

INFORMATIK



Java™

INFORMATIK

Organisatorisches

Kennenlernen

- Martin Unold
 - Wissenschaftlicher Mitarbeiter am i3mainz – Institut für raumbezogene Informations- und Messtechnik
 - martin.unold@hs-mainz.de
 - Büro in c0.04
- Schreiben Sie eine Mail an martin.unold@hs-mainz.de
 - Betreff: Informatik
 - Inhalt: Matrikelnummer
Name
Gruppe

Ablauf

- Vorlesung
 - Freitag 8:15 – 10:45 in der Aula
- Übung GuV Gruppe A
 - Donnerstag 13:30 – 16:00 in c0.06
- Übung GuV Gruppe B
 - Dienstag 12:30 – 15:00 in c0.06
- Übung GuV Gruppe C
 - Mittwoch 11:45 – 14:15 in c0.06
- Übung DMidGK
 - Montag 9:00 – 11:30 in c0.06

Ablauf

- Vorlesung
 - 14.10. - 4.11., 18.11., 2.12., 16.12.
- Übung GuV Gruppe A
 - Python-Kurs: 10.11., 17.11., 24.11.
- Übung GuV Gruppe B
 - Python-Kurs: 8.11., 15.11., 22.11.
- Übung GuV Gruppe C
 - Python-Kurs: 9.11., 16.11., 23.11.
- Übung DMidGK
 - Python-Kurs: 7.11., 14.11., 21.11.

Anforderungen

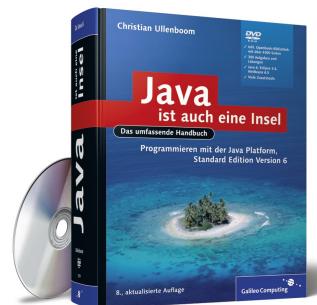
- Studienleistung
 - Anwesenheit und Mitarbeit in den Übungen
 - Höchstens zweimaliges Fehlen
 - Bestehen der Python-Übung
 - Erreichen von Level 15 in den Testaten
 - Mindestens ein Level pro Woche weiterkommen
 - Level 1 bis 16.10.
 - Level 10 bis 11.12.
 - Level 15 vor der Klausur
 - <http://unold.net/testat>

Aufwand (ganzes Modul)

- 6 ECTS (entspricht 180h Arbeitsaufwand)
 - 60 Stunden = $12 \times (2\text{h Vorlesung} + 3\text{h Übung})$
 - 120 Stunden (= ca. 10 Stunden pro Woche)
 - Vorbereitung
 - Nachbereitung
 - Prüfungsvorbereitung
- Vor- und Nachbereitung besteht NICHT nur aus der Bearbeitung von Testat-Aufgaben!
- Prüfungsvorbereitung startet schon während des Semesters!

Unterlagen

- Folien und Übungen
 - <http://unold.net/informatik>
- Lehrbuch
 - Java ist auch eine Insel (10. Auflage)
 - <http://openbook.rheinwerk-verlag.de/javainsel/>
- Recherchieren
 - WARNUNG:
Codeschnipsel
nicht einfach kopieren!!



Themen (Auszug)

- Einführung in die Informatik
 - Programmiersprachen (Java)
 - Integrierte Entwicklungsumgebungen (Eclipse)
- Variablen, Datentypen, Operatoren
- Kontrollstrukturen (Sequenz, Selektion, Iteration)
- (statische) Methoden
- Arrays
- Objektorientierung
 - Dateien schreiben und lesen
 - Fehlerbehandlung

