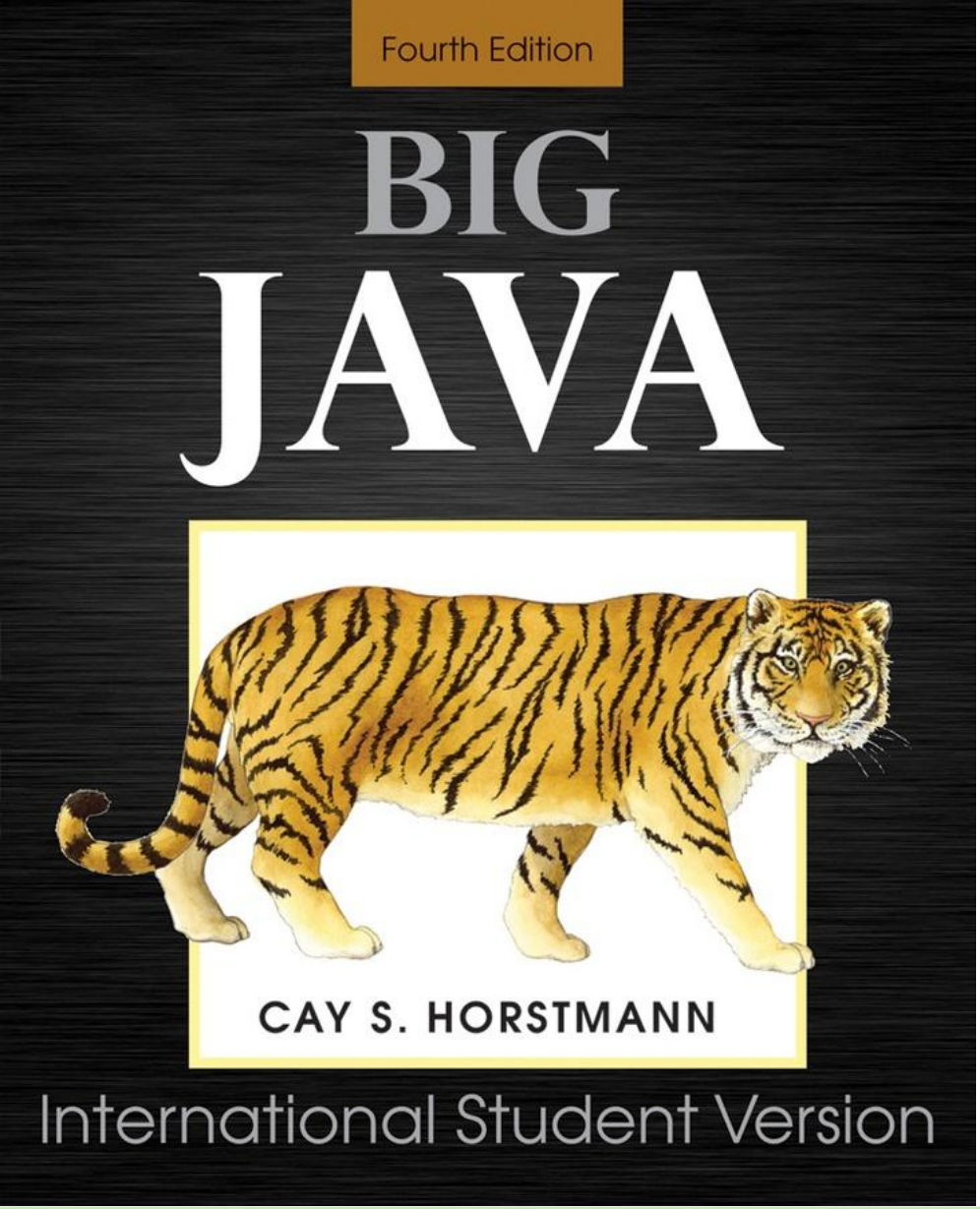


Kapitel 1 – Einführung

BFH, M. Hudritsch based on Big Java by Cay Horstmann

Horstmann: Big Java

bcs.wiley.com/he-bcs/Books?action=index&itemId=047055309X&bcsId=5653

Other bookmarks

STUDENT COMPANION SITE

Horstmann: *Big Java 4th Edition for Java 7 and 8, International Student Version*

WILEY HIGHER EDUCATION

Wiley Home | Higher Education Home | Title Home | Student Companion Site Home

Tech Support Contact Us

Browse by Chapter

Select a Chapter

Browse by Resource

- Source Code
- Worked Examples

Student Companion Site

Horstmann: *Big Java 4th Edition for Java 7 and 8, International Student Version*

Toolbox

- How to use this site
- Table of contents
- Site map

Press ctrl-D to bookmark this page

Welcome

Welcome to the Web site for ***Big Java 4e for Java 7 and 8 International Student Version*** by **Cay S. Horstmann**. This Web site gives you access to the rich tools and resources available for this text. You can access these resources in two ways:

1. Using the drop-down menu at the top left, select a chapter. A list of resources available for that particular chapter will be provided.

