Technische Hochschule Rosenheim
Technical University of Applied Sciences

Programming Basics

Introduction

Prof. Dr Silke Lechner-Greite

Table of contents - overall

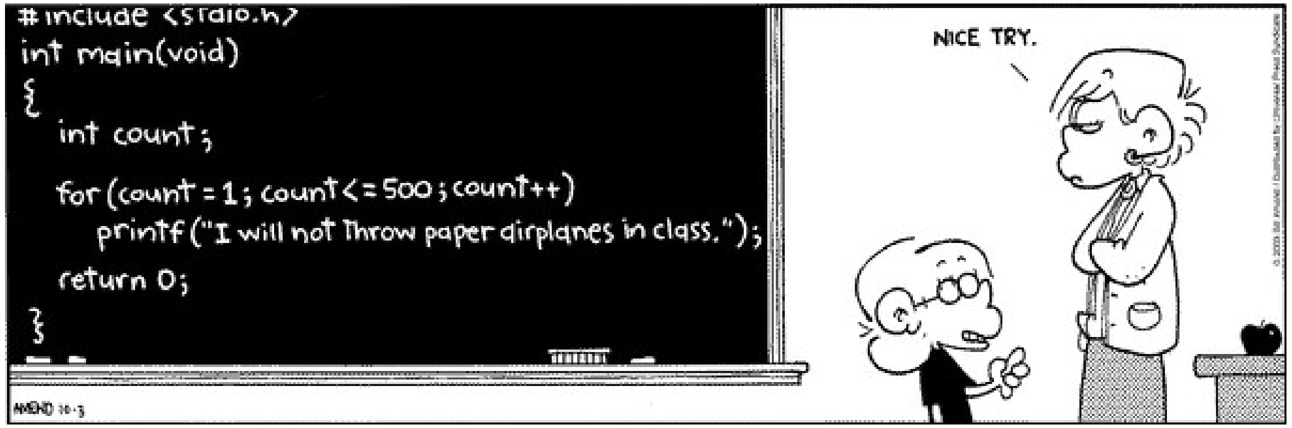
Technische Hochschule Rosenheim

Chapter 1: Introduction

Fascination with programming

"Programming is like playing God. Within the scope of the programme you can do anything."

Quote: Vinton G. Cerf



```
#include <stdio.h>
int main(void)
{
  int count;

  for (count = 1; count <= 500; count++)
     printf("I will not Throw paper airplanes in class.");
  return 0;
}
```

What is a programme?

- Operating instructions for a computer system

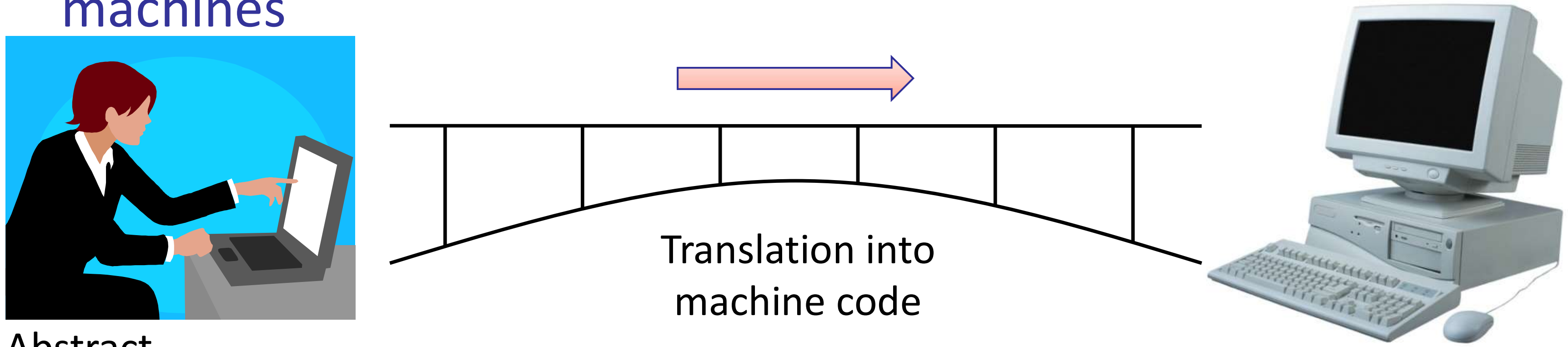
- Required characteristics:
  - Must provide clear and detailed instructions to the computer system in a precisely defined notation about what it should do step by step

$$\begin{tikzcd} & \downarrow & \\ & \text{Programming language} \arrow[dl] \arrow[dr] & \\ \text{Notation} & & \text{Meaning} \end{tikzcd}$$

of the individual instructions must be defined

What is a programming language?

