Recap: IO and Exceptions

- Input and output
  - Streams: Pipes through which data flows
    - When something has consumed, it's no longer there
    - Need to be flushed and closed at the end
  - InputStream/OutputStream: byte-wise operations
  - Readers/Writers: Used on top of streams to operate on characters
- Things can go wrong, even if our program works well
  - Many error sources with I/O: Files, disks, networks can fail
  - Exception handling
    - Mechanism to handle unexpected errors
    - `try {} catch (EX) {}`
  - Exceptions are objects of class `java.lang.Exception`

Session 14: Generics, Code Style, Java Standard Library, Closing

Softwaretechnologie: Java I

Nils Reiter
nils.reiter@uni-koeln.de

January 25, 2023

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

Section 1

Exercise

Section 2

Java Standard Library

Introduction

- Programming language core: Rather small
- A few types, some statements, some syntactic elements

Introduction

- Programming language core: Rather small
- A few types, some statements, some syntactic elements
- Libraries
  - Collections of code, useful for all kinds of things
  - Many languages have such libraries
  - To avoid reinventing the wheel, we should use them

Java Standard Library

Interesting packages

- `java.io` – Input and output
- `java.lang` – Core functions
- `java.math` – Mathematical functions
- `java.net` – Handling networks and connections
- `java.text` – Simple text processing
- `java.util` – Various utility functions, in particular collections
  - Will be discussed in depth in the summer term
- `java.awt`, `javax.swing` – Classes for graphical user interfaces

Section 3

Generics

Introduction

- Some classes in the library have < ... > in their name
- These are generic classes

Example

- We know arrays cannot be increaed in length, which is inconvenient
- We can copy the content of an array into a new longer array
- We could define a class that encapsulates an array and deals with the copying

demo

Generics

- Many classes are “container classes”: They deal with objects of specific types, but do nothing specific with the type
  - E.g., `MyArray<T>` – with `T` being a parameter that can be filled with any reference type
- If we define them as generic, the type of the objects they deal with becomes like a variable
- The variable is filled with a concrete type at compile time

Section 4

Code Style

Introduction

- Interaction between programmers is easier, if they adhere to common style
- Style: How to write and format variables, methods, classes etc.
- Java Code Style
  - No strict rules, but guidelines
  - I.e.: There are exceptions, and you're not punished for violating them
- Offical document from 1997:
  `https://www.oracle.com/technetwork/java/codeconventions-150003.pdf`
- In Eclipse, you can select the code and use Source > Format to automatically format the code nicely

Java Code Style

- `CamelCase` is used for combining words (instead of underscore or dot)
- Class and interface names start with an upper case letter (`MyArray`) and are nouns
- Methods names start with a lower case letter (`get()`) and are verb phrases
- Variables start with a lower-case letter and are as long as it needs to be for clarity
  - Variable names like `a` are dispreferred
- Indentation should be used to make the structure of the program visible
  - Substatements of a statement or declaration should be indented
  - Indentation should be four spaces wide
- Avoid lines longer than 80 characters
- Files longer than 2000 lines are cumbersome and should be avoided.
- ...

Section 5

Closing

Learning Programming

- Learning to program is hard and takes time
- It helps to
  - Regularly do it
  - Talk about it
  - Be stubborn
  - Think formalistic
  - Be fearless and disrespectful
  - Read documentation
  - Try to understand your mistakes
- It's ok to make mistakes

Looking Ahead

What happens in the summer term

- Version control (= git)
- Recursion
- Data structures
- Unit testing
- Efficient programming
- Multithreading
- ...

Looking Ahead

What happens in the summer term

- Version control (= git)
- Recursion
- Data structures
- Unit testing
- Efficient programming
- Multithreading
- ...

Programming Ideas for the Break

- A simple game such as Tic Tac Toe
  - Turn-based games are simpler than real time games
- Birthday predictor (read in a list of birthdays, calculate the next round anniversaries)
- Make algorithmic art (e.g., ASCII art)