

Informatik IIb

Objektorientierte Programmierung mit Java

Oliver Jack

Fachhochschule Jena
Fachbereich Elektrotechnik und Informationstechnik

Wintersemester 2010/11

Vorlesung 1. Einführung

Allgemeines und Organisatorisches

Modulbeschreibung

Termine

Modulprüfung

Literatur

Java im Überblick

Einführung

Erste Schritte

Java Applets

Zusammenfassung

Identifikation

Studiengang

ET/AT, KMT, TI

Modulname

Informatik 2

Teilmodul

Informatik IIb

Modulnummer

ET.1.306

Pflichtmodul

Inhalt

Weiterführung der objektorientierten Programmierung:

- ▶ Grafikprogrammierung
- ▶ Benutzeroberfläche
- ▶ Netzwerkprogrammierung
- ▶ Applets
- ▶ Vererbung und Klassenhierarchie
- ▶ Streams und Dateien
- ▶ Ausnahmen und Fehlersuche

Qualifikation, Voraussetzungen

Qualifikationsziele

- ▶ Ausbildung von Fähigkeiten und Fertigkeiten für einen systematischen Entwurf von Benutzeroberflächen
- ▶ Vermittlung von Arbeitsmethoden für die systematische Entwicklung von Netzwerkanwendungen

Prüfung, Arbeitsaufwand

Leistungspunkte (ECTS credits)

3

Voraussetzungen für die Vergabe von Leistungspunkten

Alternative Prüfungsleistung

Arbeitsaufwand (work load): 90 h gesamt

- ▶ Präsenzstunden (SWS): 45
- ▶ Selbststudium (h): 45

Verwendbarkeit des Moduls

- ▶ Informationstechnik
- ▶ Rechnerarchitektur
- ▶ Softwaretechnologie
- ▶ Echtzeitbetriebssysteme

Vorlesung

- ▶ In den Semesterwochen 1–7
- ▶ Zeit/Raum:
 - ▶ Dienstags 15:15–16:45 Uhr / 05.03.43 (HS 4)
 - ▶ Mittwochs 07:45–09:15 Uhr / 01.02.14
- ▶ Vorstellung grundlegender Konzepte und theoretischer Hintergründe
- ▶ Termin: <http://stundenplanung.fh-jena.de>

Praktikum

- ▶ ab Semesterwoche 6
- ▶ Raum: 01.01.11
- ▶ Dozenten: Herr Berger, Herr Fiedler
- ▶ Aufteilung nach Sets und Termin: <http://stundenplanung.fh-jena.de>
- ▶ Anwendung des Vorlesungsstoffs und handwerkliche Übung
- ▶ Bearbeitung einfacher Aufgaben
- ▶ Bearbeitung eines Software-Entwicklungsprojekts

Prüfungsleistung

- ▶ Alternative Prüfungsleistung, **bitte melden Sie sich im Prüfungsamt für die Prüfung an**
- ▶ Bearbeitung eines Software-Entwicklungsprojekts
- ▶ in Gruppen gemäß Set-Einteilung
- ▶ Projektthemen werden in der dritten Vorlesung vorgestellt
- ▶ Abgabe einer vollständigen Projektdokumentation inklusive der Qualitätssicherung im Entwicklungsprozess
- ▶ Präsentation und Demonstration an den letzten Praktikumsterminen
- ▶ Abgabe der Dokumentation und der entwickelten Software spätestens zum Präsentationstermin

Literatur

 Helmut Balzert.
Objektorientierte Programmierung mit Java 5.
Elsevier, München, 1. edition, 2005.

 Douglas Bell and Mike Parr.
Java für Studenten - Grundlagen der Programmierung.
Pearson Studium, München, 3. überarb. edition, 2003.

 Ian F. Darwin.
Java Kochbuch - Beispiele und Lösungen für Java-Programmierer.
O'Reilly, Beijing; Cambridge; Farnham; Köln; Paris; Sebastopol;
Taipei; Tokyo, 2. edition, 2005.

Literatur



Guido Krüger.

Handbuch der Java der Programmierung.

Addison-Wesley, München, Boston et.al, 4. edition, 2003.



Bernhard Steppan.

Einstieg in Java - Aktuell zum JDK 5.

