

Softwareentwicklung 1

Dr. Herbert Prähofer
Institut für Systemsoftware
Johannes Kepler Universität Linz



Vortragender

Dr. Herbert Praehofer

**Institute for System Software
Johannes Kepler University
Altenbergerstrasse 69
A-4040 Linz / Austria**

**Tel.: ++43 (732) 2468 7132
Fax.: ++43 (732) 2468 7138
EMail: herbert.praehofer@jku.at
http: www.ssw.uni-linz.ac.at**

Raum: HF 306 (Hochschulfondgebäude 3. Stock)



Ziel und Inhalt

Ziel Einführung des Programmieren in Java

- Inhalte**
- Grundlagen der Programmierung
 - Programmieren in Java
 - Einfache Algorithmen
 - Grundlagen der Objektorientierung
 - Programmentwurf



Motivation programmieren zu lernen

- Software erstellen können
 - Grundlagen der Programmierung verstehen
 - Java kennen
 - komplexe Software entwerfen
 - sich neue Gebiete der Softwaretechnik erarbeiten können
- Software verstehen
 - wie arbeitet Software
 - was kann Software leisten
 - wie sind Programme aufgebaut
 - wie wird Software erstellt
- Kreativität erleben
 - Programmieren kann viel Spaß machen
 - Programmieren ist eine Herausforderung



Organisation der LVA Softwareentwicklung 1

- Vorlesung
 - Stoff wird theoretisch vorgetragen
 - Mit eingebauten Programmierbeispielen
- Übung
 - Stoff wird geübt
 - Programmieraufgaben sind wöchentlich zu programmieren
 - 2 Unterrichtsstunden für
 - Wiederholung
 - Praktische Beispiele
 - Besprechung der Programmieraufgaben
 - Übungsleiter: Hr. DI Peter Hamader

**Programmieren ist eine Fähigkeit,
die nur mit viel Übung erlernt werden kann!!**



Webpage der LVA

<http://www.ssw.uni-linz.ac.at/Teaching/Lectures/UZR/SWE1/2009W/index.html>

- Alle wichtigen Infos wie
 - Termine
 - Unterlagen
 - Studienmaterial
- findet man auf dieser Seite



- P. Mössenböck, Sprechen Sie Java?, 2. Auflage, dpunkt.verlag, 2005 (Lehrbuch zur VL!)
- G. Krüger, GoTo Java 2, Addison-Wesley, 2000
On-Line Buch für Java Online Version hier Download (www.javabuch.de)
- M. Campione, K.Walrath: The Java Tutorial Online Edition (www.javasoft.com/tutorial)
- Java-Einführungskurs (<http://www.boku.ac.at/javaeinf/jein.html>)



Einheit 1: Grundlagen der Programmierung

Programmierung und Ausführung von Programmen
Algorithmen und Algorithmdarstellung
Programmiersprachen und Java
Das Java-Entwicklungssystem



Worum geht es?

