Internet-Programmierung
(mit Java)

Zhenyu Qian, MZH 8140, Tel. 2239.
[dschen-jü tjian]

Hui Shi, MZH 8120, Tel. 4684.

Inhalt

1. Objektorientierte Konzepte in Java
   - Vererbung- und Implementierungsrelation.
2. Applets in Java
   - Internet-Programmierung.
3. Graphiken, Windows, Ereignisse.
4. Multithreads.
5. Sicherheitsaspekte.

Ziele

1. Konzepte.
2. Praktische Programmierung.

Voraussetzung

Keine - Kenntnisse von OOP, insbesondere C++, vorteilhaft.

Form

- Kurs + Übung + Praktisches Programmieren

Leistungsnachweis (2+2SWS)

Schriftliche Ausarbeitung von Übungsaufgaben (Praktisches Programmieren)

Lehrbuch

- David Flanagan.
  Java in a Nutshell. O'Reilly & Associates, Inc. 1996.
- Ken Arnold und James Gosling.
  The Java Programming Language. Addison-Wesley, 1996.
- Laura Lemay und Charles L. Perkins.
  Teach Yourself Java in 21 Days. sams.net Publishing, 1996.

Folien und Übungsaufgaben

http://www.informatik.uni-bremen.de/~qian/lv3761.html

Abgeschichtete Prüfung

Gebiet: “Praktische Informatik - Programmiersprachen und Übersetzer”.

Inhalt: Internet-Programmierung + OO Entwurf (oder FP, LP, Nebenläufige Prog., Übersetzerbau)

Kapitel 1

Ein kurzer Überblick über Java

“Java is the coming thing!”

• Das Java Phänomen (von Odersky)

Frühling 1995: *“The language wars are over. C++ has won.”*
Frühling 1996: *“C++ is old hat; Java is the coming thing.”*

• Warum die plötzliche Änderung???

• Die Entwicklung der Programmiersprachen

Assembly - ... - Fortran - ... - Cobol - ... - Pascal - ...
- Modula - ... - C - ... - Ada - ... - Smalltalk - ... - C++ - ...
- Java ... ...

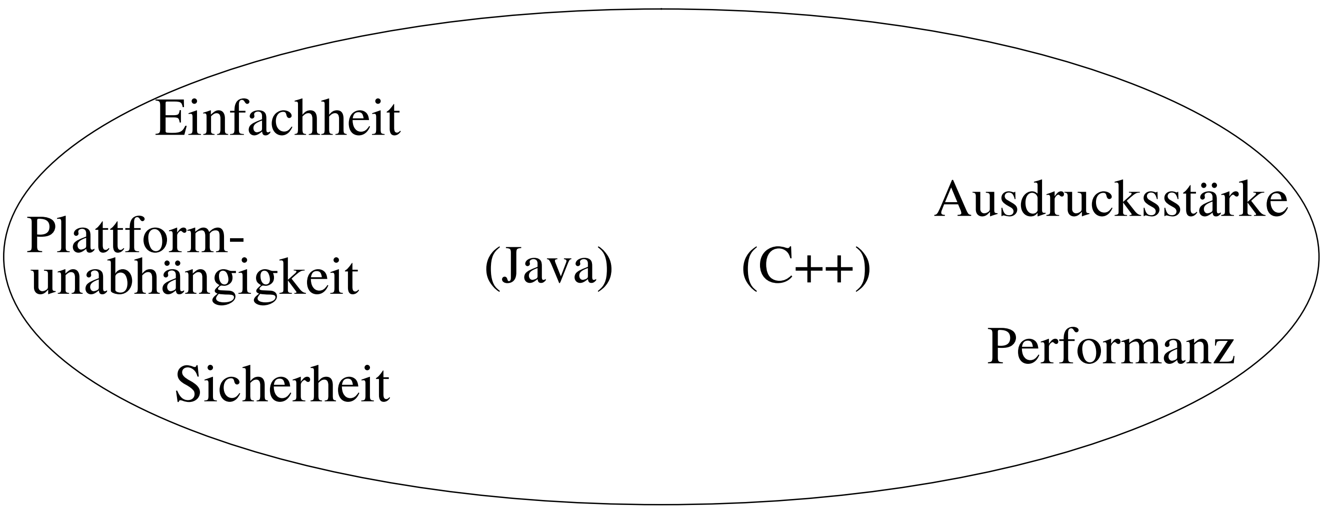
- Die Entwicklung geht immer weiter!

