

1. Einführung

- 1.1 Was ist Informatik?
- 1.2 Die Programmiersprache Java
- 1.3 Kommentare in Java

1. Einführung

- 1.1 Was ist Informatik?
- 1.2 Die Programmiersprache Java
- 1.3 Kommentare in Java

Franz. *informatique* (= information + mathématiques)

Engl. *computer science*, neuerdings auch *informatics*

- DUDEN Informatik:
Wissenschaft von der systematischen Verarbeitung von Informationen, besonders der automatischen Verarbeitung mit Computern.
- Gesellschaft für Informatik (GI):
Wissenschaft, Technik und Anwendung der maschinellen Verarbeitung und Übermittlung von Informationen.
- Association for Computer Machinery (ACM):
Systematic study of algorithms and data structures.

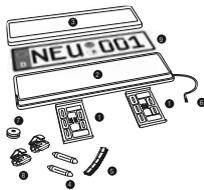
Primäres Lernziel: nicht die Komponenten eines Computers oder die Sprachen zu seiner Programmierung sondern die *Prinzipien und Techniken zur Verarbeitung von Information*.

- Theoretische Informatik:
Theoretische Durchdringung und Grundlegung von Fragen und Konzepten der Informatik (z.B. “Berechenbarkeit” von Aufgabestellungen, “Komplexität” von Aufgabenstellungen und Lösungen).
- Technische Informatik:
Beschäftigung mit der *Hardware* (z.B. maschinengerechte Darstellung von Daten und Algorithmen).
- Praktische Informatik:
Konstruktion, Darstellung und Ausführung von Algorithmen (*Software*).

- Zentraler Begriff in der Informatik.
- Abgeleitet von *Muhammad ibn Musa al-Chwarizmi*, einem bedeutenden persischen Mathematiker des 9. Jahrhunderts.
- Algorithmus = Handlungsvorschrift, um ausgehend von bestimmten Vorbedingungen ein bestimmtes Ziel zu erreichen.

Montageanleitung

Einzelteile



- Leuchtenträger
- Flächenleuchte
- Abdeckrahmen
- Lastverleiste
- Kabelmasse zur Vibrationsdämpfung
- Kabel
- Abblendschirmung
- Abzugsklemmen
- Nummernschild (nicht in Packung enthalten)

1

Die Kennzeichentafel abdecken und die Leuchtenträger **1** an Schraubstellen der Kennzeichen befestigen. Dann können die abgedeckten unteren Schrauben wieder in die Halterungen auf Frontschürze montiert werden. Die Montage von 4 Schrauben: Die Leuchten ermöglichen eine flexible Positionierung der Nummernschilder. Durch die verstellbare Ausrichtung **2** können Sie die genaue Position der Leuchtenträger an Schrauben festlegen. Wenn Sie die Markierungen ablesen, und stellen Sie auf beiden Seiten dasselbe Maß ein. Die Leuchtenträger sollten optional möglichst nahe der Leuchte und einer Ecke der Flächenleuchte liegen.

2a

Montage der Kennzeichen in der Rückleuchte: Die Innenverkleidung der Rückleuchte so weit entfernen, dass Sie ein wenig an der Kennzeichenleuchte gelangen. Die Schrauben an der Führung auf Führung unternehmlich. Tragen Sie ein Zweifelschraube nach. Stellen Sie jetzt ein neues Maßnahme **2** ein und stellen Sie die Masse unter der entsprechenden Kennzeichenleuchte. Stellen Sie ein und stellen Sie gegen Frontal ablesen. **3** Stellen Sie die für das Bereich ein Ende hinter dem Kennzeichen, die von innen eingeklinkt ist und stellen sie vor der Flächenleuchte wieder ein.

2b

Montage der Kennzeichen in die Leuchte: Stellen Sie mit einem Maßstab (27 mm) ein Maß für die Kabellänge ein. Die untere Kennzeichenleuchte, Bereich einlegen und mit Leuchte gegen Frontal ablesen.

Hinweis: Stellen Sie für das Einstecken eine Seite hinter dem Kennzeichen, die von innen bzw. unten eingeklinkt ist und stellen sie vor der Flächenleuchte wieder ein.