Programmieren

Problem so exakt beschreiben, dass es ein Computer lösen kann

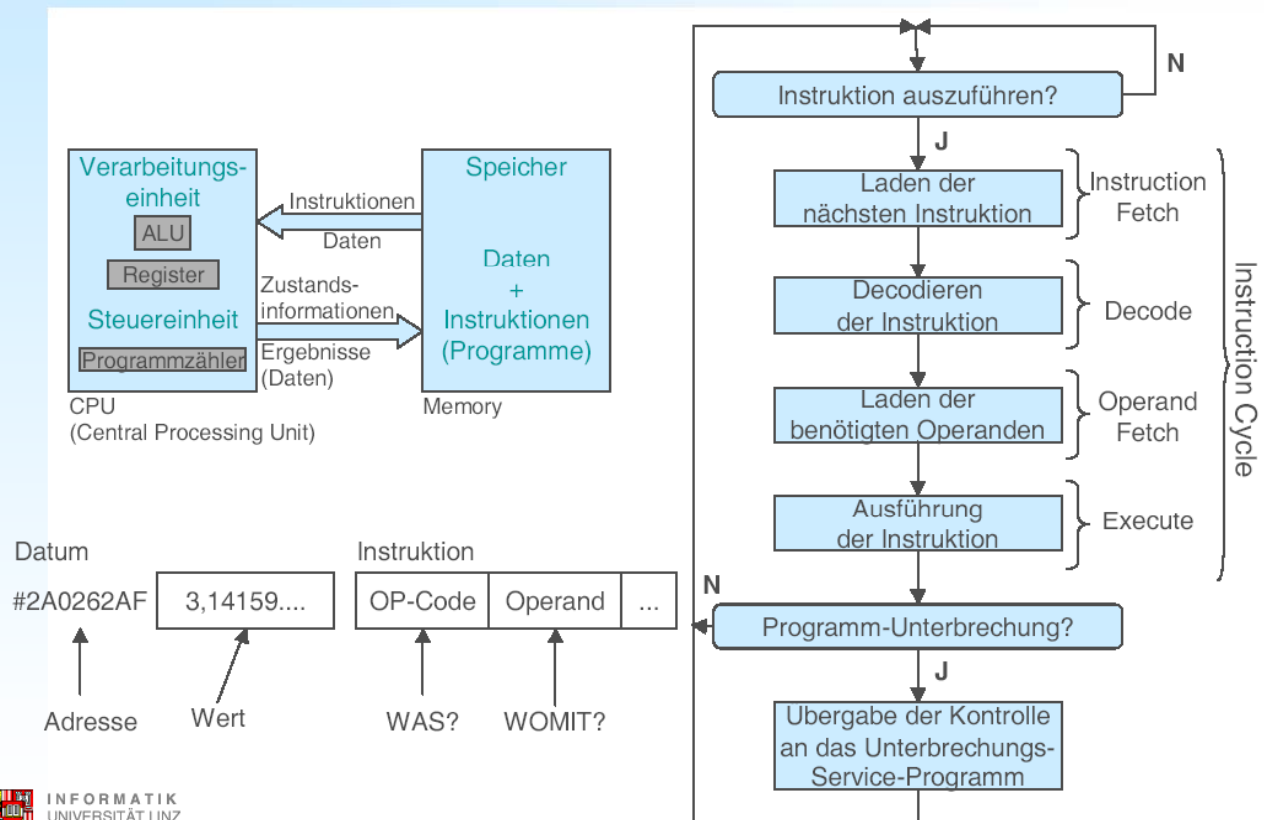
- ☞ kreative Tätigkeit
- ☞ Ingenieurtätigkeit
- ☞ Nur wenige Leute können gut programmieren

Programm = Daten + Befehle



0699 1111 1010

von Neumann Architektur



RISC Programm

(hypothetischer) RISC Befehlssatz: Suchen in einer verketteten Liste

- R1 enthält Adresse des Listenanfanges
- R2 enthält Suchmuster

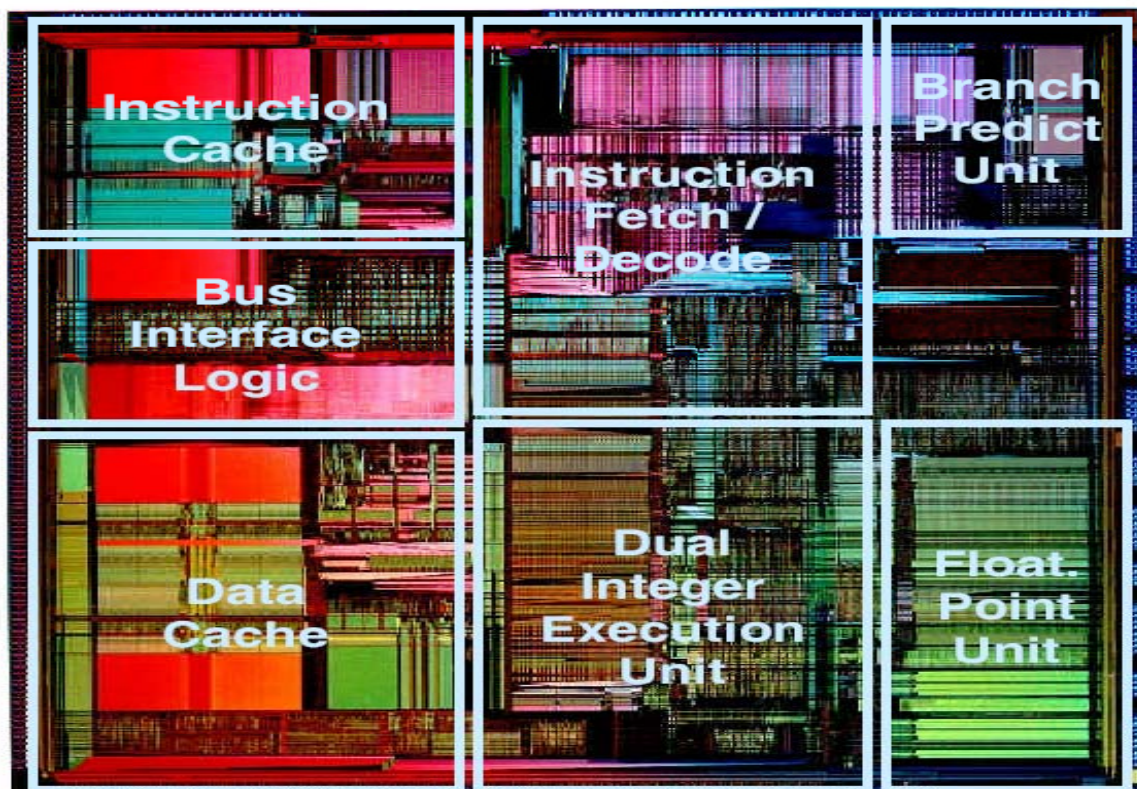
	MOVE	W R1, R0	Laufvariable R0 initialisieren
loop:	CMP	W !0, R0	Testen ob "NULL" (Liste leer)
	JEQ	nein	fertig (Listenende), nicht gefunden
	CMP	W R2, 8+!R0	Vergleich des Suchmusters mit Nutzdaten
	JEQ	ja	fertig, gefunden
	MOVE	W !R0, R0	dereferenzieren
	JUMP	loop	Schleife
ja:	...		
nein:	...		