INFORMATIK

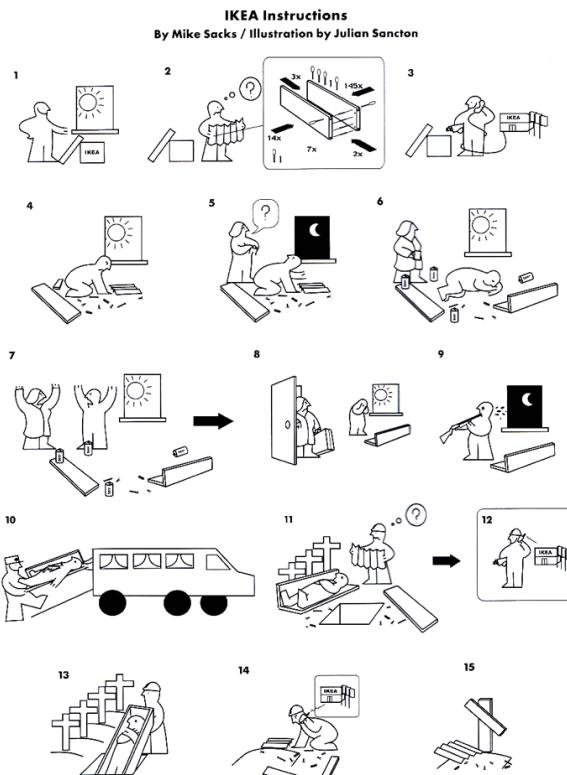
Organisatorische Fragen?

INFORMATIK

Informatik

Was ist Informatik?

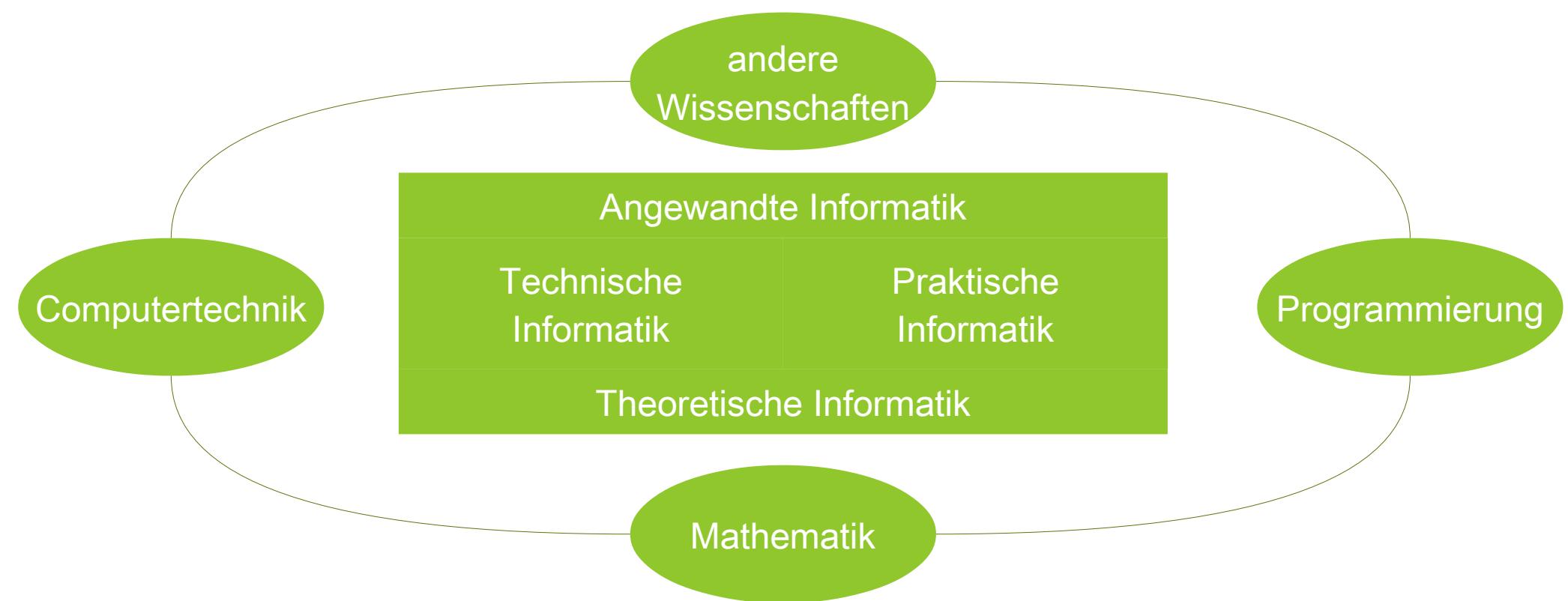
- Systematische Verarbeitung von Informationen
(Karl Steinbuch 1957)