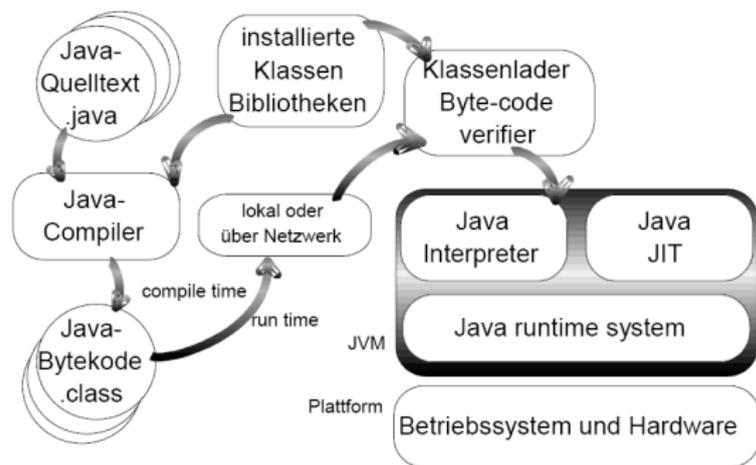
Galileo Press, Bonn, 2. edition, 2005.

Java als Programmiersprache



- ▶ Objekt-orientierte Programmiersprache
- ▶ Entwickelt der Firma *Sun Microsystems*, die 2009 von der Firma *Oracle* übernommen wurde
- ▶ 1995 eingeführt
- ▶ Entwickelt von James Gosling, Bill Joy, Patrick Naughton und Mike Sheridan
- ▶ Eine Komponente der Java-Technologie (die zweite Komponente ist die Java-Plattform)