2. Click the name of the resource you wish to use in the left navigation menu. This will allow you to access a particular resource section. You will then have the option of selecting resources within the section or going directly to a specific chapter.

Kapitelziele:

- Die Aktivität der Programmierung verstehen
- Einblick in die Architektur eines Computers
- Maschinensprache gegenüber Hochsprachen
- Compiler und Entwicklungsumgebung kennenlernen
- Ein erstes JAVA-Programm erstellen
- Syntax- und Laufzeitfehler kennenlernen
- Pseudo-Code eines einfachen Algorithmus schreiben

Was ist Programmierung?

- Computer sind programmiert um Aufgaben zu lösen
- Verschieden Aufgaben = verschiedene Programme
- Programm
  - *Abfolge von einfachen Operationen um eine Aufgabe zu lösen*
- Anspruchsvolle Programme werden von einem Team hochausgebildeter Programmierern entwickelt.
- Programmieren kann schnell eine anspruchsvolle intellektuelle Arbeit werden.

Eigenkontrolle 1.1

Was wird benötigt, um eine Musik-CD auf dem Computer abzuspielen?

Antwort: Ein Programm, das die Daten von der CD liesst, verarbeitet und die Ausgabe über die Soundkarte an die Lautsprecher schickt.

Eigenkontrolle 1.2

Warum ist ein CD-Player weniger flexibel als ein Computer?

Antwort: Ein CD-Player kann nur CDs abspielen. Er kann keine anderen Programme laufen lassen.

Eigenkontrolle 1.3

Kann ein Computer-Programm Aufgaben besser lösen, als es der Programmierer vorgesehen hat?

Antwort: Nein – Ein Computer ist dumm und kann nur Programmschritte machen, die der Programmierer vorgesehen hat.

Maschinen-Code

- Maschinen-Code hängt vom Prozessortyp ab
- Befehle für die virtuelle Java Maschine (JVM) kann auf allen Maschinen ausgeführt werden.
- Eine typische Sequenz von Maschinensprachbefehlen lautet z.B.:

1. *Lade den Inhalt der Speicheradresse 40 in das Register A.*
2. *Lade die Zahl 100 in das Register B.*
3. *Ist der Wert in Register A grösser als der Wert in Register B, dann fahre mit dem Programm weiter an der Speicheradresse Y*

Maschinen-Code

- Maschinebefehle sind als Nummern gespeichert:

```
21 40
16 100
163 240
```

- Ein Compiler übersetzt Hochsprachen in Maschinen-Code

Eigenkontrolle 1.6

Wie sieht der Code für die JVM aus: “Lade den Inhalt der Speicheradresse 100 in Register A”?

Antwort: `21 100`

Anmerkung MH: Das stimmt sicher nicht.

Der Java Compiler erzeugt einen Zwischen-Code (Byte-Code) welcher dann auf der VM in Maschinen-Code compiliert und optimiert wird. Erst dann gelangt der Maschinen-Code auf der CPU zur Ausführung.

Eigenkontrolle 1.7

Braucht eine Person, die den Computer für die Büroarbeit braucht je einen Compiler?

Antwort: Nein – ein Compiler wird nur von Programmierern gebraucht, um Programme in Hochsprachen in Maschinen-Code zu übersetzen.

Die Java Programmiersprache

- Einfach
- Sicher
- Plattform-unabhängig (“write once, run anywhere”)
- Viele Bibliotheken (Libraries)
- Wurde ursprünglich von Sun Microsystems für mobile Geräte entwickelt.