- Programming languages serve as the bridge between humans and machines



Abstract
high-level language for the programmer

Result
can be executed on a computer system

Translation is performed by a compiler

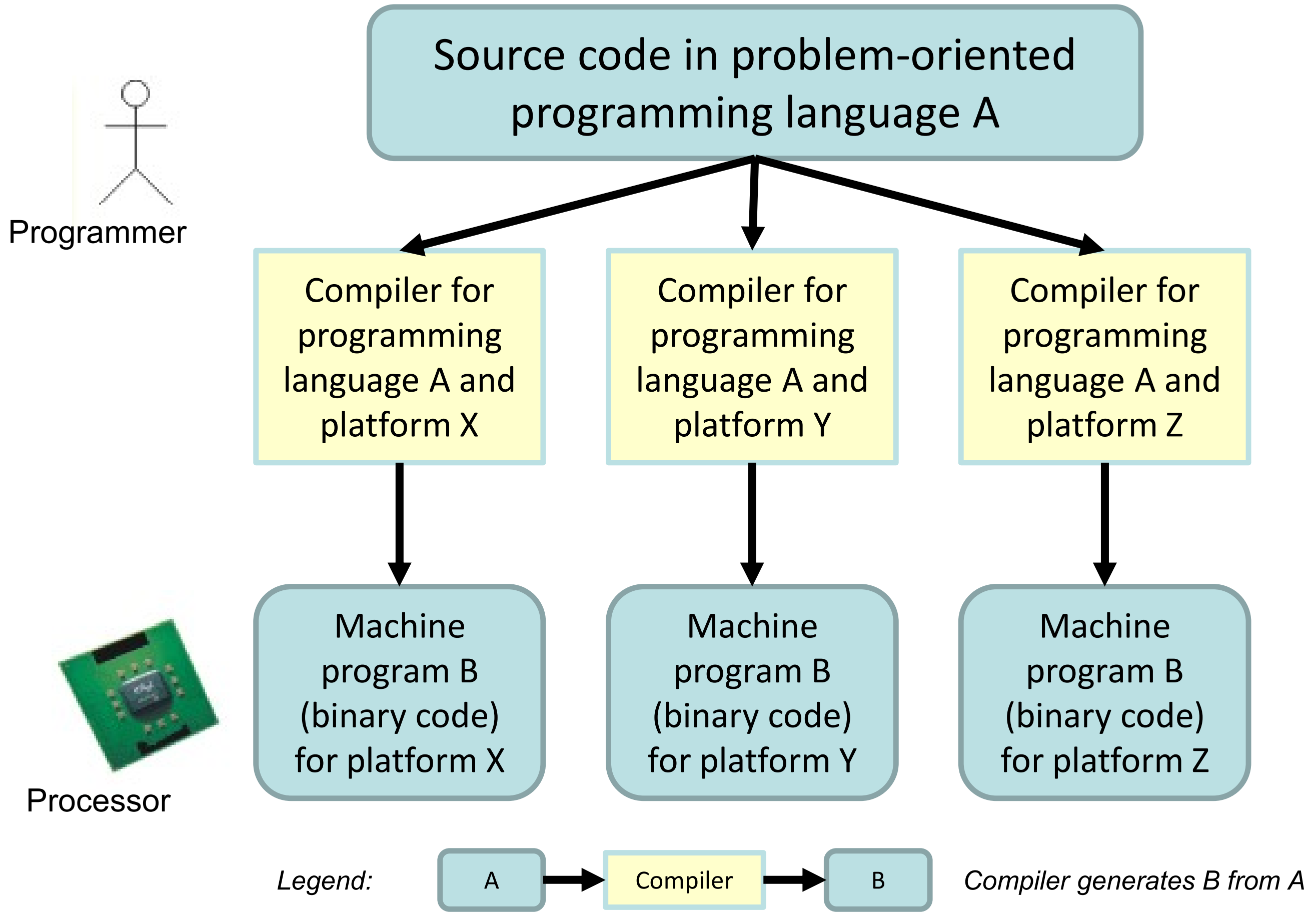
Types of programming languages

- Problem-oriented (user-oriented, higher-level)
  - Aim: to make writing programmes as easy as possible for people

- Machine-oriented
  - Aim: to make transformation into machine language (which can be executed directly by computers) as easy as possible -> tailored to a specific computer platform (processor type and operating system)

Programming language

Translation process



Source code in problem-oriented programming language A
Programmer
Compiler for programming language A and platform X
Compiler for programming language A and platform Y
Compiler for programming language A and platform Z
Machine program B (binary code) for platform X
Machine program B (binary code) for platform Y
Machine program B (binary code) for platform Z
Processor
Legend:
A
Compiler
B
Compiler generates B from A

Compiler
-> Conversion
of all sentences in a
source language into
equivalent sentences
in a target language

Translated from: Balzert (2013): Java: Der Einstieg in die Programmierung

Groups of programming languages (1)

- Can be divided into three groups:
  (1) Compiled programmes
  (Programmes are translated by a compiler and then executed, e.g. C++)
  (2) Interpreted programmes
  (Programmes are interpreted instruction by instruction and – if the respective instruction is syntactically correct – executed immediately -> script languages, e.g. JavaScript)
  (3) Compiled and interpreted programmes
  (Programmes are translated into intermediate code; instructions in the intermediate code are then analysed and executed by interpreters or compiled step by step by a just-in-time compiler -> e.g. Java and C#)

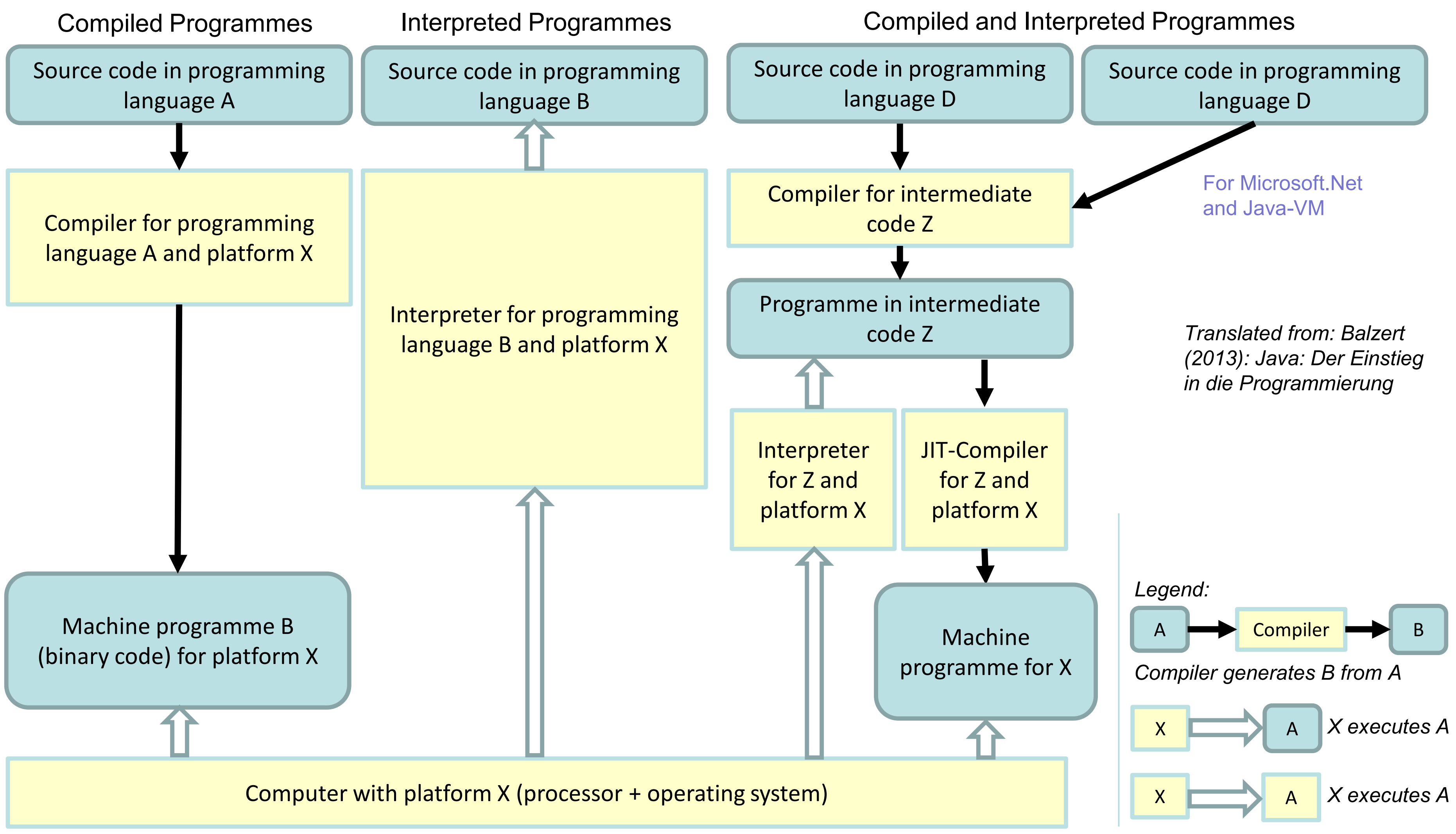
Compiled and interpreted programmes

- Java
  - The intermediate code generated is called ***bytecode***.
  - Interpreter: JVM (*Java virtual machine*) - also abbreviated as VM.
  - The interpreter conceals the properties of the respective platform