- Automatisiert mit Digitalrechnern

INFORMATIK

Was ist Informatik?



Theoretische Informatik

- Automatentheorie und Formale Sprachen
 - Auf Mathematik und Logik basierte formale Beschreibung von Rechenmaschinen
- Berechenbarkeitstheorie
 - Welche Probleme sind mit welchem Modell lösbar?
- Komplexitätstheorie
 - Wie hoch ist der Bedarf an Ressourcen (Zeit, Speicherplatz, ...) zur Lösung eines bestimmten Problems?
- Theorie für praktische und technische Informatik
 - Entwicklung von Programmiersprachen
 - Korrektheit eines Systems sicherstellen

Technische Informatik

- Grundlagen der Hardware
 - Aus welchen Komponenten baut man ein digitales Gerät am besten zusammen, um es für seine Aufgabe optimal nutzbar zu machen?
- Rechnerarchitektur
 - Welche Speicher und Recheneinheiten sind nötig?
 - Wie können Informationen komprimiert werden?
- Mikroprozessortechnik
 - Wie baut man Speicher und Prozessor am besten auf, um Zugriffe und Berechnungen schnell durchführen zu können?
- Rechnernetze
 - Wie kommunizieren Systeme untereinander über große Distanz?
 - Wie können Daten verlustfrei übertragen werden?

Praktische Informatik

- Softwaretechnik
 - Wie kann die Arbeit der Entwicklung großer Softwaresysteme effektiv auf verschiedene Personen aufteilen?
- Datenstrukturen und Algorithmen
 - Wie lauten die formalen Handlungsanweisungen, um ein reales Problem am Computer umzusetzen?
- Programmiersprachen
 - Wie können Datenstrukturen und Algorithmen auf digitalen Geräten umgesetzt werden?
 - Welche Sprachen sind für welche Klasse von Problemen und Voraussetzungen geeignet?

Angewandte Informatik

- Computational Sciences
 - Bioinformatik
 - Chemieinformatik
- Wirtschaftsinformatik
- Geoinformatik
- Ingenieurinformatik
- Digital Humanities

...

Dieses Modul „Informatik“

- Erstellen, Kompilieren und Ausführen von Programmen mit einer integrierten Entwicklungsumgebung
- Elementare Datentypen
- Zahlensysteme und boolsche Logik
- Programmierkonstrukte: Verzweigungen, Schleifen
- Felder, eindimensional und mehrdimensional
- Strukturierungskonstrukte: Methoden
- Grundlagen der Objektorientierung: Klassen und Objekte

INFORMATIK

Fragen zur Informatik?

INFORMATIK

Programmiersprachen
(Java)

Was ist ein (Computer-)Programm?

- Sammlung von Anweisungen
 - Zur Bearbeitung von Aufgaben mit einem digitalen Gerät
- Ein „ausgereiftes“ Programm nennt man auch Software



Microsoft®
Office



WhatsApp

Wie sind die Anweisungen gespeichert?