Applet auf einer Web-Seite

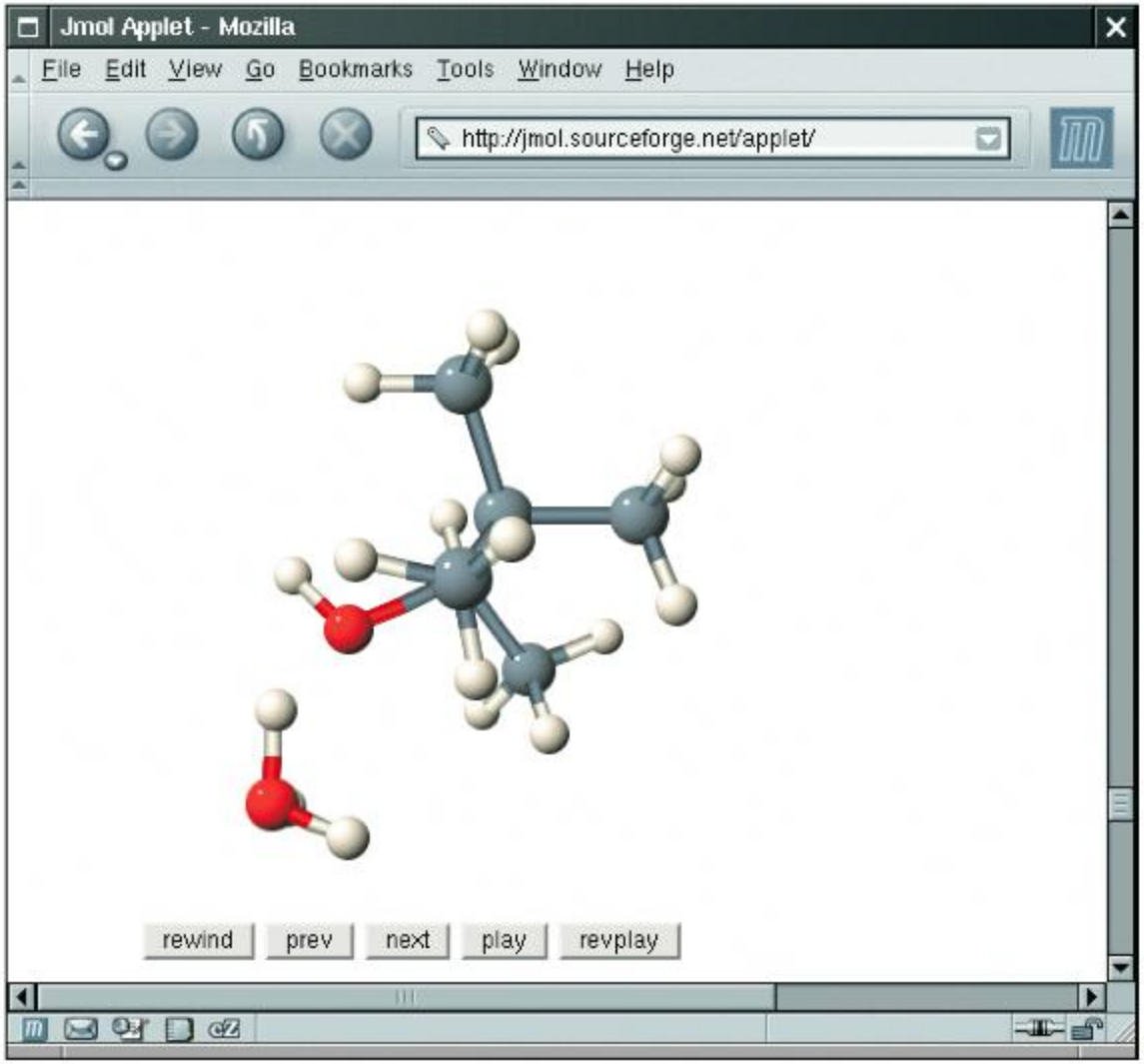


Figure 6 An Applet for Visualizing Molecules Running in a Browser (*http://jmol.sourceforge.net/applet/*)

Java Versionen

Version	Year	Important New Features
1.0	1996	First release
1.1	1997	Inner classes
1.2	1998	Swing, Collections
1.3	2000	Performance enhancements
1.4	2002	Assertions, XML
5	2004	Generic classes, enhanced for loop, auto-boxing, enumerations
6	2006	Library improvements
7	2010	Small language changes and library improvements
8	2014	Lambda Expressions, new build system, new profiles, ...

Eigenkontrolle 1.9

Wie lange dauert es die ganze Java Bibliothek zu lernen?

Antwort: Niemand kann die ganze Bibliothek lernen, sie ist zu gross.

ch01/hello/HelloPrinter.java

```
1  public class HelloPrinter
2  {
3     public static void main(String[] args)
4     {
5        // Display a greeting in the console window
6
7        System.out.println("Hello, World!");
8     }
9  }
```

Programmausgabe in der Konsole:

```
Hello, World!
```

Die Struktur eines Programms: Klassendeklaration

- Klassen sind die fundamentalen Teile eines Java-Programms:

  `public class HelloPrinter`

  Startet eine neue Klasse

- Jede Programmdatei muss min. eine **public** Klasse enthalten
- Der Klassenname muss gleich sein wie der Dateiname:
  - *Klasse* `HelloPrinter` *muss in der Datei* `HelloPrinter.java` *sein.*

Die Struktur eines Programms: Die `main` Methode

- Jede Java-Applikation muss EINE öffentliche Klasse mit EINER statischen main() Methode haben.
  - *Wenn eine Applikation startet, wird diese Methode aufgerufen.*
  - *Falls dem Programm Parameter übergeben werden, so erhält die Methode main diese im String Array args:*

```
public static void main(String[] args)
{
   . . .
}
```

Gibt nichts zurück

noch von der Kommandozeile mit Parameter

Immer auf ENGLISCH schreiben

Die Struktur eines Programms: Kommentare

- Die erste Linie in der main Methode ist ein Kommentar:

```
// Display a greeting in the console window
```

- Compiler ignorieren allen Text hinter // und zwischen /* … */
- Kommentare sollen dem Programmierer helfen den Code zu verstehen.