    → higher »abstraction layer« than an integrated platform

    → integrated processor → virtual machine
- C#
  - The intermediate code generated is called *MSIL (Microsoft Intermediate Language)* - also abbreviated as IL.
  - Execution: with .Net runtime environment *Common Language Runtime* (CLR) == VM for C#

Groups of programming languages (2)



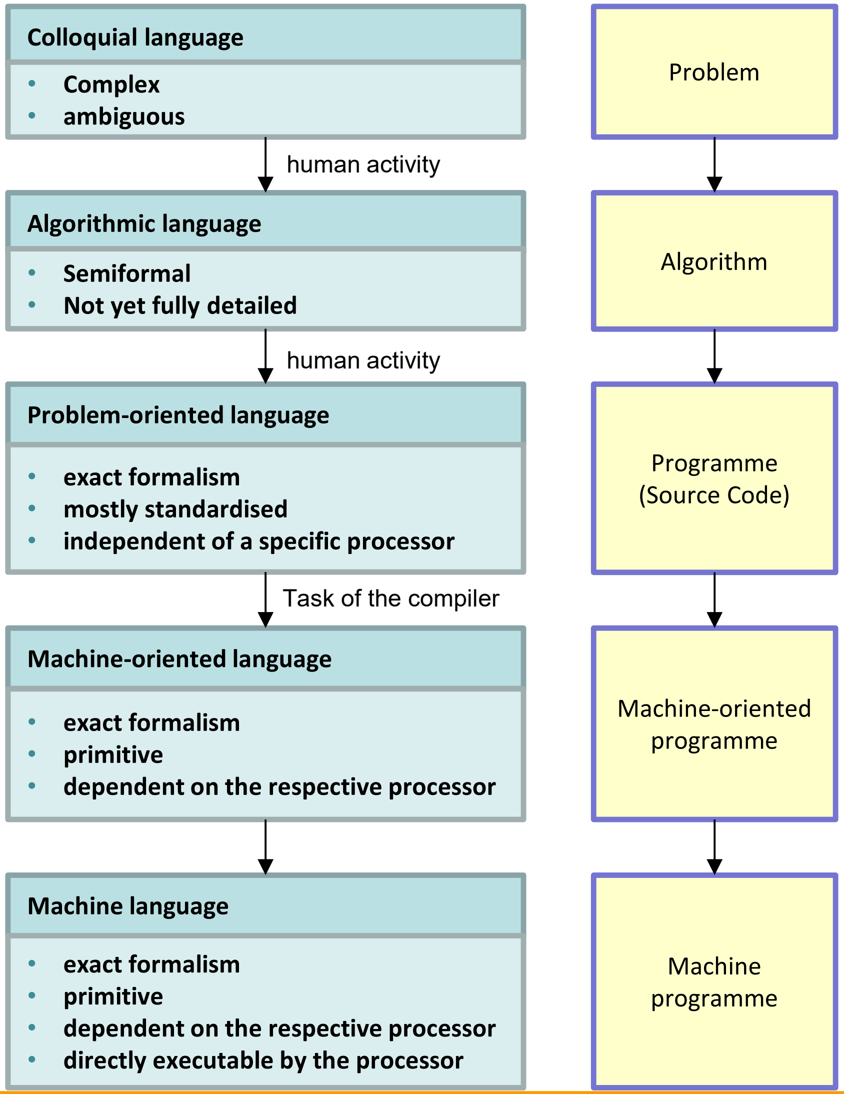
Compiled Programmes
Interpreted Programmes
Compiled and Interpreted Programmes
Source code in programming language A
Source code in programming language B
Source code in programming language D
Source code in programming language D
Compiler for programming language A and platform X
Interpreter for programming language B and platform X
Compiler for intermediate code Z
For Microsoft.Net and Java-VM
Programme in intermediate code Z
Translated from: Balzert (2013): Java: Der Einstieg in die Programmierung
Interpreter for Z and platform X
JIT-Compiler for Z and platform X
Machine programme B (binary code) for platform X
Machine programme for X
Computer with platform X (processor + operating system)
Legend:
A
Compiler
B
Compiler generates B from A
X
A
X executes A
X
A
X executes A

JIT compiler

- *Just-in-time compiler* (JIT compiler)
- Only the part that is currently required for processing is compiled step by step and *not* the entire programme.
  - → Immediate start of execution of the programme
  - → Intermediate code only needs to be generated once (stored in cache)
- Disadvantage: hot spots → it is necessary to wait for missing parts that have not yet been translated.
- *JVM:* identifies frequently accessed parts of the computer programme (*hot spots*) during execution and compiles *only* those parts. → »adaptive optimisation«.

Method

- From problem to programme



Translated from:

Balzert (2013): Java: Der Einstieg in die Programmierung

Example (1)

- Problem / task:
  Calculation of the value of goods; the quantity available, the net price and the value added tax (VAT) are given

- Algorithm:

```
1. Read in quantity, netPrice and VAT
2. Calculate: valueOfGoodsNet = quantity * netPrice
3. Calculate: ValueOfGoodsGross = valueOfGoodsNet*(VAT+100)/100
4. Output the valueOfGoodsGross.
```

Example (2)

Problem-oriented programming language

```
read(quantity);
read(netPrice);
read(VAT);
valueOfGoodsNet = quantity * netPrice;
valueOfGoodsGross = valueOfGoodsNet * (VAT+100)/100;
print(valueOfGoodsGross);
```

Machine-oriented programming language

```
INPUT quantity
INPUT netPrice
INPUT VAT
LOAD quantity
MUL netPrice
STORE valueOfGoodsNet
LOAD VAT
ADD 100.0
DIV 100.0
STORE valueOfGoodsGross
OUTPUT valueOfGoodsGross
```

Machine language programme:

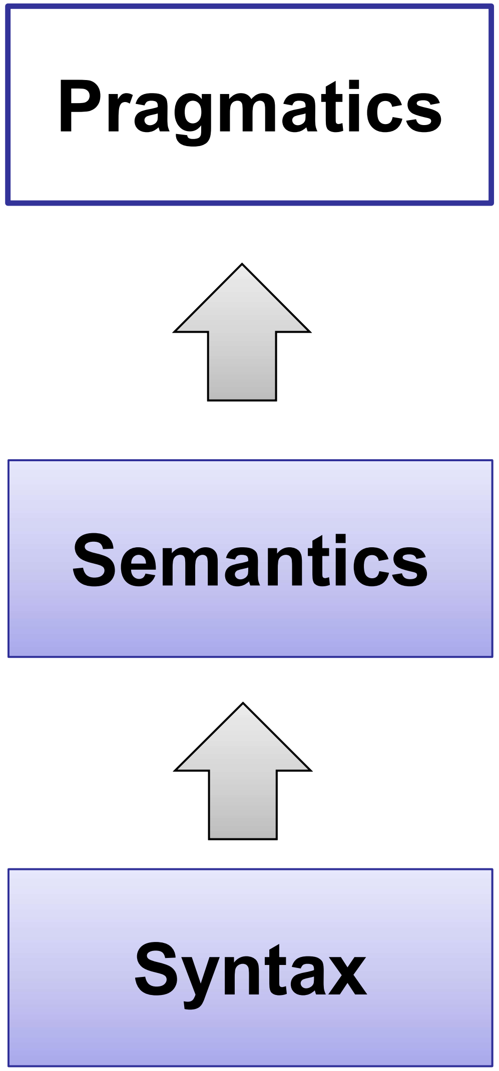
Binary code consisting of binary characters (bits: 0 and 1). Binary code is executed by the processor