• Der Technologie-Vorsprung

- Ab Aug. 1991,	die Sprache Oak für Prog. auf Eletrogeräten.
- Ab Ende 1993,	Oak für Internet
- Anfang 1995,	Oak -> Java
- 1994	HotJava - ein Java-fähiger WWW-Browser
- 1995	Netscape 2.0 - Java-fähig
	Java Beta-Release
- Anfang 1996	Java 1.0

Warum Java?

- Ähnlichkeiten zu C/C++ - *nicht-schreckerregend*
- Unterschiede zu C/C++ - *interessant*
- Einige allgemeine Aspekte bei einem Entwurf:



- Was Java wirklich interessant macht, ist

Internet-Programmierung

- Intelligente Organisation und Verteilung (von Dokumentation, Software etc.) und netzweit Berechnen.

• Vorteile von WWW:

1. Bessere Organisation von Dokumenten und SW.
   Leichtere Referenzen (unter Bildern, Filmen und Texten).

2. Leichtere Verteilung. Keine Druck- und Versandskosten.

3. Leichteres Aktualisieren.

4. Anfragen und Aufträge online

- Beispiel - WWW-Seiten einer Baufirma:

1. Beschreibungen über verschiedene Sorten von Dächern, Türen, Fenstern, Wänden, etc.

2. Musterhäuser

... ...

Neue Anwendungen von Java - Fortsetzung

• Vorteile von Java im Vergleich zu WWW

1. Leichteres Testen von Software

2. Simulieren eines Vorgangs und Konstruktion online.

3. Sichten aus unendlich vielen Perspektiven.

3. Geringerer Leistungsanspruch an eigenen Computern.

- Beispiel - Java-fähige WWW-Seiten einer Baufirma:

1. Auswahl von Dächern, Türen, Fenstern, Wänden usw.
   und Konstruktion eines Hauses auf eigenem Computer.

2. Sichten über das Haus aus unendlich vielen Perspektiven.

3. Kostenkalkulation.

... ...

Was für eine Sprache ist Java?

- Java is a

“simple,	object-oriented,	distributed,
interpreted,	robust,	secure,
architecture neutral,	portable,	high-performance,
multi-threaded,	dynamic	

language”.

• Einfach

- Ziel: Leichte Sicherheitsmechanismen,
  leichtes Programmieren,
  leichtes Lernen.

