

Exception

Terminal

Files & Path

P-Aufgaben



Folien: go.tum.de/904005

Exception

Exception

Terminal

Files & Path

P-Aufgaben

Exceptions sind Fehler, die bei unerwartetem Verhalten *geworfen* werden.

z.B. `NullPointerException`, `IndexOutOfBoundsException`, etc.

RuntimeException

Tritt unregelmäßig auf

Muss nicht speziell behandelt werden

unchecked

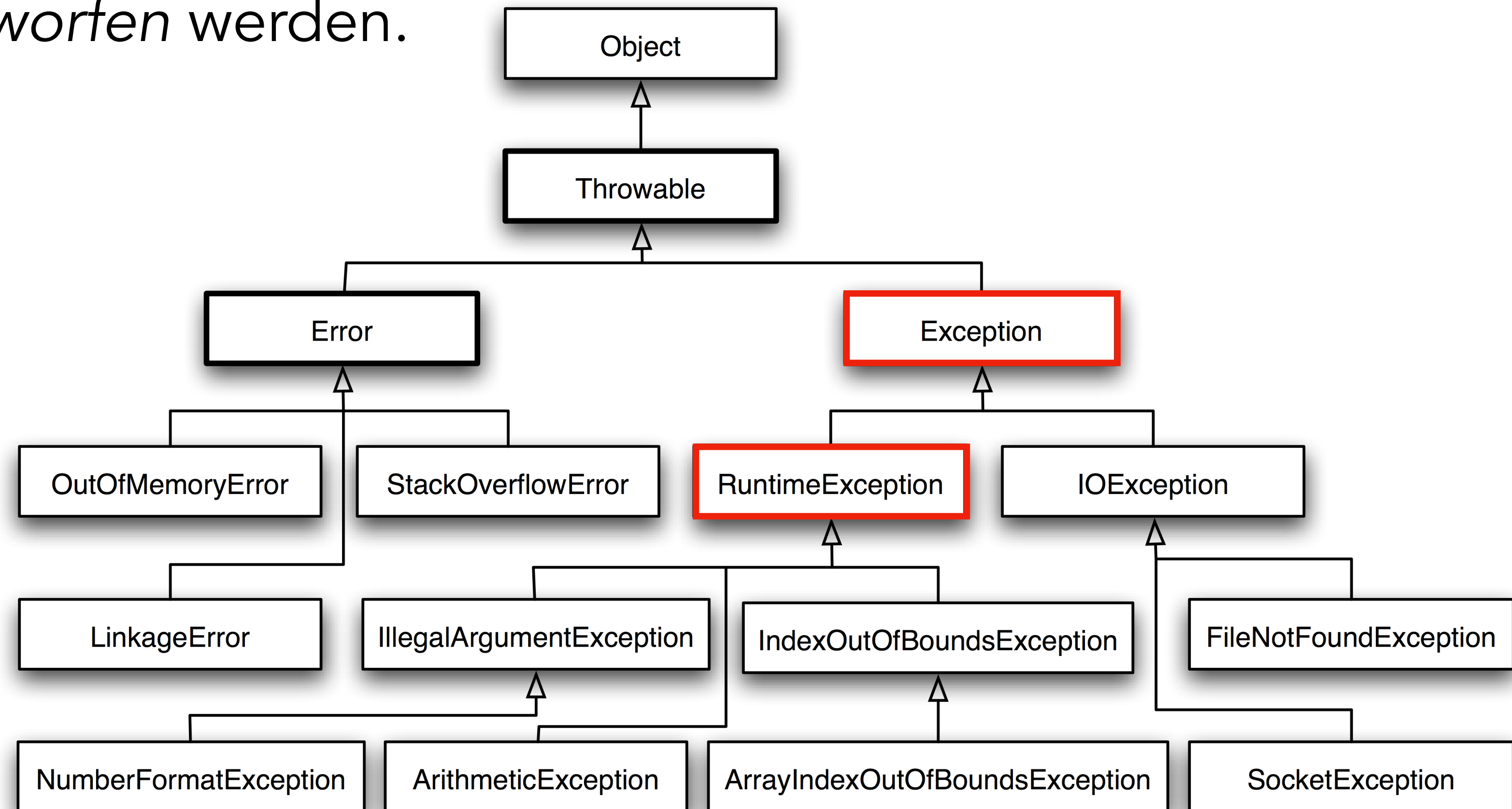
Exception

Tritt mit hoher Wahrscheinlichkeit auf
Methoden benötigen `throws`-declaration oder `try & catch`-Block

checked

Exception

Exceptions sind Fehler, die bei unerwartetem Verhalten *geworfen* werden.