- Als Datei auf der Festplatte (.apk, .exe, ...)
 - Enthält die Sammlung von Anweisungen
 - Bei Ausführung (Prozess) werden die Anweisungen an die jeweiligen Hardware-Komponenten geleitet

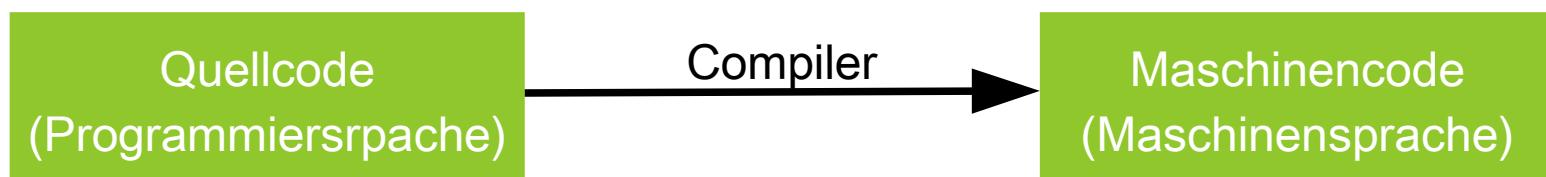
In Maschinencode

- Typischerweise sind die Anweisungen derart gespeichert, dass sie möglichst ohne weitere Bearbeitung von der jeweiligen Hardware unmittelbar ausgeführt werden können



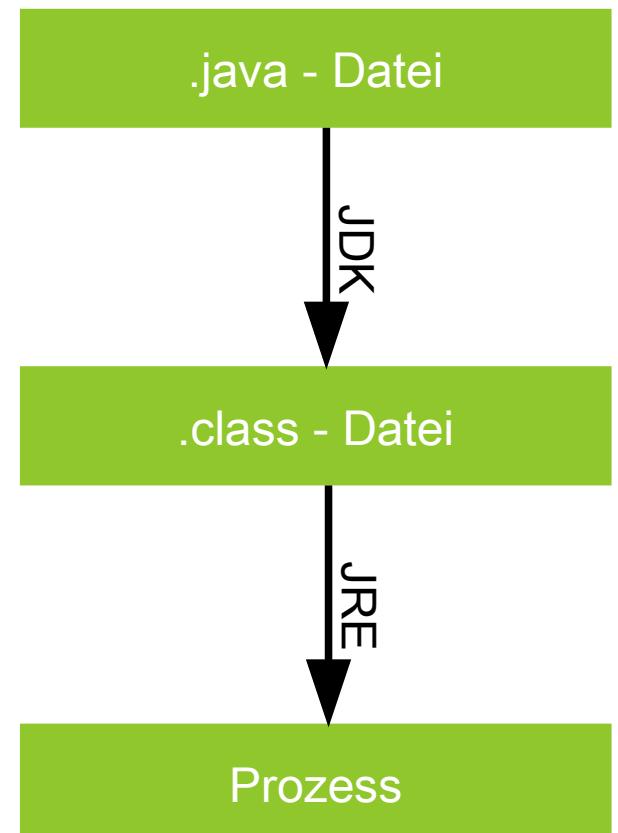
Wie werden Programme erstellt?

- Programmiersprache ist Menschen-lesbar
 - Programmiert wird in der Programmiersprache
- Maschinensprache ist Maschinen-lesbar
 - Das Fertige Programm ist in Maschinensprache
- Compiler ist ein Programm, das den Code übersetzt



Wie wird Java kompiliert?

- JDK (Java Development Kit) enthält den Java-Compiler
- JRE (Java Runtime Environment) ist eine Virtuelle Maschine
- Quellcode
 - In Programmiersprache geschrieben
 - Als .java-Datei auf der Festplatte
- Zwischencode / Bytecode
 - In Maschinennaher Sprache
 - Als .class-Datei auf der Festplatte
- Maschinencode
 - In Maschinensprache
 - Prozess wird ausgeführt



Wozu dieser Zwischencode in Java?

- Vorteil
 - Plattformübergreifend – .class-Datei kann für alle Plattformen verwendet werden, auf denen ein JRE installiert ist
- Nachteil
 - Voraussetzung ist, dass ein JRE installiert ist
 - Programme können nicht für spezielle Plattformen optimiert werden
 - Zusätzlicher Übersetzungsaufwand macht das Programm langsam
- Java ist eher ungeeignet zur Programmierung von ...
 - Anwendungen für ein spezielles Gerät
 - Extrem speicherintensiven oder zeitkritischen Anwendungen

Warum Java?