INFORMATIK
UNIVERSITÄT LINZ

SWE 1 – A. Ferscha
Begriffsfestlegungen < 32 >

Verarbeitungseinheit



Typische Befehle

- **load:** Laden von Daten aus Hauptspeicher
- **store:** Schreiben von Daten in Hauptspeicher
- **move:** Speicheroperationen
- **add, sub, mult, ...:** Arithmetische Operationen
- **cmp:** Vergleichsoperationen: Vergleich auf 0
- **jmp, jeq:** Sprünge und bedingte Sprünge im Ablauf
- ...



Variablen

Programme arbeiten mit Variablen.
Sind benannte Behälter für Werte.



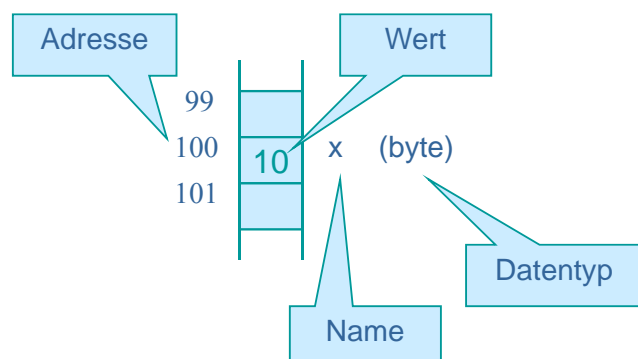
Variablen können ihren Wert ändern



Beispiel: byte x = 10







Charakterisiert durch

- Name oder Bezeichner
- Wert
- Adresse (im Hauptspeicher)
- Datentyp
 - int, char, byte, float, ...



Datentypen

Variablen haben einen Datentyp == Menge erlaubter Werte

Variablentyp	Werte	
 Zahl	 17  54 ...	Typ \cong Form - in eine Zahlenvariable passen nur Zahlen - in eine Zeichenvariable passen nur Zeichen
 Zeichen	 'a'  'x' ...	

Beispiele von Datentypen:

- char (2 Byte) [a-z][A-Z][0-9][Sonderzeichen]
- int (4 Byte) ~-2,147 Mrd. – ~+2,147 Mrd.
- boolean true, false
- float 3.14159265259



Algorithmus

Ein Algorithmus ist ein **endliches, schrittweises** Verfahren zur **Berechnung gesuchter aus gegebenen Größen**, in dem jeder Schritt aus einer Anzahl **eindeutig** ausführbarer Operationen und einer Angabe über den **nächsten Schritt** besteht.

- Berechnungsvorschrift
- schrittweise
- endlich
- ausführbare Operationen
- Angabe der Reihenfolge der Schritte

(Der Name *Algorithmus* ist abgeleitet von Al Chwarizmi, arabischer Mathematiker, ca. 800 n.Chr.)



Beispiel: Algorithmus „Sum“

Algorithmus „Sum“ zum Berechnen der Summe einer Zahlenfolge

Aufgabenstellung:

