

Grundlagen der Programmierung UE



Vortragende

Dr. Herbert Prähofer (G1)
Mag. Reinhard Wolfinger (G2 u. G3)

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

Email: herbert.praehofer@jku.at (Telefon 7132)
reinhard.wolfinger@ase.jku.at (Telefon 7134)

http: www.ssw.uni-linz.ac.at

Raum: HF 306 und 304 (Hochschulfondgebäude 3. Stock)



<http://www.ssw.uni-linz.ac.at/Teaching/Lectures/GdP/2008/index.html>

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



Bewertung

Ü1	1	12
	2	x
Ü2	1	x
	2	10
Ü3	1	x
	2	12
Ü4	1	x
	2	11
Ü5	1	x
	2	11
Ü6	1	x
	2	5
Ü7	1	x
	2	8,5
Ü8	1	
	2	9
Ü9	1	7,5
	2	2
Ü10	1	
	2	
Abgabe Schnitt		85% 20,3
Test/Pkt. Test/Note		17,5 3
Gesamt/Pkt. Gesamt/Note		18,9 2

- Pro Übung zwei Teilaufgaben mit je 12 Punkten
- Eine Teilaufgabe wird bewertet
- Mind. 80% abgeben
- Die besten 80% ergeben den Punkteschnitt
- Gesamtnote =
50% Übungspunkte
50% Test
beides muss positiv sein



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)
- Sind aber auch für den Menschen wesentlich einfacher lesbar und schreibbar als Maschinensprache
 - Programmieren wird wesentlich erleichtert
 - Programme können vom Experten gelesen und verstanden werden



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 Virtual Machine

- Java-Programme werden nicht *direkt* auf dem Computer ausgeführt sondern von einem speziellen Programm, nämlich der

Java Virtual Machine (JVM)

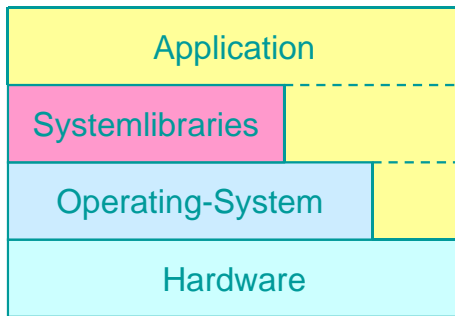
Vorteile:

- Java-Programme laufen „kontrolliert“ in der JVM ab
 - Erhöhte Sicherheit und Zuverlässigkeit
- Kann auf jeder Plattformen portiert werden.
 - Vom Super Computer bis zu Handys
 - Von Unix bis Windows-CE
 - Selbes Programm läuft (mit Einschränkungen) auf alle diesen Plattformen ohne nochmals übersetzt oder portiert zu werden, da sich die VM auf jeder Plattform gleich verhält



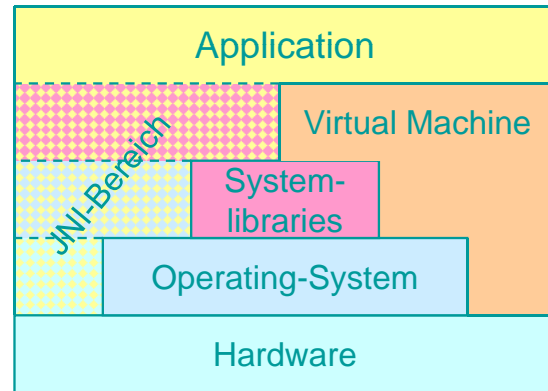
Unterschied normale Programmausführung ⇒ Virtuelle Maschine

normale Programmausführung



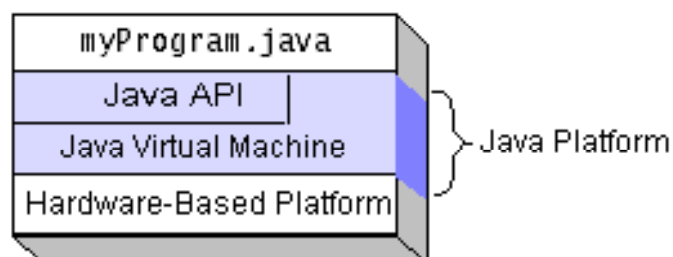
⇒

Virtuelle Maschine



Java Runtime Environment (JRE)

- Eine Plattform ist die Hard- und Softwareumgebung auf der ein Programm läuft.
- Die meisten Plattformen bestehen aus Hardware und Betriebssystem.
- Die Java-Plattform unterscheidet sich von den meisten anderen Plattformen dadurch, dass sie eine reine Software-Plattform ist, die für verschiedenste Hardware und Betriebssysteme verfügbar ist.
- Die Java Runtime Environment besteht aus zwei Teilen:
 - Die Java Virtual Machine (Java VM)
 - Das Java Application Programming Interface (Java API)



