

Übungsblatt 2

Abgabe: Die Abgabe Ihrer Lösung erfolgt über den Moodle-Kurs der Vorlesung (<https://moodle.hu-berlin.de/course/view.php?id=114664>). Nutzen Sie dort das im Abschnitt Übung angegebene **Aufgabenblatt 2: Vollmondberechnung und Flugsimulation**. Die Abgabe ist bis zum **21.11.2022** um **9:15 Uhr** möglich.

Hinweise:

- Achten Sie darauf, dass Ihre Java-Programmdateien mittels des UTF-8-Formats (ohne byte order marker (BOM)) kodiert sind und keinerlei Formatierungen enthalten.
- Verzichten Sie auf eigene Packages bei der Erstellung Ihrer Lösungen, d.h. **kein package**-Statement am Beginn Ihrer Java-Dateien.
- Quelltextkommentare können die Korrigierenden beim Verständnis Ihrer Lösung (insbesondere bei inkorrekten Abgaben) unterstützen; sind aber nicht verpflichtend.
- Testen Sie Ihre Lösung bitte auf einem Rechner aus dem Informatik Computer-Pool z.B. gruenau6.informatik.hu-berlin.de (siehe auch Praktikumsordnung: Referenzplattform)

Aufgabe 1 (Vollmondberechnung)

12 Punkte

In einem anderen Sonnensystem gibt es einen Planeten „HUpyster“ der sich genau wie die Erde verhält. Tage, Monate, (Schalt-) Jahre verhalten sich auf dem Planeten wie bei uns und auch die Kalender sind identisch. Allerdings ist der Mond des Planeten etwas schneller und benötigt exakt 29 Tage für einen Umlauf. Am 1. Januar 2022 sei dort Vollmond. Schreiben Sie ein Programm **Vollmond.java**, welches einen Kommandozeilenparameter n entgegennimmt und die ersten n Monate (jeweils in einer Zeile) ausgibt, in denen es **2-mal Vollmond gibt**. Die Jahreszahl wird hierbei vierstellig ausgegeben und gültige Monatsnamen bilden: Januar, Februar, März, April, Mai, Juni, Juli, August, September, Oktober, November und Dezember. Das Format für die Ausgabe des Programms ist:

```
YYYY, Monat-1
YYYY, Monat-2
...
YYYY, Monat-n
```

Der folgende Aufruf illustriert die Ein- und Ausgabe des Programms beispielhaft:

```
java Vollmond 2
2022, Januar
2023, Oktober
```

Hinweise:

- Für das Einlesen des Eingabeparameters können Sie die Anweisung `String n = args[0];` innerhalb der `main`-Methode verwenden:

```
public static void main(String[] args) {
    int n = Integer.parseInt(args[0]);
    // .... hier folgt Ihr weiteres Programm
}
```

- Ihr Programm kann eine valide Eingabe des Parameters n annehmen, d.h. die Eingaben bestehen immer nur aus den Ziffern 0-9. Ihr Programm muss keine Überprüfung auf Eingabefehler vornehmen.
- Beachten Sie zur Bestimmung von Schaltjahren folgende drei Regeln: (1) Die Jahreszahl muss durch 4 teilbar sein, (2) ist die Jahreszahl durch 100 teilbar, dann ist es kein Schaltjahr, es sei denn (3) die Jahreszahl ist durch 400 teilbar.

Aufgabe 2 (Simulation Fluggesellschaft)

8 Punkte

Die Flotte der Fluggesellschaft *HUB-Air* besteht aus lediglich einem Flugzeug. Das Flugzeug besitzt insgesamt 75 Plätze für Passagiere. Aus persönlichen Erfahrungen weiß die Geschäftsführerin der Gesellschaft jedoch, dass nicht alle Ticketkäufer*innen auch wirklich zum Flug erscheinen. Sie rechnet damit, dass ein/e Ticketkäufer*in nur mit einer Wahrscheinlichkeit von 0.92 tatsächlich erscheint. Basierend auf dieser Erfahrung, überbucht die Gesellschaft immer ihre Flüge und verkauft stets 78 Tickets.

Schreiben Sie ein Java-Programm `FlugSimulator.java`, welches n Flüge der Fluggesellschaft simuliert und für diese erfasst, wie häufig zu viele Passagiere zu einem Flug erscheinen. Gehen Sie in Ihrem Programm davon aus, dass für jeden Flug immer 78 Flugtickets verkauft werden.

Das Verhalten eines Passagiers, d.h. ob er/sie erscheint oder nicht, können Sie mit Hilfe von Zufallszahlen simulieren. Für die Erzeugung von Zufallszahlen besitzt Java die Methode `Math.random()`. Diese Methode liefert einen zufälligen `double` Wert der größer gleich 0.0 und kleiner 1.0 ist. Die erzeugten Zufallswerte sind gleichverteilt, das bedeutet, dass jeder Wert zwischen 0.0 und 1.0 mit gleicher Wahrscheinlichkeit auftritt. Gehen Sie davon aus, dass das Erscheinen der einzelnen Passagiere unabhängig voneinander ist, d.h. es gibt keine zusammengehörigen Gruppen wie Familien oder Firmenmitarbeiter*innen.

Die Anzahl der zu simulierenden Flüge n wird als Parameter an das Programm übergeben:

```
java FlugSimulator <n>
```

Das Programm gibt bei jeder Ausführung die Anzahl der Überbuchungen sowie deren relativen Anteil an allen Flügen aus. Ferner wird die durchschnittliche Anzahl an Passagieren, die wirklich erschienen sind, dargestellt. Eine Ausführung des Programms könnte bspw. wie folgt aussehen:

```
java FlugSimulator 150  
  
Überbuchungen: 5 (3.33%)  
  
Mittelwert: 67.5
```

Hinweise:

- Für das Einlesen des Eingabeparameters können Sie die Anweisung `int anzahlFluege = Integer.parseInt(args[0])` innerhalb der `main`-Methode verwenden:

```
public static void main(String[] args) {  
    int anzahlFluege = Integer.parseInt(args[0]);  
    // .... hier folgt Ihr weiteres Programm  
}
```

- Runden Sie den relativen Anteil der Flüge, die überbucht sind, auf 2 Nachkommastellen.
- Runden Sie den Mittelwert der Passagieranzahl auf eine Nachkommastelle.
- Ihr Programm kann eine valide Eingabe der Fluganzahl annehmen, d.h. es werden jeweils nur Ganzzahlen ≥ 1 an das Programm übergeben.
- Die dargestellten Beispiele illustrieren nur die Ausgabeformatierung des Programms und sind **nicht als Musterlösungen** zu betrachten. Beachten Sie zudem, dass durch die Verwendung von Zufallszahlen die Ergebnisse bei jedem Aufruf variieren (können).
- Geben Sie Ihre Lösung als Java-Datei `FlugSimulator.java` in Moodle ab. Achten Sie darauf, dass die Datei mittels des UTF-8-Formats (ohne byte order marker (BOM)) kodiert ist.

Viel Spaß und Erfolg bei der Lösung der Aufgaben!