Die Struktur eines Programms: Anweisungen (Statements)

- Der Block der main Methode enthält Anweisungen zwischen geschwungenen Klammern (`{}`)
- Jede Anweisung endet mit einem Semikolon (`;`)
- Anweisungen werden eine nach der anderen ausgeführt.
- Die main Methode hat nur eine Anweisung:

```
System.out.println("Hello, World!");
```

Welche den Text auf der Konsole ausgibt:

```
Hello, World
```

Die Struktur eines Programms: Methodenaufruf

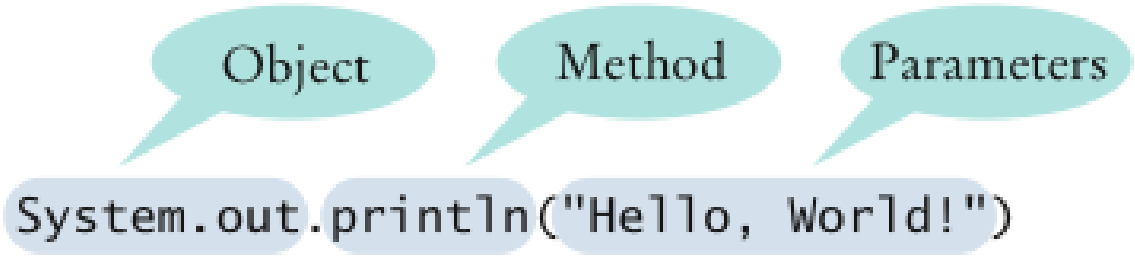
- `System.out.println("Hello, World!");`

  ist Methodenaufruf

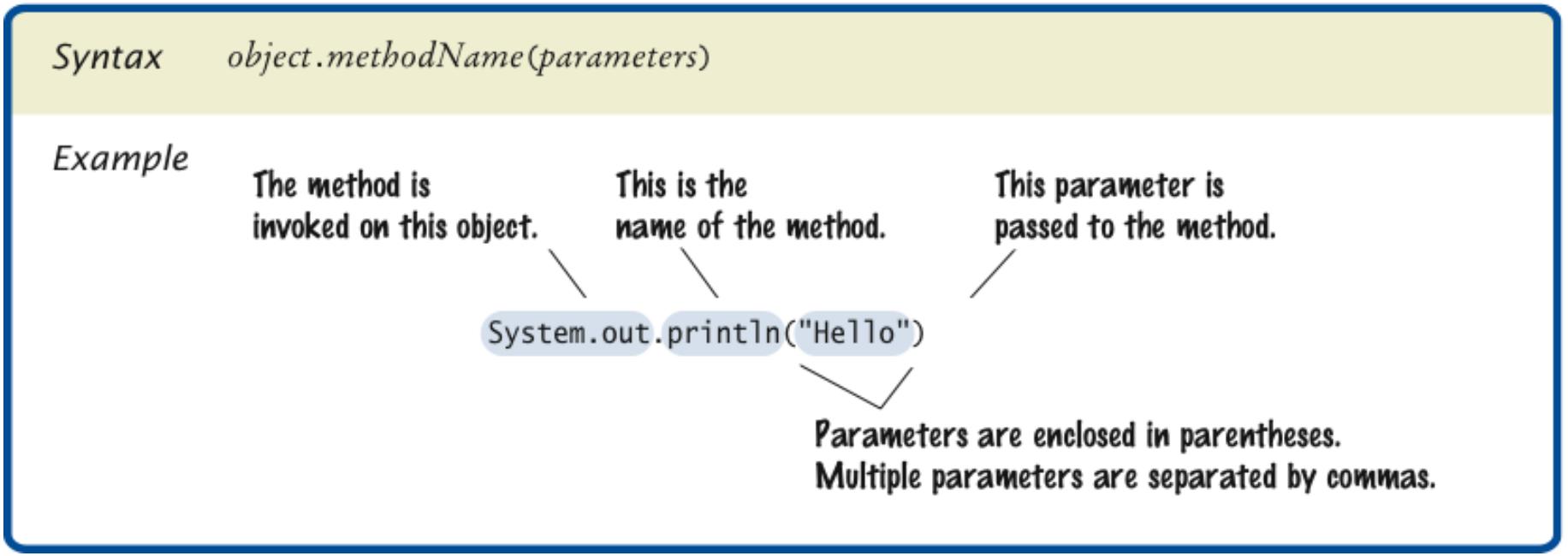
- Ein Methodenaufruf braucht:

  1. Ein *Objekt, dem die Methode gehört* (in diesem Fall `System.out`)
  2. Den *Namen der Methode* (in diesem Fall `println`)
  3. *Werte für Parameter* die die Methode braucht (in diesem Fall `"Hello, World!"`)

Figure 7
Calling a Method



Syntax 1.1 Methodenaufruf



Die Struktur eines Programms: Strings

- **String:** Eine Zeichenfolge umgeben von Anführungszeichen:

```
"Hello, World!"
```

Eigenkontrolle 1.10

Wie würden Sie das `HelloPrinter` Programm abändern, damit die Wörter `"Hello,"` und `"World!"` auf zwei Zeilen erscheinen?

