

Java I – Vorlesung 1

Einführung in Java

Alexander Koller
koller@coli.uni-sb.de

26.4.2004

Problem – Algorithmus – Programm
Programmiersprachen
Warum Java?
Unser erstes Java-Programm
Organisatorisches

Was heißt „Programmieren“?

- ◆ Programmierer will ein **Problem** auf systematische Weise lösen.
- ◆ Ein **Algorithmus** ist eine abstrakte, detaillierte Berechnungsvorschrift, die das Problem löst.
- ◆ Ein **Programm** ist eine Darstellung des Algorithmus in einer konkreten **Programmiersprache**.
- ◆ Ein Programm kann mit verschiedenen **Eingaben** ausgeführt werden.

Einige Probleme

- ◆ Berechnung von arithmetischen Funktionen
- ◆ Was ist der kürzeste Weg in einem Graphen?
- ◆ Computerlinguistische Probleme: Dialog mit einem Menschen führen; grammatische Analyse
- ◆ Wie komme ich vom Bett in die Uni?
 - Teilproblem: Anziehen

Algorithmen

- ◆ „Kochrezepte“, die Schritt für Schritt beschreiben, wie das Problem gelöst wird.
- ◆ Funktionieren für alle Eingaben, die das Problem vorsieht.
- ◆ Müssen in endlich vielen Schritten fertig werden.
- ◆ Granularität der Einzelschritte hängt davon ab, was ein Mensch als offensichtliche Schritte akzeptiert.

Ein Algorithmus fürs Anziehen

- ◆ Gehe zum Kleiderschrank.
- ◆ Nimm eine frische Unterhose.
- ◆ Stecke die Beine durch die Löcher.
- ◆ Nimm ein frisches T-Shirt.
- ◆ Stecke Kopf und Arme durch die passenden Löcher.
- ...
- ◆ Binde den rechten Schuh.

Ein Algorithmus für „Größte Zahl“

Gegeben eine Liste L von n natürlichen Zahlen; gesucht ist die größte Zahl in L.

- ◆ Merke dir die erste Zahl in L als aktuelle größte Zahl.
- ◆ Gehe der Reihe nach durch die 2-te bis n-te Zahl in L. Wenn die aktuelle Zahl größer als die bisherige größte ist, merke dir die aktuelle als die neue größte Zahl.
- ◆ Der gemerkte Wert nach Ansehen aller Zahlen in L ist die echte größte Zahl in L.

Programme

- ◆ Sind konkrete Implementierungen eines Algorithmus in einer Programmiersprache (z.B. Java).
- ◆ Verwenden Konstruktionen der Programmiersprache, um intuitive Begriffe präzise zu machen (z.B. Schleifen, Variablen).
- ◆ Einzelne Schritte hängen von Programmiersprache und verfügbaren Funktionen ab.

Ein Algorithmus für „Größte Zahl“

Gegeben eine Liste L von n natürlichen Zahlen; gesucht ist die größte Zahl in L.

- ◆ Merke dir die erste Zahl in L als aktuelle größte Zahl.
- ◆ Gehe der Reihe nach durch die 2-te bis n-te Zahl in L. Wenn die aktuelle Zahl größer als die bisherige größte ist, merke dir die aktuelle als die neue größte Zahl.
- ◆ Der gemerkte Wert nach Ansehen aller Zahlen in L ist die echte größte Zahl in L.

„Größte Zahl“-Algorithmus als Java-Programm

Gegeben eine Liste L von n natürlichen Zahlen; gesucht ist die größte Zahl in L.

```
int max = L[0];
```

- ◆ Gehe der Reihe nach durch die 2-te bis n-te Zahl in L. Wenn die aktuelle Zahl größer als die bisherige größte ist, merke dir die aktuelle als die neue größte Zahl.
- ◆ Der gemerkte Wert nach Ansehen aller Zahlen in L ist die echte größte Zahl in L.

„Größte Zahl“-Algorithmus als Java-Programm

Gegeben eine Liste L von n natürlichen Zahlen; gesucht ist die größte Zahl in L.

```
int max = L[0];
```

Variable

Zuweisung

- ◆ Gehe der Reihe nach durch die 2-te bis n-te Zahl in L. Wenn die aktuelle Zahl größer als die bisherige größte ist, merke dir die aktuelle als die neue größte Zahl.
- ◆ Der gemerkte Wert nach Ansehen aller Zahlen in L ist die echte größte Zahl in L.

