



Einführung in die Programmierung mit Java

Martin Wirsing

in Zusammenarbeit mit
Moritz Hammer und Axel Rauschmayer

SS 06

Informatik II, SS 06



Ziele

- Begriffsbildungen: Informatik, Algorithmus, Programm ...
- Warum Java als Programmiersprache verwenden?
- Ein einfaches Java-Programm erstellen, übersetzen und ausführen
- Gut dokumentierte Java-Programme erstellen können



Informatik

Informatik

ist ein Kunstwort,
das in den 60-ziger Jahren in Frankreich kreiert wurde,

entstanden aus
Information + Mathematik

englisch: **Computer Science**
neuerdings auch: **Informatics**

bedeutet

Wissenschaft der maschinengestützten Informationsverarbeitung



Teilgebiete der Informatik

Praktische Informatik

- Programmierung und Software-Entwicklung
- Datenbanksysteme
- Betriebssysteme und Middleware

Technische Informatik

- Rechenanlagen und Rechnernetze (Hardware)

Theoretische Informatik

- Formale Modelle und Semantik
- Algorithmen und Komplexität

Informatik und Gesellschaft

- Auswirkungen der Informatik auf die Gesellschaft
(Rationalisierung, Automatisierung, Datensicherheit, ...)



Algorithmen und Programme

Algorithmus (nach Al-Khwarizmi, um 800)

- Allgemeines Verfahren zur Lösung einer Klasse von Problemen, das durch eine eindeutige Vorschrift so genau festgelegt ist, dass man es anwenden kann, ohne es verstanden zu haben.
- Eigenschaften:
 - Jeder Einzelschritt ist eindeutig festgelegt und berechenbar.
 - Das Verfahren liefert nach endlich vielen Schritten eine Lösung.
- Beispiele:
 - Modellbau: Montageanleitung
 - Informatik: Sortieralgorithmus



Al'Khwarizmi
790-840
Author von
Hisab al-jabr
w'al-muqabala

Programm

- Beschreibung von Datenstrukturen und Algorithmen in einer Programmiersprache



Programme und Software-Entwicklung

Programm

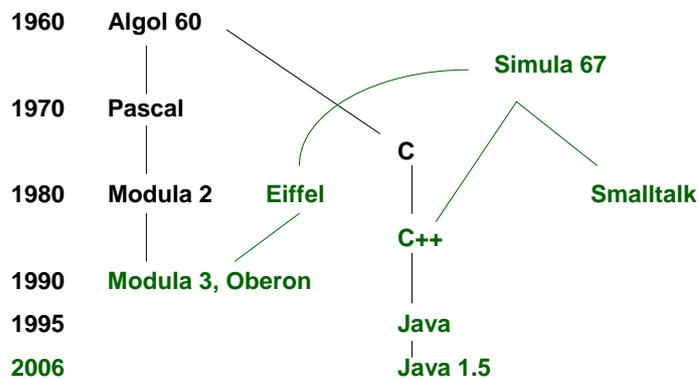
- Beschreibung von Datenstrukturen und Algorithmen in einer Programmiersprache

Software-Entwicklung

- Systematische Konstruktion von Programmen zur Lösung eines in der realen Welt gestellten Problems



Entwicklung objektorientierter Programmiersprachen



Algol 60

- Entwickelt von John Backus, F.L. Bauer, John McCarthy, Peter Naur, Alan J. Perlis, Heinz Rutishauser, Klaus Samelson.
- Imperative Sprache mit Blockkonzept, Call-by-Value und Call-by-Name, Syntaxdefinition in Backus-Naur-Form



F.L. Bauer
*1924,
Diss 1952 LMU
Entwickler von
Algol 60,
Patent auf
Kellerprinzip,
„Vater“ der dtsh.
Informatik



Pascal

- Entwickelt von Niklaus Wirth
- Imperative Sprache, Weiterentwicklung von Algol 60/68 Verbunddatentypen (Record)

N. Wirth, *1931, PhD 1963 Berkeley
Entwickler von Pascal, Modula, Oberon,
Turing-Preis 1984



Simula 67

- Entwickelt von Kristen Nygaard und Johan-Ole Dahl.
- Erste OO-Sprache, Erweiterung von Algol 60 mit Klassen-Konzept, Vererbung (damals genannt „Prefixing“)



Smalltalk 80

- Entwickelt von Alan Kay und Adele Goldberg
- 1. Dynamische OO-Sprache, mit Betriebssystem und Entwicklungsumgebung

A. Kay, *1940, PhD 1969 Utah
Entwickler von Smalltalk, Dynabook (1. PC),
Turing-Preis 2003

J.-O. Dahl
1931 – 2002
Entwickler von
Simula 67
1. Informatikprof.
in Norwegen (1968)
Turing-Preis 2001



Java

- Entwickelt von J. Gosling, Bill Joy, P. Naughton, u.a.
- Erste plattform-unabhängige OO-Sprache, insbesondere zur Programmierung von Internet-Applikationen
- Heute auch eingesetzt im Multimedia-Bereich und (beginnend) für Eingebettete Systeme
- Erste Version 1.0 1995, heute Java 1.5.
- Ursprünglicher Name: OAK.