Viewing levels for programming languages

Technische Hochschule Rosenheim

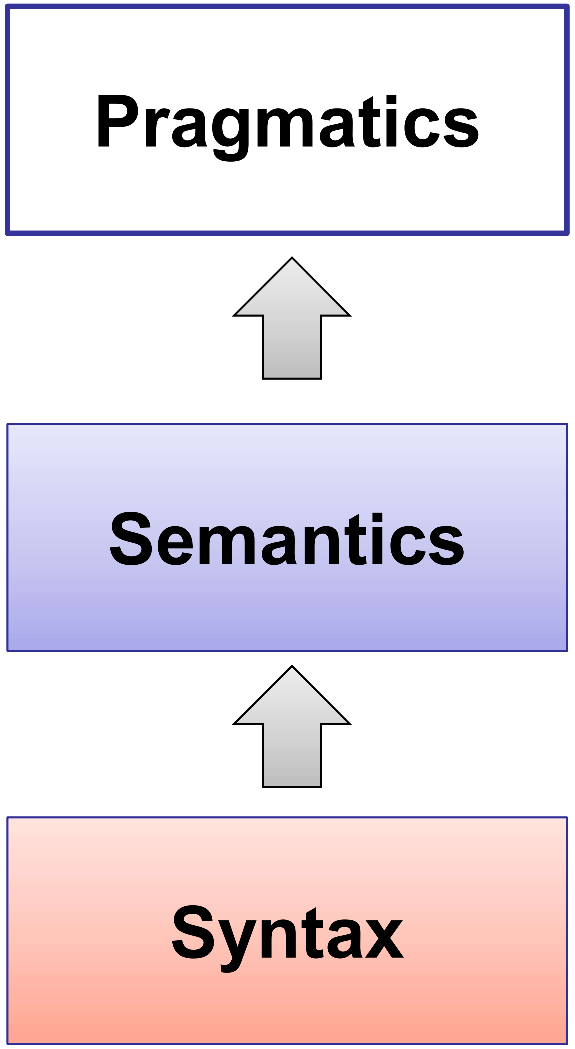
In practice, the pragmatic view is also relevant!

Traditionally two levels



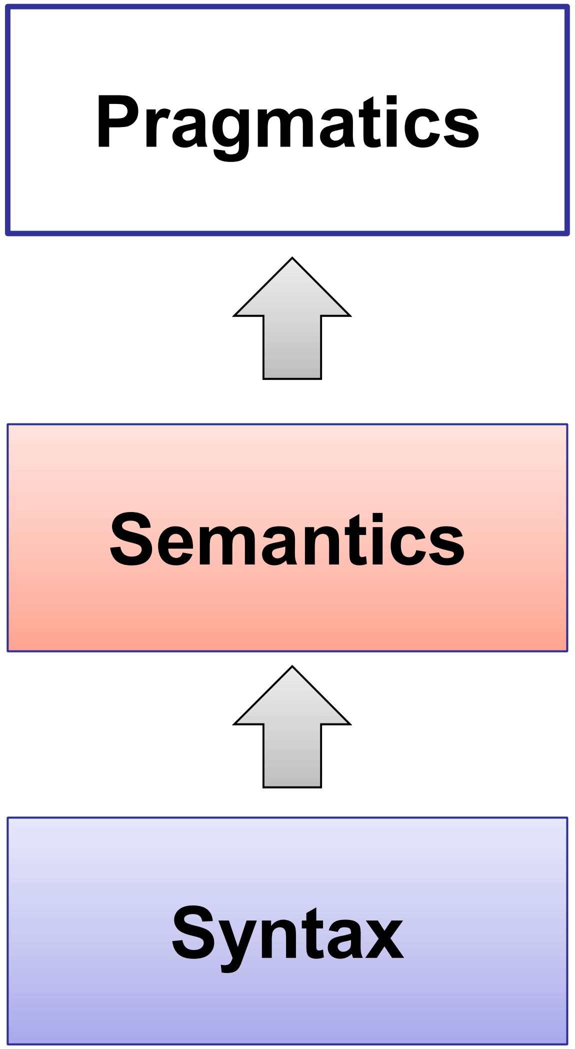
Syntax

- Syntax = notation, representation
  - Regulates spelling and grammar
    - Notation for individual words
    - Sentence construction
    - Use of special characters and fixed keywords
    - Order of elements in the programme
- Compiler checks syntax compliance
  - It checks very strictly!
  - Aborts when errors are detected, then does *not* generate binary code
- Syntax errors are relatively harmless, because no executable programme is created



Semantics

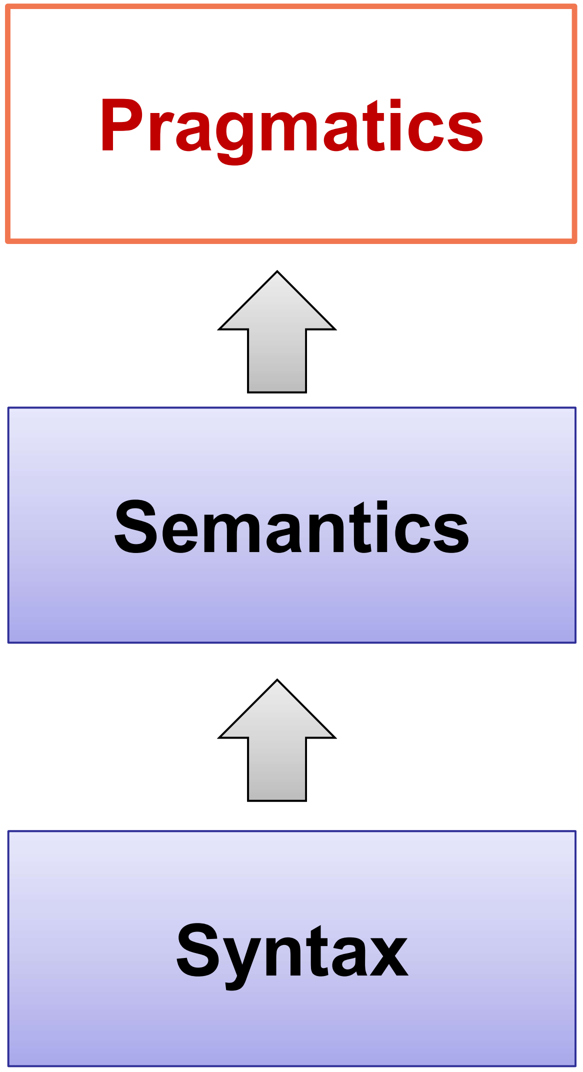
- Semantics = meaning
  - Defines meaning of syntactically correct programme fragments
  - Not everything that is syntactically correct is also semantically meaningful!!!
- Problem
  - A semantic error is an error in the meaning of the programme
  - Semantic errors can sometimes only be detected early on (e.g. during testing)
  - The rest occur as runtime errors (i.e. during execution of the programme), possibly with serious consequences
- Errors in semantics can be extremely dangerous!!!



