Computer Science for Engineers
Exercise 1

Organisation
Introduction to Java

Prof. Dr. Dr.-Ing. J. Ovtcharova
Dipl. Wi.-Ing. Dan Gutu
14th of November 2008
Content of the exercises

• Programming (Java)
  • Basics – SW development
  • Eclipse - IDE (Integrated Development Environment)
  • Elementary data types, variables
  • Object orientation: classes, inheritance
  • Objects and methods, threads
  • Iteration statements
  • Data structures (Collections) & algorithms
  • GUI (Graphical User Interface)

• Software Engineering / UML
  • Source Code versus UML
What is Software Development??
Software Development

• Software Development is more than programming
  - Different requirements, depending on different interests
    ▪ Security, Multi-user, ergonomy
  - Documentation and program
  - Architecture and program extension

- **Software Development Lifecycle**
  ▪ Planning / requirements analysis
  ▪ Implementation
  ▪ Release and maintenance (Updates, Versions)
  ▪ Leave the market
Ex. planning and requirements

- How the customer explained it
- How the Project Leader understood it
- How the Analyst designed it
- How the Programmer wrote it
- How the Business Consultant described it

- How the project was documented
- What operations were installed
- How the customer was billed
- How it was supported
- What the customer really needed
Ex. Development, software quality

Life is like a baboon’s ass; colorfull and full of shit. The same thing goes for computers. Something just happened, as you’ve probably figured yourself. I could of course give you the error code, but I know that those kind of messages usually tell you nothing else than the fact that you want to kick your computer in the face [had it had a face]. So instead I’ll just tell you to click on OK, restart the program and open it again. Then try clicking on “Error Finder” found under Tools. It should tell you if there is something wrong with the database. If not, then it’s probably just a resulst of bad programming skills. On the other hand... I could just have closed the program down and give you a blue screen.

And by the way... the error code for whatever just happened is 429 but that probably doesn't tell you anything. And this is what the code is supposed to mean: ActiveX component can’t create object

Have a nice day!

---

Error!!!

Could not Submit Control File!
Call Tommy!

Contivity VPN Client

Disconnect Europe and Middle East?

ImgBurn

Please be patient, I heard you the first time!
What-Is Programming

• Which is the structure of the machine? Which is the structure of the program?
  - Software as a “machine” that runs on a computer

• “Construction” of software (analogies)
  - Create the „big picture“, create the components, assemble the system
  - Include standard components (see frameworks)
  - Create the steps that lead to the satisfy the requirements
  - Optimize the components
Outline of the First Exercise

Exercise 1

1. General Java Information
2. Characteristics of Java
3. Java Data Types
First Cup of Java
• 1991: Sun Microsystems started the project „Green“, in which consumer electronic prototypes, with a concentration on multimedia, were built.

• As a “by-product” a new, object orientated, interpreted language called “Oak” was developed, which was renamed to “Java” („strong coffee“) for marketing reasons.

• 1995: The first public demonstration was a striking success.

• 1996: Java-Boom. All of the major software developers order Java licenses. Standardisation efforts are made in order to secure platform independence for the future („Write once, run everywhere“)

• Since that time Java has continually been expanded and improved, not only in terms of the language, but also the APIs.
2007: Java becomes Open Source after GPL v.2

- A big part of Java source code is available as Open Source
  - Javac (Java Compiler)
  - Java SE (Standard Edition)
  - Kernel of JDK (Java Development Kit)
  - JVM (Java Virtual Machine)
  - JRE (Java Runtime Environment)

Danger of “uncontrolled growth”

- For compatibility: “Technology Compatibility Kits”
  - Uniformity of the Java platform
  - Certification as Java-compatible
General Java Information (1)

- Free and compatible with many platforms (Mac, Solaris, UNIX, LINUX, Windows, ... )
- Java Editions
  - J2SE: Java Standard Edition
    - Compiler, Runtime Environment, Software Development Kit
  - J2EEK: Java Enterprise Edition
    - Architecture for scalable, distributed applications
  - J2ME: Mobile Edition
    - Java for mobile devices
General Java Information (2)