Exception

Exception

Terminal

Files & Path

P-Aufgaben

Exceptions sind Fehler, die bei unerwartetem Verhalten *geworfen* werden.

RuntimeException, NullPointerException

```
public static void main(String[] args) {  
    String s = null;  
    write(s.length());  
    // throws NullPointerException  
}
```

Exception

Exception

Terminal

Files & Path

P-Aufgaben

Exceptions sind Fehler, die bei unerwartetem Verhalten *geworfen* werden.

Exception, IOException mit try & catch

```
public static void main(String[] args) {  
    ServerSocket s;  
    try {  
        s = new ServerSocket();  
    } catch (IOException exc) {  
        // Fehlerbehandlung  
    } finally { /*wird sicher ausgeführt*/ }  
}
```

Exception

Exception

Terminal

Files & Path

P-Aufgaben

Exceptions sind Fehler, die bei unerwartetem Verhalten *geworfen* werden.

Exception, IOException mit throws-declaration

```
public static void main(String[] args)
    throws IOException {
    ServerSocket s = new ServerSocket().
}

```

Fehler wird an aufrufende Methode weitergegeben und sollte dort behandelt werden.

Exception

Exception

Terminal

Files & Path

P-Aufgaben

Eine eigene Exception:

```
public class Fehler extends Exception {  
    private String txt; | Attribute  
    public Fehler(String txt) { |  
        super(); | Konstruktor  
        this.txt = txt;  
    }  
    public String toString() { |  
        return "Info: " + this.txt;  
    }  
}
```

Exception

Exception

Terminal

Files & Path

P-Aufgaben

Eine eigene Exception:

```
public static void main(String[] args) {  
    try {  
        foo();  
    } catch (Fehler f) {  
        write(f.toString());  
    }  
}
```

Try & Catch Block
(für Exceptions vom
Typ Fehler)

```
public static void foo() throws Fehler {  
    throw new Fehler("testFehler")  
}
```


Exception

Exception

Terminal

Files & Path

P-Aufgaben

Eine eigene Exception:

```
public static void main(String[] args) {  
    try {  
        foo();  
    } catch (Exception e) {  
        write(e.toString());  
    }  
}  
  
public static void foo() throws Fehler {  
    throw new Fehler("testFehler")  
}
```

Try & Catch Block
(für alle
Exceptions)

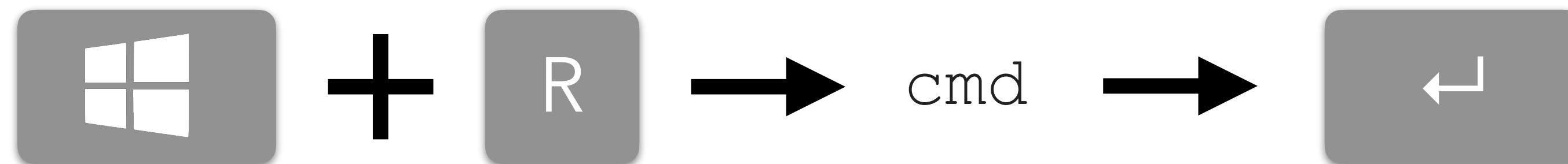
Terminal

Das Terminal starten

MacOs:



Windows



Terminal

Exception

Terminal

Files & Path

P-Aufgaben

Wichtige Terminal Befehle:

Befehl	Erklärung
<code>cd <path></code>	Navigiert in das angegebene Verzeichnis
<code>ls</code>	Listet Inhalte des aktuellen Verzeichnisses auf
<code>open <file></code>	Öffnet eine Datei
<code>cp <name> <newName></code>	Kopiert eine Datei
<code>mv <from> <to></code>	Verschiebt eine Datei
<code>mv <name> <newName></code>	Benennt eine Datei um
<code>mkdir <path></code>	Erstellt ein neues Verzeichnis (Ordner)
<code>rmdir <path></code>	Löscht leeres Verzeichnis

Terminal

Wichtige Terminal Befehle:

Befehl	Erklärung
<code>rm -R <path></code>	Löscht Verzeichnis inkl. Unterverzeichnisse
<code>sudo <command></code>	Führt Command als Admin aus
<code>top</code>	Listet alle aktiven Prozesse
<code>q</code>	Verlasse Prozess (z.B. top)
<code>clear</code>	Leert Terminal (von vorherigen Commands)
<code>help <command></code>	Hilfe zu einem bestimmten Befehl
<code>whatis <command></code>	Kurze Hilfe zu bestimmten Befehl
<code>exit</code>	Beendet die Terminal-Session