- Wann ist Java gut geeignet?
 - Allgemeine Software für eine breite Nutzergemeinde
 - Open-Source-Projekte mit vielen Programmierern
- Was wurde bereits mit Java Programmiert?
 - Die meisten Android Apps
 - Minecraft
 - Server-Komponenten diverser Web-Anwendungen
 - ...

Wie entsteht ein (Java-)Programm?

- Natürliche Sprache
 - Zu vage und komplex für Maschine
- Algorithmische Sprache (Pseudocode oder Struktogramm)
 - Hilfsmittel zur menschenlesbaren Beschreibung eines Algorithmus
- Problemorientierte Sprache (Programmiersprache)
 - Formal exakt und eindeutig, durch Computer nutzbar
- Maschinennahe Sprache (Zwischencode)
 - Plattformunabhängig, fast wie Maschinensprache
- Maschinensprache
 - Direkt vom Prozessor ausführbar

Wie sind Programmiersprachen aufgebaut?

- Syntaktische Regeln
 - Werden vom Compiler erkannt
 - Welche Zeichen dürfen wo stehen?
 - In welcher Rolle befinden sich welche Teile innerhalb des Codes?
- Semantische Regeln
 - Können vom Compiler nur in Einzelfällen erkannt werden
 - Welche Anweisung bewirkt was?
 - Kommt den einzelnen Abschnitten eine spezielle Bedeutung zu?

Beispiele in natürlicher Sprache

- Die Wetter ist heute sonnig
 - Syntaktisch korrekt
 - Semantisch falsch (Artikel von Wetter falsch, inhaltlich unzutreffend)
- Blinde Winde trinken offene Geräusche
 - Syntaktisch korrekt
 - Semantisch unsinnig
- Ich kein Vorlesung
 - Syntaktisch falsch
 - Durch die fehlerhafte Syntax kann Semantik nur geraten werden
- Regel: Ist die Syntax falsch, dann auch die Semantik!

Einschub: Skriptsprachen

- Werden nicht kompiliert,
sondern als Textdatei direkt genutzt
- Benötigen ein spezielles Programm,
das die Skripte ausführen kann
- Beispiele
 - JavaScript (benötigt Browser) – hat nichts mit Java zu tun
 - VBA (für Microsoft-Produkte)
 - PHP
 - Python
 - Matlab
- ...

Programmiersprache Java

- Definition und Formale Beschreibung der Sprache
 - <http://docs.oracle.com/javase/specs/>
- Beispiel (HelloWorld.java)

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

Programmiersprache Java

- Programm startet mit „public class“ gefolgt vom Namen und {
- Die erste Zeile ist „public static void main (String[] args)“
- Beispiel (HelloWorld.java)

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

Diese Anweisung
gibt einen Text
auf der Konsole aus

Programmiersprache Java

- Jede Anweisung wird mit ; beendet
- Kommentare werden mit // eingeleitet
- Beispiel (HelloWorld.java)

```
public class HelloWorld {  
    public static void main (String[] args) {  
        // Hier stehen die Anweisungen  
    }  
}
```

Schließende Klammern
und Einrückung
nicht vergessen!