Gegeben: Eine Zahl n größer 0

Gesucht: Die Summe dieser Zahlen von 1 bis n

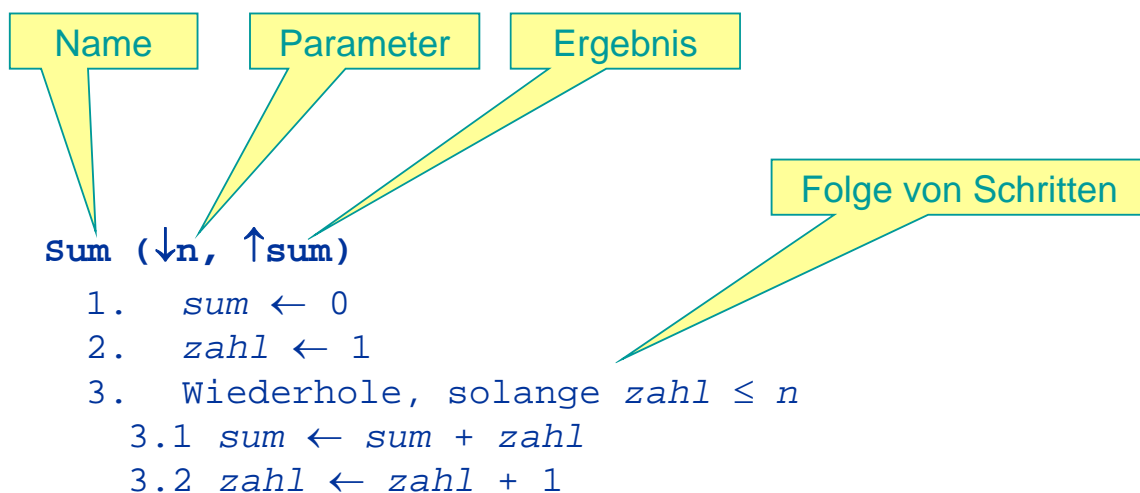
Algorithmus:

1. Addiere alle Zahlen von 1 bis n
2. Gib das Ergebnis aus

Diese Formulierung ist für einen Computer noch *zu wenig genau!*



Algorithmus „Sum“ als Verfahren in mehreren Schritten

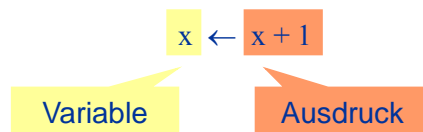


Diese Formulierung ist einem Computerprogramm schon *sehr nahe.*



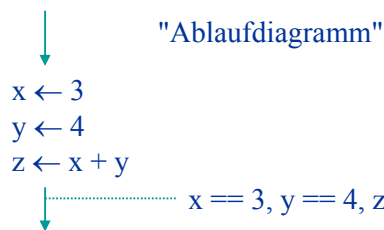
Anweisungen: Wertzuweisung und Sequenz

Wertzuweisung



1. werte Ausdruck aus
2. weise seinen Wert der Variablen zu

Anweisungsfolge (auch Sequenz)



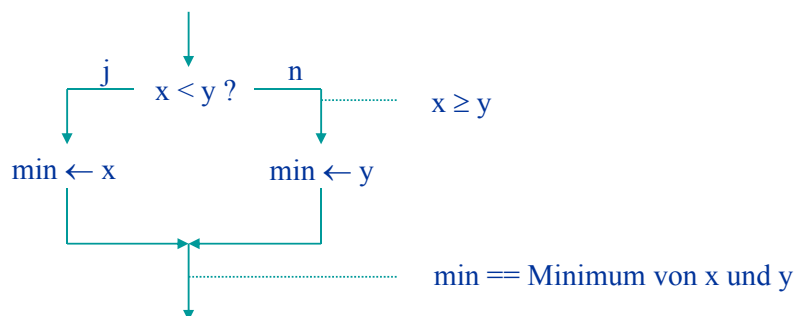
Assertion

Assertion (Zusicherung)
Aussage über den Zustand des Algorithmus
an einer bestimmten Stelle



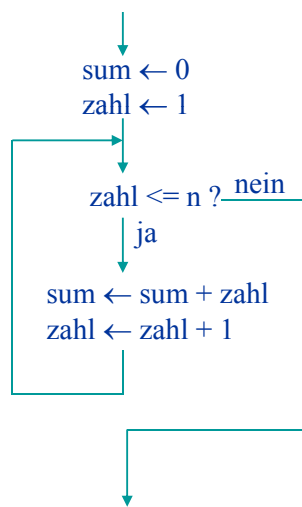
Anweisungen: Verzweigung

Auswahl (auch Verzweigung, Abfrage, Selektion)

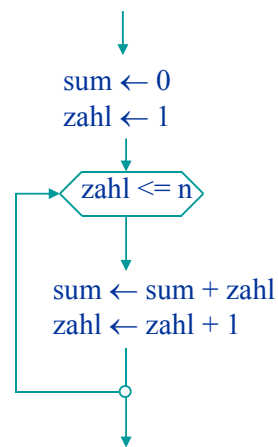


Anweisungen: Wiederholung

Wiederholung (auch Schleife, Iteration)

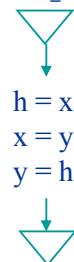


Alternative Darstellung



Beispiel: Vertauschen zweier Variableninhalte

Swap ($\uparrow x, \uparrow y$)

