



Summer Term 2010

# Softwareentwicklung in der Geoinformatik Praxis

## **Introduction**

## **Java and Eclipse**

Bernd Resch

16 March 2011



# Overview

- Organisational
- Overview – Contents & Goals
- Recap – Java
- Introduction to Eclipse
- Programming...



# Organisational ::: Dates

- Wed, 16 March 2011 09.00-12.00 → HS 3.004
- Wed, 06 April 2011 09.00-12.00 → HS 3.004
- Wed, 13 April 2011 09.00-12.00 → HS 3.004
- Wed, 04 May 2011 09.00-12.00 → HS 3.004
- Wed, 18 May 2011 09.00-12.00 → HS 3.004
- Wed, 01 June 2011 09.00-12.00 → HS 3.004
- Wed, 15 June 2011 09.00-12.00 → HS 3.004

- 
- *Fri, 15 July 2011* *End-term Assignment Due Date*

# Organisational ::: Grading Scheme

---

- For you:
  - ◆ Interim Assignments
  - ◆ End-of-term assignment
  - ◆ Active participation in class
  
- For me:
  - ◆ Evaluation!

## Overview ::: Goals

---

- Die Definition von „Geoinformatik“ als Fachgebiet wird traditionell von Seiten der physischen und Humangeografie motiviert, und damit Geoinformatik als Teilbereich der Geografie betrachtet. Die VL „Grundlagen und Konzepte der Softwareentwicklung“ versucht eine Annäherung an den Begriff von der Informatikdomäne her, also ein Verständnis von **Geoinformatik als ICT mit starker geografischer Prägung** zu begründen.

# Overview ::: Goals

---

- Das Ziel der Übung ist die Vermittlung eines Grundverständnisses für die Umsetzung der in der VL erlernten Basiskonzepte der Softwareentwicklung. Ein starker Fokus wird auf der Verwendung von **Java** liegen, wobei auch web-basierte Technologien zum Einsatz kommen.
- Vermittelt wird ein Grundverständnis für praktische Aspekte der Softwareentwicklung mit einem speziellen **Fokus auf geographischen Anwendungen**. Weiters werden spezielle Geo-Softwareanwendungen (Google Maps, Google Earth, GeoTools, GeoServer, OpenLayers u.a.) erläutert und in praktischen Beispielen angewendet. .

# Overview ::: Contents

---

- Java – object-oriented programming
- GeoTools
- HTML
- Java Server Pages
- Google Maps API, Google Earth
- GeoServer – WMS and WFS connectors

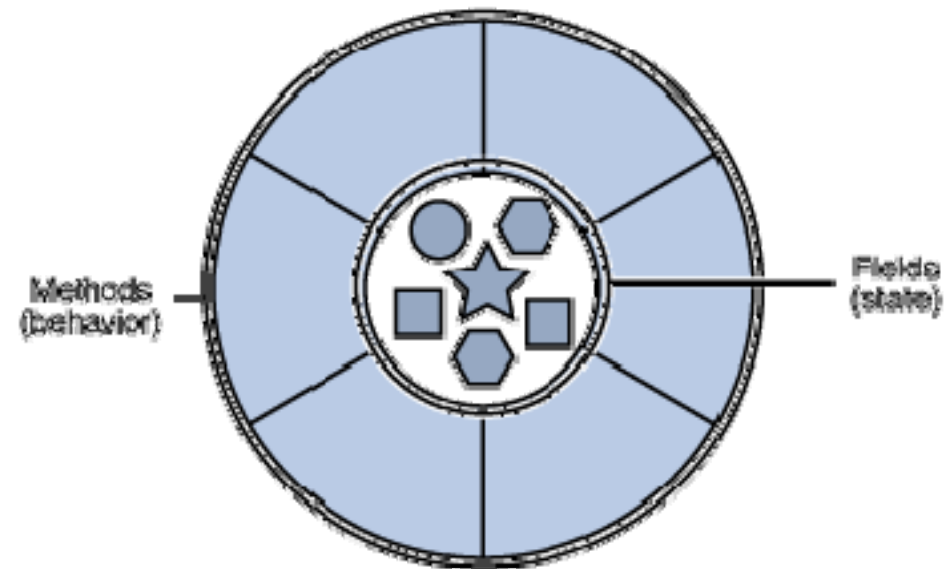
---

# Recap ::: Java

---

# Object-oriented Programming

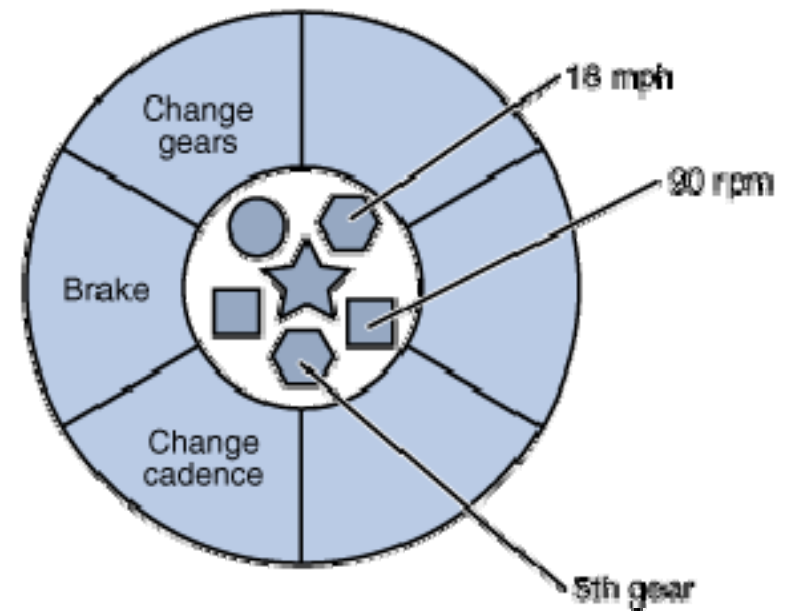
- Object
  - ◆ Identity
  - ◆ State – attributes
  - ◆ Behaviour – methods