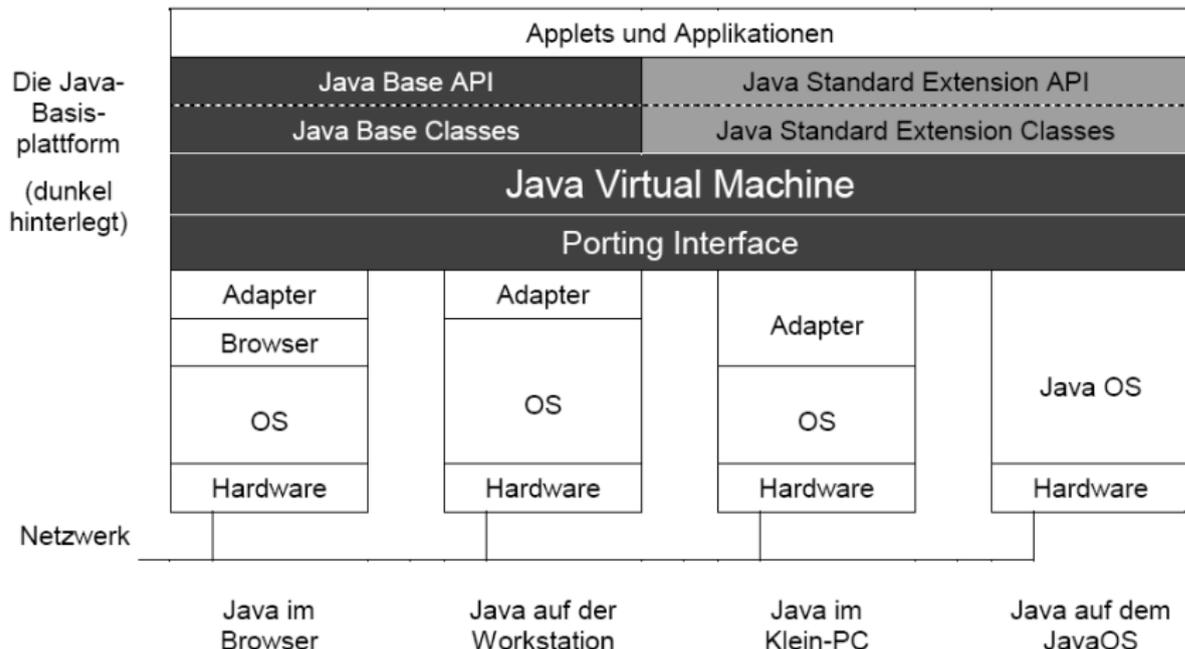
Das Ausführungsmodell von Java



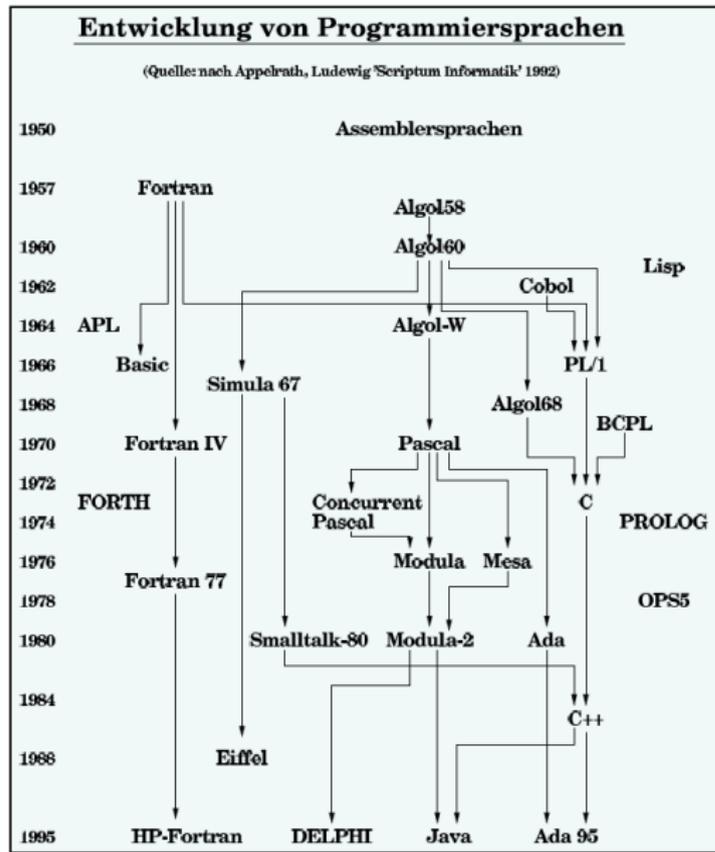
Quelle: A. Weinert. Java für Ingenieure. Fachbuchverlag Leipzig. 2002

- ▶ Maschineneunabhängige Ausführung in einer speziellen Plattform (Java-Plattform)
- ▶ Übersetzung der Quellprogramme in Bytecode
- ▶ Wichtigster Bestandteil der Java-Plattform ist die Java Virtual Machine (JVM), die Bytecode ausführt.

Die Java-Plattform



Einordnung in Programmiersprachen



Sprachmerkmale von Java

Konzepte

- ▶ Objektorientiert und imperativ
- ▶ Syntax angelehnt an C und C++
- ▶ Integration von
 - ▶ Mehrfädigkeit (Multithreading)
 - ▶ strukturierter Ausnahmebehandlung (Exception handling)
 - ▶ grafischen Fähigkeiten

Wichtige Unterschiede zu C++

- ▶ keine expliziten Zeiger (Pointer)
- ▶ keine separaten Header-Dateien
- ▶ keine Mehrfachvererbung
- ▶ keine Templates

Sprachmerkmale im Detail

Datentypen

- ▶ Elementare Datentypen wie in C (die meisten)
- ▶ Felder (Arrays) und Zeichenketten (Strings) sind als Objekte implementiert (Compiler und Laufzeitsystem)
- ▶ Keine methodenlosen Strukturtypen wie in C (struct, union)
- ▶ Elementare Datentypen vorzeichenbehaftet und in der Größe exakt spezifiziert
- ▶ Datentyp boolean für Wahrheitswerte

Sprachmerkmale im Detail

Felder

- ▶ Semidynamische Arrays
- ▶ Initiale Größe zur Laufzeit festgelegt
- ▶ Arrays werden als Objekte angesehen
- ▶ Mehrdimensionale Arrays wie in C/C++ durch Ineinanderschachtelung realisiert

Benötigte Software

JDK 6 Update 21, frei

- ▶ Windows Offline Installation (ca. 76 MB)
- ▶ Download:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
ehemals <http://java.sun.com/javase/downloads/index.jsp>
- ▶ Minimal, ohne Entwicklungsumgebung, Kompilieren und Ausführen über Kommandozeile, Quelltext muss mit irgend einem separaten (Text-)Editor erstellt werden.
- ▶ (MS Windows) Standard-Installation
- ▶ (MS Windows) Nach der Installation ggf. Umgebungsvariable um `<JDK-Installationspfad> \ bin` ergänzen.