Pragmatics

- Pragmatics = qualitatively meaningful use
  - Quality criteria
  - Proven and tested templates and patterns
  - Requires experience, time and lots of practice

- Not every programme that works is also good quality!



Exercise – Important terminology

- Live exercise
  - Complete Task 1 on the live exercises sheet “Introduction”.
  - Complete the first four sentences of the text in Task 2.
  - You have 5 minutes.

Programming Basics

Technische Hochschule Rosenheim

Chapter 1: Introduction

Categorisation of Java

- Problem-oriented and object-oriented programming language

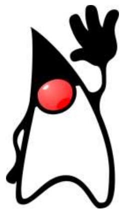
Language constructs are chosen in such a way that high-level / simpler programming is possible

Concepts of object-oriented programming such as classes, objects and inheritance are supported

Java - short historical outline

Technische Hochschule Rosenheim

- 1990: Predecessor "Oak"
  - Software for interactive television and consumer electronics devices
  - Goal: Reliable, compact, platform independent
- 1993: "Oak" becomes Java
  - Further development for the World Wide Web
  - Focus: Internet applications (predecessors of applets)
- 1996: Java JDK 1.0 is released
  - Implementation within Netscape Navigator
- Class library is constantly expanded
- 2008: Android appears, which is based on Java, among other things.
- 2009: Oracle acquires Sun.
  - Current version: Java Standard Edition 17
  - Look here for more details



Different types of Java programmes

- Java applications
  - run independently on a computer system
  - also: Android apps
- Jakarta Servlets (formerly Java Servlets)
  - are run on a web server, usually triggered by commands in an HTML document
- Java applets *(historical)*
  - are loaded from a web server via the Internet and run in a web browser

⇨ From mobile phone programme to business applications

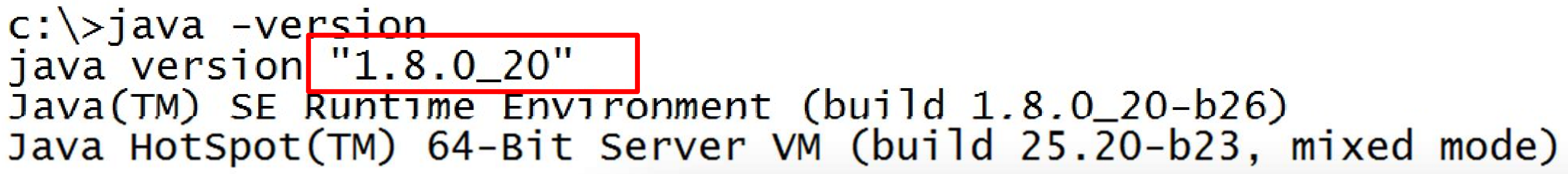
Java - editions

- Java runs on different systems (platforms, operating systems)
- There are three different bundled Java systems for different target groups
  - Standard Edition (SE)
    - Java for use on desktop systems
  - Enterprise Edition (EE)
    - Java for use on servers
  - Micro Edition (ME)
    - Java for use on devices with limited resources, e.g. personal digital assistant (PDAs) or mobile phones
- The Java *language* is the same in all editions – but with a *different* selection and scope of the tools and libraries supplied

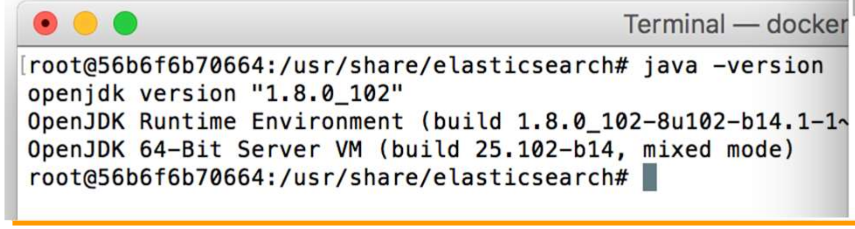
Java - versions

- There are multiple versions, which are upward compatible
  - 1.1 – 1.5 (became version 5)
  - Java 6, Java 7 and Java 8
  - Oracle JDK - Java 11/12/13 … currently **Java SE 17**

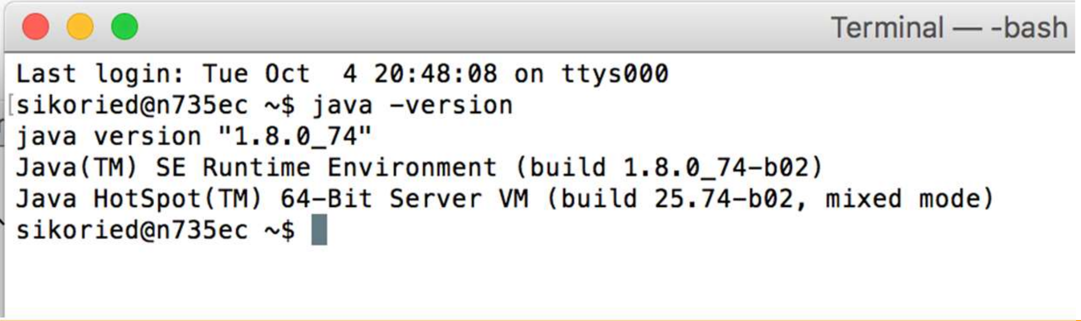
- Check your installed version



```
c:\>java -version
java version "1.8.0_20"
Java(TM) SE Runtime Environment (build 1.8.0_20-b26)
Java HotSpot(TM) 64-Bit Server VM (build 25.20-b23, mixed mode)
```