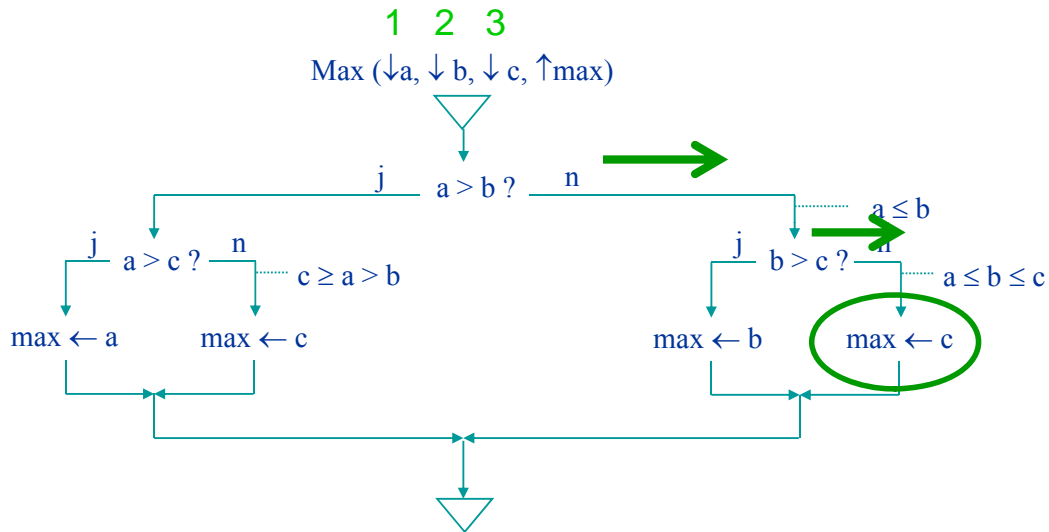


Schreibtischtest

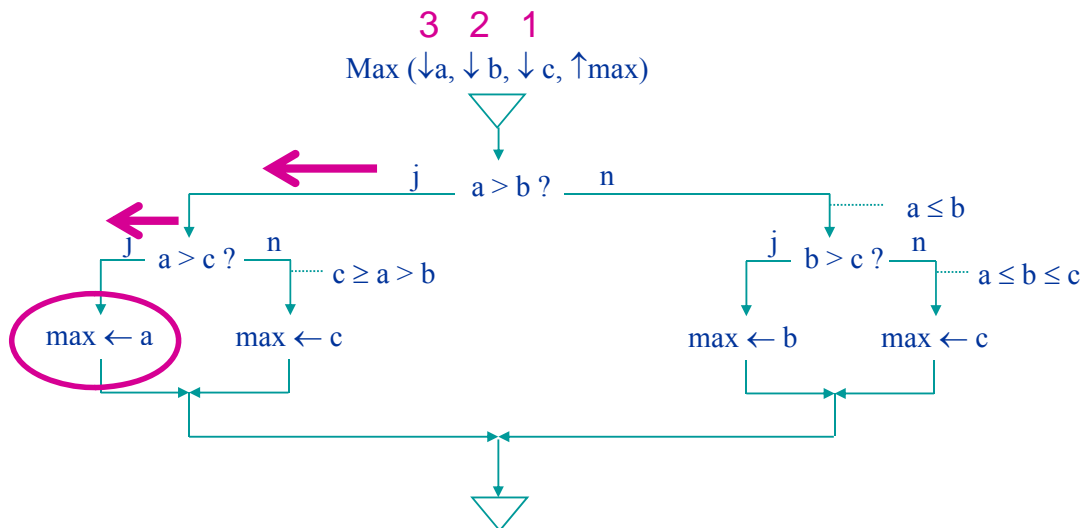
x	y	h
3	2	3
2	3	



Beispiel: Maximum dreier Zahlen bestimmen

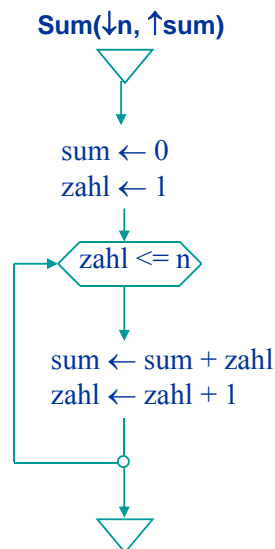


Beispiel: Maximum dreier Zahlen bestimmen



Beispiel: Summe

- Summe von 1 bis n



Schreibtischtest ($n = 5$)

	sum	zahl
0	0	1
1	1	2
2	3	3
3	6	4
4	10	5
5	15	6



Algorithmen und Programmierung

Programm

Ein für die Lösung einer bestimmten Aufgabe mit einer Datenverarbeitungsanlage geeigneter Algorithmus

d.h.