Benötigte Software

Eclipse, frei

- ▶ <http://www.eclipse.org/downloads/>
- ▶ Eclipse (aktuelle Version: Helios, mit Werkzeugen zu modellbasierter Entwicklung)
- ▶ Sehr komfortable Entwicklungsumgebung, Quasi-Standard für Java-Entwicklung
- ▶ Zusätzlich zu Java Plattform JDK
- ▶ ZIP-Datei, einfach auspacken, im Verzeichnis eclipse die Datei eclipse.exe ausführen
- ▶ Beim ersten Start gibt es Hinweise und Anleitungen zur Konfiguration und Tutorials

Das erste Java-Programm

```
1 public class HelloWorld {  
2     // A program to display the message  
3     // "Hello World!" on standard output  
4  
5     public static void main(String[ ] args) {  
6         System.out.println("Hello World!");  
7     }  
8 } // end of class HelloWorld
```

Das erste Java-Programm

Bedeutung der Programmzeilen

- ▶ 1 Definition einer Klasse mit Namen HelloWorld.
- ▶ 2, 3 Kommentare
- ▶ 5-7 Definition der Funktion main, das ist eine besondere Funktion, die vom Java-Interpreter als erstes aufgerufen wird.
- ▶ 6 Funktionsaufruf `System.out.println(...)`.
 - ▶ Funktion aus der Java System library wird aufgerufen.
 - ▶ Ausgabe des übergebenen Argumentes „Hello World!“ auf Standard-Ausgabe, anschließend Zeilenvorschub.
- ▶ 1, 5, 7, 8 Geschweifte Klammern `{ }` schließen Block von Code und Deklarationen ein.

Struktur einer Java-Klasse

Class

gespeichert in File mit gleichem Namen und .java Endung

Klassen Variablen

'main' Methode

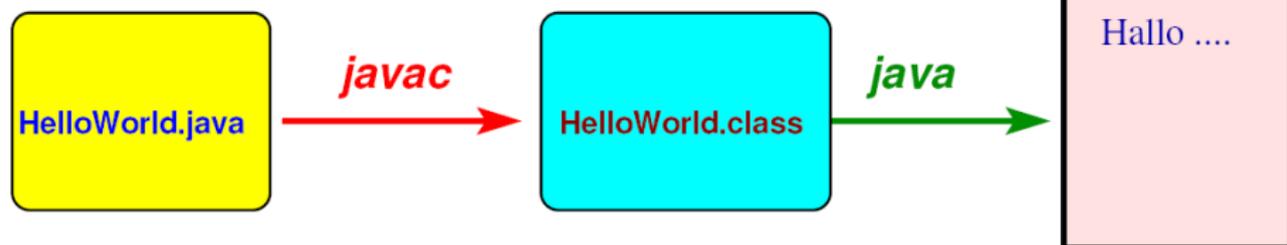
Weitere Methoden

Java-Quellcode

Java-Konvention

- ▶ **Dateiname** des Quellprogramms hat Endung **.java**
- ▶ **Basisname** des Files ist der **Klassenname** (HelloWorld)

Kompilieren und Ausführen

*Source-file**Byte-compiler**Byte-code**Interpreter*

Kompilieren

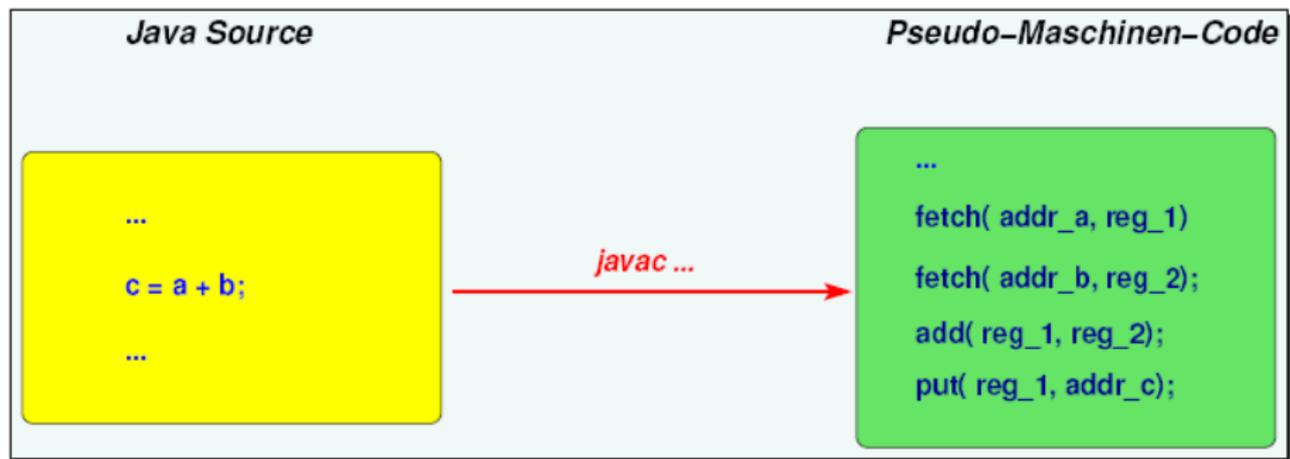
```
javac HelloWorld.java  
erzeugt byte-code HelloWorld.class
```

