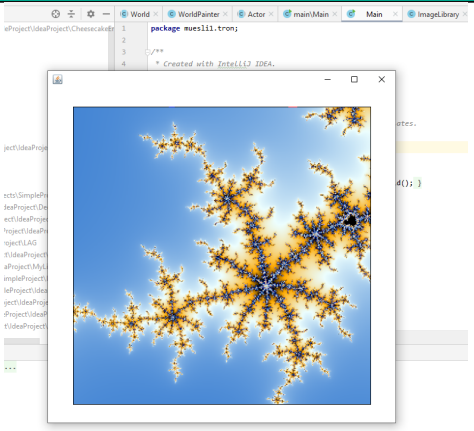
FOP Recap #13

File-IO



Moin Moin!

Das steht heute auf dem Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

File-IO
- Wie? Was? IO?
- Habe ich da `Stream` gehört?
- `BufferedReader` `readLine`
- `BufferedWriter` `write`
- mit Streams

`IOException?`

`String`

`String` <-> Zahl

File-IO

Wie? Was? IO?

TECHNISCHE UNIVERSITÄT DARMSTADT

- IO steht für Input/Output
- Heißt: Lesen und Schreiben von Dateien
- Unterschiedliche Typen:
  - Textdateien (z.B. `.txt`)
  - Bilddateien (z.B. `.png`, `.jpg`)
  - Sounddateien (z.B. `.ogg`, `.mp3`)
  - Videodatein (z.B. `.mp4`)
  - ....

File-IO

Wie? Was? IO?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Typische Klassen mit denen man hier arbeitet:
  - `File`
  - `InputStream`
  - `OutputStream`
  - `Reader`
  - `Writer`
  - `IOException`
  - ....

File-IO

Habe ich da `Stream` gehört?

TECHNISCHE UNIVERSITÄT DARMSTADT

- `InputStream` und `OutputStream` waren zuerst da!
- Teilen sich dieselbe Grundidee: Funktionieren wie ein Fließband
- Haben jedoch nichts miteinander zu tun!

File-IO

BufferedReader readLine — Gut für Textdateien

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
// ! Absoluter Pfad ist nicht empfehlenswert !
File myFile = new
    ↪  File("C:\\Users\\Marc\\Daten\\TestVerzeichnis\\TextDatei.txt");
FileReader fileReader = new FileReader(myFile);
```

```java
// Relativer Pfad zum Programm
FileReader fileReader = new FileReader("TextDatei.txt");

BufferedReader bufferedReader = new BufferedReader(fileReader);

String firstLine = bufferedReader.readLine();

bufferedReader.close();
```

File-IO

`BufferedWriter` `write` — Gut für Textdateien

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
// Relativer Pfad zum Programm
FileWriter fileWriter = new FileWriter("TextDatei.txt");

BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);

bufferedWriter.write("First line of the document!\n");

bufferedWriter.close();
```

File-IO

`BufferedWriter` `write` — Gut für Textdateien

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
// Relativer Pfad zum Programm
FileWriter fileWriter = new FileWriter("TextDatei.txt");
BufferedWriter writer = new BufferedWriter(fileWriter);

writer.write(String.valueOf(true));
writer.write(",")
writer.write(String.valueOf(5.2));
writer.write(",")
writer.write(String.valueOf(10));

writer.close();
```

File-IO

mit Streams — `Files.lines`

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
Stream<String> stream = Files.lines(Paths.get("TextDatei.txt"));

List<String> lines = stream.collect(Collectors.toList());

stream.close();
```

Das steht heute auf dem Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

File-IO

`IOException?`
- Das Problem
- `try-with-resources`

`String`

`String` <-> Zahl

IOException?

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
try {
    FileReader fileReader = new FileReader("TextDatei.txt");

    BufferedReader bufferedReader = new
     ↪  BufferedReader(fileReader);

    String firstLine = bufferedReader.readLine();

    // !!!
    bufferedReader.close();
}
catch(IOException e) {
    e.printStackTrace();
}
```

IOException?

Das Problem

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
FileReader reader = null;
try {
    reader = new FileReader("TextDatei.txt");
    // ....
}
catch(IOException e) {
    e.printStackTrace();
}
finally {
    // Close Reader
}
```

IOException?

Das Problem

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1  FileReader reader = null;
2  ....
3  finally {
4      if(reader != null) {
5          try {
6              reader.close();
7          }
8          catch(IOException e) {
9              e.printStackTrace();
10         }
11     }
12 }
```

IOException?

try-with-resources

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1  try(FileReader fileReader = new FileReader("TextDatei.txt");
2      BufferedReader bufferedReader = new
       ↪  BufferedReader(fileReader)) {
3
4      String firstLine = bufferedReader.readLine();
5
6  }
7  catch(IOException e) {
8      e.printStackTrace();
9  }
```

IOException?

try-with-resources

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
try(BufferedReader bufferedReader = createANewReader()) {

    String firstLine = bufferedReader.readLine();

}
catch(IOException e) {
    e.printStackTrace();
}
```

Das steht heute auf dem Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

File-IO

IOException?

```
String
   split
   contains
   indexOf
   substring
```

String <-> Zahl

String
split

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
String text = "Dieser Text ist toll!";

String[] split = text.split(" ");

System.out.println(Arrays.toString(split));
// -> [Dieser, Text, ist, toll!]
```

String
split

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
String text = "A,B,C,";

String[] split = text.split(",");

System.out.println(Arrays.toString(split));
// -> [A, B, C]
```

String
contains

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
String text = "A,B,C,";

System.out.println(text.contains("B"));
// -> true
System.out.println(text.contains("Z"));
// -> false
```

String
indexOf

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1  String text = "A,B,C,";
2
3  System.out.println(text.indexOf("B"));
4  // -> 2
5  System.out.println(text.indexOf("Z"));
6  // -> -1
7  System.out.println(text.indexOf(","));
8  // -> 1
```

String
substring

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
String text = "0123456789";

String a = text.substring(1);
// -> 123456789
String b = text.substring(2);
// -> 23456789
String c = text.substring(2, 5);
// -> 234
```

Das steht heute auf dem Plan

TECHNISCHE UNIVERSITÄT DARMSTADT

File-IO

`IOException`?

`String`

`String` <-> Zahl

String <-> Zahl

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
String text = "257";

// NumberFormatException?
int num = Integer.parseInt(text);

String other = "25.112";

// NumberFormatException?
double otherNum = Double.parseDouble(other);
```

String <-> Zahl

TECHNISCHE UNIVERSITÄT DARMSTADT

```java
int num = 257;

String text = String.valueOf(num);

double otherNum = 25.112;

String other = String.valueOf(otherNum);
```

Live-Coding!