

# Übungen zu „Programmieren“

## Exceptions & geschachtelte Klassen

Profs. Dres. Rathke & Thies

In der nachfolgenden Übung beschäftigen wir uns mit Laufzeitfehlern. Die Programmiersprache Java integriert mit den so genannten Exceptions ein mächtiges Konzept für den Umgang mit potentiellen Laufzeitfehlern. Diese Übung gibt zunächst einen einführenden Überblick über die einzelnen Aspekte des Exception-Konzeptes.

1. Was sind *Exception* im Kontext von Java?
2. Welche vordefinierten Fehler-Klassen gibt es in Java?
3. Was ist der Unterschied zwischen *Ausnahmen(Exception)* und *Fehlern(Error)*?
4. Was sind die wichtigsten Methoden welche in der Klasse *Throwable* zur Verfügung stehen?
5. Wie sind *Exception* aufgebaut?
6. Definiere folgende Konstrukte: *try*, *catch*, *finally*, *throw* und *throws*
7. Kopieren sie folgendes Programm in Eclipse und führen es aus

```
public class Konto {  
  
    private double guthaben = 0;  
  
    public void abheben(double betrag) {  
        guthaben -= betrag;  
    }  
  
    public void einzahlen(double betrag) {  
        guthaben += betrag;  
    }  
}
```

```

public String toString() {
    return "Kontostand: " + guthaben;
}

public static void main(String[] a) {
    Konto k = new Konto();
    k.einzahlen(-100);
    k.abheben(20);
    System.out.println(k);
}
}

```

- a) Was gibt das Programm aus und wo liegt der Fehler?
- b) Erweitern Sie die Methode `public void einzahlen(double betrag)` mit einem Mechanismus der eine Exception wirft falls ein negativer Beitrag eingezahlt wird.

## 8. Kopieren sie folgendes Programm in Eclipse und führen es aus

```

public class TestAusnahme {

    public static void main(String[] args) {

        double d;

        d = Double.parseDouble("12,99");
        System.out.println("Der Inhalt des übergebenen Strings wurde in eine Gleitkommazahl umgewandelt.");
    }
}

```

- a) Was gibt das Programm aus und wo liegt der Fehler?
- b) Erweitern Sie das Programm mit einem try und catch Block und führen sie das Programm erneut aus.

In der nachfolgenden Übung beschäftigt sich mit dem Thema „innere Klassen“.

## 9. Welche Arten von verschachtelten Klassen gibt es in Java?

## 10. Welche Vorteile ergeben sich durch die Anwendung von inneren Klassen in Java-Programmen?

## 11. Wie unterscheiden sich Innere Klassen (nichtstatisch verschachtelte Klassen) zu von statisch verschachtelten Klassen?

## 12. Öffnen sie ihre Entwicklungsumgebung und erstellen sie ein Java-Programm das die nachfolgend beschriebenen Anforderungen erfüllt.

Als Vorarbeit für unseren Blog benötigen wir zunächst vier Klassen (TextBeitrag, BlogEintrag, Person und Kommentar).

- 1) Erstellen Sie zunächst eine Java-Datei mit dem Titel "TextBeitrag" in Ihrer Entwicklungsumgebung: Gehen Sie dabei für diese und die nächste Klasse wie folgt vor:: TextBeitrag hat die Attribute "Person autor", "Date datum" und "String text". BlogEintrag erbt von TextBeitrag und hat zusätzlich das Attribut "String Titel". Die Klasse Person kann frei gestaltet werden.
- 2) Die Klasse Kommentar soll als innere Klasse von BlogEintrag angelegt werden, ebenfalls von Text Beitrag erben und als einziges Attribut einen "BlogEintrag blogEintrag" haben
- 3) Erstellen Sie Getter und Setter für die jeweiligen Klassen. Überlegen Sie sich einen sinnvollen Konstruktor.

Test des Programms:

- a) Erstellen Sie einen zweite Java-Datei mit dem Namen "start.java"
- b) Implementieren Sie eine Main-Methode, um das Programm anschließend zu starten
- c) Erstellen Sie zuerst einen BlogEintrag um im Anschluss daran einen Kommentar zu diesem BlogEintrag zu erstellen. Geben Sie anschließend alle relevanten Werte aus.