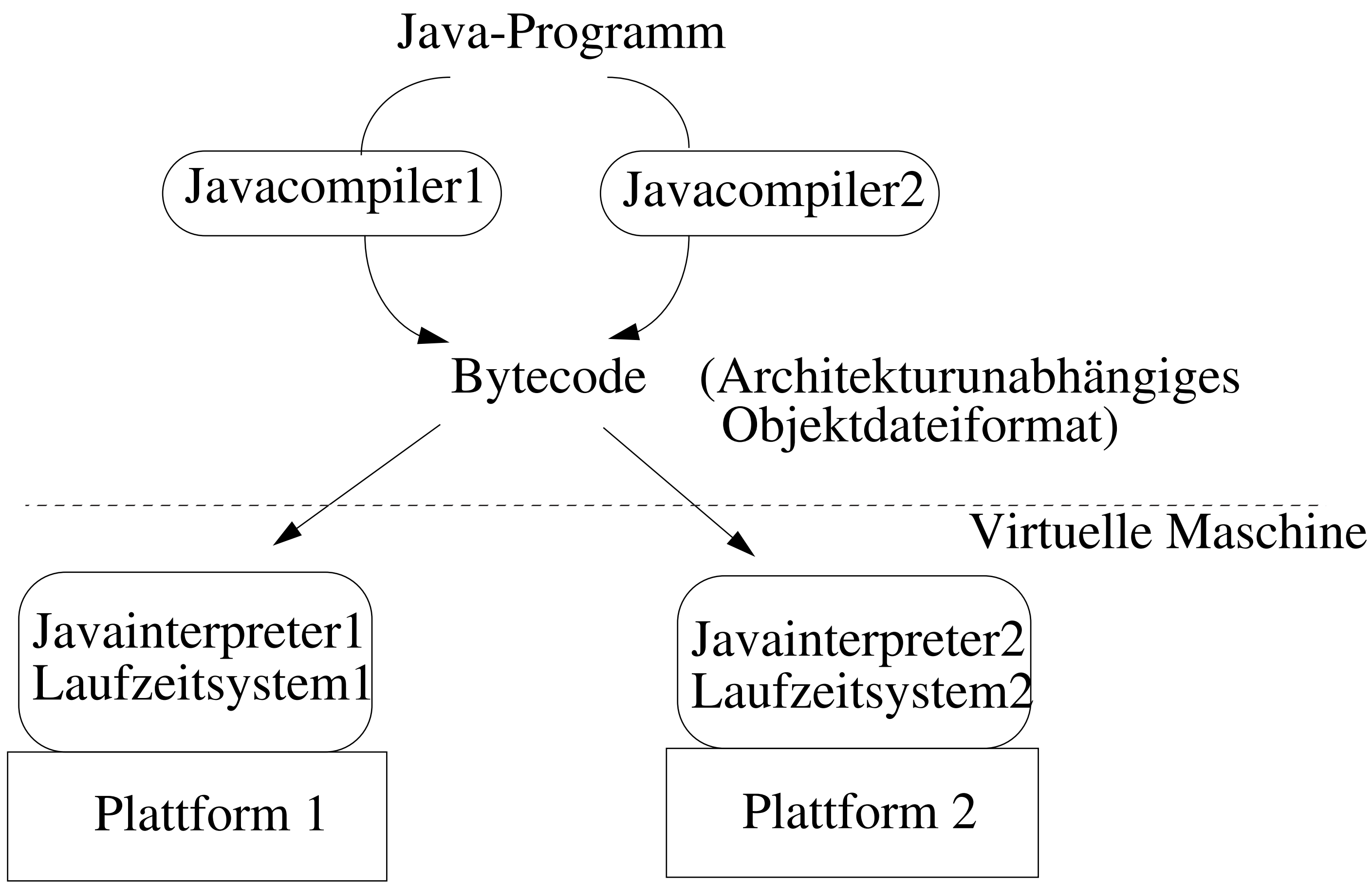
- Konkrete Maßnahmen:
  keine expliziten Zeiger,
  mit Garbage Collection,
  keine `goto`-Anweisung,
  keine überladenen Operatoren wie in C++,
  keine mehrfachen Vererbungen wie in C++,
  ... ...
- Ist jede der konkreten Maßnahmen gut begründet?

• Objektorientiert - später.

• Verteilt - später.

Was für eine Sprache ist Java? - Fortsetzung 1

- Interpretiert und achitekturunabhängig.



- Portabilität.

- Keine implementationsabhängigen Aspekte in der Sprachdefinition.
  Z.B. feste Größe für Interger, Float, Boolean, etc.)

- Java-Umgebung (Compiler, Laufzeitumgebung, Laufzeitsystem, etc.) ist portable.

Was für eine Sprache ist Java? - Fortsetzung 2

• Robustheit.
- Leicht, zuverlässige Software zu schreiben.
- Konkrete Maßnahmen:
  - Strenge Typisierung (strenger als C++)
  - Keine expliziten Zeiger
  - Mit Garbage Collection
    - Explizite Zeiger und Speicherdeallozierung sind Quellen von Fehlern
  - Ausnahmebehandlung für Fehlerbehandlung
  - ... ...

