

Klasse: Übung

Klasse als Grundlage

Die gesamte objektorientierte Programmierung fusst auf dem Konzept der Klasse – einem Konzept, dem man sich auf zwei verschiedenen Wegen nähern kann:

- über die Philosophie der objektorientierten Programmierung
- über die Klasse als neuen Datentyp

Klassen und Objekte

Im objektorientierten Denkmodell ist ein Objekt ein reales Ding, beispielsweise sie selbst, Ihr Nachbar, oder die Bäume, die Sie sehen, wenn Sie aus dem Fenster schauen. Einerseits ist jedes dieser Objekte einzigartig, andererseits gibt es zu jedem dieser Objekte auch gleichartige Objekte. Für eine Gruppe von gleichartigen Objekte verwenden wir den Begriff Klasse. Sie beschreiben die Dinge, die wir in dem Programm verwenden wollen

Sie dienen als Vorlage oder Gussform, aus der wir die Objekte, mit denen wir arbeiten wollen, erst erzeugen.

Klassen beschreiben ihre Objekte mit Eigenschaften und Verhalten

Die Verhaltensweise sind Tätigkeiten, die die Objekte der Klasse ausführen können, beziehungsweise die Operationen, die wir beim Programmieren auf die Objekte der Klasse anwenden können.

Wie Sie sehen, werden die Eigenschaften als Variablen und die Verhaltensweisen als Methoden in der Klasse definiert.

Anlegen einer Instanz

Instanzen einer Klasse werden wie bei einer normalen Variablendeklaration gebildet. Bei der Deklaration einer Klasse wird, wie bei einer Typendefinition üblich kein Speicherplatz reserviert. Erst durch das Anlegen von Instanzen wird Speicher bereitgestellt bzw. muss mit `new` angefordert werden.

```
// KlassenDemo

import java.lang.* ;

class Vektor

{
// Die Eigenschaften

    int x;

    int y; // Die Verhaltensweisen

void addieren ( Vektor v )

    {

    x += v.x;

    y += v.y ;

    }

    } /* end class Vektor */

public class KlassenDemo

{

    public static void main(String[] args )

    {

        Vektor v1 = new Vektor();
    }
}
```

```

        Vektor v2 = new Vektor();
    }
} /* end class KlassenDemo */

```

In der `main()` – Methode der Rahmenklasse `KlassenDemo` werden zwei Instanzen unserer Klasse `Vektor` erzeugt. Dies erfolgt mit dem `new`-Operator.

Konstruktor

Bei der Instanzbildung geschieht folgendes:

ein Objekt der Klasse wird erzeugt

der Konstruktor der Klasse wird aufgerufen

Die Aufgabe des Konstruktors besteht darin, dafür zu sorgen, dass das erzeugte Objekt korrekt initialisiert wird. Wenn kein Konstruktor aufgesetzt wird, erzeugt der Compiler intern einen Standardkonstruktor für die Klasse.

```

// KlassenDemo

import java.lang.* ;

class Vektor

{

// Die Eigenschaften

    int x;

    int y;

// Die Verhaltensweisen

void addieren ( Vektor v )

    {

        x += v.x;

        y += v.y ;

    }

// Konstruktor

void Vektor ()

    {

        x = 0 ;

        y = 0 ;

    }

} /* end class Vektor */

```

Konstruktoren besitzen keinen Rückgabewert.

In Java gibt es keine Destruktoren (Die Aufhebung der Zuordnung der Ressourcen wird von Java selber erledigt)

Zugriffsspezifizierer

Objekte sind selbständige, abgegrenzte Einheiten. Der Zugriff auf das Innenleben der Klassen wird daher mit Zugriffsspezifizierer, d.h. bestimmten Schlüsselwörtern bestimmt. Zugriffe von innen sind bei einer Klasse immer erlaubt. Anders liegt der Fall bei den Zugriffen von aussen. Hier kann der Ersteller der Klasse bei der Definition festlegen, wie weit die Zugriffsrechte (auch Sichtbarkeit genannt) gehen soll.

- **public:** Derart markierte Methoden und Instanzvariablen sind in den Methoden der Klasse selbst und von aussen sichtbar
- **protected:** sichtbar in der Klasse selbst und in davon abgeleiteten Klassen sowie in Klassen desselben Pakets (Package). Aufrufende Klassen aus anderen Packages können aber nicht zugreifen
- **private:** sind lediglich in der Klasse selbst sichtbar. Für abgeleitete Klassen und für Aufrufer von Instanzen bleiben private-Variablen verdeckt.

Methoden/Variablen ohne Zugriffsspezifizierer haben Standard-Sichtbarkeit (package scope) d.h. sie sind von Klassen (genauer: in deren Methoden) aufrufbar, die zum gleichen Package gehören.

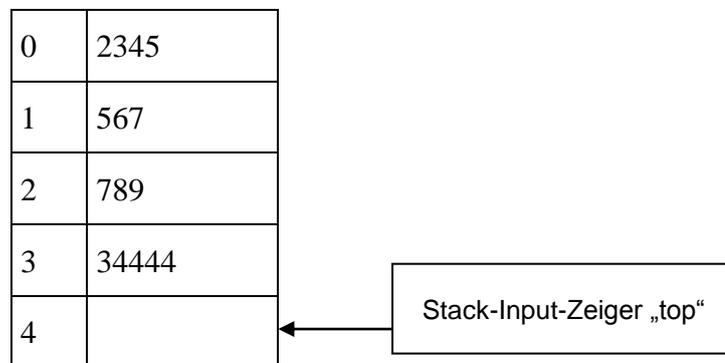
Zusammenfassung

Die Definition einer Klasse besteht aus folgenden Bestandteilen:

- Klassennamen
- Eigenschaften
- Verhaltensweisen
- Konstruktor
- Zugriffsspezifizierern

Aufgabe Klasse Stack

Wir wollen einen einfachen Stack programmieren. Schreiben Sie dazu eine Klasse Stack bestehend aus der Datenbasis und den Zugriffsmethoden auf diese Datenbasis. Durch einen Konstruktor soll die Datenbasis initialisiert werden. Schützen Sie die Datenbasis mit den entsprechenden Zugriffsspezifizierern. Methoden die nur intern in der Klasse verwendet werden, sollen ebenfalls gegen Zugriffe von aussen geschützt werden. (Gegen aussen bekannte Methoden sind fett markiert)



Datenbasis

```
final int anz_elemente = 5;

int [] stack = new int[anz_elemente]; // Speicher für Stack

int top = 0; // nächstes freies Feld -->

.....
```

Methoden

```
public int pop ( ); // Wert vom Stack holen

public void push ( int value ); // Wert auf Stack setzen

private bool full(); // Test ob Stack voll ist
```

```
private bool empty(); // Test ob Stack leer ist
public void display(); // Anzeige aller Elemente auf Stack
public int size(); // Anzahl Elemente auf Stack
```

Aufgabe Klasse Stacktest

Mit der Klasse Stacktest soll die Klasse Stack auf „Herz und Nieren“ getestet werden.