

Programmieren I: Java

Wintersemester 2021/2022

Übungsblatt P03

(Bearbeitung im Rechnerpraktikum in der Zeit vom 16.11.2021 bis zum 22.11.2021)

Dieses P-Übungsblatt soll Ihnen Grundlagen im Umgang mit Entscheidungsanweisungen vermitteln sowie Ihnen ermöglichen, die Verwendung von Variablen und Einlesemethoden erneut zu üben. Außerdem werden alle bisherigen Erkenntnisse zusammengefasst und eine erste Schleife erstellt.

Übrigens: Fehleingaben vom Benutzer müssen auf den P-Übungsblättern nur dann abgefangen werden, wenn explizit darauf hingewiesen wird.

Aufgabe P3.1 *

Arithmetische Operationen

Schreiben Sie ein Programm, das einen Sekunden-Wert in einer `int`-Variablen `sec` speichert und diesen Wert in Jahre, Tage, Stunden, Minuten und Sekunden zerlegt. Das Programm soll beispielsweise folgendes ausgeben:

```
Sekunden: 123456789
```

```
123456789 Sekunden entsprechen:  
3 Jahren,  
333 Tagen,  
21 Stunden,  
33 Minuten und  
9 Sekunden.
```

Hinweis: Verwenden Sie den modulo-Operator (%). Ein Jahr hat für diese Aufgabe immer 365 Tage.

Aufgabe P3.2 */**

Entscheidungsanweisungen

Sie sind als Praktikantin bzw. Praktikant bei der Hans Wiwi GmbH tätig. Da bei der Hans Wiwi GmbH nur Karlsruher Wiwis arbeiten, hat das Unternehmen bereits einen Monat nach seiner Gründung seine maximale Effizienz erreicht. Ihre Vorgesetzte langweilt sich somit die meiste Zeit ihres Arbeitstages. Da sie ebenfalls mitbekommen hat, dass Sie seit neuestem Grundlagenwissen im Programmieren in Java besitzen, gibt sie Ihnen die Aufgabe, ein simples Spiel zu programmieren.

a) Das Zahlenraten-Spiel soll dabei wie folgt aussehen:

- Sie denken sich ein Intervall aus und lesen es in Ihr Programm ein (zwei positive Werte, eine Unter- und eine Obergrenze).
- Danach gibt das Programm ein paar (beliebige) Zeilen aus, damit Ihre Vorgesetzte nicht sehen kann, welche Grenzen Sie eingegeben haben.
- Ihre Vorgesetzte gibt dann einen Wert ein und Ihr Programm gibt aus, ob sich dieser Wert in dem von Ihnen definierten Bereich befindet.

- Sollte der Punkt im Bereich – und Ihre Vorgesetzte damit richtig – liegen, terminiert das Programm mit einer *Glückwunsch*-Meldung.
- Sollte Ihre Vorgesetzte falsch geraten haben, terminiert das Programm mit einer entsprechenden Meldung.

Eine Beispielausgabe könnte wie folgt aussehen:

```
Bitte untere Grenze eingeben: 2
Bitte obere Grenze eingeben: 10
*****
*****ZAHLEN*****
*****RATEN*****
*****
Was ist die magische Zahl? 3
Herzlichen Glückwunsch!
```

b) Nach ein paar Spielen beschwert sich Ihre Vorgesetzte darüber, dass das Spiel zu schwer und unfair sei. Außerdem möchte sie zwei Versuche. Das Zahlenraten soll daher wie folgt abgeändert werden:

- Zusätzlich zu einem Intervall geben Sie nun als erster Spieler zu Spielbeginn eine beliebige, positive Zahl ein, die in diesem Intervall liegt („*Die magische Zahl*“).
- Unter der bekannten Ausgabe aus Teil a) wird nun ausgegeben, in welchem Intervall sich die magische Zahl befindet.
- Ihre Vorgesetzte tippt nun – analog zu a) – eine Zahl.
- Sollte sie die korrekte Zahl erraten haben, zeigt das Programm eine *Glückwunsch*-Meldung.
- Für den anderen Fall soll das Programm ihr mitteilen, ob die zu erratende Zahl größer oder kleiner als ihr Tipp war.
- Sollte sie falsch liegen, erhält Ihre Vorgesetzte maximal einen weiteren Versuch, die Zahl zu erraten.
- Sollte es ihr auch nach dem zweiten Versuch nicht geglückt sein, die Zahl zu erraten, zeigt das Programm eine Meldung. Ebenfalls soll dann am Ende verraten werden, welches die *magische Zahl* war.