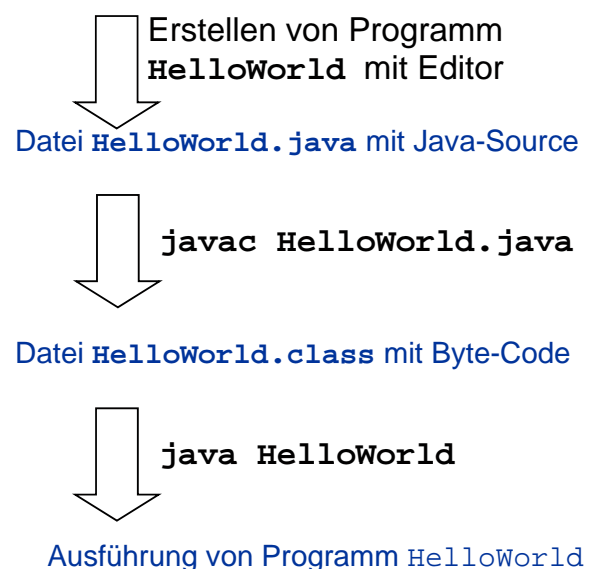
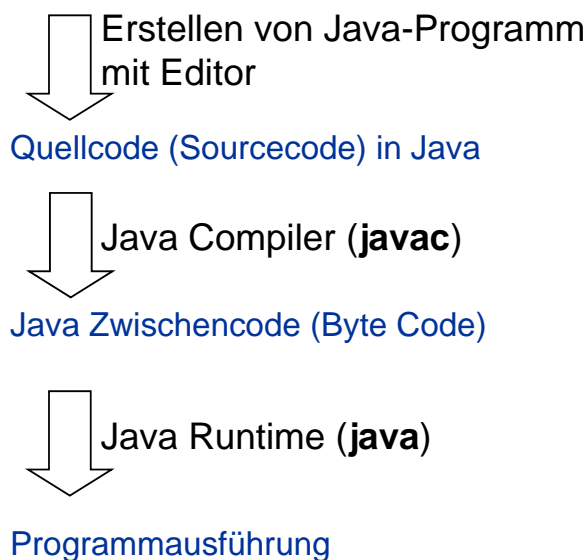
== Programmwerkzeuge zum Erstellen, Übersetzen und Testen von Java-Programmen

- 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)



Erstellen und Ausführen von Java-Programmen

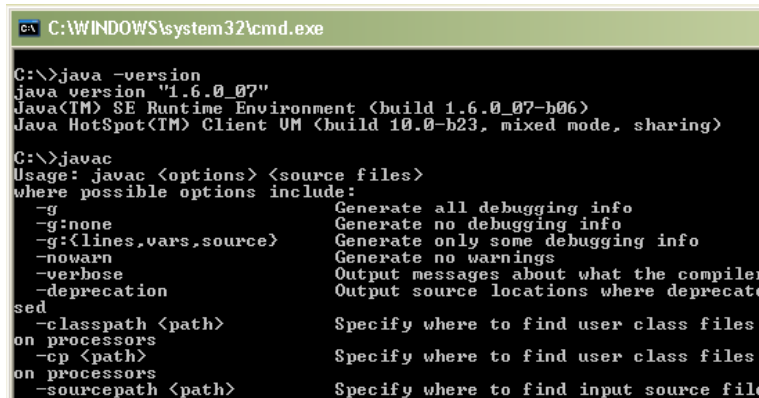
Beispielprogramm HelloWorld



- JDK (Java Development Kit)
- Download von www.java.sun.com
- Installation
- Umgebungsvariable
 - Path = ...;C:\Programme\Java\jdk1.6.0_06\bin
- Testen in Eingabeaufforderung (Konsole, CommandLine)

```
java -version
```

```
javac
```