- 2 central components:
  - **javac**: Java-Compiler, compiles source codes to JAVA-Bytecode, checks syntax and correctness of source code
  - **java** *(Java-Runtime)*: Interpreter, interprets JAVA-Bytecode, „runs the class“. „Virtual Machine“.
  - JAVA-Bytecode is not directly executable, not being a machine language
- **JDK**: Java Development Kit - standard library with data structures and algorithms that are used very often („do not re-invent the wheel each time“)
- **JRE**: Java Runtime Environment – Container that runs the bytecode
Java and other languages

• Better structured than C++
  • No compromises as for C++ (compatibility issue for C)
  • Hiding of complex language properties:
    • Pointer
    • Managed Code (Garbage Collector)
    • Multiple inheritance
    • Operator overloading
  • Real platform independence, compared to C#:
    • “C# runs on every Windows platform”
Outline of the First Exercise

Exercise 1

1. General Java Information
2. Characteristics of Java
3. Java Data Types
Characteristics of Java

• **Platform independent:**
  Runs on any platform under arbitrary operating systems, if a Java interpreter is installed

• **Robust:**
  Contains mechanisms to avoid uncontrolled system crashes

• **Strictly object oriented (OO) programming language:**
  There are only classes, methods, objects, and variables, no mixed code is possible, like in C++

• **Distributed:**
  A Java application can open and access arbitrary remote objects on the internet

• **Multithreaded:**
  Supports “multi-threading”: a task can be executed simultaneously on different parallel threads
Strictly object oriented programming

**Classic programing languages**

- **O - O Code**
  - Klasse A
  - Klasse C
  - Class B
  - Class C
  - Self defined classes
- **Compiler**
  - Machine code
  - 010010010010
  - 010101010001
  - 000011101001
  - 011010101001

- **O - O and mixed procedural code**

- **Java**
  - **O - O Code**
    - Class A
    - Class C
    - Self defined classes
  - **Compiler**
    - Java Bytecode
    - 010010010010
    - 010101010001
    - 000011101001
    - 011010101001
  - **Error message**

- **O - O and mixed procedural code**

- **Error message**

---

*Prof. Dr. Dr.-Ing. Jivka Ovtcharova - CSE – Exercise 1 - WS 08/09*
Robust

- **Mechanisms for error processing (Exception Handling):**
  control mechanisms to catch unexpected situations

- **No critical pointer structures:**
  - Java has no pointers as they are known from other programming languages. In order to accomplish object orientation, references are used. References are of an unchangeable type and are stored in a variable
  - A reference (pointer) is either NULL or points to a valid object
  - Access to NULL-references are recognized and trapped by the Java interpreter

- **Automatic storage de-allocation (Garbage Collection):**
  Objects are deleted automatically, if they are not referred by other variables or attributes any more. Ensures that no pointer refers to „nothing“.
Java source code → Java compiler → Java bytecode → Java interpreter → Platform independent
Conventional programming techniques

- One source code
- Multiple versions or run-time code

Java - Programming

- One source code
- One version of run-time code (Java Bytecode)

Binary file (Pentium)
Binary file (Macintosh)
Binary file (SPARC)
Comparison between Java and Pascal

Java
- compiler
  - Virtual Machine
  - Java byte code
  - Stack
  - PC, SP
  - Command interpreter
  - Virtual Machine

Pascal
- compiler
  - Machine code
  - CPU
  - executes

Java code
- generates
  - Java byte code

Pascal code
- generates
  - Machine code

010010010010
010101010001
000011101001
011010101001

011010010010
000101010011
110011101001
Example: Applications and Applets

Java Applet:

1. Java application
2. Java Interpreter executes
3. Java byte code
4. WWW-Server transfers request
5. executes Java bytecode
6. HTML with Java byte code
7. WWW-Client
8. HTML with Java byte code
Internet Based Applications

- A Java application (applet) can be embedded in an HTML page.
- A Java-capable browser can download the HTML from the internet and run the embedded applet.