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

# Object-oriented Programming

- Object
  - ◆ Identity: „Bicycle“
  - ◆ Attributes
    - Velocity
    - Gear
    - Rotations per minute
  - ◆ Methods
    - Change gears
    - Brake



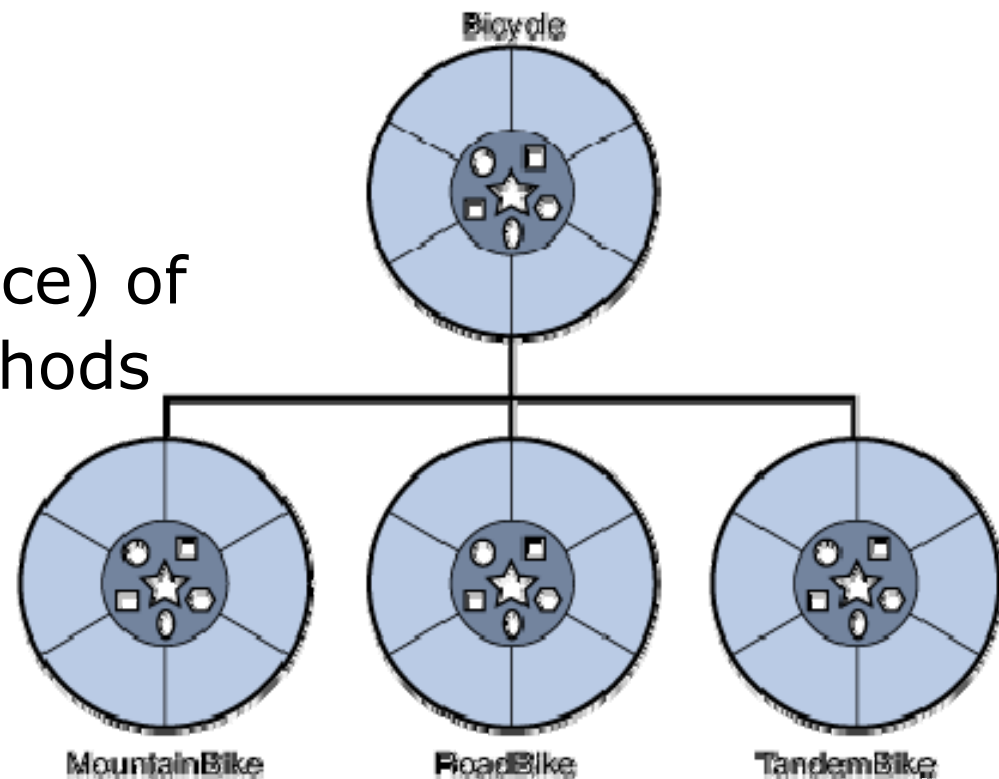
Source: <http://java.sun.com>

# Object-oriented Programming

- Inheritance

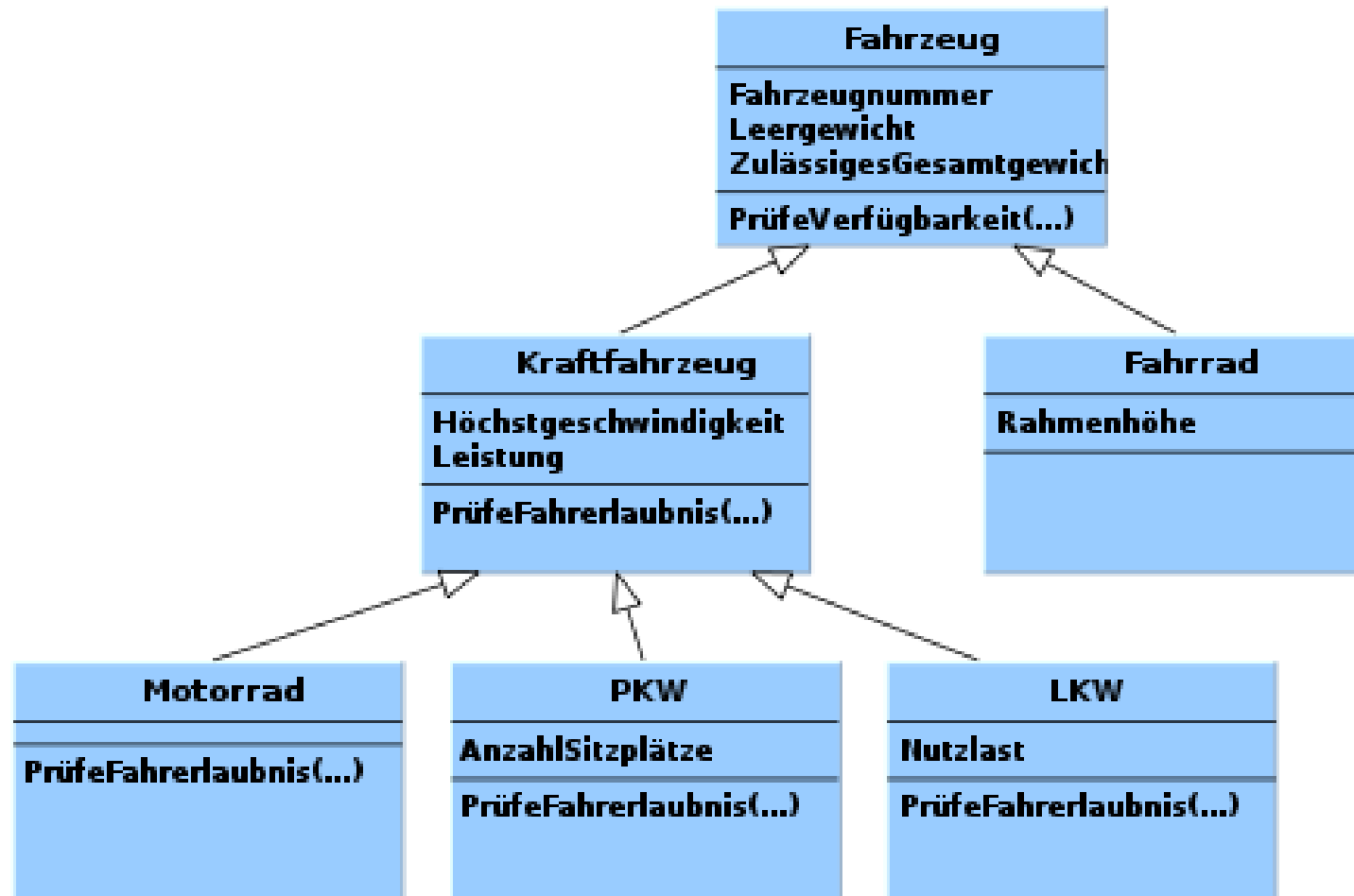
- ◆ Object hierarchy
- ◆ Transfer (inheritance) of attributes and methods

➔ Re-usability



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

# Object-oriented Programming



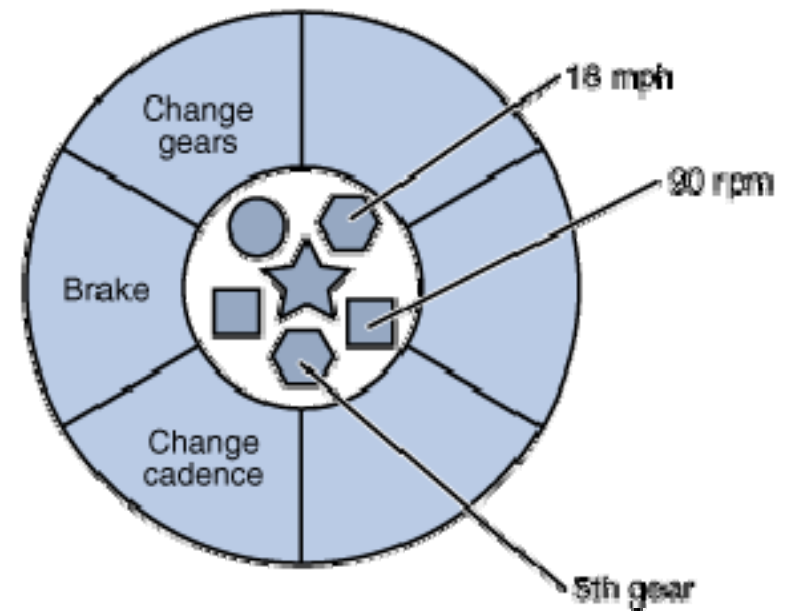