Antwort:

```
System.out.println("Hello,");
System.out.println("World!");
```

oder

```
System.out.println("Hello,\nWorld!");
```

println macht versteckt ein new line

Eigenkontrolle 1.12

Was ergeben folgende Anweisungen?

print keine new line

```
System.out.print("My lucky number is");
System.out.println(3 + 4 + 5);
```

Antwort: Die Ausgabe wäre:

```
My lucky number is12
```

überladene Methode,
für alle Ausgabenformate (String, int, double etc.) eine Version

Editieren eines Java Programms

- Mit jedem Programm das Textdateien schreiben kann.
- Java ist case sensitive
- Achten Sie auf ein sauberes Text-Layout des Programm

Installation von Java SE Development Kit 7 (JDK)

http://www.oracle.com/technetwork/java/javase/downloads

Java SE Development Kit 7u40

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

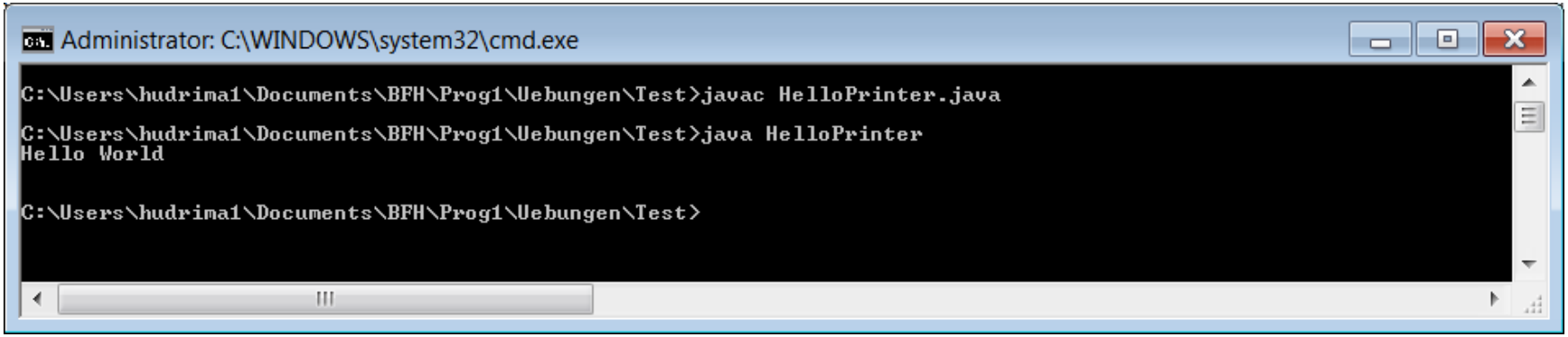
○ Accept License Agreement ◉ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM v6/v7 VFP Hard Float ABI	67.62 MB	jdk-7u40-linux-arm-vfp-hflt.tar.gz
Linux ARM v6/v7 VFP Soft Float ABI	67.62 MB	jdk-7u40-linux-arm-vfp-sflt.tar.gz
Linux x86	115.55 MB	jdk-7u40-linux-i586.rpm
Linux x86	132.83 MB	jdk-7u40-linux-i586.tar.gz
Linux x64	116.83 MB	jdk-7u40-linux-x64.rpm
Linux x64	131.63 MB	jdk-7u40-linux-x64.tar.gz
Mac OS X x64	183.35 MB	jdk-7u40-macosx-x64.dmg
Solaris x86 (SVR4 package)	139.84 MB	jdk-7u40-solaris-i586.tar.Z
Solaris x86	95.29 MB	jdk-7u40-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.43 MB	jdk-7u40-solaris-x64.tar.Z
Solaris x64	16.17 MB	jdk-7u40-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	139.06 MB	jdk-7u40-solaris-sparc.tar.Z
Solaris SPARC	98.07 MB	jdk-7u40-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.74 MB	jdk-7u40-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.18 MB	jdk-7u40-solaris-sparcv9.tar.gz
Windows x86	123.46 MB	jdk-7u40-windows-i586.exe
Windows x64	125.25 MB	jdk-7u40-windows-x64.exe

- Sie brauchen keine JRE, diese ist im JDK enthalten.