Java – Grundgerüst

- So sieht ein leeres Java-Programm ohne Anweisungen aus
- Der Programmname sollte immer groß geschrieben werden

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

Java – Ausgabe

- Mit den Befehlen System.out.print oder System.out.println (Zeilenumbruch) kann man etwas auf der Konsole ausgeben
- Texte müssen in Anführungszeichen gesetzt werden

```
public class Testprogramm {  
  
    public static void main (String[] args) {  
  
        System.out.print("Hallo Java ");  
  
        System.out.println(4711);  
  
    }  
  
}
```

Java – Variablen

- Variablen für Zahlen können mit vorangestelltem „double“ deklariert werden (mehr dazu später)
- Auch Variablen können ausgegeben werden

```
public class Testprogramm {  
  
    public static void main (String[] args) {  
  
        double x = 4711;  
  
        System.out.println(x);  
  
    }  
  
}
```

Java – Eingabe

- Die Eingabe ist etwas komplizierter, es ist zunächst eine kleine Erweiterung des Grundgerüsts vorzunehmen

```
import java.util.Scanner;

public class Programmname {

    public static void main (String[] args) {

        Scanner scanner = new Scanner(System.in);

        scanner.close();

    }

}
```

Java – Eingabe

- Nun kann jederzeit mit dem Befehl „scanner.nextDouble()“ eine Zahl vom Benutzer eingelesen werden

```
import java.util.Scanner;

public class Programmname {

    public static void main (String[] args) {

        Scanner scanner = new Scanner(System.in);

        double x = scanner.nextDouble();

        System.out.println(x);

        scanner.close();

    }

}
```

Java – Programm

- Dieses Programm bildet die Summe zweier vom Benutzer einzugebenden Zahlen

```
import java.util.Scanner;
public class Summe {
    public static void main (String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Wie lautet die erste Zahl? ");
        double zahl1 = scanner.nextDouble();
        System.out.print("Wie lautet die zweite Zahl? ");
        double zahl2 = scanner.nextDouble();

        double summe = zahl1 + zahl2;
        System.out.print("Die Summe ist ");
        System.out.println(summe);

        scanner.close();
    }
}
```

INFORMATIK

Entwicklungsumgebung
Eclipse

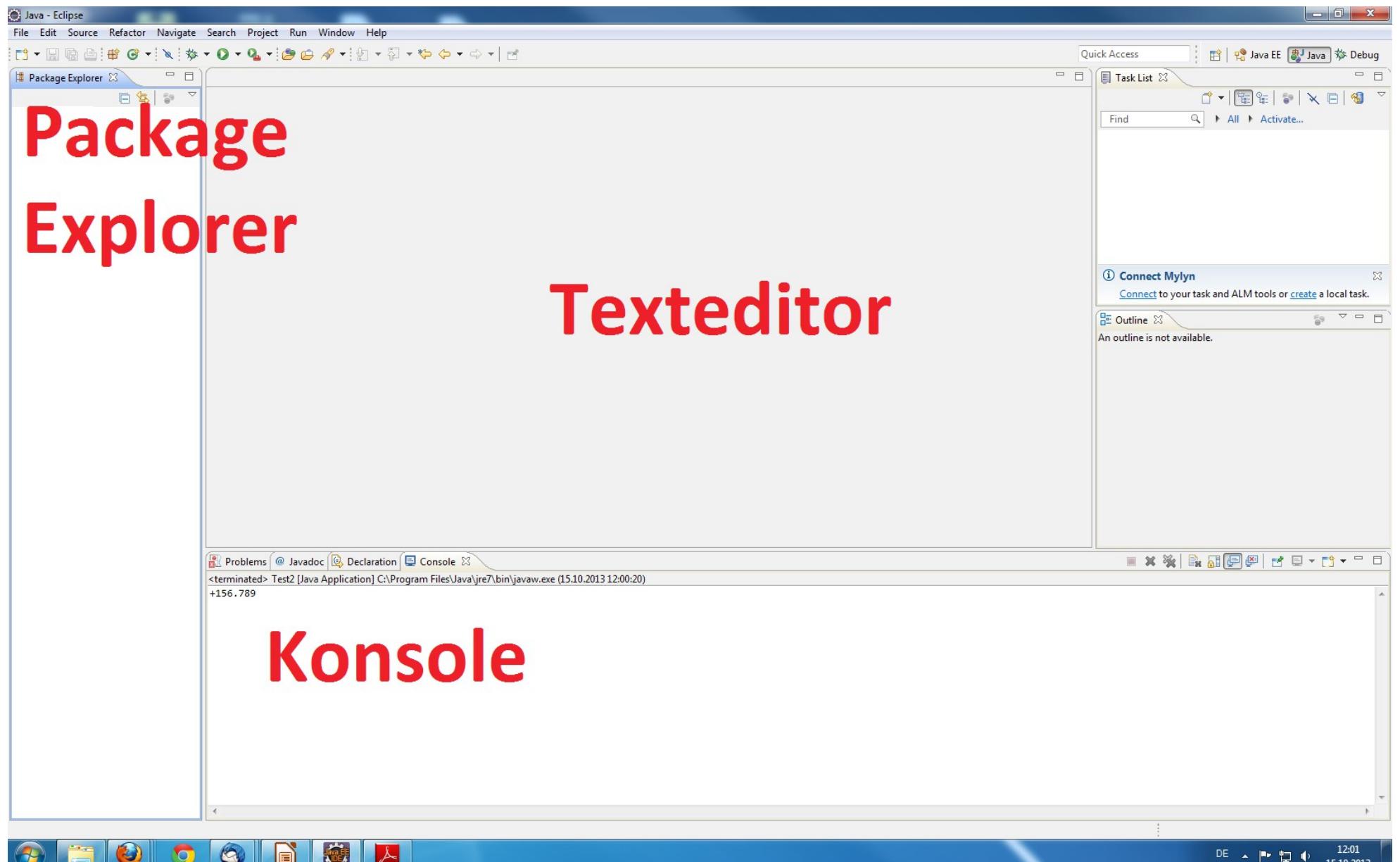
Was ist eine Entwicklungsumgebung?

