Humboldt-Universität zu Berlin
Institut für Informatik
Prof. Dr. Jens-Peter Redlich
M. Bauer, A. Heuts, Dr. W. Müller, M. Sänger

Berlin, den 23.11.2020

Übungen zur Vorlesung
Grundlagen der Programmierung

Übungsblatt 2

Abgabe: Die Abgabe Ihrer Lösungen erfolgt über den Moodle-Kurs der Vorlesung (https://moodle.hu-berlin.de/course/view.php?id=98193). Nutzen Sie dort das im Abschnitt Übung angegebene **Aufgabenblatt 2: Prüfzifferberechnung und Simulation.** Die Abgabe ist bis zum **07.12.2020** um **9:15 Uhr** möglich und erfolgt in Gruppen.

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 Korrektoren 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 (Prüfzifferberechnung) 12 Punkte

Die Krankenversicherung *HUB-KV* plant die Vergabe von neuen Versicherungsnummern für Ihre Mitglieder*innen. Um Falscheingaben und Übermittlungsfehler in Ihrem System zu vermeiden, sollen die Versicherungsnummern mit einer Prüfziffer versehen werden. Die Prüfziffer wird aus der neustelligen Basisnummer berechnet und dann (als 10. Stelle) zu dieser hinzufügt. Die Basisnummer kann aus Ziffern (0-9) sowie Großbuchstaben (A-Z) bestehen. Die Berechnung der Prüfziffer erfolgt nach dem folgenden Schema:

- Der Grundwert eines Zeichens der Basisnummer entspricht dem Wert der Ziffer (0-9). Buchstaben werden darauf aufbauend fortführend bewertet, d.h. A=10, B=11, C=12, ..., Z=35.
- Die Zeichen der Basisnummer werden von links nach rechts sich wiederholend mit 7, 3 und 1 gewichtet.

- Der Wert eines Zeichens ergibt sich dann als Produkt aus dem Grundwert des jeweiligen Zeichens und dem Gewicht.
- Die Prüfziffer ist dann die Differenz der Summe der Zeichenwerte zum nächstgrößeren Vielfachen von 10. Ist die Summe durch 10 teilbar, ist die Prüfziffer die Ziffer 0.

Betrachten Sie im Folgenden ein Berechnungsbeispiel der Prüfziffer für die Basisnummer TF2000129:

Zeichen	Basiswert	Gewicht	Zeichenwert
T	29	7	203
F	15	3	45
2	2	1	2
0	0	7	0
0	0	3	0
0	0	1	0
1	1	7	7
2	2	3	6
9	9	1	9
		Summe	272
		Nächstes 10er Vielfaches	280
		Differenz	8

Die Prüfziffer für die Basisnummer TF2000129 ist dementsprechend die Ziffer 8 und die komplette Versicherungsnummer lautet TF20001298.

Schreiben Sie ein Java-Programm `KVNummerPruefer.java`, welches eine 10-stellige Versicherungsnummer als Eingabe erhält und prüft, ob diese gültig ist (d.h. ob die Prüfziffer (nach dem vorgegebenen Schema) korrekt ist). Solltet die Prüfziffer korrekt sein, gibt das Programm „Korrekt“ aus. Im Falle einer inkorrekten Prüfziffer wird „Inkorrekt“ ausgegeben und die korrekte Prüfziffer dargestellt. Die folgenden zwei Aufrufe illustrieren die Ein- und Ausgabe des Programms für eine korrekte und eine inkorrekte Prüfziffer beispielhaft:

```
java KVNummerPruefer TF20001298

Korrekt
```

```
java KVNummerPruefer TF20001293

Inkorrekt

Korrekte Ziffer: 8
```

Hinweise:

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

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

- Um auf das *i*-te Zeichen in einer Zeichenkette zuzugreifen, können Sie die Methode `charAt(i)` der `String`-Klasse, die einen Buchstabe (`char`) zurückliefert, verwenden:

```
String nachricht = "Hallo Welt";

char ersterBuchstabe = nachricht.charAt(0);
System.out.println(ersterBuchstabe); // Ausgabe "H"

char fuenfterBuchstabe = nachricht.charAt(4);
System.out.println(fuenfterBuchstabe); // Ausgabe "o"
```

  Beachten Sie, dass die Nummerierung der Buchstaben in einem String bei 0 beginnt.

- Ihr Programm kann eine valide Eingabe der Versicherungsnummer annehmen, d.h. die Eingaben bestehen immer nur aus den Ziffern 0-9 und den Buchstaben A-Z und sind immer genau 10 Zeichen lang. Ihr Programm muss keine Überprüfung auf Eingabefehler vornehmen.
- Geben Sie Ihre Lösung als Java-Datei `KVNummerPruefer.java` in Moodle ab. Achten Sie darauf, dass die Datei mittels des UTF-8-Formats (ohne byte order marker (BOM)) kodiert ist.

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, 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!