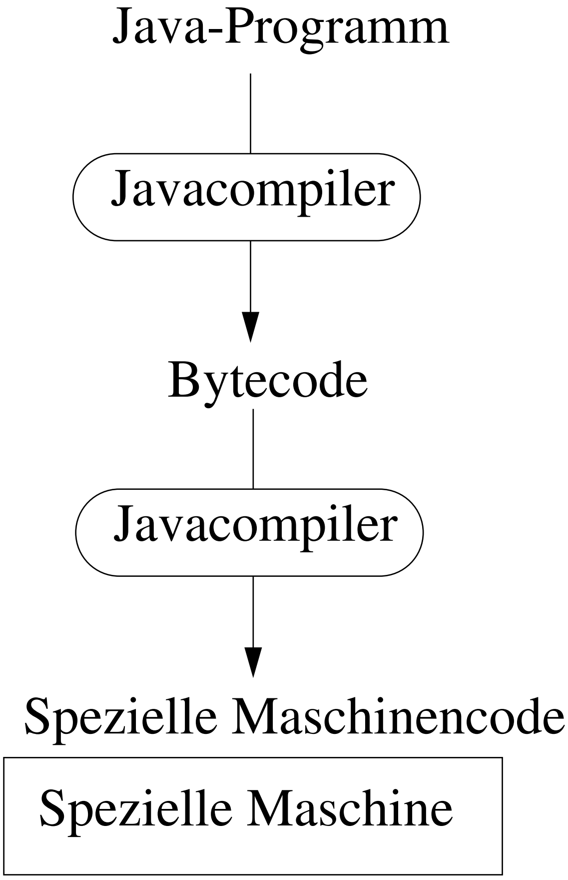
• Sicherheit (relevant zu Robustheit).
- Schutz vor fremden gefährlichen Software.
- Konkrete Maßnahmen:
  - Keine expliziten Zeiger
    - Keine Tricks durch Zeiger
  - Speicherallozierung durch Interpreter
    - Compiler bestimmt nicht Aufteilung des Speichers, sondern der Interpreter zur Laufzeit.
  - Mit Garbage Collection
  - ... ...
- Relativ sicher - “verhältnismäßig sicher”.
  - Gibt große Mühe für Sicherheit, hat bis jetzt gelungen.

Was für eine Sprache ist Java? - Fortsetzung 3

- Hochperformanz?

- Ablauf via Interpreter ist 20 mal langsamer als C,
  aber reicht für interaktive Anwendungen auf Netz.

- "Just in time" Compiler:



(Vergleichbare Effizienz im Vergl. mit C++)

- Multithread-Fähigkeit - später.

- Dynamisch - später.

Eine einfache Applikation

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

• Analyse

- Programme nur als einer Klassen. Der Name ist `HelloWorld`
  Rein objektorientiert, im Gegensatz zu C++, Oberon.

- Mit der Methode `main()` beginnt Ausführung der Klasse.
  Ähnlich wie in C++, anders als in Oberon.

• Übersetzen und Ausführen in Sun JDK

- Speichern das Programm in eine Datei `HelloWorld.java`

- Übersetzen (`javac` ist der Compiler)

```
% javac HelloWorld.java
```

- Das Ergebnis ist eine Datei `HelloWorld.class`

- Interpretieren (`java` ist der Interpreter)

```
% java HelloWorld
```

- Das Ergebnis ist Ausgabe der Meldung

```
Hello World!
```

Ein einfaches Applet - Das Programm selber

```
public class HelloWorldApplet
                extends java.applet.Applet{
    public void paint(java.awt.Graphics g){
         g.drawString("Hello World!", 5, 25);
    }
}
```

- Analyse

  - `... extends ...`
    Unterklasse `extends` Oberklasse

  - Mehrere Standard-Methoden für Applet
    (in `java.applet.Applet`):
    `init()`, `start()`, `paint()`, ...
    Sie werden von einem WWW-Browser aufgerufen.

  - Eigene Definitionen für Standard-Methoden:
    Die `paint()` Methode in `HelloWorldApplet`.

- Übersetzen
  - Speichern in eine Datei `HelloWorldApplet.java`

    `% javac HelloWorldApplet.java`
  - Das Ergebnis ist eine Datei `HelloWorldApplet.class`

Ein einfaches Applet - Aufruf von WWW-Browser

```
<HTML>
<HEAD><TITLE>Hello to Everybody
</TITLE></HEAD>
<BODY>
<P>My Java applet says:
<APPLET CODE="HelloWorldApplet.class"
              WIDTH=150 HEIGHT=25></APPLET>
</BODY>
</HTML>
```

- Analyse
  - Der `APPLET` Teil ist für Aufruf eines Applets zuständig.
  - `CODE="..."` - wo ist die übersetzte Applet-Klasse
  - `WIDTH=... HEIGHT=...` - Größe des Applets.

- Laden der HTML-Datei durch Netscape-Browser
  - `Open File` öffnet die HTML-Datei.