- Elementare Operationen sind die durch den Computer ausführbaren Befehle
- Programm muss in einer für den Computer lesbaren Form definiert sein (\implies Programmiersprache)

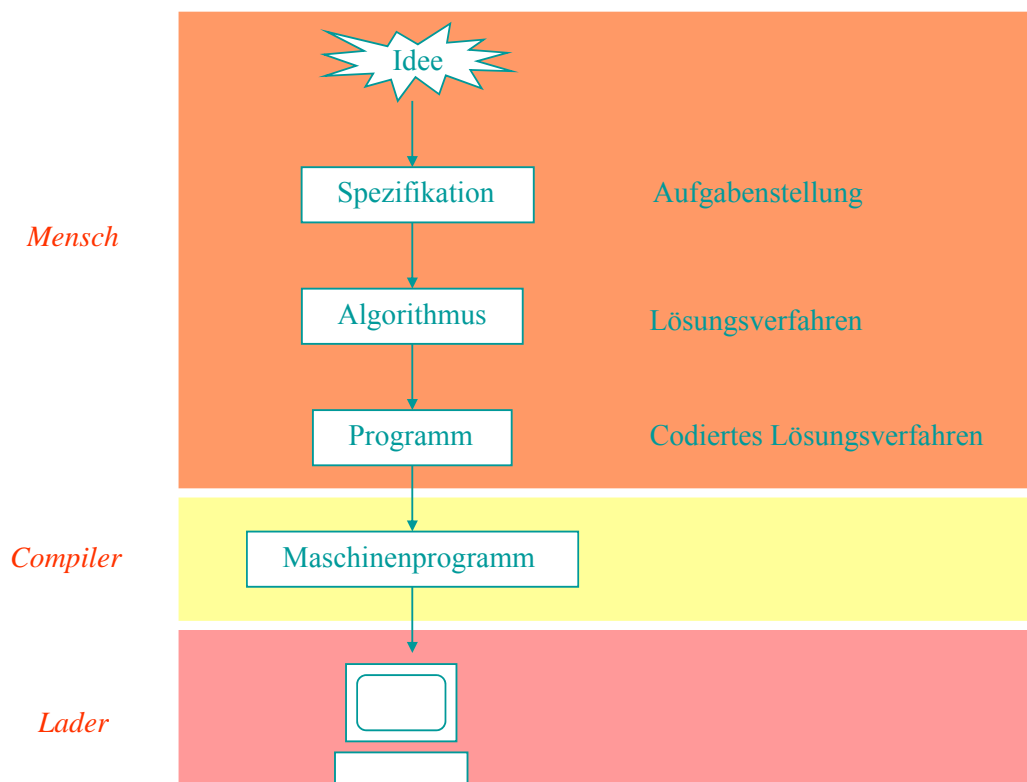


Höhere Programmiersprachen

- Höhere Programmiersprachen dienen der Formulierung von Programmen
- Sind für den Computer lesbar und verarbeitbar
 - können vom Computer gelesen werden
 - können vom Computer in ein ausführbares Programm übersetzt werden (*Compiler*)
- für den Menschen wesentlich einfacher lesbar und schreibbar als Maschinensprache
 - Programmieren wird wesentlich erleichtert
 - Programme können vom Experten gelesen und verstanden werden



Programmerstellung



Geschichte der Programmiersprachen

Jahr	Sprache	Bemerkung
1957	Fortran	technische Anwendungen
1960	Algol-60	"Algol-Familie"
	Cobol	kaufmännische Anwendungen
1962	Basic	Anfänger-Sprache
	APL	Vektoren und Matrizen
	Lisp	funktionale Sprache
1965	PL/I	Allzwecksprache
1968	Algol-68	Nachfolger von Algol-60
1971	Pascal	Einfluß auf spätere Sprachen
1972	Prolog	Wissen und Schlußregeln
1973	C	"Unix-Sprache"
1980	Modula-2	Nachfolger von Pascal
	Ada	Allzwecksprache; DoD
	Smalltalk	objektorientierte Sprache
1983	C++	objektorientierte Erweiterung von C
1987	Oberon	Weiterentwicklung von Modula-2
1995	Java	Weiterentwicklung von C++; WWW
2000	C#	Weiterentwicklung von Java durch Microsoft



Programmiersprache Java

- Java ist eine höhere, problemorientierte Programmiersprache, die laut Sun durch folgende Eigenschaften charakterisiert ist:
 - Simple: einfacher als C++
 - Architecture-neutral: läuft auf Windows-Rechner als auch auf Unix/Linux, Mac
 - Object-oriented
 - Portable
 - Distributed
 - High-performance
 - Interpreted
 - Multithreaded
 - Robust
 - Secure
 - Dynamic



Java Development Kit installieren

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

Java SE Development Kit (JDK)

JDK 6 Update 16
This special release provides a few key fixes. » Learn more

Download

Docs ▾

Java SE Runtime Environment (JRE)