„Größte Zahl“-Algorithmus als Java-Programm

Gegeben eine Liste L von n natürlichen Zahlen; gesucht ist die größte Zahl in L.

```
int max = L[0];
```

```
for( int i = 1; i < L.length; i++ )
```

Wenn die aktuelle Zahl größer als die bisherige größte ist, merke dir die aktuelle als die neue größte Zahl.

- ◆ Der gemerkte Wert nach Ansehen aller Zahlen in L ist die echte größte Zahl in L.

„Größte Zahl“-Algorithmus als Java-Programm

Gegeben eine Liste L von n natürlichen Zahlen; gesucht ist die größte Zahl in L.

```
int max = L[0];
```

Schleife

```
for( int i = 1; i < L.length; i++ )
```

Wenn die aktuelle Zahl größer als die bisherige größte ist, merke dir die aktuelle als die neue größte Zahl.

- ◆ Der gemerkte Wert nach Ansehen aller Zahlen in L ist die echte größte Zahl in L.

„Größte Zahl“-Algorithmus als Java-Programm

Gegeben eine Liste L von n natürlichen Zahlen; gesucht ist die größte Zahl in L.

```
int max = L[0];
```

```
for( int i = 1; i < L.length; i++ )  
    if( L[i] > max )
```

merke dir die aktuelle als die neue
größte Zahl.

- ◆ Der gemerkte Wert nach Ansehen aller Zahlen in L ist die echte größte Zahl in L.

„Größte Zahl“-Algorithmus als Java-Programm

Gegeben eine Liste L von n natürlichen Zahlen; gesucht ist die größte Zahl in L.

```
int max = L[0];
```

```
for( int i = 1; i < L.length; i++ )  
    if( L[i] > max )
```

Bedingung

merke dir die aktuelle als die neue größte Zahl.

- ◆ Der gemerkte Wert nach Ansehen aller Zahlen in L ist die echte größte Zahl in L.

„Größte Zahl“-Algorithmus als Java-Programm

Gegeben eine Liste L von n natürlichen Zahlen; gesucht ist die größte Zahl in L.

```
int max = L[0];
```

```
for( int i = 1; i < L.length; i++ )  
    if( L[i] > max )  
        max = L[i];
```

- ◆ Der gemerkte Wert nach Ansehen aller Zahlen in L ist die echte größte Zahl in L.

„Größte Zahl“-Algorithmus als Java-Programm

Gegeben eine Liste L von n natürlichen Zahlen; gesucht ist die größte Zahl in L.

```
int max = L[0];
```

```
for( int i = 1; i < L.length; i++ )  
    if( L[i] > max )  
        max = L[i];
```



Zuweisung

- ◆ Der gemerkte Wert nach Ansehen aller Zahlen in L ist die echte größte Zahl in L.

„Größte Zahl“-Algorithmus als Java-Programm

Gegeben eine Liste L von n natürlichen Zahlen; gesucht ist die größte Zahl in L.

```
int groessteZahl(int[] L) {  
    int max = L[0];  
  
    for( int i = 1; i < L.length; i++ )  
        if( L[i] > max )  
            max = L[i];  
  
    return max;  
}
```

„Größte Zahl“-Algorithmus als Java-Programm

Gegeben eine Liste L von n natürlichen
Zahlen: gesucht ist die größte Zahl in L.

Methode

Argumente

```
int groessteZahl(int[] L) {  
    int max = L[0];  
  
    for( int i = 1; i < L.length; i++ )  
        if( L[i] > max )  
            max = L[i];  
  
    return max;  
}
```

Rückgabewert der
Methode

„Größte Zahl“-Algorithmus als Java-Programm

Gegeben eine Liste L von n natürlichen Zahlen; gesucht ist die größte Zahl in L.

```
class GroessteZahl {
    int groessteZahl(int[] L) {
        int max = L[0];

        for( int i = 1; i < L.length; i++ )
            if( L[i] > max )
                max = L[i];

        return max;
    }
}
```

„Größte Zahl“-Algorithmus als Java-Programm

Gegeben eine Liste L von n natürlichen
Zahlen, ... bestimmt die größte Zahl in L.