C++

- Entwickelt von Bjarne Stroustrup
- Effiziente OO-Sprache, Erweiterung von C

B. Stroustrup, *1950 in Arhus (DK), PhD 1979 Cambridge
Entwickler von C++

James Gosling, *1956
PhD 1983 CMU
Entwickler von
Emacs (Texteditor),
Java



Aspekte von Java

- Objektorientiert: Klassenkonzept, strenge Typisierung
- Unabhängig von Plattform: Durch Übersetzung in Virtuelle Maschine (JVM)
- Netzwerkfähig, nebenläufig
- Sicherheitskonzept

- **Nachteile:** Laufzeithandicap durch Interpretation der JVM
(aber z.T. ausgeglichen durch Just-in-Time Übersetzung)
- **Vorteile:**
 - Verteilte Anwendungen, Web-Anwendungen
 - Rechnerunabhängigkeit von Graphikanwendungen



Grober Aufbau eines Java-Programms

- Ein **Java-Programm** besteht aus
 - einer Menge von **Klassen**
 - (und **Schnittstellen**, siehe später)
- Eine **Klasse** besteht aus
 - **Attributen** („fields“): Beschreiben den **Zustand eines Objekts**
 - **Methoden**: Beschreiben die **Operationen**, die ein Objekt ausführen kann
 - **Konstruktoren**: Operationen zur Erzeugung von Objekten



Einfaches imperatives Java-Programm

- Ein **imperatives** Java-Programm besteht aus
 - **Klassendeklaration** mit einer
 - **einzigsten Methode** namens **“main”**:

```
public class <KlassenName>
{
    public static void main(String[] args)
    {
        <Anweisungen>
    }
}
```



Beispiel: [Hallo](#)

```
public class Hallo
{
    public static void main(String[] args)
    {
        System.out.println(„Hallo!“);
    }
}
```



Methodenaufruf

- **Methodenaufruf allgemein:**

```
object.methodName(actual parameters);
```

- **Beispiel:**

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



Konventionen

- Klassennamen beginnen mit großen Buchstaben

Bsp. Hallo

- Methodennamen, Attributnamen und Variablennamen beginnen mit kleinen Buchstaben

Bsp. println, out

- Konstantennamen bestehen NUR aus großen Buchstaben.

Bsp. BLUE

- Zusammengesetzte Namen werden zusammengeschrieben, jeder (innere) Teilname beginnt mit einem großen Buchstaben

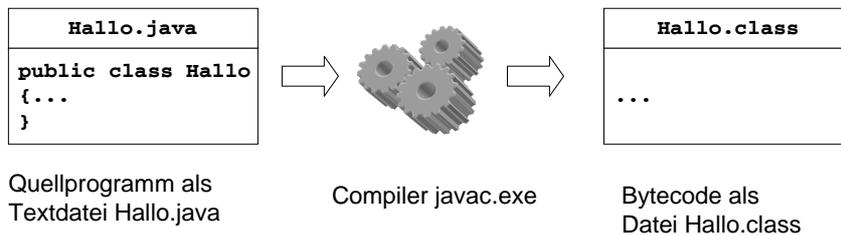
Bsp. Klasse HalloWelt, Methoden getName, getMyObject



Übersetzung und Ausführung von Java-Programmen

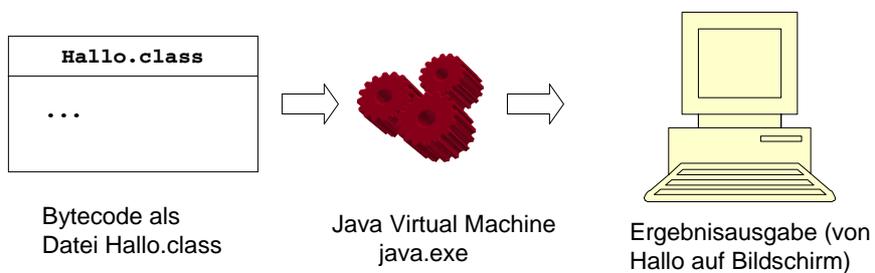
Übersetzung in Bytecode

- Aus einer Textdatei mit Endung „.java“ erzeugt der Compiler javac eine Datei mit gleichem Namen, aber Endung „.class“
- Diese enthält den Bytecode für die JVM



Übersetzung und Ausführung von Java-Programmen

- Die Datei mit dem Bytecode wird der JVM übergeben und von der JVM ausgeführt (d.h. interpretiert).





Übersetzung und Ausführung von Hallo.java (unter Windows)

Übersetzung von Hallo.java:

```
C: > javac Hallo.java
```

Interpretation von Hallo.class:

```
C: > java Hallo
```

Gibt auf Bildschirm zurück:

```
Hallo!
```



Zusammenfassung

▪ Geschichte:

- Objektorientierte Programmiersprachen seit 1967: Simula
- OO-Programmierung populär seit Ende der 80er Jahre mit Smalltalk und C++.
- Heute vor allem Java, C++ und C# .

▪ Java

- OO-Programmiersprache,
- vor allem zur Programmierung im Internet eingesetzt
- Java ist plattform-unabhängig, interpretierend, unterstützt Sicherheitskonzepte und besitzt eine reichhaltige Klassenbibliothek (API, engl. "Application Programming Interface").



Zusammenfassung (2)

- Ein **Java-Programm** besteht aus einer oder mehreren **Klassen** (und **Schnittstellen**).
 - Klassen enthalten **Attribute** und die Deklarationen von **Methoden** und **Konstruktoren**.
 - Eine Methode besteht aus einer **Sequenz von Anweisungen**, die den Berechnungsablauf festlegen.
- Jede **selbstlaufende** Java-Anwendung enthält eine Methode „**main**“.
- Ein Java-Programm wird mit einem **Übersetzer in Byte-Code** übersetzt, der dann mit einem **Interpreter**, der **JVM**, ausgeführt wird.