JRE 6 Update 16
This special release provides a few key fixes. » Learn more

Download

Docs ▾

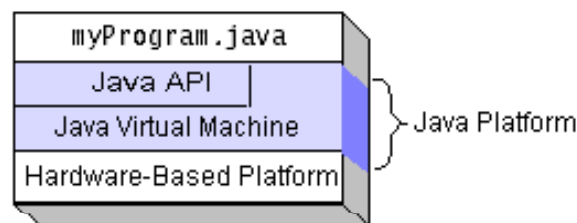


JOHANNES KEPLER UNIVERSITY LINZ



Java Runtime Environment (JRE)

- Die Java Virtual Machine (Java VM)
- Das Java Application Programming Interface (Java API)



Java Virtual Machine (JVM)

- Java-Programme nicht *direkt* auf dem Computer ausgeführt
- sondern durch **java-Programm**

Vorteile:

- Java-Programme laufen „kontrolliert“ in der JVM ab (erhöhte Sicherheit und Zuverlässigkeit)
- Kann auf jede Plattformen portiert werden.



JOHANNES KEPLER UNIVERSITY LINZ



Java Development Toolkit (JDK)

== Programmwerkzeuge zum Erstellen, Übersetzen und Testen von Java-Programmen

- Java Compiler ⇒ `javac`
 - Übersetzt den Quellcode in Java Byte-Code
- Java Debugger ⇒ `jdb`
 - Tool zur Fehlersuche
- Java Disassembler ⇒ `javap`
 - Erzeugt aus Byte-Code lesbaren Quellcode
- Weitere Tools
 - ⇒ `jar`, `javah`, `javadoc`, `javakey`, `serialver`, ...
- Java Dokumentation (separates Download)



Java Development Kit (JDK)

Erstellen und Ausführen von Java-Programmen.

Beispielprogramm HelloWorld

↓ Erstellen von Java-Programm
mit Editor
Quellcode (Sourcecode) in Java

↓ Java Compiler (**javac**)
Java Zwischencode (Byte Code)

↓ Java Runtime (**java**)
Programmausführung

↓ Erstellen von Programm
HelloWorld mit Editor
Datei **HelloWorld.java** mit Java-Source