Applet storage

Local execution of applets
• „Counterpart“ to the Applet

• Java application, that runs on a server and accepts queries from applets
  - Processes the query on the server (performance)
  - Encapsulates the query on the server

• Runs inside a Servlet Container (Application Server)
Java Frameworks

• Spring
  - Application Framework - Basic structure for Java applications
  - Assist „Best Practices“ for Software architectures
  - Simplifies the development of distributed applications

• Hibernate
  - Persistence framework, simplifies database access
  - Database must not compulsory be adressed in SQL
  - Overrides the „Impedance Mismatch“

• Strut
  - Framework for the Presentation Layer
Potential Java Applications

• Minus:
  - As an interpreted language, Java has a run-time handicap. Although it is about 10-20 times faster than other interpreted languages, it is still 2-10 times slower than the compiled machine code produced by other languages.

• Plus:
  - Can process many applications on the internet or intranet with relatively high performance.
  - Run-time can be shortened by using just-in-time compilers to compile the bytecode while it is being loaded.
  - Reduction of administration problems in client-server environments.
  - Programming of graphical environments is platform independent.
  - Web sites can be made interactive even with low bandwidth and storage
Outline of the First Exercise

Exercise 1

1. General Java Information
2. Characteristics of Java
3. Java Data Types
Basic elements in Java

- Variables
  - Ex.:
    - \( F = m \times a \)
      - \( F \,[N] = m \,[kg] \times a \,[m/s^2] \)
      - \( 392.4 = 50 \times 9.81 \)
    - Name: \( F \); Value: 392.4 ; Typ: \([N]\)
    - „A table is made of a board and four legs“
      - \( \rightarrow \) Table = Board + 4x Legs (as data types)

- Variables must be defined:
  - The name and the data type are determined
  - A value is assigned at initialization

Ex.:

\[
\text{double } \text{Force; double } \text{Mass}
\]
/**
 * converts a collection of Jena Ontology Classes to OntClasses
 * @param ontClasses the jena OntClasses to convert
 * @return the cs3d ontology classes
 * @throws ConverterException when the conversion blows
 */

protected Set<FeatureType> convertJenaClassToConcept(
    Collection<OntClass> ontClasses) throws ConverterException {

    Iterator<OntClass> iter = ontClasses.iterator();
    Set<FeatureType> result = new HashSet<FeatureType>();
    while (iter.hasNext()) {
        OntClass ont = iter.next();
        FeatureType ftResult = this.convertJenaClassToConcept(ont);
        result.add(ftResult);
    }
    return result;
}
Naming Rules (mandatory):
- Names are made of letters, digits and _ (number_2, aField)
- First character: letter, following characters according to above rule
- In principle, infinitely long names are possible
- Case sensitive
- Reserved keywords may not be used as names

Naming Conventions:
- First letter of variable names: lower case
- First letter of method names: lower case
- First letter of class names: upper case
- All letters of constant names: upper case
- Use long and meaningful names (numberOfIterations, MyFrame1, MAXIMUM)
The Data Type of a variable defines how it is represented in memory. We have:

- **primitive data types**
  - Numerical types:
    - Integers:
      - byte: $-2^7$ to $2^7-1$,
      - short: $-2^{15}$ to $2^{15}-1$,
      - int: $-2^{31}$ to $2^{31}-1$,
      - long: $-2^{63}$ to $2^{63}-1$,
  - Floating point types:
    - float: ca. $1.4 \times 10^{-45}$ to $3.4 \times 10^{38}$
    - double: ca. $4.9 \times 10^{-324}$ to $1.7 \times 10^{308}$
  - Logical data type:
    - boolean: false, true
  - Character:

- **non-primitive data types (reference)**
  - Character string: String (is a class)
  - Arrays
  - Classes.
A variables data type defines the possible operations that can be applied to it (or operations that the variable can be used for)

- numerical data types: arithmetic operator (+,-,*,/,%)
- logical data types: logical operators (&, |, !)
- specially defined operators for character (-string)

Example: Division operator / and %:

<table>
<thead>
<tr>
<th>Type</th>
<th>Expression</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>int / int</td>
<td>14 / 5</td>
<td>2</td>
</tr>
<tr>
<td>int % int</td>
<td>14 % 5</td>
<td>4</td>
</tr>
<tr>
<td>double / int</td>
<td>14.0 / 5</td>
<td>2.8</td>
</tr>
<tr>
<td>int / double</td>
<td>14 / 5.0</td>
<td>2.8</td>
</tr>
<tr>
<td>double / double</td>
<td>14.0 / 5.0</td>
<td>2.8</td>
</tr>
</tbody>
</table>