Klasse

```
class GroessteZahl {  
    int groessteZahl(int[] L) {  
        int max = L[0];  
  
        for( int i = 1; i < L.length; i++ )  
            if( L[i] > max )  
                max = L[i];  
  
        return max;  
    }  
}
```

Vorsicht: Wir haben den
Spezialfall missachtet, bei
dem die Liste leer ist!

Effiziente Algorithmen

- ◆ Manche Algorithmen lösen ein Problem **effizienter** (mit weniger Zeit- oder Speicherverbrauch) als andere.
- ◆ Beispiel: Finde die **zwei** größten Zahlen in L.
- ◆ Algorithmus 1: Verwende zwei Variablen, um die aktuell beiden größten Zahlen zu speichern.
- ◆ Algorithmus 2: Berechne zunächst die größte Zahl; dann lösche sie aus L und berechne nochmal die größte Zahl.

Programmiersprachen

- ◆ Es gibt eine sehr breite Auswahl an verschiedenen Programmiersprachen.
- ◆ CPU versteht sehr eingeschränkten Befehlssatz, der für Menschen unbequem ist.
- ◆ Programmiersprachen verbergen Komplexität und stellen abstraktere Konstrukte zur Verfügung.
- ◆ Verfolgen verschiedene Ziele; unterstützen verschiedene Programmierstile; haben verschiedene Vor- und Nachteile.

Imperativ vs. funktional vs. deklarativ

- ◆ Imperativ: Programme sind Sequenzen von Anweisungen. (Algol, Fortran, C, C++, Java)
- ◆ Funktional: Programme sind mathematische Funktionen, die aus einfacheren Funktionen zusammengesetzt sind (SML, Lisp, Haskell)
- ◆ Deklarativ: Programme sind logische Formeln, die das Problem spezifizieren; Programmiersystem kümmert sich um eigentliche Berechnung (Prolog)

Prozedural vs. objektorientiert

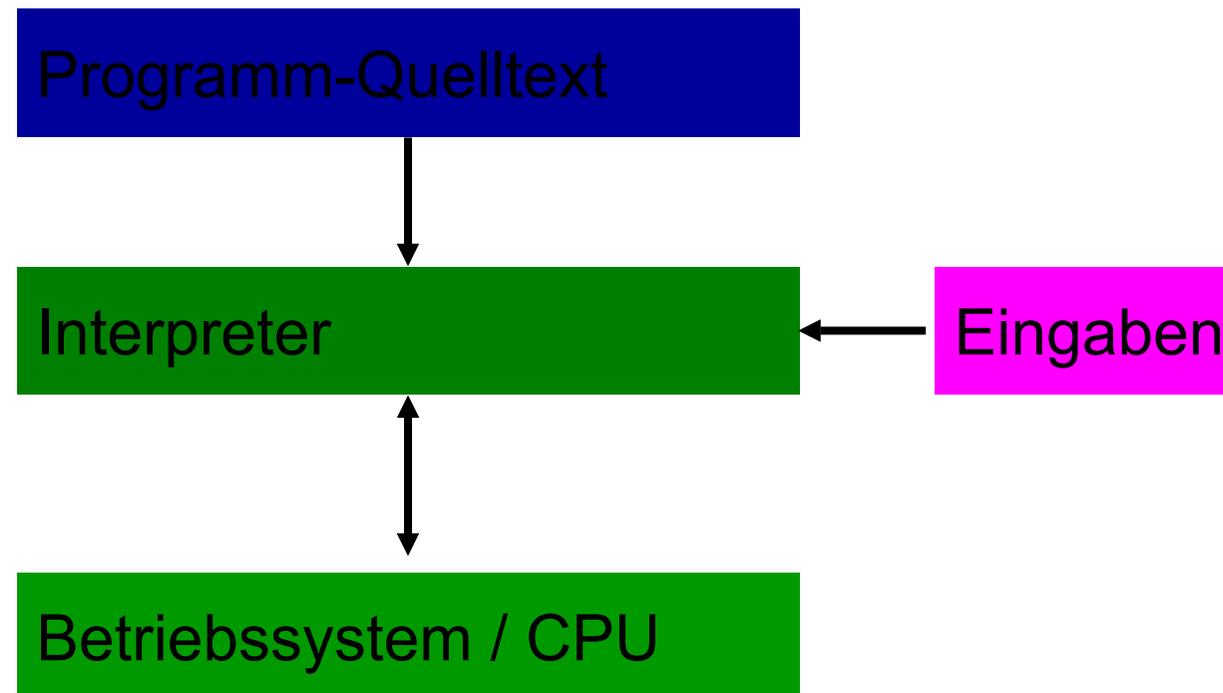
- ◆ Prozedural: Die wesentliche Gliederungseinheit des Programms ist die **Prozedur**, d.h. Gruppe von Anweisungen (C, SML)
- ◆ Objektorientiert: Die wesentliche Gliederungseinheit ist die **Klasse**, d.h. Repräsentation eines Objekts mit Sammlung von Methoden (Java, C++, Smalltalk)