↓ **javac HelloWorld.java**
Datei **HelloWorld.class** mit Byte-Code

↓ **java HelloWorld**
Ausführung von Programm **HelloWorld**



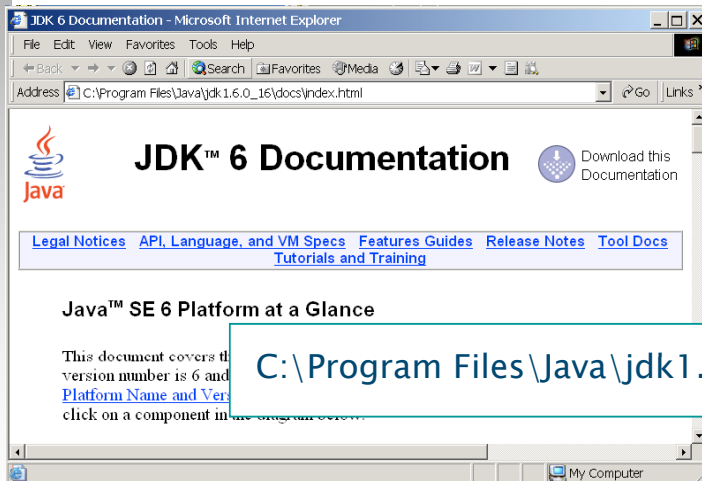
Java Dokumentation installieren

Java SE 6 Documentation

Download

Docs

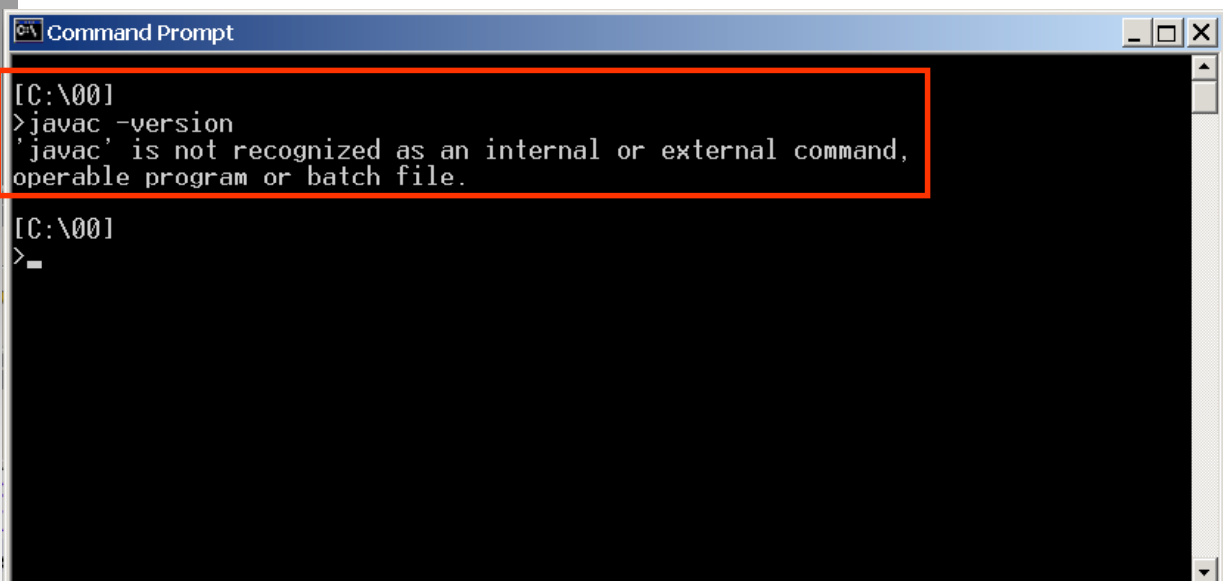
<http://java.sun.com/javase/downloads/>



JOHANNES KEPLER UNIVERSITY LINZ



Java Development Kit probieren



findet javac.exe nicht, weil JDK nicht im PATH



JOHANNES KEPLER UNIVERSITY LINZ



Java Development Kit probieren

```
Command Prompt

[C:\>]
>set path
Path=C:\WINNT\system32;C:\WINNT;C:\WINNT\System32\Wbem;c:\Program Files\Microsoft SQL Server\90\Tools\bin\;C:\PROGRAM\1\ATT\Graphviz\bin;C:\Program Files\Java\jdk1.6.0_16\bin;C:\Program Files\Subversion;C:\Program Files\TortoiseSVN\bin;C:\Documents and Settings\reinhard\bin;C:\Program Files\OpenVPN\bin;C:\Program Files\Microsoft Visual Studio 8\Common7\Tools;c:\Program Files\Subversion;C:\plux\Applications\Tutorial.MovieFinder\06-Sofa\cushion-0.6.1\bin;C:\plux\Applications\Tutorial.MovieFinder\06-Sofa\sofa-0.6.1\bin
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH

[C:\>]
>javac -version
javac 1.6.0_16

[C:\>]
>
```

Informationen

- Probleme beim Ausführen des Java-Compilers `javac` in der Eingabeaufforderung? Sie müssen die Path-Umgebung-Verzeichnis Ihrer JDK-Installation setzen. [Lesen Sie hier](#) wie man das in Windows 2000/XP macht.
- **Übungsmodus** für die Übungen
- **Stilrichtlinie** für die Übungen



Java Development Kit verwenden

```
HelloWorld.java - WordPad
File Edit View Insert Format Help

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

Command Prompt

[C:\00]
>dir
HelloWorld.java

[C:\00]
>javac HelloWorld.java

[C:\00]
>dir
HelloWorld.class
HelloWorld.java

[C:\00]
>java HelloWorld
Hello World!

[C:\00]
>
```

Übersetzen

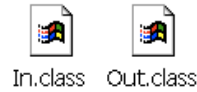
Ausführen



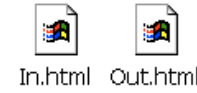
Daten einlesen und ausgeben

Zusatzinformationen

In, Out Paket
Dokumentation zu In und Out
Folien aus erster Stunde



Bytecode



Dokumentation



Quelltext

```
HelloWorld.java - WordPad
File Edit View Insert Format Help

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

Out.println("Hello World!");
```

Out.print(5)
Out.println("GdP")

In.readInt()
In.readWord()
In.readString()

Text-Editor Notepad++

<http://notepad-plus.sourceforge.net/>

```
C:\00\HelloWorld.java - Notepad++
File Edit Search View Format Language Settings Macro Run Window ?

HelloWorld.java

1 public class HelloWorld {
2     public static void main(String[] args) {
3         System.out.println("Hello World!");
4     }
5 }
```

nb char : 118 nb line : 5 Ln : 5 Col : 2 Sel : 0 Dos\Window ANSI INS

- Problem erfassen und beschreiben
- Lösungsidee erarbeiten und niederschreiben
- Lösungsidee in einen schrittweisen Ablauf überführen (= Algorithmus)
- Algorithmus so weit verfeinern und konkretisieren, dass Umsetzung in Java direkt möglich ist
- Programmierung in Java
- Überlegen von Testfällen (dabei alle möglichen Sonderfälle, Grenzfälle betrachten)
- Testen und Verbessern
- Dokumentieren der Ergebnisse

