E.Tutorial
www.et.ethz.ch

Programmieren mit Java Modul 0

Programme erstellen in Java

Theorieteil

Inhaltsverzeichnis

Begriffe

Programmiersprache	Quelltext	Syntax
Programm	Compiler	
Programmierumgebung	Bytecode	Semantik
Editor	Klasse	
Algorithmus	Kommentar	Anweisung

Autoren:

Lukas Fässler, Barbara Scheuner

E-Mail:

et@ethz.ch

Datum:

4 April 2025

Version: 1.1

Hash: 98d9199

Trotz sorgfältiger Arbeit schleichen sich manchmal Fehler ein. Die Autoren sind Ihnen für Anregungen und Hinweise dankbar!

1 Modulübersicht

Die Entwicklung des Computers ermöglicht uns, Rechenarbeit durch Maschinen erledigen zu lassen. Der Computer kann jedoch allein keine Probleme lösen, sondern ihm muss ein Lösungsweg (eine Bearbeitungsvorschrift) gegeben werden. Dieser Lösungsweg wird ihm in Form eines **Programms** mitgeteilt. Dies geschieht wiederum in einer speziellen Sprache, der **Programmiersprache**. Eine Bearbeitungsvorschrift zur Lösung einer Aufgabe wird **Algorithmus** genannt. Hierbei fordern wir, dass ein Algorithmus seine Arbeit immer beendet, also nicht unendlich lange braucht, wenn er ausgeführt wird und für jede Eingabe eine sinnvolle Ausgabe generiert. *Algorithmus* ist somit ein recht abstrakter Begriff. Wir können z.B. ein Kuchenrezept oder eine Wegbeschreibung als einen Algorithmus verstehen. Wir betrachten hier hingegen nur Alogrithmen, die konkret in einer Programmiersprache ausformuliert worden sind.

2 Schreiben von Computerprogrammen

Wenn zwei Menschen miteinander kommunizieren, wird dies von vielen Dingen, wie beispielsweise Mimik und Gestik, begleitet. Auf die Frage „*Wie geht es dir?*“ kann eine Antwort „*Gut.*“ ganz unterschiedlich interpretiert werden, abhängig davon, wie der Antwortende dies zum Beispiel betont. Menschen besitzen einen Intellekt, der es ihnen ermöglicht, einen Dialog zu interpretieren und in einen Kontext zu setzen. Computer haben diese Fähigkeit nicht. Um mit einem Rechner zu kommunizieren, müssen wir uns exakt ausdrücken. Der Computer weiss nicht, was wir eigentlich gemeint haben, sollten wir uns falsch ausgedrückt haben. Für die ersten Computer war dies eine sehr mühselige Aufgabe, denn die Sprache, die ein Computer versteht, ist für Menschen nicht sehr intuitiv. Deshalb wurden sogenannte **Hochsprachen** entwickelt, die unserer natürlichen Sprache näher sind. In diesem Kurs werden Sie eine solche Sprache, nämlich **Java**, verwenden, um Algorithmen als Computerprogramme umzusetzen.

2.1 Computerprogramme bestehen aus Daten und Instruktionen

Ein **Computerprogramm** ist im Wesentlichen eine Auswahl von Daten und eine Folge von **Instruktionen**, die – wenn sie ausgeführt werden – jeweils eine bestimmte Funktion erfüllen. Eine Instruktion kann beispielsweise eine Berechnung ausführen. Zum besseren Verständnis können Sie sich, wie oben erwähnt, ein Kochrezept vorstellen. Es enthält als erstes die Mengenangaben der Zutaten (Daten) und danach die Reihenfolge der Schritte (Instruktionen), die man ausführen muss, um ein bestimmtes Gericht zu kochen. Das Grundschema eines Rezepts ist meistens dasselbe: zuerst die Zutaten, danach die einzelnen Arbeitsschritte.

Mit einem Computerprogramm verhält es sich ähnlich. Jedes Programm folgt ebenfalls einem Grundschema. Bei der Programmierung spricht man allerdings nicht von Schema, sondern von der **Syntax einer Programmiersprache**, d.h. von den Regeln, die für

den Aufbau eines Programms befolgt werden müssen. Wie bereits erwähnt, gibt es allerdings einen wesentlichen Unterschied zu den Schritten in einem Kochrezept. Bei den Instruktionen müssen wir präzise sein. Vorschriften analog zu „nach eigenem Ermessen würzen“ werden wir hier nicht finden, da der Computer sie nicht eindeutig auswerten kann.