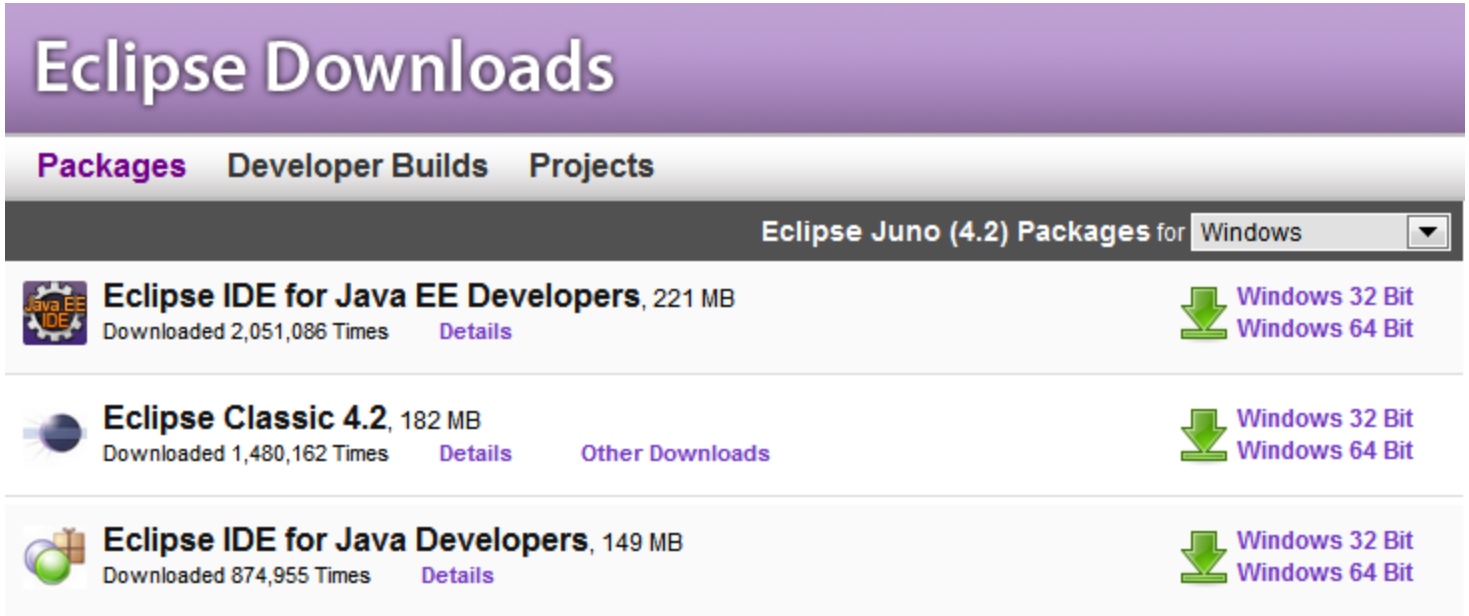
Compilieren und starten eines Java Programms

- Der Java Compiler übersetzt den Quellcode in Class-Dateien
- Eine Class-Datei hat die Erweiterung `.class`
- Der Compiler erzeugt keine Class-Datei, wenn er einen Fehler gefunden hat.
- Die JVM wird mit dem Befehl `java HelloPrinter` gestartet:

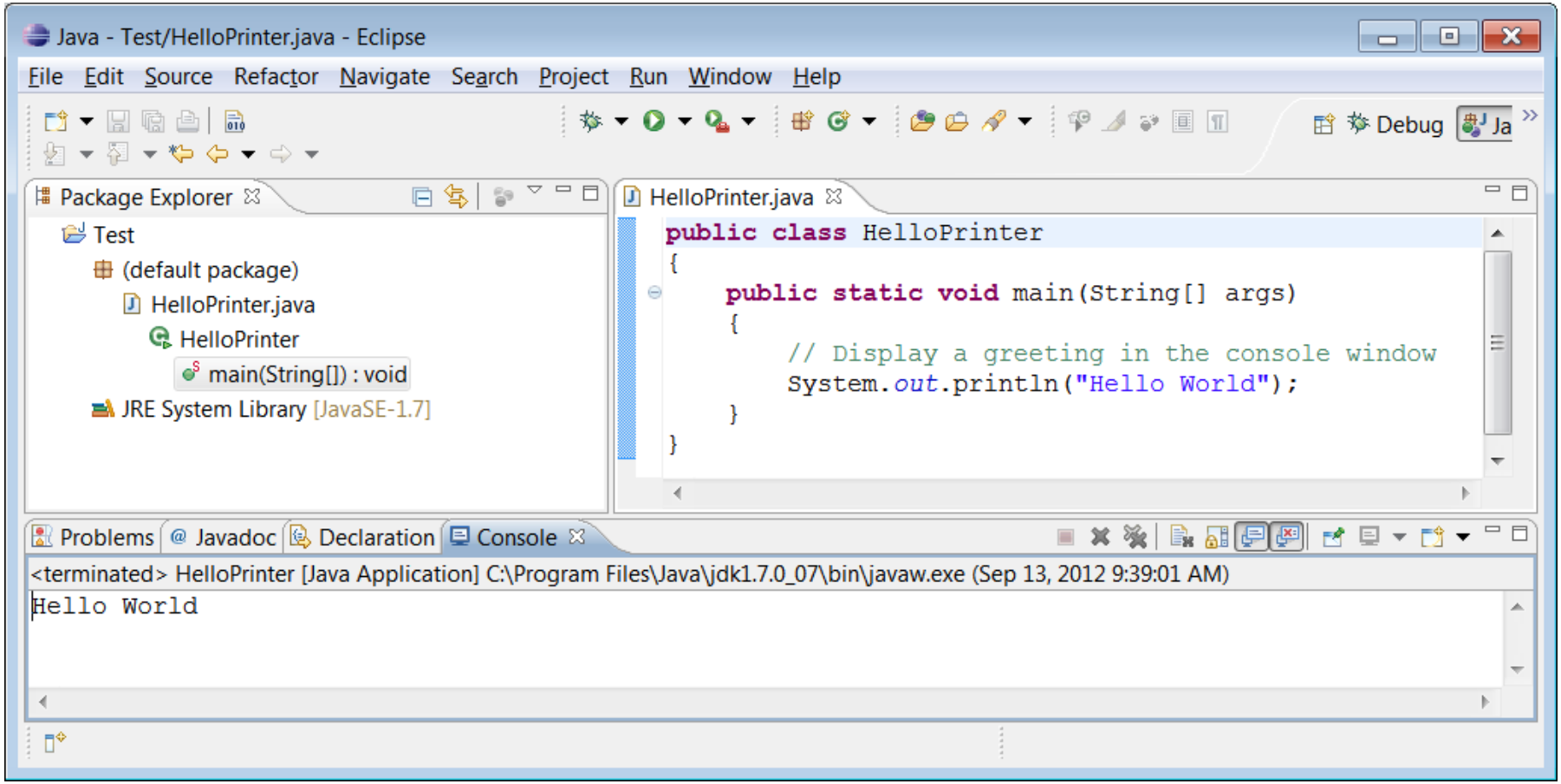


Installieren Sie die Eclipse for Java (EE) Developers

http://www.eclipse.org/downloads/



`HelloPrinter` in der Eclipse IDE



Vom Quellcode zum laufenden Programm

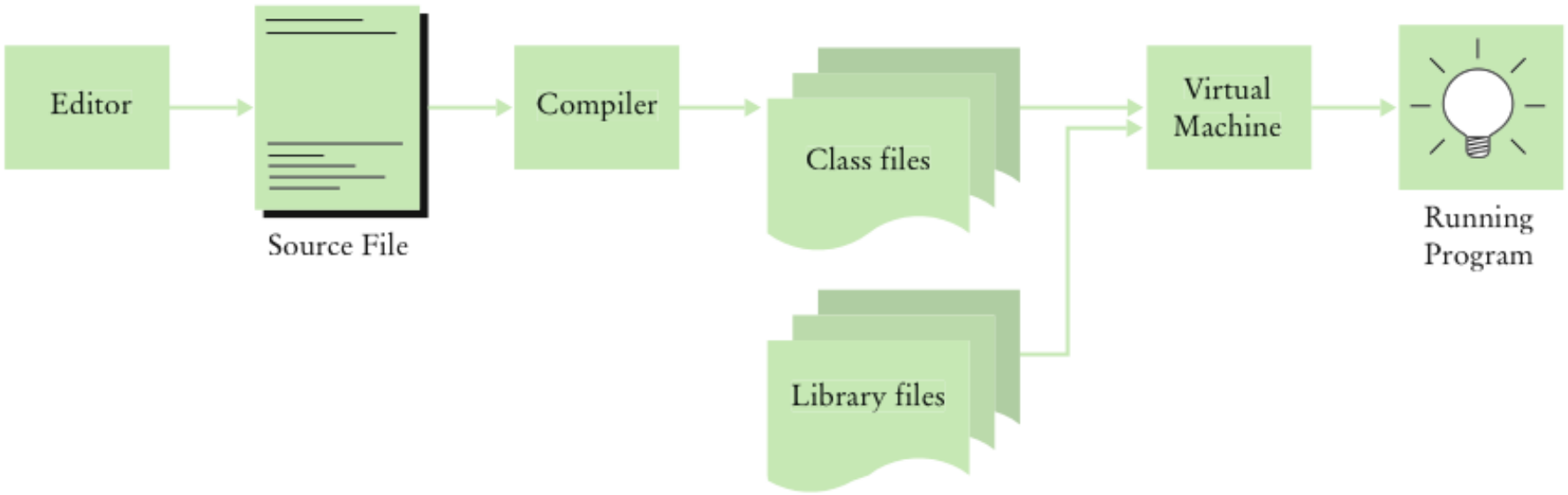


Figure 10 From Source Code to Running Program

Eigenkontrolle 1.13

Kann man mit MS Word ein Java Programm schreiben?