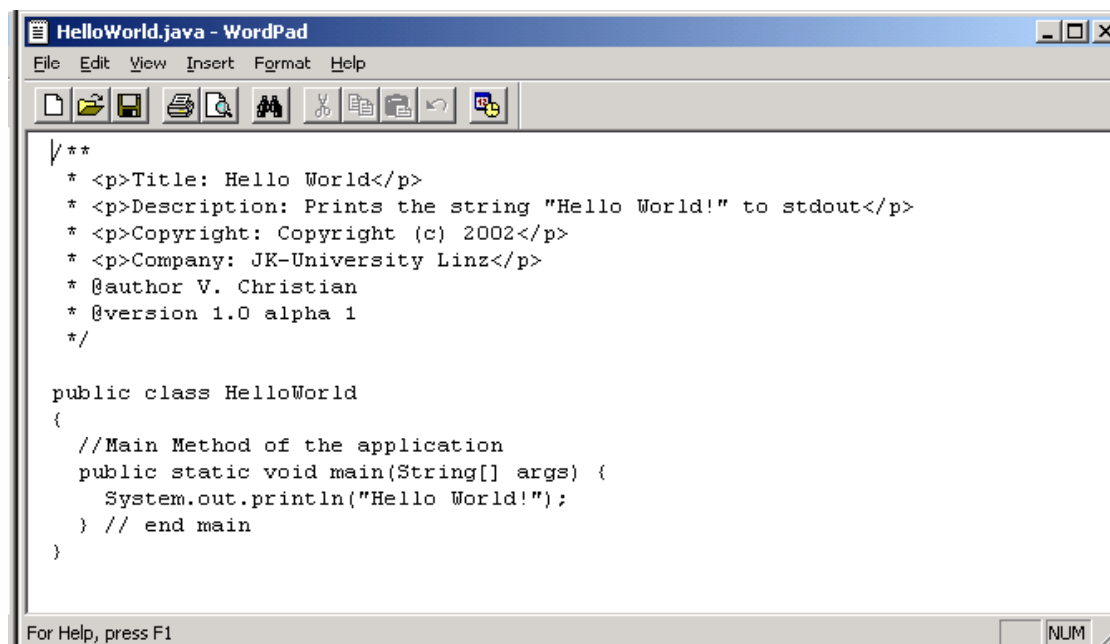
```
C:\WINDOWS\system32\cmd.exe

C:\>java -version
java version "1.6.0_07"
Java(TM) SE Runtime Environment (build 1.6.0_07-b06)
Java HotSpot(TM) Client VM (build 10.0-b23, mixed mode, sharing)

C:\>javac
Usage: javac <options> <source files>
where possible options include:
-g          Generate all debugging info
-g:none    Generate no debugging info
-g:<lines,vars,source> Generate only some debugging info
-nowarn    Generate no warnings
-verbose   Output messages about what the compiler is doing
-deprecation Output source locations where deprecated code was found
-sed
-classpath <path> Specify where to find user class files
-processorpath <path> Specify where to find user class files
-cp <path> Specify where to find user class files
-processorpath <path> Specify where to find input source files
-sourcepath <path> Specify where to find input source files
```



Programm schreiben (Wordpad)



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

/**
 * <p>Title: Hello World</p>
 * <p>Description: Prints the string "Hello World!" to stdout</p>
 * <p>Copyright: Copyright (c) 2002</p>
 * <p>Company: JK-University Linz</p>
 * @author V. Christian
 * @version 1.0 alpha 1
 */

public class HelloWorld
{
    //Main Method of the application
    public static void main(String[] args) {
        System.out.println("Hello World!");
    } // end main
}

For Help, press F1
```



Programm kompilieren (javac)

```
C:\WINNT\System32\cmd.exe

C:\Documents and Settings\voc\HelloWorld>dir
Volume in drive C has no label.
Volume Serial Number is 74FF-D96B

Directory of C:\Documents and Settings\voc\HelloWorld
30.09.2002  16:34         <DIR>          .
30.09.2002  16:34         <DIR>          ..
30.09.2002  16:15             392 HelloWorld.java
                392 bytes
                1 File(s)
                2 Dir(s)  2.759.196.672 bytes free

C:\Documents and Settings\voc\HelloWorld>javac HelloWorld.java

C:\Documents and Settings\voc\HelloWorld>dir
Volume in drive C has no label.
Volume Serial Number is 74FF-D96B

Directory of C:\Documents and Settings\voc\HelloWorld
30.09.2002  16:35         <DIR>          .
30.09.2002  16:35         <DIR>          ..
30.09.2002  16:35             426 HelloWorld.class
30.09.2002  16:15             392 HelloWorld.java
                818 bytes
                2 File(s)
                2 Dir(s)  2.759.196.672 bytes free

C:\Documents and Settings\voc\HelloWorld>
```



Programm starten (java)

```
C:\WINNT\System32\cmd.exe

C:\Documents and Settings\voc\HelloWorld>dir
Volume in drive C has no label.
Volume Serial Number is 74FF-D96B

Directory of C:\Documents and Settings\voc\HelloWorld
30.09.2002  16:35         <DIR>          .
30.09.2002  16:35         <DIR>          ..
30.09.2002  16:35             426 HelloWorld.class
30.09.2002  16:15             392 HelloWorld.java
                818 bytes
                2 File(s)
                2 Dir(s)  2.759.192.576 bytes free

C:\Documents and Settings\voc\HelloWorld>java HelloWorld
Hello World!