Ausführen

```
java HelloWorld
```

Byte-code (.class-Datei) überall lauffähig (Linux, Unix, Windows, ...)
durch Java Interpreter (Java VM)

Compiler erzeugt Pseudo-Maschinen-Code

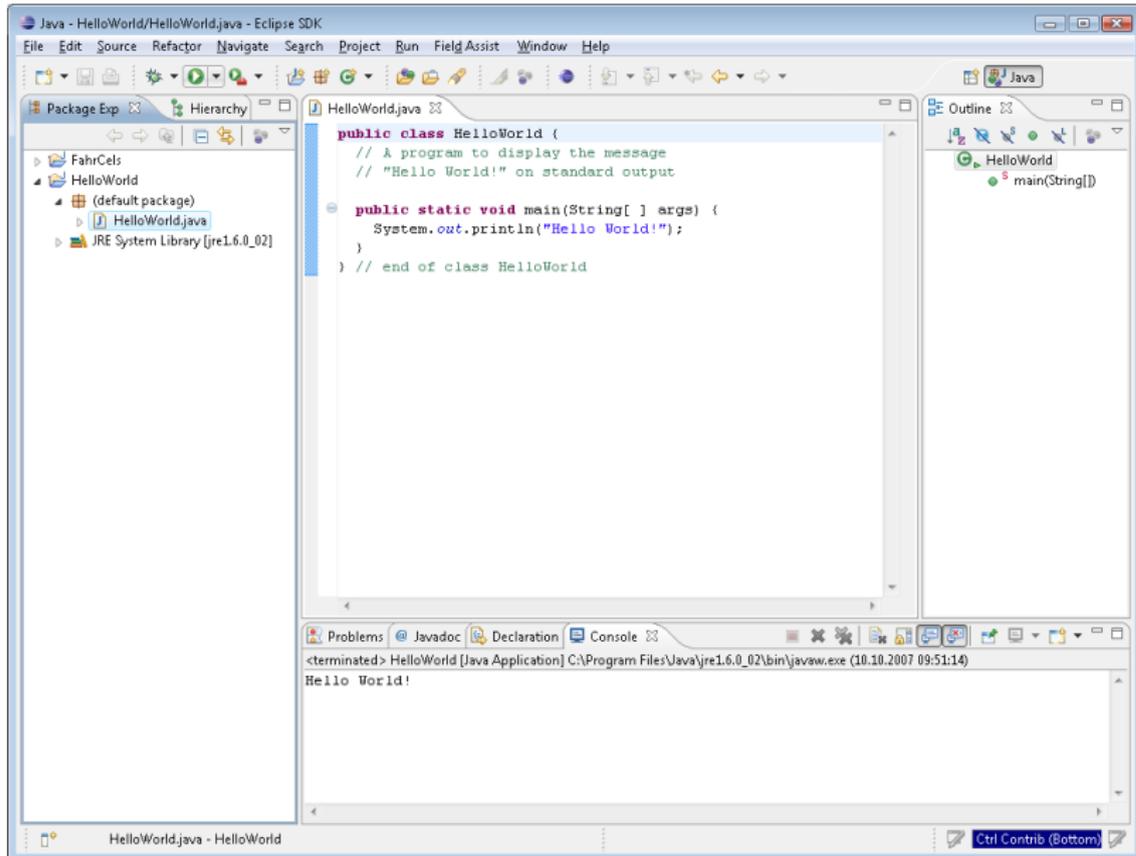


Kompilieren „Zu Fuß“

Administrator: Eingabeaufforderung

```
c:\Users\Oliver Jack>javac HelloWorld.java  
c:\Users\Oliver Jack>java HelloWorld  
Hello World!  
c:\Users\Oliver Jack>
```