3

Platzieren Sie die verstellbare Montage der Nummernschilder **1** auf der Schürzung, wenn übermäßige Abdeckung **2** ist möglich. **Abdecken:** auf der Flächenleuchte **3** einlegen. Die Kennzeichen aus 40 Grad Höhe einlesen. Stellen Sie das Schild ein und prüfen Sie, ob die Position angepasst sind. Die Flächenleuchte **1** kann um Kennzeichenleuchte die zur Montage gelöst werden. Kennzeichen Sie mit einem Schraubenzieher aus dem **4** BSW von passenden Öffnungen der Kennzeichenleuchte hoch ziehen.

4

Stellen Sie die Original-Kennzeichenleuchte außer Betrieb (angegebenes Modell S022), wenn Sie beide Leuchte mit der Kennzeichenleuchte durch die originale Kennzeichenleuchte **4** einlegen. Stellen Sie Ihre Montage, andere Leuchte angepasst werden, stellen Sie die verstellbare originale Kennzeichenleuchte durch Montage der Leuchte wieder ein.

5

Jetzt die verstellbare Kabelschirmung **5** im Kabel montieren und in den Innenschicht führen. Prüfen Sie, ob die Anordnung der Kabelschirmung für ein Kennzeichen ist.

6

Stellen Sie nach der Montage der Kabelschirmung **5** und bringen Sie diese auf die Taste des Kennzeichens ein. Die gesamte Menge können Sie je nach Bedarf wieder freibleiben und Kennzeichen.

7

Prüfen Sie die Flächenleuchte **6** auf den Leuchtenträger montiert aus. Jetzt möglich, verbleiben (je nach Vorhandensein) und **7** können ablesen.

8

Verbinden Sie mit dem verstellbaren Abzugsklemmen **8** im Kabel mit der Spannungsversorgung der ursprünglichen Kennzeichenleuchte. 1 m mit „plus“ 1 m mit „minus“ Abzug Spannungsversorgung 12 V Gleichstrom, Nennleistung 2,0 Watt.

9

Platzieren Sie beide Abzugsklemmen mit einer Zange fest zusammen. Führen Sie einen Funktionen in abgewandelter Richtung durch. **9** können allgemein identifizieren und die Flächenleuchte an Steuerung und Lichtteil einlegen. Von diesem Grad werden Sie ein Einstecken der Kabelschirmung der Fläche der Flächenleuchte nicht ohne weiteres erkennen. Anschließbar die Innenverkleidung wieder montieren.

Kochrezept für einen Afrikanischen Hühnertopf (von www.chefkoch.de)

- Zutaten:
1 Huhn (küchenfertige Poularde), 125 ml Hühnerbrühe, 8 Tomaten, 150 g Erdnussbutter, 2 Zucchini, Salz und Pfeffer, Rosmarin.
- Zubereitung:
 - Von der Poularde die Haut abziehen, Huhn zerteilen.
 - Hühnerteile abspülen und in einen breiten Topf legen.
 - Die Brühe angießen und langsam zum Kochen bringen.
 - Die Tomaten mit kochendem Wasser übergießen, enthäuten und unzerteilt zum Fleisch geben.
 - Zugedeckt bei geringer Hitze ca. 40 Minuten köcheln.
 - Nach 30 Minuten Garzeit die Erdnussbutter (Erdnüsse) einrühren.
 - Mit Salz, Pfeffer und Rosmarin abschmecken.
 - Die Zucchini putzen, Blüten und Stengelansatz entfernen.
 - Zucchini waschen und in kleine Würfel schneiden.
 - 10 Minuten vor Garzeitende in den Topf geben und alles noch einmal abschmecken.