- Programm,
das bei der Herstellung von Programmen unterstützt
- Funktionsumfang
 - Texteditor mit Quelltextformatierung
 - Compiler
 - Debugger
 - Linker
 - Versionsverwaltung
 - Projektmanagement
 - UML-Modellierung
 - Erstellung grafischer Benutzeroberflächen
 - ...

Welche Entwicklungsumgebung für Java?

- Eclipse
- NetBeans
- IntelliJ IDEA
- JBuilder
- JCreator
- ...

Eclipse

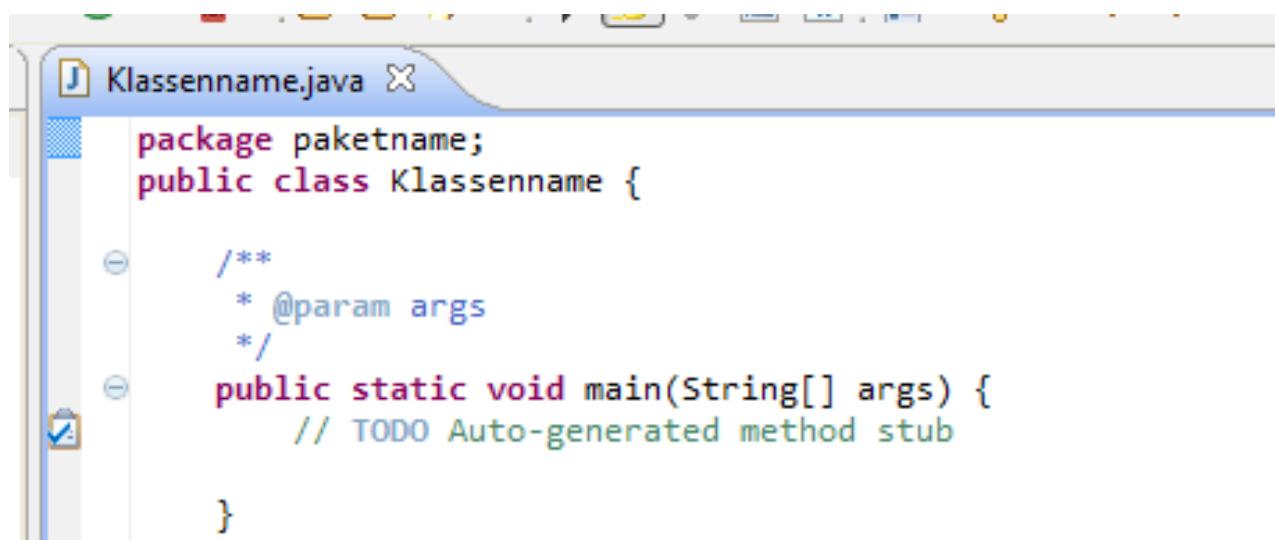


Eclipse – Vorbereitung

- Neues Projekt anlegen
 - Ein Projekt ist eigentlich gedacht als Einheit zum Erstellen großer Software mit verschiedenen kleineren Programmen
 - Wir verwenden ein Projekt „Informatik“ für alle Übungen
 - File → New → Java Project
 - Project name eingeben
 - Finish
- Neues Paket anlegen
 - Ein Paket ist eine Einheit zur Strukturierung (test, uebung1, ...)
 - Im Package Explorer Rechtsklick auf das Projekt
 - New → Package
 - Name eingeben
 - Finish

Eclipse – Programm erstellen

- Neue Klasse anlegen
 - Eine Klasse entspricht in den ersten Wochen dieser Vorlesung auch einem kompletten Programm
 - Im Package Explorer Rechtsklick auf das Paket
 - New → Class
 - Name eingeben
 - [x] public static void main(String[] args) anklicken
 - Finish

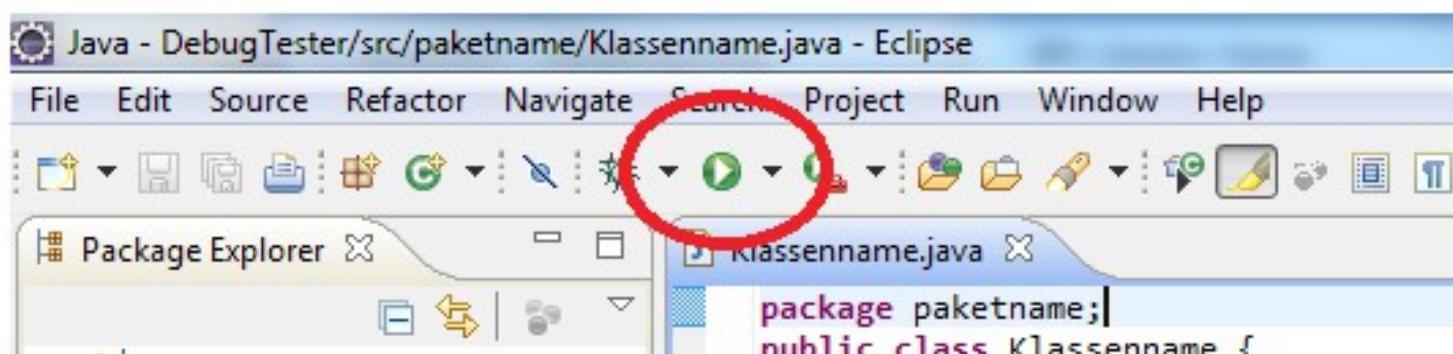


```
package paketname;
public class Klassenname {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }
}
```

Eclipse – Programm starten

- Kompilieren und Ausführen
 - Beide Schritte (Kompilieren mit JDK und Ausführen mit JRE) werden von Eclipse gleichzeitig und auf Knopfdruck durchgeführt
 - In der Konsole befinden sich Ein- und Ausgaben
 - Es wird die gerade aktive Klasse ausgeführt
 - Run → Run





Hausaufgaben

1.2 Warum Java gut ist: die zentralen Eigenschaften

1.7 Eclipse im Speziellen

2.2 Von der Klasse zur Anweisung