Terminal

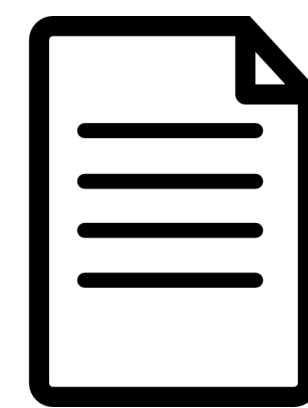
Exception

Terminal

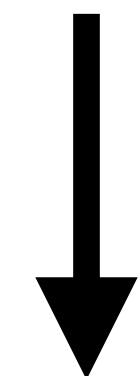
Files & Path

P-Aufgaben

Java Programme mit dem Terminal ausführen:



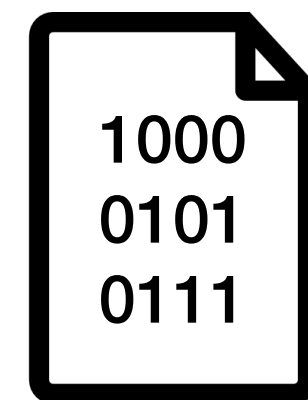
Sourcecode (.java Textdateien)



javac *<filename>*

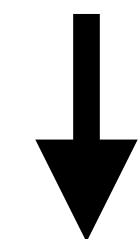
javac *.java

alle .java Dateien im
Verzeichnis kompilieren

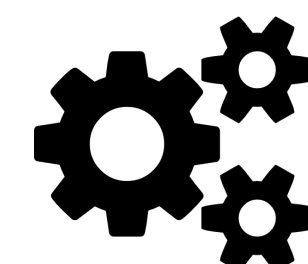


(Java-)Bytecode

```
public static void  
main(String[] args) {}
```



java *<filename>* *<args>*



Programm wird ausgeführt

Files & Path

Exception

Terminal

Files & Path

P-Aufgaben

Eine Möglichkeit, um in Java Dateien von 'außen' als Stream oder Liste zu erhalten und zu verarbeiten.

```
import java.nio.file.*;
import java.io.IOException;
try {
Files.lines(Path.of(args[0]))
    .forEach(System.out::println);
} catch (IOException e) {
    // Fehlerbehandlung
}
```

Files & Path

Exception

Terminal

Files & Path

P-Aufgaben

Eine Möglichkeit, um in Java Dateien von 'außen' als Stream oder Liste zu erhalten und zu verarbeiten.

Nützliche Methoden auf Files:

- **static** `Stream<String> lines(Path path);`
- **static** `List<String> readAllLines(Path path);`
- **static** `String readString(Path path);`

Nützliche Methoden auf Path

- **static** `Path of(String first, String... more);`

Möglicherweise leere Folge weiterer Strings, die mit einem Seperator '/' zu einem String zusammengesetzt werden