Source: <http://de.wikipedia.org>

# Object-oriented Programming ::: Java

```
// these are just syntactical examples!
package vehicles;
import org.geotools.*;
```

## ■ Java Code

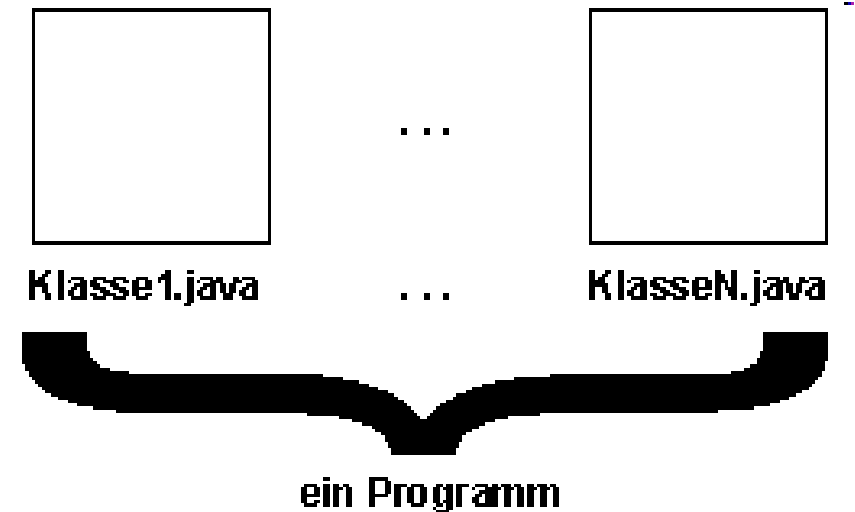
```
// Klassendefinition
class Bicycle {
    int velocity = 0;
    int gear = 1;
    void changeGears(int gear_new) {
        gear = gear_new;
    }
    void accelerate(int increase) {
        velocity = veolocity + increase;
    }
    void brake(int decrease) {
        veloocity = velocity - decrease;
    }
    void printStates() {
        System.out.println("Velocity: " + velocity + " Gear: " + gear);
    }
}
```