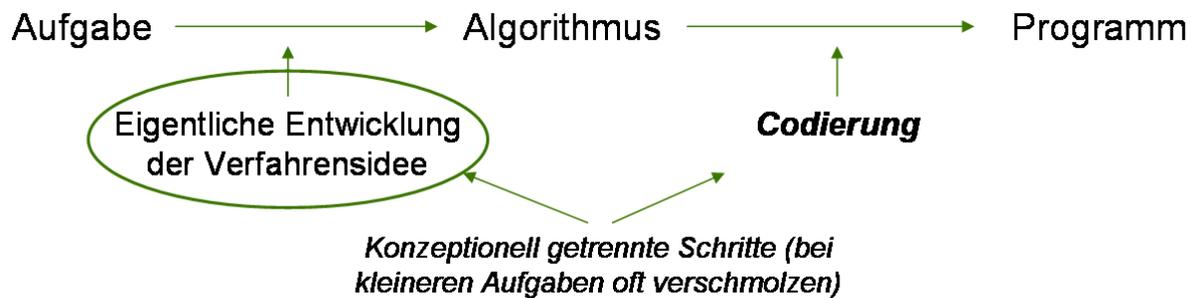
Ein *Algorithmus* ist ein Verfahren zur Verarbeitung von Daten mit einer präzisen, endlichen Beschreibung unter Verwendung effektiver, elementarer Verarbeitungsschritte.

- Daten:
Die Repräsentation und der Wertebereich sowohl der Eingabe als auch des Ergebnisses müssen eindeutig definiert sein.
- Präzise, endliche Beschreibung:
In einem endlichen Text in einer eindeutigen Sprache genau festgelegte Abfolge von Schritten.
- Effektiver Verarbeitungsschritt:
Jeder einzelne Schritt ist tatsächlich ausführbar, benötigt z.B. zu jedem Zeitpunkt der Ausführung nur endlich viel Speicherplatz.
- Elementarer Verarbeitungsschritt:
Jeder Schritt ist entweder eine Basisoperation, d.h. muss von der zugrundeliegenden "Verarbeitungseinheit" (z.B. einem Computer) ausführbar sein, oder selbst durch einen Algorithmus spezifiziert.

Zentrale Aufgabe des Informatikers: Entwicklung von Algorithmen (und oft auch deren Realisierung auf dem Rechner als Programm)

Programm: Formale Darstellung eines Algorithmus (oder mehrerer) in einer Programmiersprache

Programmiersprache: Formale (eindeutige) Sprache, stellt insbesondere elementare Verarbeitungsschritte und eindeutig definierte Datentypen für die Ein-/Ausgabe zur Verfügung



1. Einführung

1.1 Was ist Informatik?

1.2 Die Programmiersprache Java

1.3 Kommentare in Java

- objektorientiert: Klassenkonzept, strenge Typisierung
- plattformunabhängig: Übersetzung in Virtuelle Maschine (JVM)
- netzwerkfähig, nebenläufig
- Sicherheitsaspekt in der Entwicklung der Sprache wichtig

Nachteile:

Laufzeithandicap durch Interpretation (JVM), wird aber stetig verbessert

Vorteile:

- Plattformunabhängigkeit
- verteilte Anwendungen, Web-Anwendungen
- Rechnerunabhängigkeit von Graphikanwendungen

- Ein Java-Programm besteht aus Klassen und Schnittstellen
- Eine Klasse besteht aus
 - Klassenvariablen: beschreiben Eigenschaften aller Objekte dieser Klasse.
 - Attributen (fields, Instanzvariablen): beschreiben den Zustand eines Objekts.
 - statischen Methoden: Prozeduren einer Klasse, unabhängig vom Zustand eines Objekts
 - Objekt-Methoden: Operationen, die ein Objekt ausführen kann, abhängig vom Zustand des Objektes
 - Konstruktoren: Operationen zur Erzeugung von Objekten einer bestimmten Klasse

Wir betrachten nun zunächst nur die statischen Elemente. Zu den Aspekten der Objektorientierung kommen wir zu einem späteren Zeitpunkt der Vorlesung.

Keine Angst!!!

All das lernen wir später noch genauer kennen. Für den Anfang merken wir uns: ein einfaches imperatives Java-Programm besteht aus nur einer Klassendeklaration und einer Methode “main”:

```
public class KlassenName
{
    public static void main(String[] args)
    {
        // Hier geht's los mit
        // Anweisungen (elementare Verarbeitungsschritte)
        // ...
    }
}
```

Die Textdatei, die den Java-Code enthält, heißt `KlassenName.java`, also genauso wie die enthaltene Klasse, mit der Endung `java`.