Das Spiel könnte beispielsweise wie folgt aussehen:

```
Bitte untere Grenze eingeben:
1
Bitte obere Grenze eingeben:
9
Bitte magische Zahl eingeben:
4
*****
*****
*****
*****ZAHLEN*****
*****RATEN*****
*****
*****
*****
*****
*****
*****
Die magische Zahl liegt zwischen 1 und 9.
Erster Tipp: was ist die magische Zahl?
3
Leider falsch! Die magische Zahl ist größer.
Zweiter Tipp: was ist die magische Zahl?
4
Herzlichen Glückwunsch! Die Zahl wurde erraten.
```

c) Versehen Sie Ihr fertiges Programm aus Aufgabe b) an der richtigen Stelle mit den Javadoc Kommentaren @author und @version. Exportieren und betrachten Sie dann die Javadoc-HTML-Datei.

Aufgabe P3.3 *

Schleifen

- a) Schreiben Sie ein Programm, das vom Nutzer eine ganze Zahl einliest und daraus die Fakultät berechnet. Nutzen Sie hierfür eine while-Schleife.

Ausgabebeispiel:

```
Bitte ganze Zahl eingeben:
5
Die Fakultät der eingegebenen Zahl ist 120
```

- b) Ergänzen Sie Ihr Programm mit einer do-while-Schleife, welche die gleiche Funktion hat. Geben Sie beide Ergebnisse aus.

Ausgabebeispiel:

```
Bitte ganze Zahl eingeben:
5
while: Die Fakultät der eingegebenen Zahl ist 120
do-while: Die Fakultät der eingegebenen Zahl ist 120
```

- c) Ergänzen Sie Ihr Programm mit einer for-Schleife, welche die gleiche Funktion hat. Geben Sie alle Ergebnisse aus.

Ausgabebeispiel:

```
Bitte ganze Zahl eingeben:
5
while: Die Fakultät der eingegebenen Zahl ist 120
do-while: Die Fakultät der eingegebenen Zahl ist 120
for: Die Fakultät der eingegebenen Zahl ist 120
```

Aufgabe P3.4 *

Schleifen, Switch

Die for-Schleife in der Klasse GeradeUngerade wird 10 Mal ausgeführt. Es wird jeweils ausgegeben, ob die eingegebene Zahl gerade oder ungerade ist. Schreiben Sie die in der Schleife enthaltene switch-Anweisung entsprechend der neu in Java 13 eingeführten Änderungen um. Nutzen Sie dafür zuerst den Pfeil-Operator, danach das Schlüsselwort yield.

```
import java.util.Scanner;

public class GeradeUngerade {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        for(int i = 0; i <= 10; i++) {
            System.out.println("Bitte ganze Zahl zwischen 1 und 9 eingeben:");
            int eingabe = scan.nextInt();

            switch(eingabe) {
                case 1: System.out.println("Ungerade"); break;
                case 2: System.out.println("Gerade"); break;
                case 3: System.out.println("Ungerade"); break;
                case 4: System.out.println("Gerade"); break;
                case 5: System.out.println("Ungerade"); break;
                case 6: System.out.println("Gerade"); break;
                case 7: System.out.println("Ungerade"); break;
                case 8: System.out.println("Gerade"); break;
                case 9: System.out.println("Ungerade"); break;
                default: System.out.println("Falsche Eingabe"); break;
            }
        }
        scan.close();
    }
}
```

Aufgabe P3.5 **/****Schleifen, Entscheidungsanweisungen**

- a) Schreiben Sie ein Programm, das eine positive ganze Zahl einliest, sie in ihre Ziffern zerlegt und die Ziffern in umgekehrter Reihenfolge als Text ausgibt. Eine Beispielausgabe könnte wie folgt aussehen:

```
Positive ganze Zahl: 53024
Zerlegt rückwärts: Vier Zwei Null Drei Fünf
```

Überlegen Sie sich zunächst Ihr Vorgehen und planen Sie Ihren Algorithmus. Verwenden Sie für die benötigte Schleife eine `while`-Schleife sowie für Entscheidungsanweisungen `switch-case`.

- b) Erweitern Sie das Programm so, dass die Ziffern nach der ersten (umgekehrten Reihenfolge) ebenfalls in richtiger Reihenfolge als Text ausgegeben werden. Es sollten keine neuen `switch-case`-Anweisungen hinzugefügt werden.

Ausgabebeispiel:

```
Positive ganze Zahl: 53024
Zerlegt rückwärts: Vier Zwei Null Drei Fünf
Zerlegt vorwärts: Fünf Drei Null Zwei Vier
```

Viel Erfolg!