Quelle: <http://www.java.com>

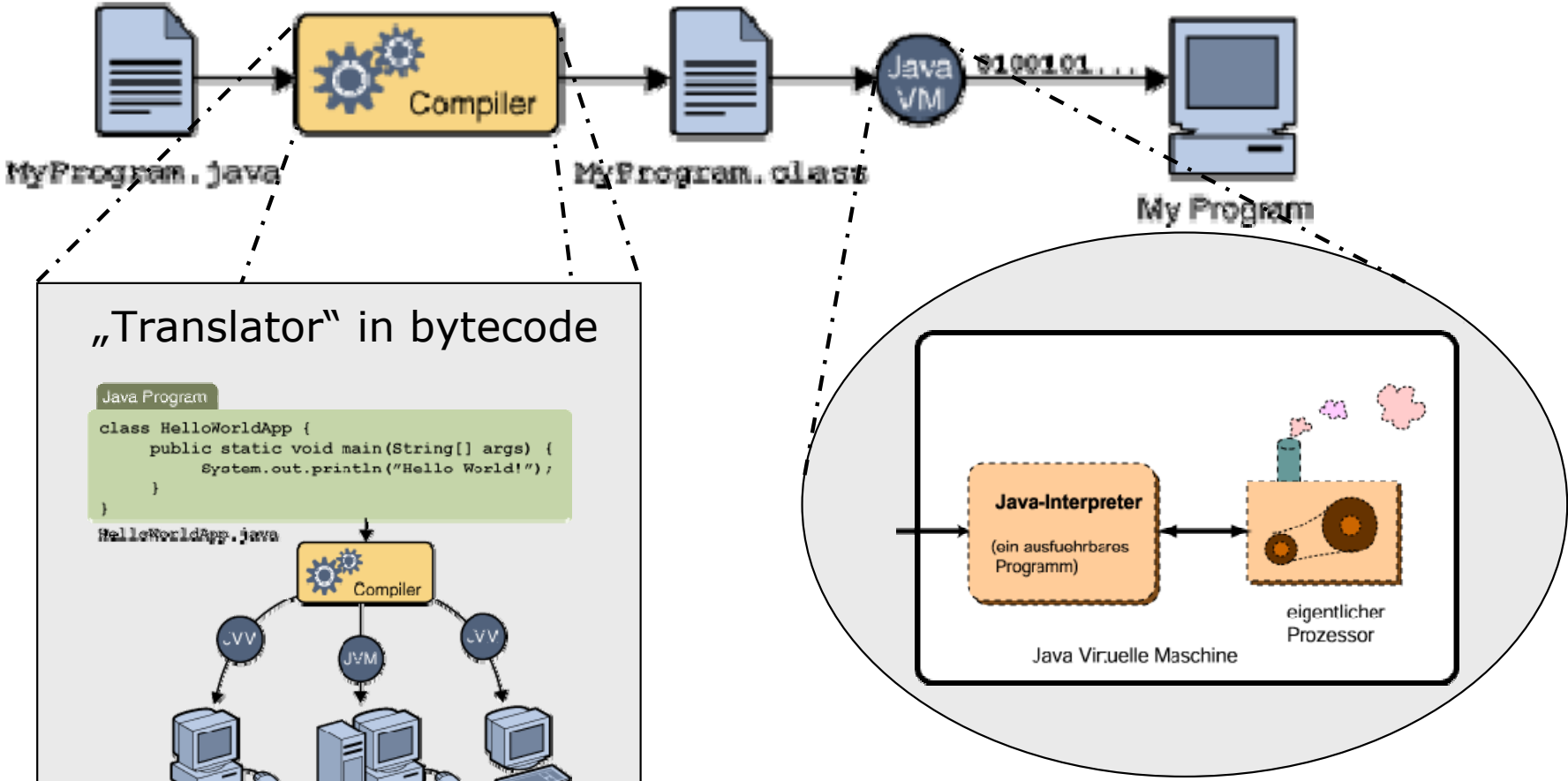
# Object-oriented Programming ::: Java

- File extension of source code files: .java
- Files extension of compiled files: .class