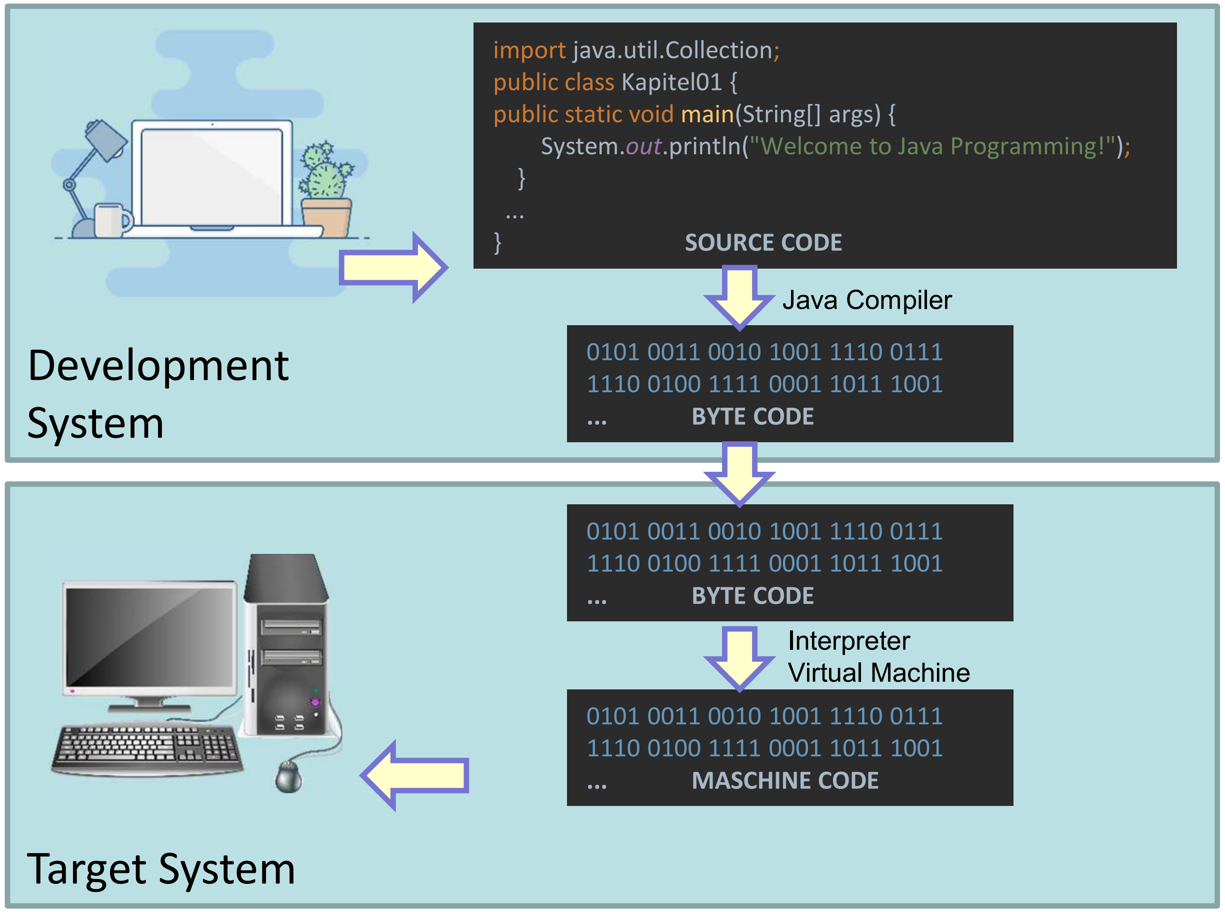


```
root@56b6f6b70664:/usr/share/elasticsearch# java -version
openjdk version "1.8.0_102"
OpenJDK Runtime Environment (build 1.8.0_102-8u102-b14.1-1~
OpenJDK 64-Bit Server VM (build 25.102-b14, mixed mode)
root@56b6f6b70664:/usr/share/elasticsearch#
```



```
Last login: Tue Oct  4 20:48:08 on ttys000
sikoried@n735ec ~$ java -version
java version "1.8.0_74"
Java(TM) SE Runtime Environment (build 1.8.0_74-b02)
Java HotSpot(TM) 64-Bit Server VM (build 25.74-b02, mixed mode)
sikoried@n735ec ~$
```

Java - characteristic features (1)



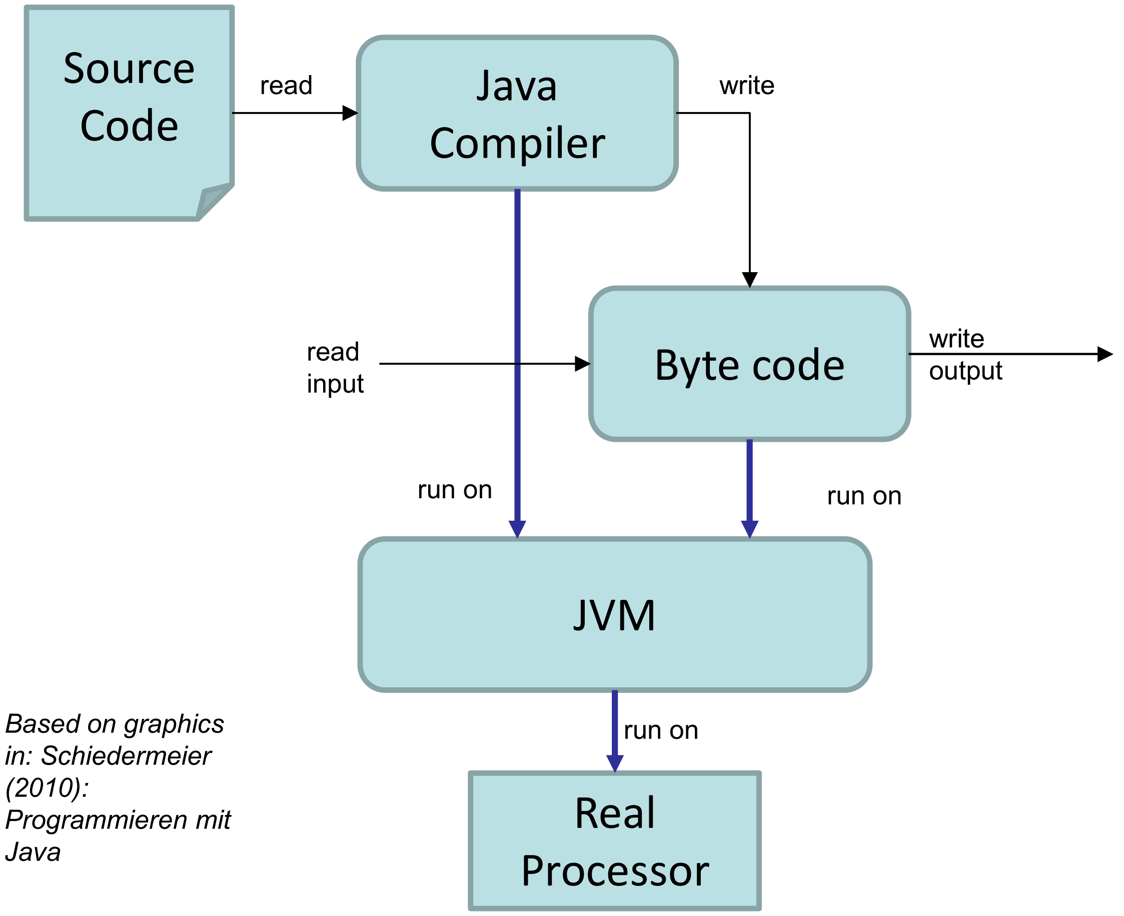
```
import java.util.Collection;
public class Kapitel01 {
public static void main(String[] args) {
        System.out.println("Welcome to Java Programming!");
    }
  ...
}
```

