



Aufgabenblatt 4

Aufgabe 1 (Exceptionmechanismus, 5 Punkte)

Auf der Homepage der Veranstaltung ist der Quelltext einer Java-Klasse zu Aufgabe 1 verlinkt. Was wird ausgegeben, wenn die Methode `testMethod1()` dieser Klasse aufgerufen wird? Erläutern Sie, wie es zu dieser Ausgabe kommt.

Aufgabe 2 (Exceptionmechanismus, 5 Punkte)

Auf der Homepage der Veranstaltung ist ein Java-Programm zu Aufgabe 2 verlinkt. Welche Ausgabe erzeugt dieses Programm? Erläutern Sie, wie es zu dieser Ausgabe kommt.

Aufgabe 3 (Exceptions deklarieren und behandeln, 10 Punkte)

Auf der Homepage der Veranstaltung finden Sie ein kleines aber nicht ganz korrektes Java-Programm, das zum Kopieren einer Datei gedacht ist. Es soll (von der Kommandozeile aus) wie folgt aufgerufen werden können:

```
> java FileCopy quelldatei zieldatei
```

Wenn Sie versuchen, dieses Programm zu übersetzen, meldet der Java-Compiler Fehlermeldungen, z. B.:

```
FileCopy.java:7: unreported exception java.io.FileNotFoundException;  
                                     must be caught or declared to be thrown  
    FileInputStream fis = new FileInputStream(args[0]);
```

- (a) Informieren Sie sich zunächst mit Hilfe der Java-API-Dokumentation, welche Exceptions von den Anweisungen in der `main`-Methode des Programms ausgelöst werden könnten.

Angenommen, Sie möchten sich um mögliche Exceptions im Programm nicht weiter kümmern und einfach nur mit möglichst wenigen Änderungen des Quelltextes das Programm übersetzen und ausführen können. Was müssen Sie tun? Oder genauer: Was fehlt im Quelltext? Es ist nur eine Kleinigkeit. Passen Sie das Programm entsprechend an!

- (b) Wenn Sie Teilaufgabe (a) richtig gelöst haben, haben Sie zwar ein Programm, mit dem Sie Dateien kopieren können, es kommt aber zu unschönen Programmabbrüchen, wenn ein Benutzer Fehler beim Programmaufruf macht. So würde z.B. der Versuch, eine nicht existierende Datei zu kopieren, zu einem Programmabbruch mit der folgenden Ausgabe führen:

```
> java FileCopy gibtsNicht irgendwohin
Exception in thread "main" java.io.FileNotFoundException:
                                     gibtsNicht (No such file or directory)
    at java.io.FileInputStream.open(Native Method)
    at java.io.FileInputStream.<init>(FileInputStream.java:106)
    at java.io.FileInputStream.<init>(FileInputStream.java:66)
    at FileCopy.main(FileCopy.java:8)
```

Implementieren Sie einen Exception-Handler, der alle möglichen geprüften Exceptions innerhalb von `FileCopy` behandelt. Achten Sie auch darauf, dass keine `ArrayIndexOutOfBoundsException` (dies ist keine geprüfte Exception) mehr auftritt.

Aufgabe 4 (Selbstdefinierte Exception-Klasse, 10 Punkte)

- (a) Definieren Sie eine Klasse `NegativeNumberException` als geprüfte Exceptions. Sehen Sie mindestens zwei Konstruktoren vor, einen parameterlosen Konstruktor und einen Konstruktor, der eine Fehlermeldung als Parameter akzeptiert.

`NegativeNumberException` werden wir in den folgenden Teilaufgaben verwenden.

- (b) Schreiben Sie eine Methode, die zu einer endlichen Folge von nicht-negativen reellen Zahlen a_1, \dots, a_n den Wert

$$\sqrt{\prod_{i=1}^n a_i}$$

als Resultat liefert. Prüfen Sie dabei, ob eine der Zahlen a_i negativ ist und lösen Sie in diesem Fall eine `NegativeNumberException` aus.

- (c) Schreiben Sie ein kleines Programm, das die Argumente der Kommandozeilen in Zahlen wandelt, für diese Zahlen die Methode aus Teilaufgabe (b) aufruft und das Ergebnis auf die Standardausgabe ausgibt. Beispiel:

```
> java WurzelVonProdukt 1 2 3 4
Ergebnis: 4.89897949
```

Behandeln Sie in Ihrem Programm alle möglichen Exceptions, die auftreten können, insbesondere `NegativeNumberException`.

Hilfe zur Lösung der Aufgaben erhalten Sie in den Übungen der KW 17. Abgabe der Lösungen bis **Sonntag, 28. April 2013, 24 Uhr** via E-Mail an die Adresse `dat-alg13@mail.inf.h-brs.de`.