Java Programmentwicklung mit eclipse



Ein weiteres Programm: FahrCels.java

```
1 // Drucke Fahrenheit->Celsius Tabelle
2 public class FahrCels {
3     public static void main(String[ ] args) {
4         double lower = 0., upper = 300., step = 20.;
5         double fahr, celsius;
6         fahr = lower;
7         while ( fahr < upper ) {
8             celsius = (5./9.) * (fahr - 32.);
9             System.out.println("Fahrenheit_" + fahr +
10                "_=" + celsius + "_Celsius");
11             fahr += step;
12         } // end-while
13     }
14 } // end of class FahrCels
```

Kompilieren und Ausführen FahrCels

C:\ Eingabeaufforderung

```
C:\Users\Oliver Jack\workspace\FahrCels\src>javac FahrCels.java
```

```
C:\Users\Oliver Jack\workspace\FahrCels\src>java FahrCels
```

```
Fahrenheit 0.0 = -17.77777777777778 Celsius  
Fahrenheit 20.0 = -6.666666666666667 Celsius  
Fahrenheit 40.0 = 4.444444444444445 Celsius  
Fahrenheit 60.0 = 15.555555555555557 Celsius  
Fahrenheit 80.0 = 26.666666666666668 Celsius  
Fahrenheit 100.0 = 37.77777777777778 Celsius  
Fahrenheit 120.0 = 48.88888888888889 Celsius  
Fahrenheit 140.0 = 60.0 Celsius  
Fahrenheit 160.0 = 71.11111111111111 Celsius  
Fahrenheit 180.0 = 82.22222222222223 Celsius  
Fahrenheit 200.0 = 93.33333333333334 Celsius  
Fahrenheit 220.0 = 104.44444444444444 Celsius  
Fahrenheit 240.0 = 115.55555555555556 Celsius  
Fahrenheit 260.0 = 126.66666666666667 Celsius  
Fahrenheit 280.0 = 137.77777777777777 Celsius
```

```
C:\Users\Oliver Jack\workspace\FahrCels\src>_
```

Ausführung im Web-Browser

Wichtiger Grund für die Popularität von Java ist die Möglichkeit interaktive Anwendungen zu schreiben, die über das Internet mit entsprechenden Browsern gelesen und ausgeführt werden können.

HelloWorldApplet (HelloWorldApplet.java)

```
1 import java.applet.Applet;
2 import java.awt.*;
3 public class HelloWorldApplet extends Applet {
4     public void paint(Graphics g) {
5         g.drawString("Hello□world!", 100, 50);
6     }
7 }
```

Bedeutung der Programmzeilen

- ▶ 1, 2 Java Applet Klasse und awt package wird verwendet und muss vorher angegeben werden.
- ▶ 3 Definition der Klasse HelloWorldApplet, die von der Klasse Applet abgeleitet ist bzw. diese erweitert.
- ▶ 4 Eine Methode paint wird definiert, die als Argument ein Objekt der Klasse Graphics erwartet. paint ist in Applet schon definiert und wird hier überschrieben.
- ▶ 5 Die Methode drawString() der Graphics Klasse wird gerufen.

