(String attribuut1, int attribuut2, double attribuut3) {

// Roep de constructor van de Superklasse aan

super(attribuut1, attribuut2);

this.attribuut3 = attribuut3;

}

// Getter en Setter voor attribuut3

public double getAttribuut3() {

return attribuut3;

}

public void setAttribuut3(double attribuut3) {

this.attribuut3 = attribuut3;

}

}

// Voorbeeldgebruik

public class Main {

public static void main(String[] args) {

// Instantie van Subklasse1

Subklasse1 subklasse1 = new Subklasse1("waarde1", 2, true); // Instantie van Subklasse1 (object van Subklasse1)

// Instantie van Subklasse2

Subklasse2 subklasse2 = new Subkl

#### Stappen

De code begint met het definiëren van de Superklasse. Dit is de hoofdklasse waarvan andere klassen erven.

De Superklasse bevat twee attributen: attribuut1 (van het type String) en attribuut2 (van het type int). Deze attributen zijn privé en kunnen alleen worden benaderd via getters en setters.

De Superklasse heeft een constructor die attribuut1 en attribuut2 als parameters accepteert en de waarden ervan toewijst aan de overeenkomstige attributen.

Daarna volgen de getters en setters voor attribuut1 en attribuut2, waarmee de waarden kunnen worden opgehaald en gewijzigd.

Vervolgens worden de subklassen gedefinieerd: Subklasse1 en Subklasse2. Deze klassen erven van de Superklasse.

Subklasse1 voegt een extra attribuut attribuut3 (van het type boolean) toe.

Subklasse1 heeft een constructor die attribuut1, attribuut2 en attribuut3 als parameters accepteert. Deze constructor roept de constructor van de Superklasse aan met behulp van super(attribuut1, attribuut2) om de waarden van attribuut1 en attribuut2 in te stellen.

Subklasse1 bevat ook getters en setters voor attribuut3.

Op een vergelijkbare manier voegt Subklasse2 een extra attribuut attribuut3 (van het type double) toe en heeft het een constructor en getters/setters voor attribuut3.

In de Main-klasse worden objecten (subklasse1 en subklasse2) van zowel Subklasse1 als Subklasse2 geïnstantieerd.

De waarden van attributen worden opgehaald en gewijzigd met behulp van de getters en setters.

#### Zonder superklasse

public class MyClass {

// Variabelen

private int myVariable;

private String myString;

// Constructor

public MyClass(int variable, String string) {

myVariable = variable;

myString = string;

}

// Methoden

public void myMethod() {

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

}

public static void main(String[] args) {

// Instantie van MyClass maken

MyClass myObject = new MyClass(10, "Hello");

// Toegang tot variabelen en methoden

int variableValue = myObject.myVariable;

String stringValue = myObject.myString;

myObject.myMethod();

}

}

In dit voorbeeld hebben we een klasse genaamd MyClass met enkele privévariabelen, een constructor en een methode. In de main-methode maken we een instantie van MyClass genaamd myObject met behulp van de constructor. Vervolgens hebben we toegang tot de variabelen en de methode van de instantie.

Opmerking: De privévariabelen myVariable en myString kunnen alleen binnen de klasse MyClass worden gebruikt. Als je toegang tot deze variabelen wilt bieden buiten de klasse, kun je ze als public of protected declareren, afhankelijk van de vereisten van je programma.

Dit voorbeeld dient als basis voor het begrijpen van klassen, instanties, constructoren en variabelen in Java. Je kunt de waarden van de variabelen en de implementatie van de methode aanpassen en de resultaten observeren.

Ik hoop dat dit voorbeeld nuttig is! Als je verdere vragen hebt, sta ik klaar om te helpen.