

Programmierung
Prof. Dr.-Ing. Nobert Fuhr

Gudrun Fischer
Sascha Kriewel
programmierung@is.informatik.uni-duisburg.de

Übungsblatt Nr. 9

Aufgabe 18: Ausnahmen

- (a) Schreibe eine Exception `FalscheIsbnException`. Erweitere die Klasse `Buch` so, dass bei der Erzeugung eines neuen Buchs eine `FalscheIsbnException` geworfen wird, falls es sich nicht um eine gültige ISBN handelt.
- (b) Das folgende kleine Programm realisiert die Eingabe von Buchtitel und ISBN über die Eingabeaufforderung. Dabei wird die `do`-Schleife solange durchlaufen, bis eine gültige ISBN eingegeben und das Buch erfolgreich angelegt wurde.

Teste das Programm mit Deiner erweiterten Klasse `Buch`. Der Compiler beschwert sich über eine *unreported exception java.io.IOException; must be caught or declared to be thrown*.

Das bedeutet, dass die Methode `readLine()` des `BufferedReader` eine Ausnahme werfen kann, die bisher noch nicht behandelt wird. Wirf die Ausnahme weiter und fange sie in der `main`-Methode geeignet ab.

Listing 1: BuchEingabeTastatur.java

```
1 import java.io.*;
2
3 public class BuchEingabeTastatur {
4
5     private static BufferedReader in =
6         new BufferedReader (
7             new InputStreamReader(System.in));
8
9     public static Buch readBuch() {
10
11         Buch b;
12
13         do {
14             String titel = "", isbn = "";
15             System.out.print("Der_Buchtitel:_");
16             titel = in.readLine();
17             System.out.print("Die_ISBN:_____");
```

```

18     isbn = in.readLine();
19
20     try {
21         b = new Buch(titel, isbn);
22         return b;
23     } catch (FalscheIsbnException e) {
24         System.out.println("Ungueltige_ISBN:_" +
25                             "Neue_Eingabe.");
26     }
27 } while (true);
28 }
29
30 public static void main(String [] args) {
31     System.out.println("Mein_Buch:_" + readBuch());
32 }
33 }

```

Aufgabe 19: Dateiaus- und eingabe

Swing bietet eine sehr komfortable und einfache Möglichkeit, Dateien auszuwählen, mit der Klasse `javax.swing.JFileChooser`. Lies in der API nach, wie man diese Klasse benutzt, und konsultiere gegebenenfalls das Tutorium *How to Use File Choosers* (das Tutorium ist auf der Vorlesungsseite verlinkt). Die folgende ganz simple Anwendung zum Ausprobieren bietet nur einen Auswahldialog und gibt nach der Benutzerauswahl den Dateinamen aus:

Listing 2: DateiWahl.java

```

1 import javax.swing.*;
2
3 class DateiWahl {
4
5     public static void main (String [] args) {
6         JFileChooser chooser = new JFileChooser ();
7         int returnVal = chooser.showOpenDialog (null);
8         if (returnVal == JFileChooser.APPROVE_OPTION) {
9             System.out.println ("Dateiname:_" +
10                                chooser.getSelectedFile ().getName ());
11         }
12     }
13 }

```

- (a) Benutze den `JFileChooser` und schreibe ein Programm, das eine benutzergewählte Textdatei einliest, jeder Zeile eine Zeilennummer voranstellt, und die Datei dann wieder zurückschreibt.
- (b) Die abstrakte Klasse `javax.swing.filechooser.FileFilter` bietet die Möglichkeit, einen Dateifilter (z.B. `'*.html'`) mit einem Dateiauswahldialog zu verbinden. Es gibt keine vorgegebene Erweiterung dieser Klasse,

schreibe nun eine für Textdateien. Dazu müssen die beiden abstrakten Methoden

- `boolean accept(File f)`
- `String getDescription()`

ausimplementiert werden. Der Filter soll Dateien mit der Endung `txt` unabhängig von Groß- oder Kleinschreibung akzeptieren.

- (c) Verknüpfe nun Deinen Dateifilter mit dem Dateiauswahldialog aus (a), so dass nur Textdateien zur Auswahl angezeigt werden.