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello, world!");
    }
}
```

In der Java-Programmierung gibt es einige Konventionen. Deren Einhaltung erleichtert das Lesen von Programmen. Beispiele solcher Konventionen sind:

- Klassennamen beginnen mit großen Buchstaben.
 - HelloWorld
- Methodennamen, Attributnamen und Variablennamen beginnen mit kleinen Buchstaben.
 - Methoden: `main`, `println`,
 - Klassenvariable: `out`,
 - Variable: `args` (weder Instanz- noch Klassenvariable, sondern Parametervariable)
- Zusammengesetzte Namen werden zusammengesrieben, jeder innere Teilname beginnt mit einem großen Buchstaben.
 - Klasse HelloWorld

Warum Konventionen?

- In der “Lebenszeit” eines Programmes werden typischerweise 80% der Kosten von Wartung verursacht.
- Selten wird ein Software-Produkt dauerhaft vom ursprünglichen Autor gewartet.
- Code-Konventionen erleichtern die Lesbarkeit von Programmen und ermöglichen Programmierern, fremden (und eigenen!) Code schneller und besser zu verstehen.

Die von Sun empfohlenen Konventionen für Java finden Sie unter:

<http://java.sun.com/docs/codeconv/>

Momentan ist dort vieles vielleicht noch unverständlich, aber schlagen Sie im Lauf des Semesters immer mal wieder dort nach!

- Ein Programm einer höheren Programmiersprache ist für Menschen lesbar, folgt aber auch einer festen Syntax (Grammatik), so dass *elementare Verarbeitungsschritte* klar definiert sind.
- Ein *Übersetzer (Compiler)* erzeugt aus der Programmdatei *Maschinencode*, der von Rechnern ausgeführt werden kann.
 - Der Compiler kennt die Syntax der Programmiersprache und bemerkt *syntaktische* Fehler (Verstöße gegen die Grammatik).
 - Nur, wenn das Programm syntaktisch korrekt ist, wird auch Maschinencode erzeugt. Andernfalls reagiert der Compiler mit Fehlermeldungen.
 - Achtung: Der Compiler erkennt nur *syntaktische*, nicht aber *semantische* Fehler, d.h. Fehler durch die das Programm eine andere *Bedeutung* annimmt als vom Programmierer beabsichtigt.
- Bei vielen Sprachen (z.B. C/C++) erzeugt der *Compiler* plattformabhängigen Maschinencode (kann nur auf bestimmten Rechnerarchitekturen/Betriebssystemen ausgeführt werden).

- Sogenannte Skript-Sprachen (z.B. Perl, PHP, SML) werden nicht kompiliert, sondern von einem plattformspezifischen *Interpreter* interpretiert, wobei auch eine Syntaxprüfung durchgeführt werden muss.
- Vorteil: Die Programme sind plattformunabhängig.
- Nachteil: Der Sourcecode bleibt unübersetzt und wird bei jeder Ausführung des Programmes von neuem interpretiert.

- Plattformunabhängigkeit eines Java-Programmes wird durch einen Kompromiß erreicht:
 - Der Sourcecode wird durch einen Compiler übersetzt in Bytecode, der plattformunabhängig verwendet werden kann.
 - Der Bytecode ist also bereits syntaktisch überprüft, kann aber nicht direkt von einem Computer verarbeitet werden.
 - Bytecode wird von einer *virtuellen Maschine* ausgeführt (interpretiert).
 - Diese Interpretation muss aber keine syntaktische Prüfung mehr vornehmen.
 - Die virtuelle Maschine gibt es in verschiedenen Versionen für verschiedene Plattformen (JVM = Java Virtual Machine, Teil des JRE = Java Runtime Environment).

- Aus einer Textdatei `KlassenName.java` erzeugt der Java Compiler `javac` eine Binärdatei `KlassenName.class`.
Beispiel:

```
javac HelloWorld.java
```

erzeugt die Binärdatei `HelloWorld.class`.
- Die Binärdatei `KlassenName.class` enthält den Bytecode für die JVM.
- Der Compiler `javac` ist Teil des JDK (= Java Development Kit). Das JDK enthält JRE, Sie benötigen also das JDK für die Übungen zu dieser Vorlesung.

Die Binärdatei `KlassenName.class` wird der JVM übergeben und von dieser ausgeführt (interpretiert). Durch den Aufruf `java KlassenName` wird die `main`-Methode der Klasse `KlassenName` aufgerufen.

Beispiel:

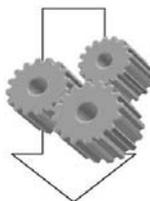
```
java HelloWorld
```

gibt "Hello, World!" auf dem Bildschirm aus.

Programmierer:

```

HelloWorld.java
public class HelloWorld
{
    ...
}
    
```



Compiler:

```
javac HelloWorld.java
```

```

HelloWorld.class
0100011101010010001001
0010100010000010000010
1110010111101010111010
1111010111101101101011
    
```

Anwender:



JVM Windows



JVM Linux



JVM Mac

Aufruf und Ausgabe

```
java HelloWorld
Hello World!
```

1. Einführung

- 1.1 Was ist Informatik?
- 1.2 Die Programmiersprache Java
- 1.3 Kommentare in Java

Prinzip: Programmcode und Kommentar gehören zusammen

“The view that documentation is something that is added to a program after it has been commissioned seems to be wrong in principle, and counterproductive in practice.

Instead, documentation must be regarded as an integral part of the process of design and coding.”

C. A. R. Hoare (Turing-Preisträger):
Hints on Programming Language Design,
1973



C. A. R. Hoare, *1934
Erfinder von Quicksort, Hoare Logik, Strukt. Programmierung,
CSP, Occam
Turing-Preis 1980

Es gibt in Java 3 Arten von Kommentaren:

- **Einzeilige Kommentare** beginnen mit `//` und enden am Ende der aktuellen Zeile
- **Mehrzeilige Kommentare** beginnen mit `/*` und enden mit `*/`
- **Dokumentationskommentare** beginnen mit `/**` und enden mit `*/` und können sich ebenfalls über mehrere Zeilen erstrecken.

Kommentare derselben Art sind nicht schachtelbar. Ein Java-Compiler akzeptiert aber einen einzeiligen innerhalb eines mehrzeiligen Kommentars.

Dokumentationskommentare dienen dazu, Programme im Quelltext zu dokumentieren. Sie werden in den mit dem Befehl `javadoc` erzeugten Report mit aufgenommen.

Beispiel: Die Klasse HelloWorld dokumentiert

```
/**
 * HelloWorld Klasse um eine einfache Benutzung einer java-Klasse zu illustrieren.
 *
 * Diese Klasse dient nur dem Anzeigen des Strings
 * "Hello, world!" auf dem Bildschirm
 *
 * @author Arthur Zimek
 */
public class HelloWorld
{
    /**
     * Die main-Methode wird automatisch aufgerufen, wenn die Klasse mit
     * java HelloWorld
     * aufgerufen wird.
     *
     * Die Methode main druckt "Hello, world!" auf die Standard-Ausgabe.
     *
     * @param args Array mit Parametern - wird von dieser Methode nicht verwendet.
     */
    public static void main(String[] args)
    {
        // Ausgabe von "Hello World!" auf die Standard-Ausgabe
        System.out.println("Hello World!");
    }
}
```

Mit dem Befehl

```
javadoc HelloWorld.java
```

wird automatisch eine Beschreibung der Klasse `HelloWorld` erzeugt und in die Datei

```
HelloWorld.html
```

geschrieben.

Die Klassenbeschreibung wird eingebettet in eine organisierte Menge von html-Dokumenten:

http://www.dbs.ifi.lmu.de/Lehre/EIP/WS_2008/skript/programmbeispiele/einfuehrung/helloWorld/index.html

Diese Dokumentation kann auch für viele Klassen gleichzeitig erfolgen (`javadoc *.java`).

Die durch `@` eingeleiteten Elemente in einem Dokumentationskommentar haben eine besondere Bedeutung, z.B.:

- `@see` für Verweise
- `@author` für Name des Autors / Namen der Autoren
- `@version` für die Version
- `@param` für die Methodenparameter
- `@return` für die Beschreibung des Ergebnisses einer Methode

Auch die Bibliothek der Standard Edition ist mit javadoc erzeugt:

<http://java.sun.com/javase/6/docs/api/>

Für das fortgeschrittene Programmieren mit Java ist diese API Doc ein sehr wichtiges Hilfsmittel.