Combination of a compiler and an interpreter system

Based on the graphic from : Habelitz (2012): Programmieren lernen mit Java

Java - characteristic features (2)

- Processing model



Based on graphics in: Schiedermeier (2010): Programmieren mit Java

Java is suitable for:

– Platform independent, robust and secure programmes
– Distributed systems and networks
– Internet-based applications

Java is not suitable for:

– Very hardware-related programmes, such as drivers
– No direct access to hardware - only accessible via VM

Assessment of the JVM approach

- Advantages:
  - Platform independence: translated bytecode runs unmodified on any system (provided that JVM and the required resources are available)
  - Programmes can be used virtually anywhere, from PDAs right through to supercomputers
- Disadvantages:
  - In addition to the actual programme, the JVM also runs (higher resource usage)

  But:

  - Resources are constantly getting cheaper
  - Programmes are becoming increasingly complex
  - Instead of "slow or fast", the focus is on "doable or not"

Exercise – Java programming language

Technische Hochschule Rosenheim

- Live exercise
  - Complete sentences from (5) up to and including (8) of the text in Task 2.
  - You have 3 minutes.

Programming Basics

Technische Hochschule Rosenheim

<u>Chapter 1: Introduction</u>

Required tools (1)

Execution of finished/pre-compiled Java programmes:

- **JRE** (Java Runtime Environment)
  - JVM (Java virtual machine)
  - Runtime library
  - Typically as a `.jar` file: bundle/archive of multiple subclasses and programmes

Required tools (2)

Write, compile and execute your own Java programmes:

- Text editor or word processor, e.g. Notepad++
- JDK (Java Development Kit)
  - JRE
  - `javac` compiler and
  - other tools for developing new Java programmes

What's happening with Java?

- Currently, **version 8u201** is the latest Java version for the virtual machine (JRE), for which development can take place using the free *Java SE Development Kit 8u201.*
  https://java.com/de/download/

  Old licensing conditions
  Oracle grants you a non-exclusive, non-transferable, limited license without license fees to reproduce and use internally the Software completely and unmodified for the sole purpose of running Programs.

- **Version 15** is the latest stable version of the SDK.
  https://www.oracle.com/technetwork/java/javase/downloads/index.html

  New licensing conditions:
  Furthermore, You may not use the Programs for any data processing or any commercial, production, or internal business purposes other than developing, testing, prototyping, and demonstrating your Application.

- Alternative: OpenJDK
  https://jdk.java.net/

Quiz

- What is the name of the source file where the following programme should be stored?

```
public class Output {

   public static void main(String[] args) {
      System.out.println(" Good Luck! ");
   }

}
```

A – GoodLuck.java
B – Output.class
C – Output.java
D – HelloWorld.txt

Quiz

To compile the file Output.java in the DOS window, which of the following command must be entered?

A – javac Output.java

B – java Output

C – java Output.java

Folie 39

LH1 The following command must be entered in order to compile the Ausgabe.java file in the DOS window:
Lauren Hahn; 16.08.2021

Quiz

Technische Hochschule Rosenheim

What interprets the Java bytecode?

A – Java Standard Edition

B – IntelliJ

C – OpenJDK

D – Java Virtual Machine

Quiz

The main method is the point at which a programme is started.

A – Correct

B – Incorrect

Quiz

Why must the bin directory of Java be set in the system path?

A - By setting the path, the compiler is executed automatically.

B - This enables the Java programmes to be called from anywhere in the console.

C - This will install an IDE.

D - To annoy us.

Quiz

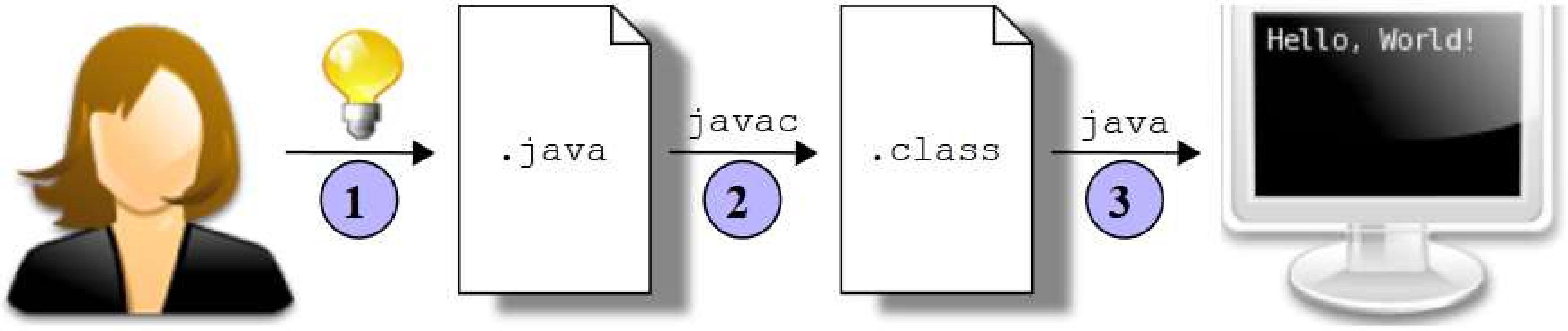
For a Java programme to compile, the class name must be different from the file name.

A – Correct

B – Incorrect

Required steps

- From the creation of a programme to the execution of a Java programme, three steps must always be followed:
  1. Creating the source code
  2. Compiling the source code into the bytecode
  3. Starting the programme by passing the bytecode to the Java interpreter (executing bytecode)



