

**Ausgabe:** 19.11.2018  
**Abgabetermin:** Montag, 26.11.2018, 11:00 Uhr

Prof. Dr.-Ing. Norbert Siegmund  
M.Sc. André Karge

Bitte lesen Sie die folgenden Informationen zum Übungsablauf **sorgfältig** durch.

Grundsätzlich – wenn nicht anders angegeben – sind die Lösungen zu den Übungen zu Einführung in die Programmierung jeden **Montag bis spätestens 11:00 Uhr** an André Karge per E-Mail zu schicken.

Schreiben Sie bitte im Betreff Ihrer E-Mail Ihre **Teamnummer** sowie die Nummer des Übungsblattes. In der E-Mail schreiben Sie bitte zusätzlich Ihren **Namen** und **Matrikelnummer**. Die Lösungen für Sie bitte als Java Dateien als Anlage hinzu. Es werden **keine** kompilierten Dateien, wie \*.class oder \*.jar angenommen.

Übungen müssen von **minimal zwei** und **maximal drei** Studierenden aus derselben Übungsgruppe in einem festen Team bearbeitet werden (Ausnahmen nur auf Anfrage beim Übungsleiter). Pro Team soll die Lösung nur einmal abgegeben werden. Aufgaben sollen **im Team gelöst** und nicht nur vom Team abgegeben werden. Sie müssen mindestens **50%** dieser Punkte für eine Zulassung zur Prüfung erreichen. Das **Abschreiben** identischer Lösungen wird jeweils mit 0 Punkten bewertet.

Bei Fragen oder Unklarheiten wenden Sie sich bitte **vor der Abgabe** des Übungsblattes an den Übungsleiter (per E-Mail oder persönlich). Es soll nie jemand sagen müssen: „Wir haben die Aufgabe nicht verstanden und konnten sie daher nicht bearbeiten.“

Weitere Informationen, wie aktuelle Ankündigungen, finden Sie online (<https://www.uni-weimar.de/de/medien/professuren/intelligente-softwaresysteme/lehre/>) unter Einführung in die Programmierung

## Aufgabe 1 Array Referenzen (4 Punkte)

Es sollen von sechs Primzahlen die Wurzel und das Quadrat berechnet werden und jeweils in einem Array gespeichert werden. Dazu wurde folgender Code implementiert:

```
double[] primeNumbers = {3, 5, 7, 11, 13, 17};

double[] sqrtOfPrimeNumbers = primeNumbers;
for (int i = 0; i < 6; ++i)
{
    sqrtOfPrimeNumbers[i] = Math.sqrt(sqrtOfPrimeNumbers[i]);
}

double[] squaredPrimeNumbers = primeNumbers;
for (int i=0; i<6; ++i)
{
    squaredPrimeNumbers[i] = squaredPrimeNumbers[i] * squaredPrimeNumbers[i];
}
```

Lassen Sie sich die Ergebnisse des obigen Codes ausgeben um deren Korrektheit zu überprüfen. Erklären Sie warum die Ergebnisse (nicht) korrekt sind und implementieren Sie gegebenenfalls eine korrekte Lösung.

## Aufgabe 2 Clone (6 Punkte)

Gegeben sind die Klassen `Sheep` und `Fur`. Jedes Schaf-Objekt hat neben seinem Namen als weitere Membervariable ein Objekt der Klasse `Fur`, welches die Eigenschaften seines Fells darstellt.

Ziel ist nun Schaf-Objekte zu klonen. In der Vorlesung wurde besprochen, dass es dafür zwei verschiedene Möglichkeiten gibt. Implementieren Sie die Methoden `public Sheep cloneShallow()` und `public Sheep cloneDeep()` in der Klasse `Sheep`. Erstellen Sie eine weitere Klasse `SheepCloner`, die ein Schaf-Objekt erzeugt und dieses klonet.

Beispiel:

```
Fur furOfDolly = new Fur(10, "white");
Sheep dolly = new Sheep(furOfDolly,"Dolly");
Sheep dollyShallowClone = dolly.cloneShallow();
Sheep dollyDeepClone = dolly.cloneDeep();
```

Was passiert mit den Klonen, wenn man bei Dolly die Farbe ändert? Erklären Sie die Auswirkungen.

Beispiel:

```
dolly.fur.color = "pink";
```

Die Klassen `Sheep.java` und `Fur.java` finden Sie auf der Seite der Professur unter `material_ex05.zip`.

## Aufgabe 3 Advent Advent (7 Punkte)

(a) Implementieren Sie eine Klasse `Adventskalender`.

- Die 24 Türchen (Array) sind mit Objekten vom Typ `Sweetie` gefüllt.
- `Sweetie` ist ein Aufzählungstyp. Es gibt: Lebkuchen, Plaetzchen, Praline, Schoko-Nikolaus, Gummibaerchen und Spekulatius.
- Der Konstruktor initialisiert die Türchen per Zufallsgenerator.
- Stellen Sie sicher, dass auf die Türchen nicht direkt von außerhalb der Klasse zugegriffen werden kann.
- Die Methode `public String openDoor(int day)` öffnet das entsprechende Türchen, falls `day` heute ist oder in der Vergangenheit liegt, sonst erfolgt die Rückgabe: Heute ist noch nicht der `day`. Dezember. (Hinweis: `(new java.util.GregorianCalendar()).get(java.util.Calendar.DAY_OF_MONTH)` ist der aktuelle Tag).
- Verbirgt sich hinter dem Türchen ein Schoko-Nikolaus wird zusätzlich ein Sound abgespielt (Verwenden Sie die Datei `MyAudio.java` und die `jingle-bells.wav` aus dem `material_ex05.zip` Archiv. Der Aufruf erfolgt folgendermaßen: `MyAudio.play("/jingle-bells.wav");` )

(b) Erstellen Sie eine weitere Klasse mit einer `main`-Methode zur Interaktion mit dem Benutzer. Es wird abgefragt, welches Türchen er öffnen will. Der Inhalt wird angezeigt. Und danach ob er noch weitere Türchen öffnen will, bis er dies verneint und das Programm so beendet.