Interpretiert vs. kompiliert

- ◆ Programmiersprachen sind dafür da, dass Menschen Algorithmen bequem aufschreiben können.
- ◆ Programm-**Quelltexte** können nicht direkt vom Computer ausgeführt werden.
- ◆ Zu einer Programmiersprache gehört deshalb Software, die Programme für Computer ausführbar macht.

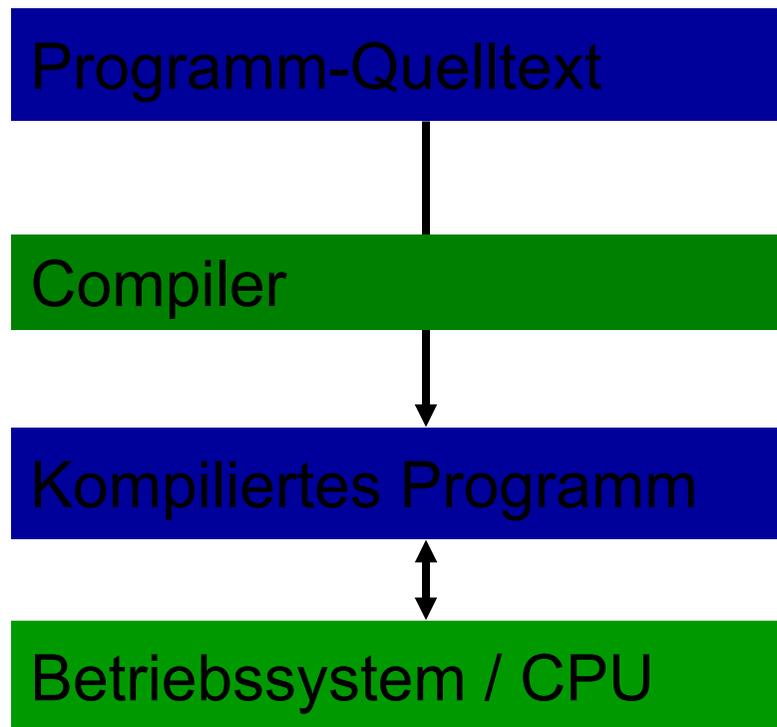
Interpretation

- ◆ Interpreter sind Programme, die den Quelltext Anweisung für Anweisung ausführen. Belieb z.B. für Skriptsprachen wie Perl.