P10.01

Exception

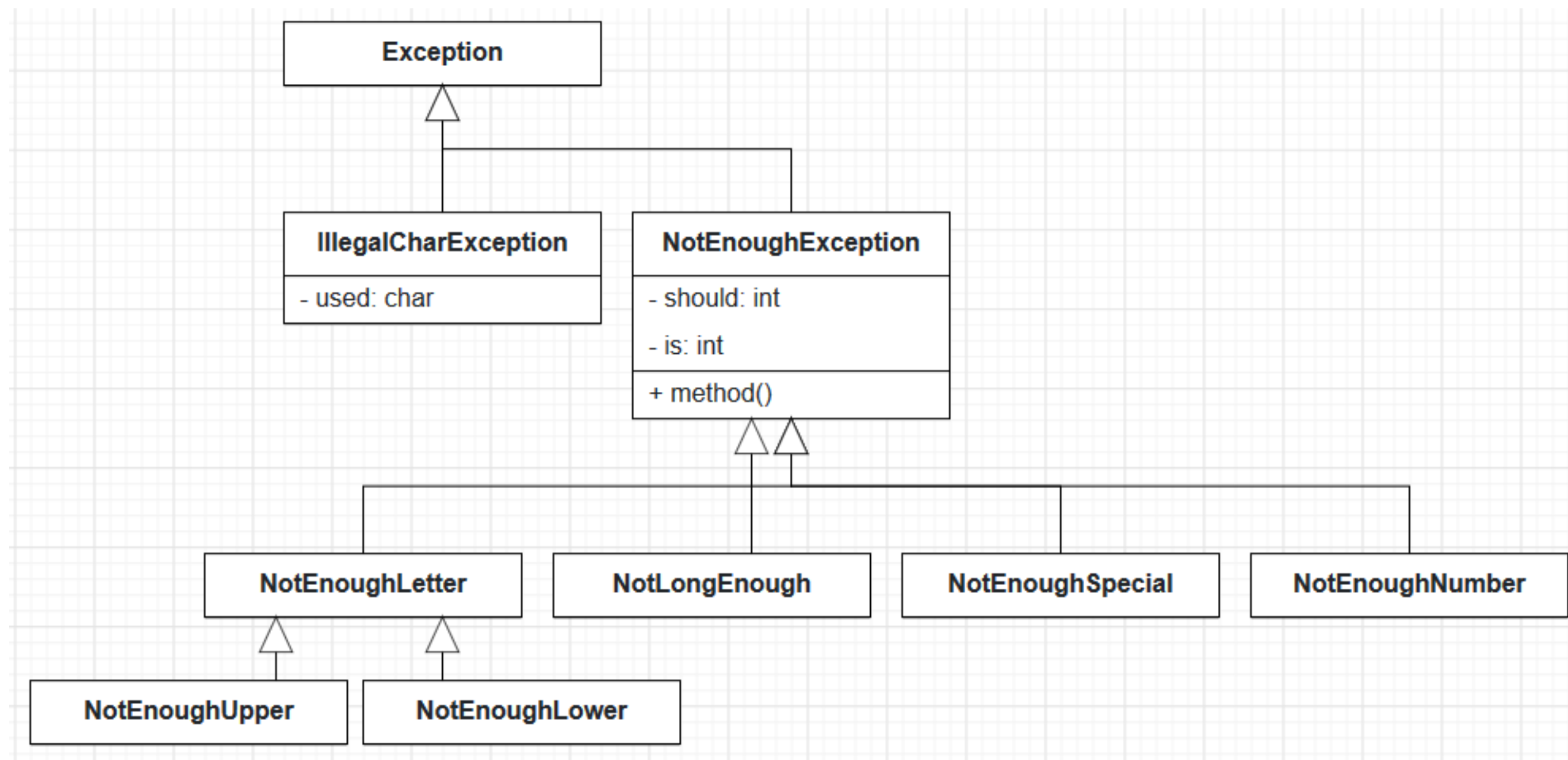
Terminal

Files & Path

P-Aufgaben

Passwortüberprüfung, bei Nichteinhaltung der Konvention Fehler werfen.

```
checkFormat (String pwd) ;
```



P10.02

Exception

Terminal

Files & Path

P-Aufgaben

Konkatenieren zweier Dateien

- Das Programm soll über das Terminal gestartet werden

```
java <filename> <in1>, <in2>, <out>
```

- Alle *checked* Exceptions sollen abgefangen werden und eine sinnvolle Fehlermeldung verursachen.

Hinweis: Stream.concat

```
Stream.concat(Stream<? extends T> a,  
              Stream<? extends T> b)
```

P10.03

Exception

Terminal

Files & Path

P-Aufgaben

Telefonbuch, Exkurs HTML:

```
<html>
  <body>
    <table>
      <tr><td>Z1, S1</td><td>Z1, S2</td></tr>
      <tr><td>Z2, S1</td><td>Z2, S2</td></tr>
    </table>
  </body>
</html>
```

Exception

Terminal

Files & Path

P-Aufgaben

Telefonbuch, Exkurs HTML:

```
<html>
```

```
  <body>
```

```
    <table>
```

```
      <tr><td>Z1, S1</td><td>Z1, S2</td></tr>
```

```
      <tr><td>Z2, S1</td><td>Z2, S2</td></tr>
```

```
    </table>
```

```
  </body>
```

```
</html>
```

Jedes Element muss
geöffnet und
geschlossen werden.

P10.03

Exception

Terminal

Files & Path

P-Aufgaben

Telefonbuch, Exkurs HTML:

```
<html>
  <body>
    <table>
      <tr><td>Z1, S1</td><td>Z1, S2</td></tr>
      <tr><td>Z2, S1</td><td>Z2, S2</td></tr>
    </table>
  </body>
</html>
```

table → Tabelle
tr → table row
td → table data

P10.03

Exception

Terminal

Files & Path

P-Aufgaben

Telefonbuch, csv-Datei in HTML-Tabelle parsen.

```
// TODO
```

```
- interface CsvEntry
```

```
- class PhoneBookEntry
```

```
- String toHtml()
```

```
- CsvEntry cellsToEntry(String[] cells)
```

```
- class CsvToHtml
```

```
- CsvEntry parseCsv(String line)
```

```
- convert(Path in, Path out)
```