Applet im Browser

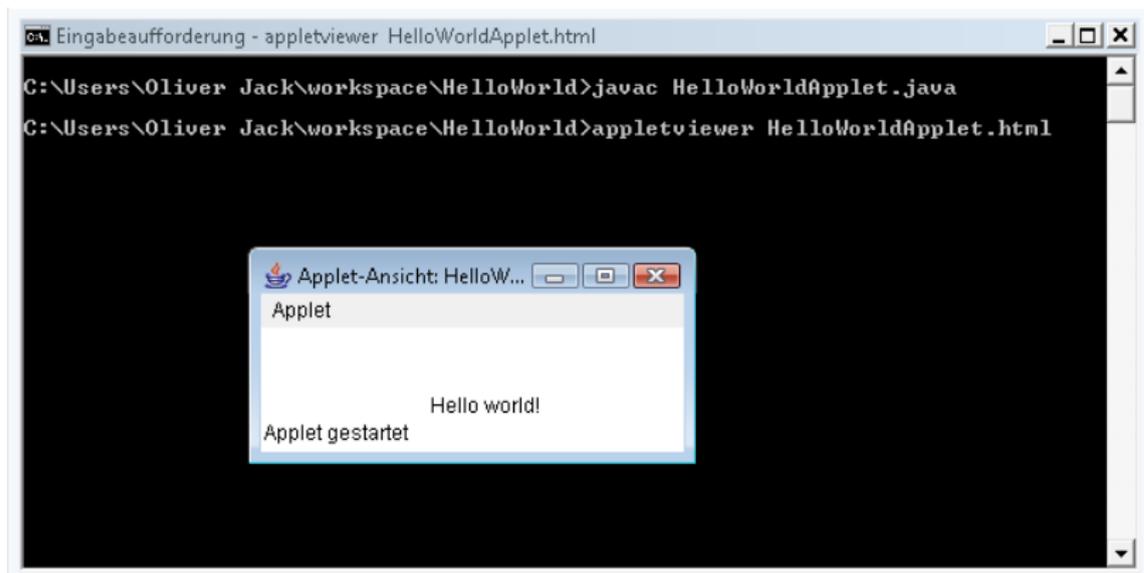
Einbettung in HTML-Datei (HelloWorldApplet.html)

```
<HTML>
<HEAD>
<TITLE> Hello World </TITLE>
</HEAD>
<BODY>
<h1>Mein erstes Applet:</h1>
<applet code="HelloWorldApplet.class"
  width="250" height="50">
</applet>
</BODY>
</HTML>
```

Ausführung

- ▶ Applet kompilieren
`javac HelloWorldApplet.java`
- ▶ Zum Ausführen des Applets entweder im Web-Browser die .html-Datei öffnen oder mit dem Java appletviewer:
`appletviewer HelloWorldApplet.html`

Ausführung



The screenshot shows a command prompt window titled "Eingabeaufforderung - appletviewer HelloWorldApplet.html". The command prompt displays the following commands and their outputs:

```
C:\Users\Oliver Jack\workspace\HelloWorld>javac HelloWorldApplet.java
C:\Users\Oliver Jack\workspace\HelloWorld>appletviewer HelloWorldApplet.html
```

An "Applet-Ansicht: HelloW..." window is overlaid on the command prompt. It displays the text "Applet" at the top, "Hello world!" in the center, and "Applet gestartet" at the bottom.

Eigenschaften von Java

- ▶ Java ist objekt-orientierte, plattform-unabhängige Programmiersprache. Entwickelt in den 1990er Jahren, stark verbreitet für Internet Anwendungen.
- ▶ Grundlegende Java-Syntax ähnlich wie C++. Aber konzeptionell und im Detail viele Unterschiede.
- ▶ Java erzwingt objekt-orientiertes Programmieren, d.h. Arbeiten mit Klassen und Objekten, Vererbung und Polymorphismus.

Eigenschaften von Java

- ▶ Java steht zwischen Compiler-Sprachen, wie z.B. C/C++, Fortran, Cobol und Interpreter- bzw. Skript-Sprachen wie Basic oder Perl, Shells.
- ▶ Java wird von manchen immer noch als Sprache für kleinere Web oder Grafikanwendungen angesehen, weniger geeignet für ernsthafte wissenschaftliche und technische Anwendungen. Falsch: Java bietet für technische Anwendungen alle Eigenschaften, die Fortran, C/C++, Pascal haben, und noch einiges darüber hinaus.

Vergleich Java und C++

Vorteile von Java

- ▶ Wohldefinierter, überschaubarer Sprachumfang
- ▶ Vielzahl von Hilfspaketen zu I/O, Netzwerkprogrammierung, Graphik, Datenbanken in der Standarddistribution integriert.
- ▶ Automatische Speicherverwaltung
- ▶ Keine Pointer
- ▶ Plattform-unabhängig
- ▶ Gut und schnell erlernbar, hohe Programmiereffizienz

Vergleich Java und C++

Nachteile von Java

- ▶ Eingeschränkter Sprachumfang (pointer, operator overloading, templates)
- ▶ Performanz-Nachteile (meist nur wenig)
- ▶ Hardwarenahe Programmierung erschwert (aber nicht unmöglich)