Source: <http://www.it-infothek.de>

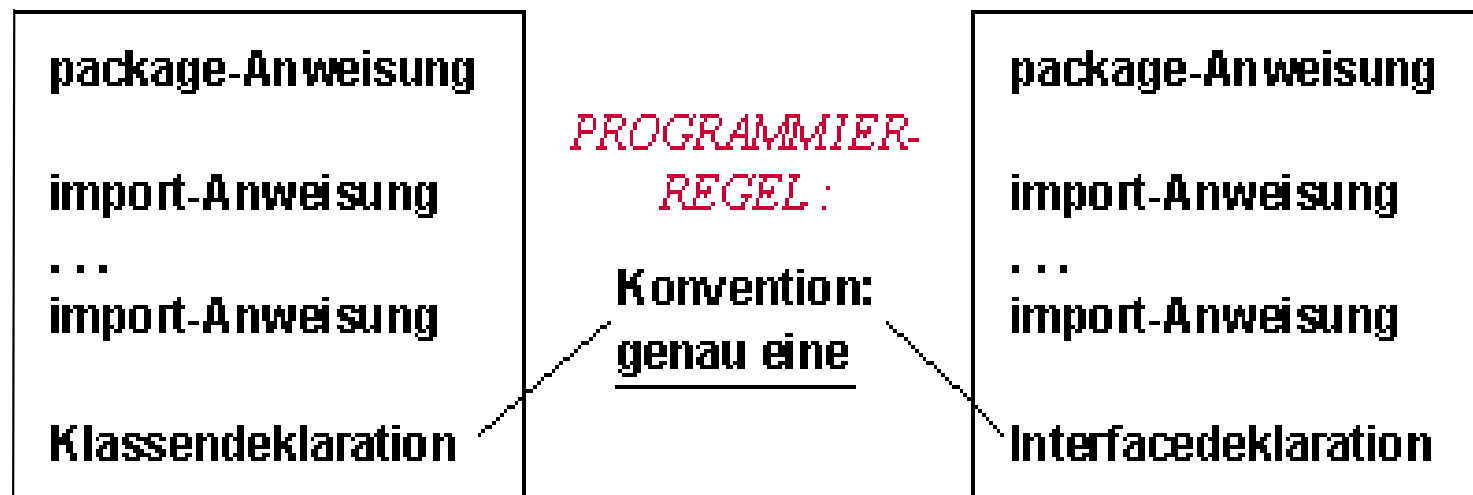
# Object-oriented Programming ::: Java



Sources: <http://www.java.com>  
<http://www.gailer-net.de>

# Object-oriented Programming ::: Java

- Basic structure of a Java programme



Source: <http://www.it-infothek.de>

# Object-oriented Programming ::: Java

---

- Class
- Object
- Interface
- Package
- Method
- Import
- Data type
- Loops and conditions
- Virtual Machine
- Inheritance – „extends“
- Multi-threading
- Servlet

---

# Introduction to Eclipse

---

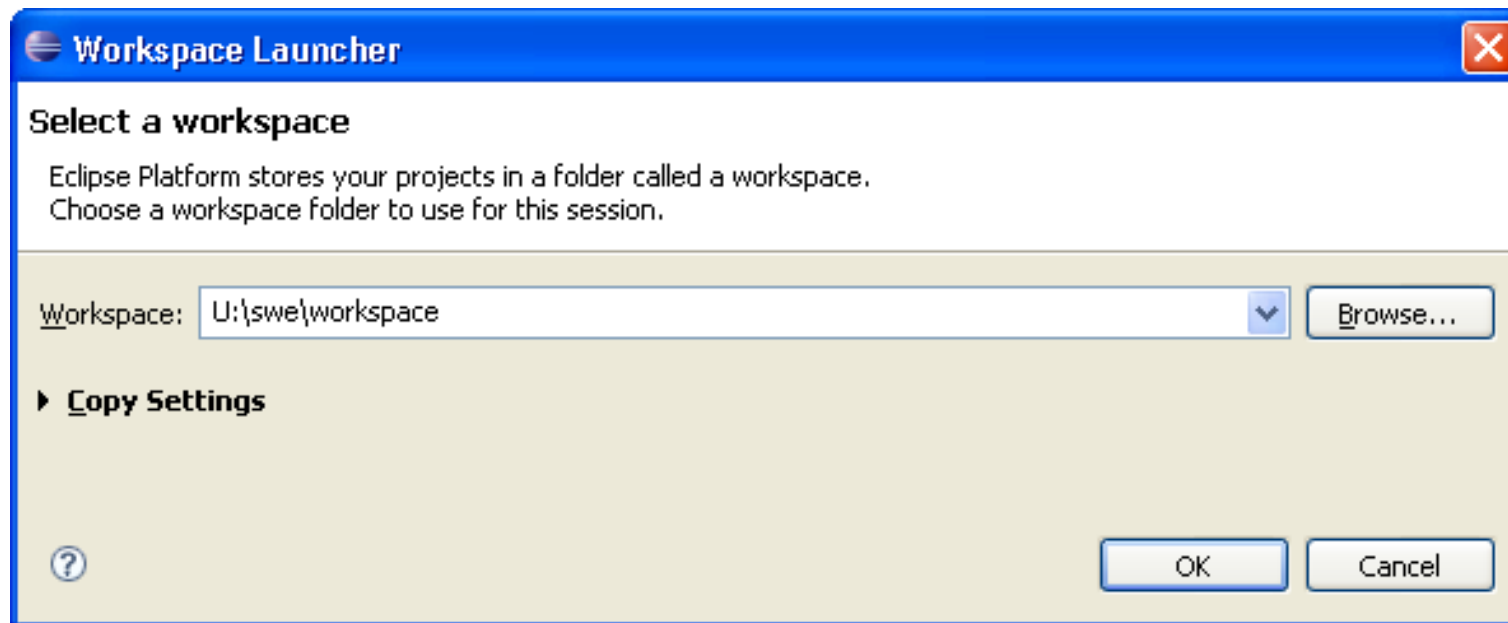
# Introduction to Eclipse

---

- IDE – Integrated Development Environment
- One-click:
  - ◆ Compilation
  - ◆ Execution
  - ◆ Debugging