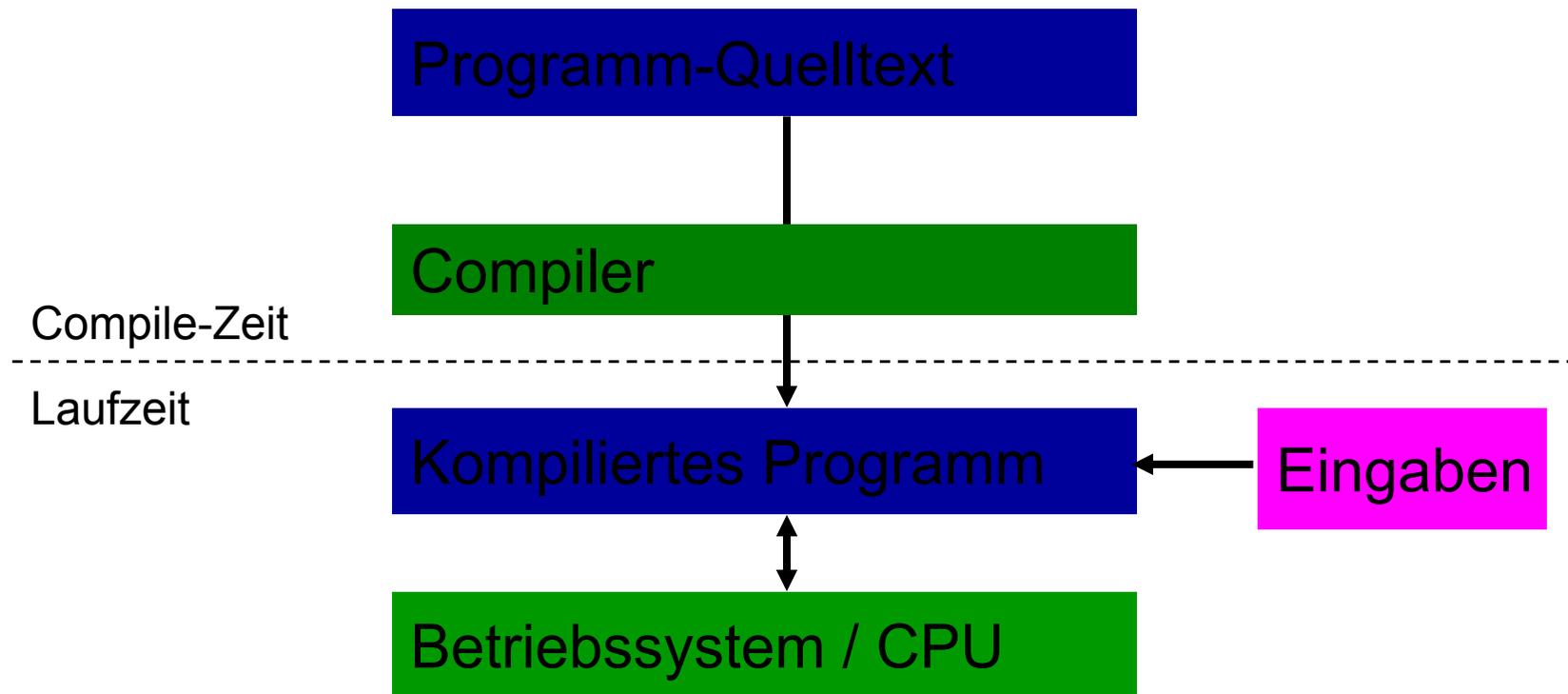
Kompilation

- ◆ Compiler sind Programme, die den Quelltext in ein direkt ausführbares Maschinenprogramm übersetzen. Beliebte z.B. für C, C++ usw.



Kompilation

- ◆ Compiler sind Programme, die den Quelltext in ein direkt ausführbares Maschinenprogramm übersetzen. Beliebte z.B. für C, C++ usw.