Folgende Zeilen zeigen ein sehr einfaches Beispiel für ein Programm in der Programmiersprache Java:

```
public class HalloWelt {
  public static void main(String[] args) {
    System.out.println("Willkommen zur Javaprogrammierung.");
  }
}
```

Unser Programm enthält in diesem Fall

- ein **Grundgerüst**, bestehend aus einer Klasse mit dem Namen „*HalloWelt*“. Der Name der Klasse muss zwingend mit dem Namen der Datei übereinstimmen, in der das Programm gespeichert ist. Unser Code wird deshalb in der Datei *HalloWelt.java* gespeichert. Die Klasse enthält eine Methode, die „Hauptmethode“ (*main*) genannt wird.
- eine **Instruktion** in der Hauptmethode (`System.out.println()` als Anweisung).
- die **Daten** (hier den Text `Willkommen zur Javaprogrammierung.`).

Wird dieses Programm nun ausgeführt, wird folgende Zeile in die Konsole ausgegeben:

```
Willkommen zur Javaprogrammierung.
```

Das, was ein Programm ausführt, also seine Bedeutung, nennt man die **Semantik** des Programms.

2.2 Programme müssen übersetzt werden

Programme in einer Programmiersprache wie Java sind für uns Menschen lesbar und verständlich. Wie bereits erwähnt, versteht ein Computer sie aber nicht direkt, sondern nur nach einer Umwandlung in Instruktionen für seinen Prozessor. Diese sind für uns nicht nur schwer verständlich, sondern auch wesentlich simpler als die Anweisungen eines Programms in einer Hochsprache wie Java. Das heisst, eine einzelne Instruktion eines Programms führt zu einer Folge mehrerer Prozessor-Instruktionen.

Damit nun ein Computer unser Programm ausführen kann, müssen die Anweisungen des Programms in Instruktionen des Computers übersetzt werden. Für das Übersetzen von Programmen aus einer Programmiersprache in eine Folge von Prozessor-Instruktionen gibt es spezielle Computerprogramme, so genannte **Kompilierer** (*Compiler*, Übersetzer). Der Vorgang des Übersetzens wird deshalb auch **kompilieren** genannt.

2.2.1 Schreiben und Ausführen eines Java-Programms

Programme werden in Dateien gespeichert. Um diese Dateien editieren und abspeichern zu können, brauchen wir einen **Editor**. Für Java gibt es eine Vielzahl von Editoren und Entwicklungsumgebungen. Nachdem Sie ein Programm geschrieben haben, wird es als **Quellcode** gespeichert. Dateien, die Java-Quellcode enthalten, haben die Erweiterung *.java*. Im nächsten Schritt übersetzt der Compiler den Quellcode in ein Format namens **Bytecode**, das für die Anwenderin oder den Anwender nicht lesbar ist. Dieser bekommt die Endung *.class*.

3 Anweisung

Eine **Anweisung** (*statement*) ist die kleinste ausführbare Einheit eines Programms. Wie in vielen anderen Programmiersprachen auch, wird eine Anweisung mit einem Strichpunkt oder **Semikolon** (`;`) abgeschlossen.

Schreibweise:

```
Anweisung;
```

Beispiel:

```
System.out.println("Hallo Welt");
```

4 Kommentare

Kommentare sind Lesehilfen für uns Menschen. Sie dienen der Dokumentation des Programmcodes. Es können beliebig viele Kommentare eingefügt werden. Der Compiler liest über die Kommentare hinweg und ignoriert diese vollständig. Es muss festgelegt werden, wo ein Kommentar beginnt und wo er endet. In Java können Kommentare auf zwei Arten geschrieben werden. Einerseits gibt es **Zeilenkommentare**, welche nur eine Zeile lang sein können (also ohne Zeilenumbruch). Andererseits gibt es **Blockkommentare**, welche über mehrere Zeilen gehen können und ein einleitendes sowie ein abschliessendes Zeichen besitzen.

Schreibweise Zeilenkommentar: Im folgenden Beispiel werden die Zeilen 1 und 3 vom Compiler ignoriert, die 2. Zeile wird hingegen übersetzt.

```
// Dies ist ein Kommentar und wird vom Compiler ignoriert.
System.out.println("Zeile wird vom Compiler übersetzt.");
// Dies ist ein Kommentar und wird vom Compiler ignoriert.
```

Schreibweise Blockkommentar: Im folgenden Beispiel werden alle drei Zeilen vom Compiler ignoriert.

```
/* Dies ist ein Kommentar und wird vom Compiler ignoriert.
Diese Zeile wird vom Compiler ebenfalls ignoriert.
Diese Zeile wird vom Compiler ebenfalls ignoriert. */
```