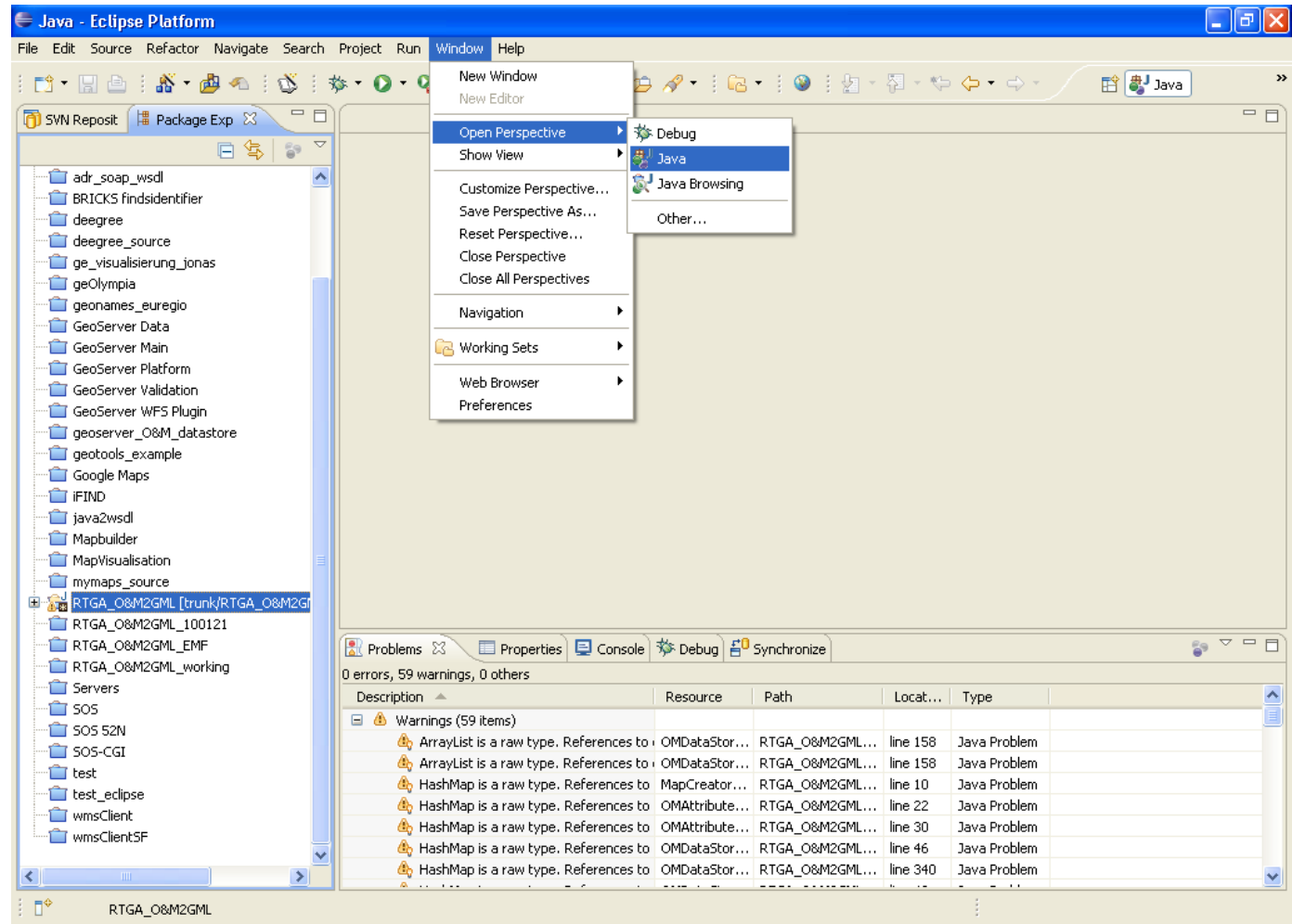
# Introduction to Eclipse

- \\dgeo09\dfs\home\studenten\- Start-up – Workspace Launcher
  - ◆ U:\swe\workspace