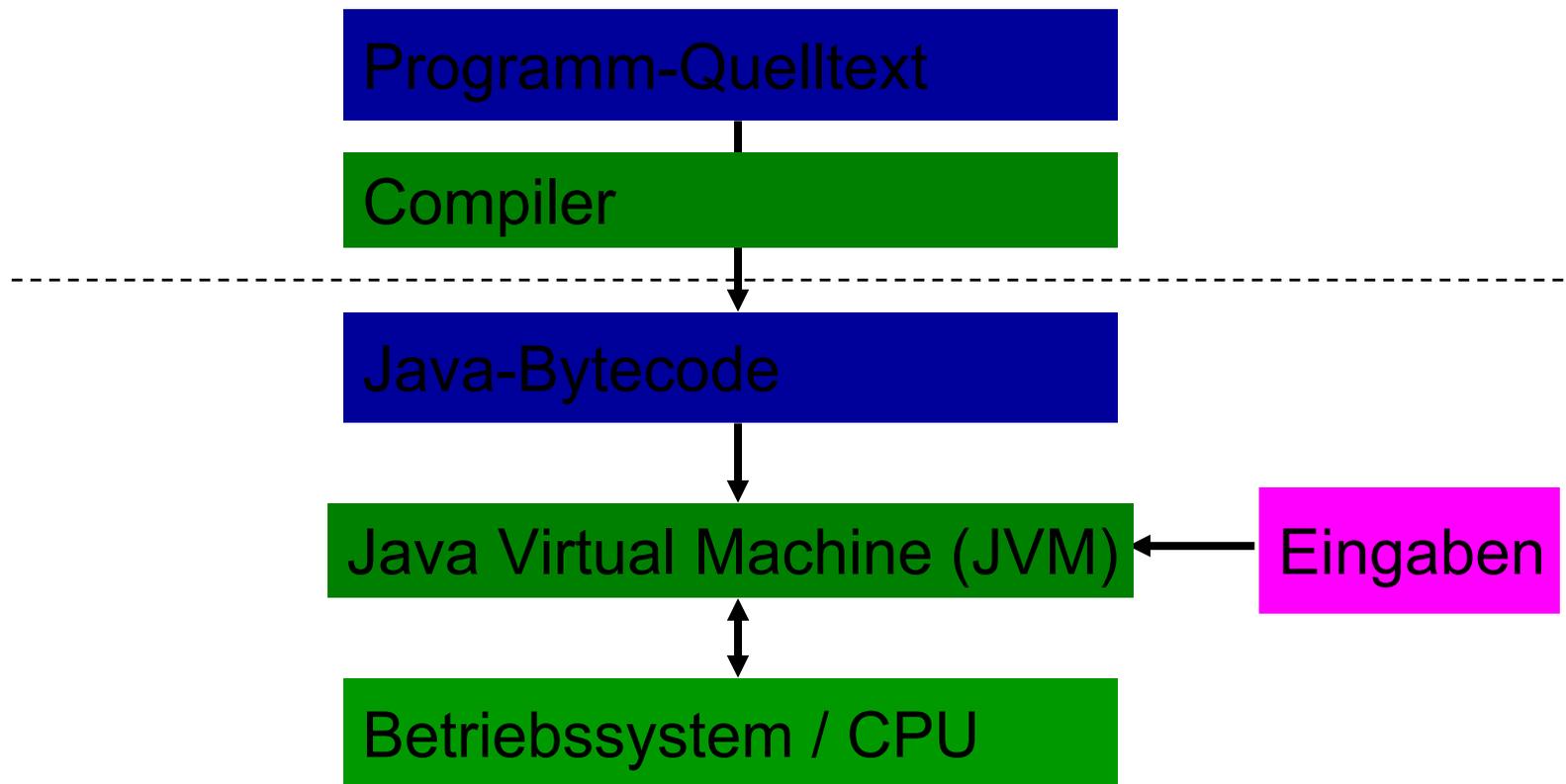


Interpretation vs. Kompilation

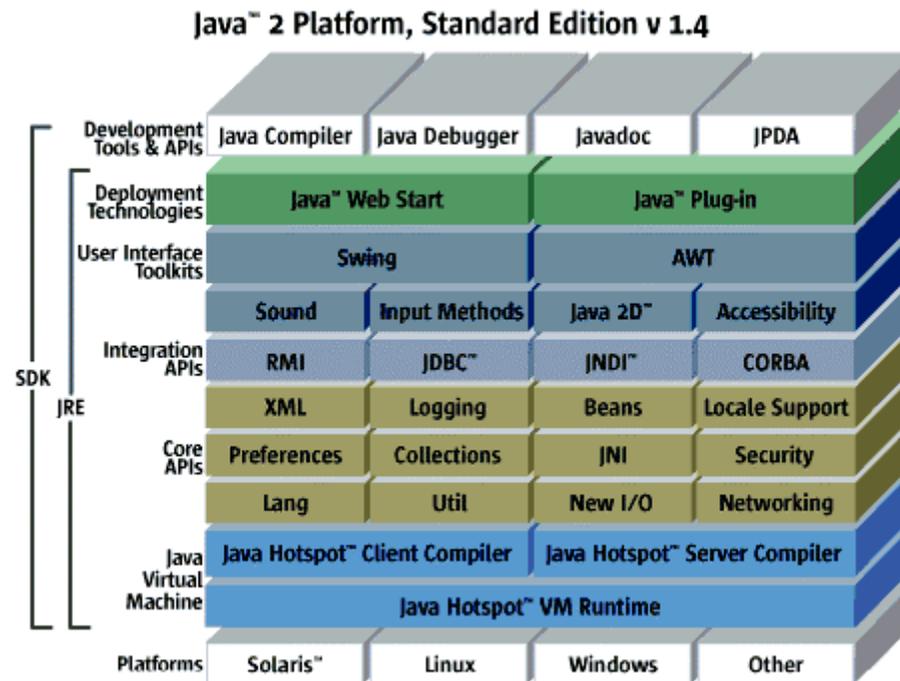
- ◆ Interpretation ist viel langsamer, da Interpreter selbst Rechenzeit verbraucht. Compiler wendet einmal Zeit bei der Kompilation auf, danach sehr effizienter Programmablauf.
- ◆ Ein kompiliertes Programm läuft nur auf einer einzigen Plattform (Betriebssystem + CPU). Quelltext muss daher auf neuen Plattformen neu kompiliert werden (Portierung). Das ist oft problematisch.

