

# Einführung in die Programmierung

Wolfgang Schreiner

Research Institute for Symbolic Computation (RISC)

Johannes Kepler University, Linz, Austria

[Wolfgang.Schreiner@risc.jku.at](mailto:Wolfgang.Schreiner@risc.jku.at)

<http://www.risc.jku.at>

# Inhalt und Organisation

## Grundlagen der Informatik

Informatik als ein wichtiges Element von KWM.

- 1. Semester: Prozedurale Programmierung
  - Grundlagen der prozeduralen Programmierung, die Ausführung von Programmen in Computern, Einführung in die Programmiersprache Java.
- 2. Semester: Objektorientierte Programmierung.
  - Objekte mit Methoden, Schnittstellen, Vererbung, Algorithmen und Datenstrukturen, ereignisorientierte Programmierung.

Problemlösen mit Hilfe des Computers.

## Prozedurale Programmierung

- Grundlagen der Programmierung.
  - Variablen, Zuweisungen, Kontrollstrukturen, Ablaufdiagramme.
- Die Ausführung von Programmen in Computern.
  - Speicher, Prozessor, Maschinsprache, Übersetzung, Interpretation.
- Grundlagen von Java.
  - Datentypen, Kontrollstrukturen, einfache Ein/Ausgabe, Applets.
- Weiterführendes Java.
  - Klassen, Methoden, Objekte.
  - Strukturierte Programmentwicklung.

Einführung in die prozedurale Programmierung.

## Organisation

- Vorlesung (Wolfgang Schreiner)
  - Schriftliche Gesamtklausur am Ende der LVA.
  - Vorgezogene Teilklausur in der Mitte.
    - \* Hälfte des LVA-Stoffes (50% der Punkte der Gesamtklausur).
    - \* Entsprechender Teil der Gesamtklausur ist optional.
- Übung (2 Gruppen, Mirjam Augstein und Wolfgang Schreiner).
  - 10 Übungen (Minimum: 8 Abgaben und 8 positive Bewertungen).
  - Positiver Erfolg ist Voraussetzung für die Zulassung zur Klausur.
- Freiwillige Förderstunde (Matthias Neuwersch).

Fragen in/vor/nach einer LVA, jederzeit im Moodle-Forum, im Büro nach Terminvereinbarung per Email.

## Lernunterlagen

Registrierung im entsprechenden Moodle-Kurs!

- Folien:

- Selbst ausdrucken und mitnehmen zum Mitlesen und Annotieren.

- Skript:

- Selbststudium zu Hause.

- Literatur:

**Mössenböck** *Sprechen Sie Java? — Eine Einführung in das systematische Programmieren.*

- Sehr gute Einführung in die Programmierung (330 Seiten).

**Oracle** *Java Tutorials* und *Java Language Reference.*

- Frei verfügbar (<http://docs.oracle.com/javase>).

# Introduction

## Introduction

Problem solving with computers.

- **Computers:** special kind of machines.

- Behavior not controlled by their physical construction (the **hardware**).
- Behavior controlled by **programs**.

- **Programs:** formal descriptions of behavior.

- Programs are **software**, i.e., intangible units.

Software can be stored on a physical medium like a thought can be written on paper; however like a thought is not the paper on which it is written, software is not the medium on which it is stored.

- **Computer system:** hardware and software.

- For interaction, systems are linked by **networks** within an organization (Intranet) and across the globe (Internet).



## Programming

- **Programming language:** formal notation for writing programs.

Like a **thought** may be formulated in different human languages, a program may be written in different programming languages

- **Algorithm:** method for solving a problem.

Like an **idea** may be expressed by different words (or in different languages), an algorithm may be implemented by different programs or in different programming languages.

**A program expresses an algorithm in a programming language.**

## Systematic Program Development

1. analysis of the problem **requirements**;
2. unambiguous **specification** of the problem to be solved;
3. **design** of the problem solution by decomposing the problem into subproblems;
4. use of known or (design of new) **algorithms** that solve the subproblems;
5. analysis of the design, consideration of alternatives, **refinement** of the design;
6. **implementation** of the design as a program in a particular programming language;
7. running the program and **debugging** it,
8. systematic **testing** of the program;
9. **packaging** the program with the documentation of its implementation and use.

At any step we may iterate back to some previous step.

## Role of Programming

- Just a part of **software engineering**.

The construction of large programs that solve complex problems by teams of programmers.

- A **core part** in this process:

- A **craft** that must be trained to free our minds from technical details.

The writer of a book does not have to think any more about how to put letters on paper.

- An **art** which requires creativity and intuition.

Most people learn reading or writing but only few are good writers.

- Educational effects:

- Shapes mind and strengthens discipline.

We cannot cheat by omitting unpleasant parts of a problem or by not refining a solution in sufficiently much detail

- Good training methodology for analyzing and solving (mostly but not only) technical problems.

## The Programming Language Java

- Programming:

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

- Compiling:

```
javac HelloWorld.java
```

- Executing:

```
java HelloWorld
```

What are the basic elements of programming?