Antwort: Ja, wenn Sie die Datei als reine Textdatei speichern.

Fehler (Errors)

- **Fehler zur Compilier-Zeit (compile-time):** Entsteht wenn der Compiler eine Verletzung der Sprachsyntaxregeln feststellt:
  - *Beispiel*:

```
System.ou.println("Hello, World!);
```

  - *Syntaxfehler*
- **Fehler zu Laufzeit (run-time):** Alle anderen Fehler, die eine Weiterführung des Programms verunmöglichen.
  - *Beispiel:*

```
System.out.println("Hello, Word!");
System.out.println(1/0);
```

  - *Logischer Fehler (Division durch Null)*

Fehler-Management Strategie

- Lernen wie man klassische Fehler vermeidet
- Defensive Programmierstrategie um die Wahrscheinlichkeit von Fehlern zu reduzieren.
- Test- und Fehlerbehebungsmethoden (Debugging) lernen.

Eigenkontrolle 1.15

Angenommen Sie unterlassen die `//`-Zeichen im `HelloPrinter.java` Programm. Wird es ein Fehler zur Compilier-Zeit oder zur Laufzeit geben?

Antwort: Ein Syntaxfehler zur Compilier-Zeit. Der Compiler wüsste nicht was er mit dem Wort `Display` Anfangen soll.

Eigenkontrolle 1.16

Wenn ein Programm abstürzt, ist es ein Syntaxfehler oder ein Laufzeitfehler?

Antwort: Wenn ein Programm gestartet werden kann, ist es immer ein Laufzeitfehler.

Algorithmen

- **Algorithmen:** Eine Sequenz von Anweisungen:
  - *eindeutig*
  - *ausführbar*
  - *endlich*
- Algorithmus um zu entscheiden, welches Auto zu kaufen ist auf Grund der Kosten:

```
For each car, compute the total cost as follows:
   annual fuel consumed = annual miles driven / fuel efficiency
   annual fuel cost = price per gallon x annual fuel consumed
   operating cost = 10 x annual fuel cost
   total cost = purchase price + operating cost
If total cost1 < total cost2
   Choose car1
Else
   Choose car2
```

Pseudocode

- **Pseudocode:** Informelle Beschreibung eines Algorithmus:
  - *Wie berechnen sich die totalen Kosten:*

    total cost = purchase price + operating cost
  - *Entscheidungen und Repetitionen beschreiben:*

    For each car
      operating cost = 10 x annual fuel cost
      total cost = purchase price + operating cost

    Einrücken verwenden, um einen Block zu kennzeichnen
  - *Resultat aus- oder zurückgeben:*

    Choose car1

Programm Entwicklungsprozess

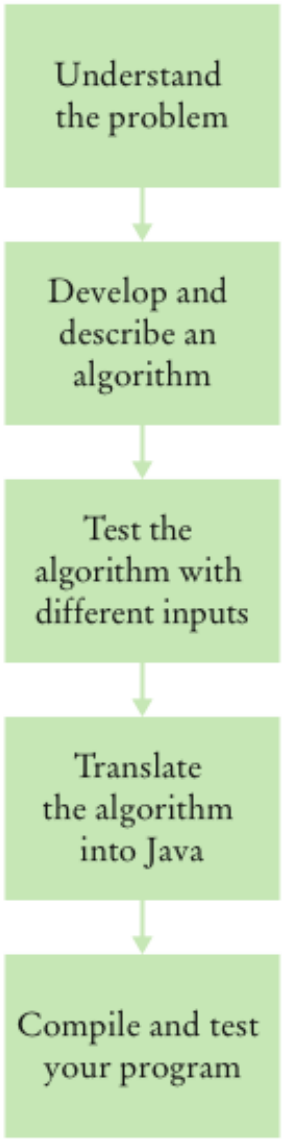


Figure 12
The Program Development Process

Eigenkontrolle 1.18

Zinsertrag: 10’000 Fr auf einem Konto erhalten 5% Zins pro Jahr. Wann hat sich der Kontostand verdoppelt?

Algorithmus:

Start with a year value of 0 and a balance of 10,000 Fr.
Repeat the following steps while the balance is less than 20,000 Fr..
  Add 1 to the year value.
  Multiply the balance value by 1.05 (a 5 percent increase).

Angenommen der Zins wäre 20%. Wann hat sich der Kontostand verdoppelt?

Antwort: 4 Jahre:

```
0 10,000
1 12,000
2 14,400
3 17,280
4 20,736
```

Übung zu Kapitel 1

- Schliessen Sie die Installationen ab.
- Besorgen Sie das Buch «*BigJava International Student Version*»
- Machen Sie sich mit der Eclipse IDE vertraut.
  - Workspace, Projekt, Import Java Files, Preferences
- Erstellen Sie eine schöne ASCII-Art Ausgabe mit System.out.println
- Machen Sie die Aufgaben P1.8 – P1.11