Virtuelle Maschinen

- ◆ In Java wird der Quelltext zunächst in Bytecode kompiliert; dieser wird von einer virtuellen Maschine interpretiert oder weiter ("just in time") kompiliert.



Die Java-Plattform



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

Geschichte von Java

- ◆ Seit 1995 verfügbar.
- ◆ Ursprünglich von Sun entwickelt als Plattform für Home Entertainment.
- ◆ Zunächst v.a. als Internet-Plattform verkauft.
- ◆ Heute für alle möglichen Anwendungen sehr weit verbreitete Sprache.
- ◆ Syntax basiert auf C/C++, aber viele objektorientierte Konzepte etwas anders als in C++ umgesetzt.

Warum Java?

- ◆ Portabilität: Ein (in Bytecode) kompiliertes Programm kann ohne Änderung auf verschiedenen Betriebssystemen ausgeführt werden. „Java-Plattform“.
- ◆ Objektorientierung: Unterstützt Modularisierung und Wiederverwendbarkeit von Programmcode.
- ◆ Statische Typisierung: Compiler kann viele Programmierfehler abfangen.

Warum Java?

- ◆ Mächtige Standardbibliothek: Standard-Installation von Java enthält viele nützliche Klassen.
- ◆ Effizienz: Moderne Java-Systeme kommen in die Nähe von C++-Implementierungen.
- ◆ Sicherheit: Kann verhindern, dass Programme Schaden anrichten.
- ◆ Weit verbreitet: Daher Zugriff auf Programme vieler anderer Programmierer.

Unser erstes Java-Programm

Definiere die Klasse HelloWorld in der Datei HelloWorld.java:

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

Unser erstes Java-Programm

Definiere die Klasse HelloWorld in der Datei

HelloWorld.java:

Klasse

Hauptprogramm

```
class HelloWorld {
```

Kommandozeilen
-Argumente

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

Anweisungen

```
}
```

Wie führt man ein Java-Programm aus?

```
[koller@cicero]$ javac HelloWorld.java
```

- ◆ Kompiliert Quelltext in HelloWorld.java in Bytecode und schreibt das Ergebnis in die Datei HelloWorld.class.

```
[koller@cicero]$ java HelloWorld
```

- ◆ Lädt den Bytecode für die Klasse HelloWorld aus der Datei HelloWorld.class und führt ihn auf der virtuellen Maschine aus.

Literatur

- ◆ **Einschlägige Literatur:**
 - Java-Bücher von Sun
 - O'Reilly („... in a nutshell“)
 - und viele andere
- ◆ **Die meisten Sun-Bücher sind online zugänglich:**
`http://java.sun.com/docs/books/`
- ◆ **Java-Tutorial:**
`http://java.sun.com/docs/books/tutorial/`

Zusammenfassung

- ◆ Algorithmen lösen Probleme; Programme implementieren Algorithmen in einer konkreten Programmiersprache.
- ◆ Es gibt viele Programmiersprachen. Java zeichnet sich v.a. durch Portabilität, Objektorientierung und weite Verbreitung aus.
- ◆ Zweistufiger Prozess: Erst Kompilation in Bytecode, dann Ausführen von Bytecode auf virtueller Maschine.

Organisatorisches

- ◆ Wöchentliche Übungen (wichtig!) in Teams von max. drei Studenten.
- ◆ Jeder Student muss Programmcode selbst kommentieren und in der Lage sein, seine Übung zu erklären.
- ◆ Voraussetzung zur Klausurzulassung sind 50% der Punkte in den Übungen.
- ◆ Klausur am Ende des Semesters

Termin Übungsgruppen

- ◆ Problem: Finde eine hinreichende Anzahl von Übungsterminen, so dass möglichst wenige Studenten zu keinem der Termine Zeit haben.
- ◆ Wir verwenden jetzt folgenden Algorithmus: Finde zunächst Termin, bei dem die meisten Zeit haben; finde dann Termin, bei dem von den übrigen die meisten Zeit haben.
- ◆ Löst dieser Algorithmus das Problem korrekt?