C:\Documents and Settings\voc\HelloWorld>
```



In und Out

- Die Programme In und Out unterstützen eine einfache Ein- und Ausgabe
- Dateien (verfügbar von der Homepage der LVA)
 - In.java
 - Out.javains gleich Verzeichnis wie Programmdatei
- Mit Methoden wie
 - In.read()
 - In.readLine()
 - In.readWord()können Werte eingelesen haben
- Mit Methoden wie
 - Out.print()
 - Out.println()können Werte ausgegeben haben

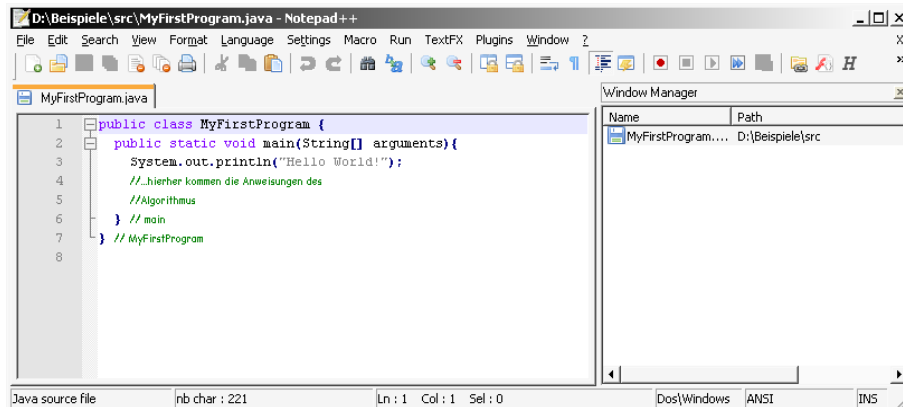


Java-Programm "Sum" (Sum.java)

```
public class Sum {  
  
    public static void main(String[] args) {  
        int sum = 0;  
        int number;  
        Out.print("Bitte Zahlen eingeben: ");  
        number = In.readInt();  
        while (number > 0) {  
            sum = sum + number;  
            number = In.readInt();  
        }  
        Out.print("Die Summe ist:");  
        Out.println(sum);  
    }  
}
```



- Windows Editor ist etwas „unbequem“
- Notepad++ als einfaches Entwicklungswerkzeug.
- Download von <http://notepad-plus.sourceforge.net/>
- Datei `MyFirstProgram.java` öffnen und auszugebenden Text ändern.
- Notepad++ bietet gutes "Syntax-Highlighting"



- Ausführen von beliebigen Programmen ist per Knopfdruck möglich
- 1) .bat Dateien im Installationsverzeichnis von Notepad++ erzeugen:

CompileJava.bat:

```
javac %1  
pause
```

RunJava.bat:

```
java %1  
pause
```

Menü Ausführen->Ausführen (F5)

\$(NPP_DIRECTORY)\CompileJava.bat \$(FILE_NAME)

Speichern als Kommando **CompileJava** für Taste F9

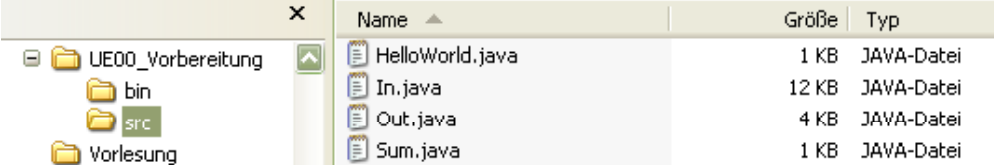
Menü Ausführen->Ausführen (F5)

\$(NPP_DIRECTORY)\RunJava.bat \$(NAME_PART)

Speichern als Kommando **RunJava** für Taste F10



- java- und .class-Dateien in unterschiedlichen Verzeichnissen verwalten
 - Damit bleibt Verzeichnis mit java-Dateien übersichtlich
 - Vor allem bei großen Projekten bzw. vielen Quellcode-Dateien
- Erstellen Sie zwei Verzeichnisse (bin und src)



Name	Größe	Typ
HelloWorld.java	1 KB	JAVA-Datei
In.java	12 KB	JAVA-Datei
Out.java	4 KB	JAVA-Datei
Sum.java	1 KB	JAVA-Datei

1) .bat Dateien im Installationsverzeichnis von Notepad++ folgend:

CompileJava.bat:

```
javac -d ..\bin %1  
pause
```

RunJava.bat:

```
java -cp ..\bin %1  
pause
```