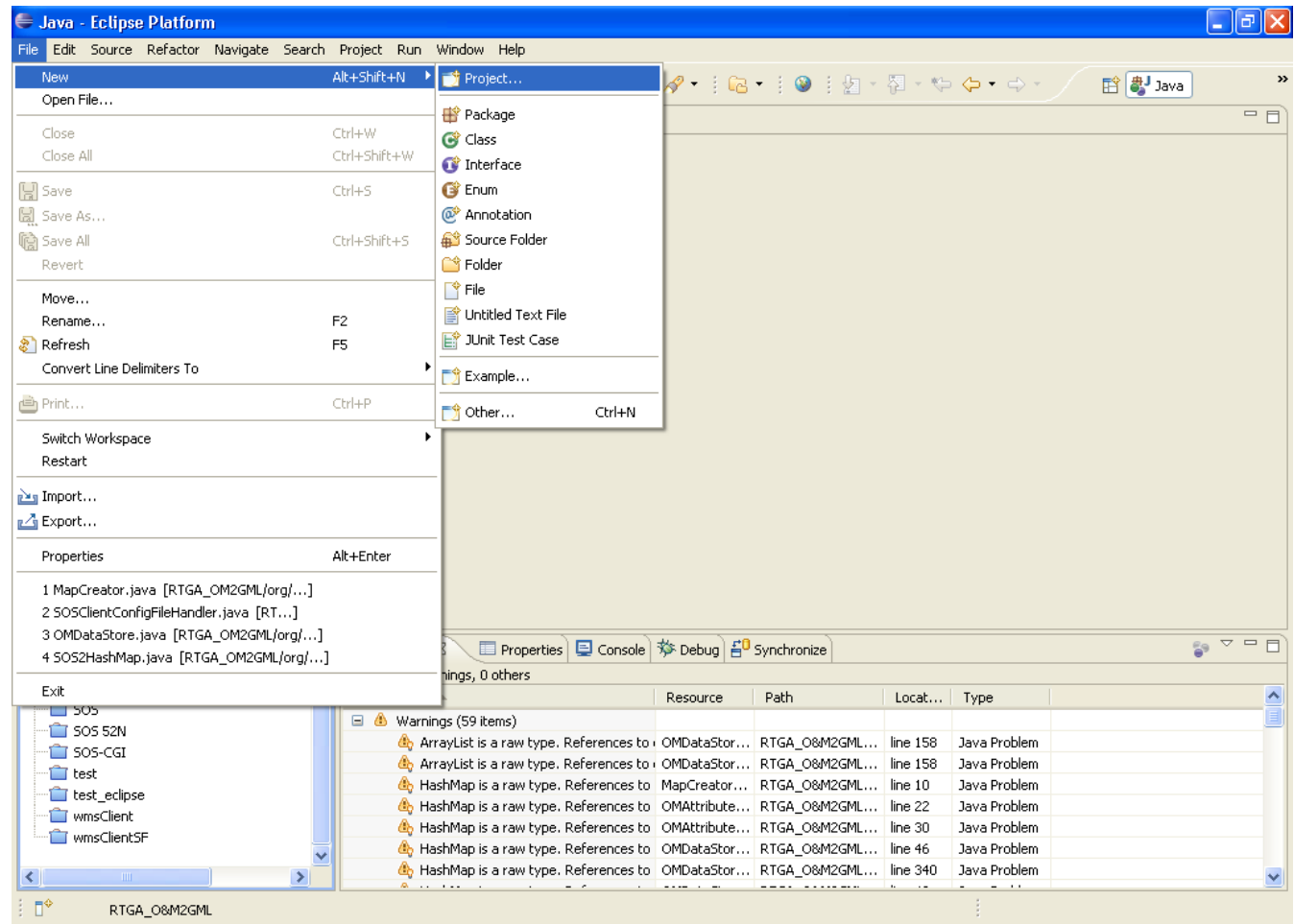
# Introduction to Eclipse

- Switch to „Java Perspective“



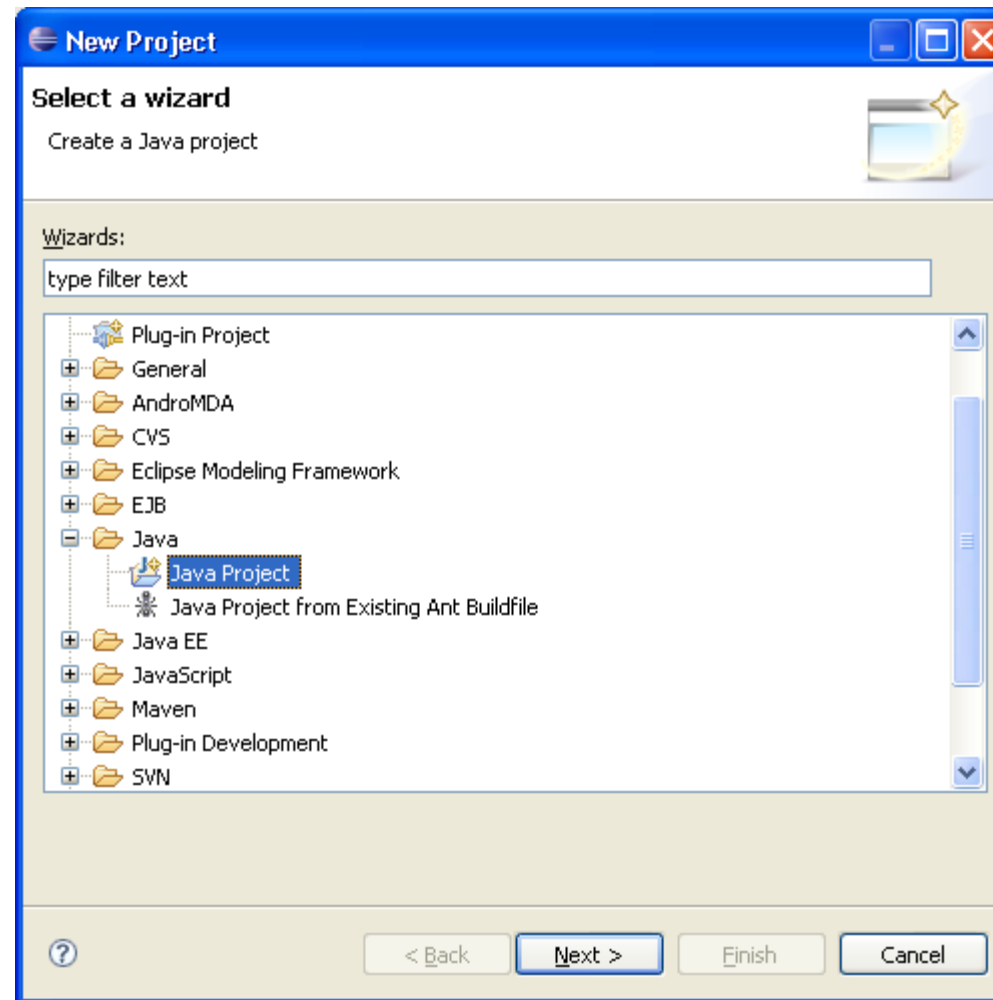
# Introduction to Eclipse

- Create a new project



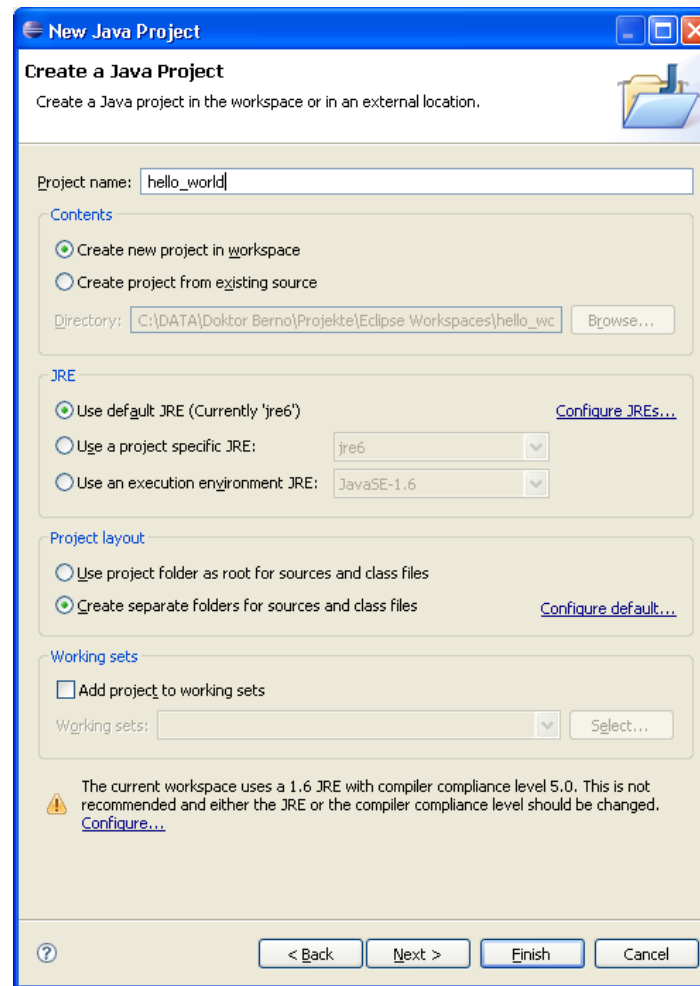
# Introduction to Eclipse

- Create a new project



# Introduction to Eclipse

- Create a new project
- “unit1”



# Java Programming Conventions

---

- Class names start with capital letters
- Method names start with lower case letters
- No umlauts and symbols
- Inline documentation ( `//` , `/*` ... `*/` )
- Java is case sensitive!

# Java Programming in Eclipse

---

- Let's get started...
- Java tutorials:
  - ➔ <http://www.tutorialspoint.com/java/index.htm>
  - ➔ <http://download.oracle.com/javase/tutorial>
  - ➔ <http://home.cogeco.ca/~ve3ll/jatutor0.htm>

# Java ::: Exercise Hello World!

---

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

# Java ::: Exercise Variable Definition

---

- `String text = "Hello World!";`
- `System.out.println(text);`

## Java ::: Exercise

---

- Create a class called “Multiplier”, which multiplies numbers from 0-9 with 10 and prints the result on the screen.

# Java ::: Exercise Methods

---

- Create a method called “multiply()”, which
  - ◆ contains the code from the assignment above.
  - ◆ is called from the “main()” method in the same class.

## Java ::: Exercise

---

- Create a class called “EvenNumbers”, which prints even numbers between 0 and 19 on the screen.
  - ◆ The method shall iterate through all numbers from 0-19 and check for each of them whether it is even or odd.