Simple example programme

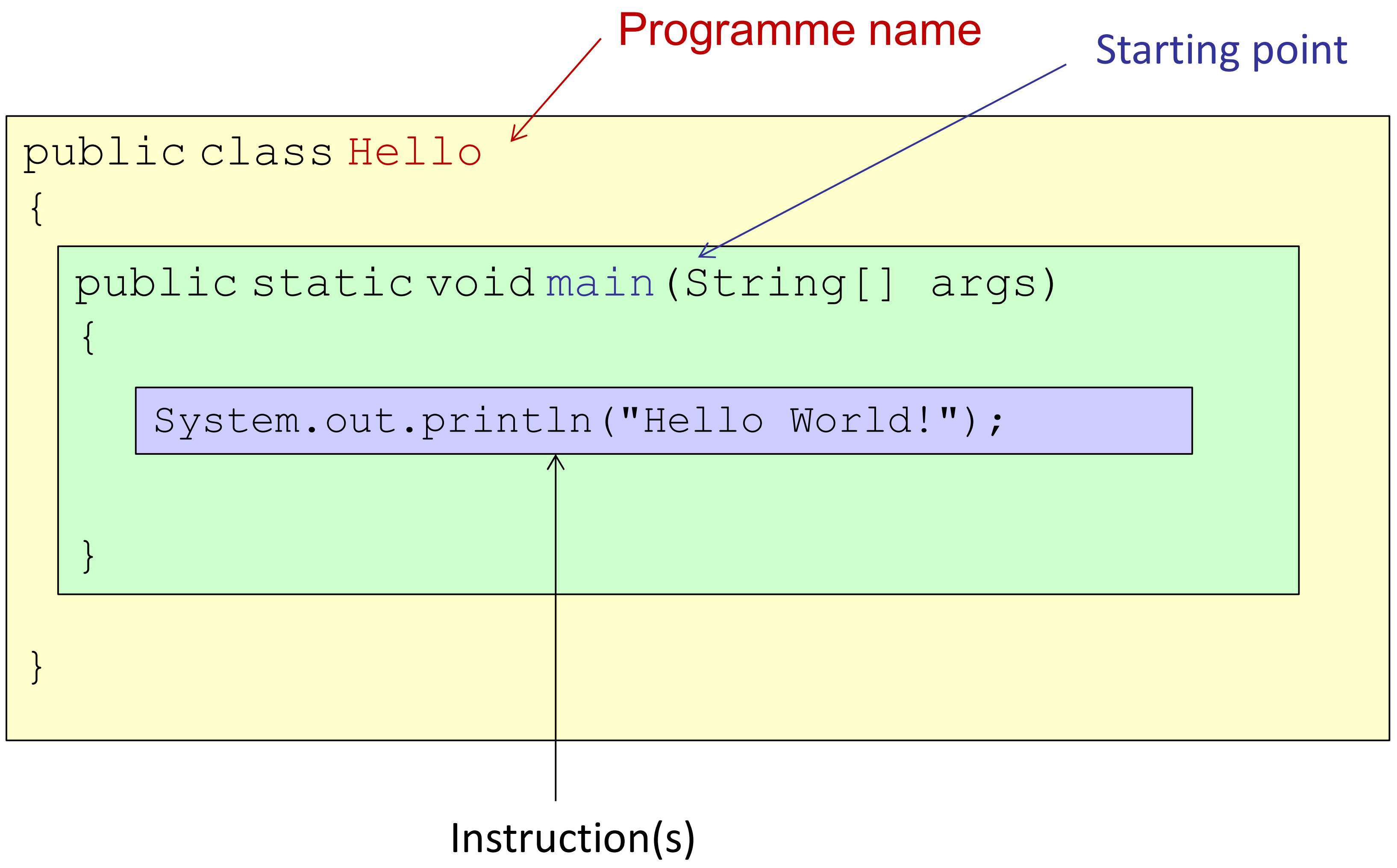
- The programme `Hello` displays the text "Hello World!" on the screen:

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

- Save the source code in a text file named `Hello.java`

Important: pay attention
to upper/lower case!
Always!

Structure of a simple Java programme (Java source code)



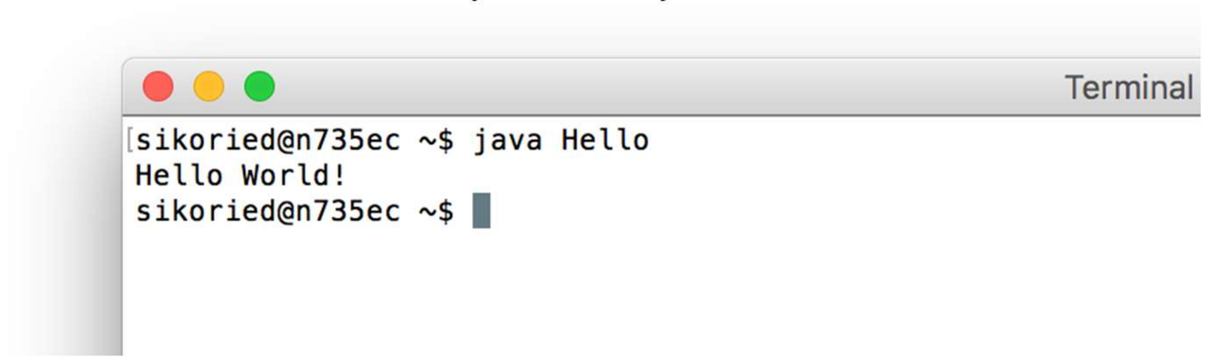
```
public class Hello
{
  public static void main(String[] args)
  {
      System.out.println("Hello World!");

  }

}
```

Compile and start

- Java compiler *compiles* source code into bytecode

```
$ javac Hello.java
```

- Java JRE *interprets* bytecode



```
[sikoried@n735ec ~$ java Hello
Hello World!
sikoried@n735ec ~$ 
```

Let's take a look at the given example …

- Demo on PC

Aim: Easy-to-read source code

- Source code is plain text
  - Readable for humans and machines (i.e. the compiler)
  - Readability for humans is decisive for usability
    - Error correction / maintenance
    - Extensions
    - Exchanges in the project
    - …

Compiler ensures readability for machines, ***not*** for humans!!!

- Maintainability
  - Readable, non-functional source code can be corrected.
  - Functional, unreadable source code is a dead end for development!

Formatting programmes

- Guidelines for easy-to-read programmes:
  - All statement blocks are enclosed in brackets
  - All lines within a pair of brackets are indented to the right by a fixed number of characters, e.g. two spaces, one tab stop (IDE takes care of this!)
  - In the line containing a closing curly bracket, there is usually nothing else
  - As a rule, one statement (instruction) per line
- Comments improve readability
  - More complicated numerical or logical operations
  - Forks in the control flow
  - "Explain it to me while I read it"

Comments in programmes

- Three different types:
  - Line comments (`//comment`)
    - From **//** to the end of the line
  - Block comments (`/*comment*/`)
    - From **/*** to the next ***/**
    - Block comments may not be nested
  - Javadoc comments (`/**comment*/`)
    - From /** to the next */
    - Can only be used in certain places
    - Enable generation of HTML documentation using Javadoc

Exercise – Java programmes

- Live exercise
  - Complete sentences from (9) up to and including (18) of the text in Task 2.
  - You have 5 minutes.

Integrated development environments (IDEs)

- Tools:
  - Text editor, ideally several windows or tabs
  - Console to compile and execute
  - Analysis of the output
    - Compiler: syntax errors
    - Programme output: semantics errors
- Integrated development environments
  - Syntax highlighting
  - Integrates the compiler and output analysis
  - Helps with navigation through large projects
- Popular IDEs: **IntelliJ**, NetBeans, Eclipse, Visual Studio Code

Get to know **IntelliJ**

- IntelliJ can be downloaded from this source:
- https://www.jetbrains.com/de-de/idea/download/#section=windows
- Free Edition is the Community edition.
- The Ultimate Edition is also free for students. Follow the steps listed here: https://www.jetbrains.com/de-de/community/education/#students

Let’s try these steps:

- Creation of a new project
- Input the source text, compile, execute
- Typical errors on an example

Technische Hochschule Rosenheim

Exercise – Java programmes

- Live exercise
  - Find the mistakes in Task 3.
  - You have 5 minutes.