## Java ::: Exercise

---

- Create a class called "ArraySort", which
  - ◆ Defines an integer array {4, 3, 5, 0, 2, 1}
  - ◆ Prints the values of the array in a sorted manner (0-5)

# HOME STUDY !!!

---

- <http://www.tutorialspoint.com/java/index.htm>
  - ◆ Java Basics

# Java ::: Assignment 1

---

- Create a class called “CircleArea”, which calculates the area of a circle:
  - ◆ The radius of the circle is 4 units

# Java ::: Assignment 1

---

- Create a class called "CircleAreaUserInput", which calculates the area of a circle:
  - ◆ The radius of the circle shall be entered by the user
  - ➔ BufferedReader
  - ➔ Exception Handling (readLine() method)
  - ➔ String to Integer conversion

# Java ::: Assignment 1

---

- Create a class called “Fibonacci”, which prints the Fibonacci sequence:
  - ◆ Fibonacci: the printed number is the sum of the two previous numbers
  - ◆ Definition: the first two numbers are 0 and 1
  - ◆ Print 20 iterations

# Java ::: Assignment 1

---

- Put all the java files in a package named "assignment1\_<name1>\_<name2>"
- Submission by email:
  - ◆ Java code files
  - ◆ Including inline documentation of key parts
  - ◆ Short description of approach (directly in the email)
- Due 04 April 2011

# REMINDER

---

- Evaluation!!
- Lecture 3<sup>rd</sup> exam date:  
Fri, 25 March 2011, 10.00-11.30 → SE Geogr



Summer Term 2010

# Softwareentwicklung in der Geoinformatik Praxis

## **Introduction**

## **Java and Eclipse**

Bernd Resch

16 March 2011

