

## Aufgabenblatt 10

Bitte beachten Sie, dass während der Weihnachtsferien Fragen im Forum evtl. erst nach einigen Tagen beantwortet werden. Bei der Lösung der folgenden Aufgaben wird der Programmierstil wie üblich bewertet.

- a) `CentralizedStart.java`, `CentralizedPlayer.java` In dieser Aufgabe soll eine zentralisierte objektorientierte Variante unseres bereits aus früheren Aufgabenblättern bekannten Ziffernratespiels implementiert werden.

Lösen Sie zum folgenden Programmtext die darunter angegebenen Teilaufgaben (1) bis (6). Der vorgegebene Programmtext darf nicht verändert, sondern nur erweitert werden (an beliebigen Stellen). Die Teilaufgaben können zum Teil unabhängig voneinander bearbeitet werden. Bitte geben Sie, wenn möglich, nur je eine Klasse `CentralizedStart.java` und `CentralizedPlayer.java` zum gesamten Aufgabenteil a) ab.

```
class CentralizedStart {  
  
    public static void main(String[] args) {  
        CentralizedPlayer player0 = new CentralizedPlayer();  
        CentralizedPlayer player1 = new CentralizedPlayer();  
        CentralizedPlayer player2 = new CentralizedPlayer();  
        player0.setName(args[0]);  
        player1.setName(args[1]);  
        player2.setName(args[2]);  
        player0.printGreeting();  
        player1.printGreeting();  
        player2.printGreeting();  
        /*  
        for (int round = 0; round < 3; ++round) {  
            player0.nextTurn();  
            player1.nextTurn();  
            player2.nextTurn();  
        }  
        int[] results = new int[3];  
        results[0] = player0.getResult();  
        Out.println("Endergebnis " + player0.getName() + ": " + results[0]);  
        results[1] = player1.getResult();  
        Out.println("Endergebnis " + player1.getName() + ": " + results[1]);  
        results[2] = player2.getResult();  
        Out.println("Endergebnis " + player2.getName() + ": " + results[2]);  
        // int max = findMax(results[0], results[1], results[2]);  
        */  
    }  
}
```

- (1) Das Programm muss mit drei Kommandozeilenparametern (den Spielernamen) aufgerufen werden. Überprüfen Sie diese Bedingung im Programm und brechen Sie andernfalls die Programmausführung mit einer aussagekräftigen Fehlermeldung ab.

(2 Punkte)

- (2) Programmieren Sie eine Klasse `CentralizedPlayer.java` mit Methoden `setName()` und `printGreeting()` sowie einem geeigneten Attribut, so dass das Programm fehlerfrei übersetzt und ausgeführt werden kann. Zur Lösung dieser Teilaufgabe darf die Klasse `CentralizedStart` **nicht verändert** werden (die Änderungen aus (1) sind okay). Bei Aufruf mit

```
java CentralizedStart Susi Fritz Egon
```

sollte sich als Ausschrift ergeben:

```
Viele Gruesse von Susi
Viele Gruesse von Fritz
Viele Gruesse von Egon
```

(8 Punkte)

- (3) Für diese und die weiteren Teilaufgaben entfernen Sie bitte die Kommentarzeichen `/*` und `*/` aus dem vorgegebenen Programmtext. Der Programmtext enthält die “magische Zahl” 3, die einerseits die Anzahl der Ziffern und andererseits die Anzahl der Spieler bezeichnet. Verwenden Sie geeignete Konstanten, um diese Doppeldeutigkeit zu vermeiden.

(2 Punkte)

- (4) Erweitern Sie Ihre Klasse `CentralizedPlayer.java`, so dass sich das Programm wieder fehlerfrei übersetzen und ausführen lässt.

In der Methode `nextTurn()` soll der Spieler selbständig einmal würfeln und auch selbst entscheiden, an welcher Position die neue Ziffer platziert wird. Hierfür genügt eine einfache Strategie: Der Spieler platziert eine 6 vorzugsweise auf der Hunderterstelle und weicht, falls diese bereits besetzt ist, auf die nächstmögliche niedrigwertigere Stelle aus. Ähnlich platziert er alle übrigen Ziffern vorzugsweise auf der Einerstelle und weicht bei Bedarf auf die nächstmögliche höherwertigere Stelle aus. Alternativ können Sie eine cleverere Strategie wählen. In jedem Fall muss der Spieler nach den drei Runden eine gültige Zahl erzeugt haben.

(10 Punkte)

- (5) Entfernen Sie nun auch die `//-`Kommentarzeichen und ergänzen Sie die Methode `findMax`. Hinweis: Diese muss mit dem Schlüsselwort `static` markiert werden.

(4 Punkte)

- (6) Ergänzen Sie weiteren Programmcode zur Ermittlung des oder der Gewinner und geben Sie den/die Gewinner aus. Ein vollständiger Programmlauf kann beispielsweise wie folgt aussehen:

```
java CentralizedStart Susi Fritz Egon
Viele Gruesse von Susi
Viele Gruesse von Fritz
Viele Gruesse von Egon
Endergebnis Susi: 112
Endergebnis Fritz: 452
Endergebnis Egon: 642
Gewinner: Egon
```

(4 Punkte)

- b) RingStart.java, RingPlayer.java: Diese Aufgabe sollte erst nach Aufgabe a), und nur wenn Sie a) weitgehend erfolgreich gelöst haben, in Angriff genommen werden.

Programmieren Sie eine zweite Variante “unseres” Spiels, bei der die Startklasse das Spiel lediglich initiiert. Danach soll jeder Spieler nach seinem Zug selbständig dem Nachfolger Bescheid geben, dass dieser an der Reihe ist. Gespielt wird wieder mit 3 Ziffern und 3 Spielern, deren Namen als Kommandozeilenparameter übergeben werden. Am Spielende soll jeder Spieler reihum sein Ergebnis am Bildschirm ausgeben. Danach muss das Spiel enden.

Die Klasse RingStart.java ist vorgegeben und darf **nicht verändert** (also auch nicht erweitert) werden:

```
class RingStart {  
  
    public static void main(String[] args) {  
        RingPlayer player0 = new RingPlayer();  
        RingPlayer player1 = new RingPlayer();  
        RingPlayer player2 = new RingPlayer();  
        player0.setName(args[0]);  
        player1.setName(args[1]);  
        player2.setName(args[2]);  
        player0.setSuccessor(player1);  
        player1.setSuccessor(player2);  
        player2.setSuccessor(player0);  
        player0.nextTurn();  
    }  
}
```

Beispielausgabe:

```
java RingStart Susi Fritz Egon  
Susi wuerfelt 5  
Fritz wuerfelt 4  
Egon wuerfelt 6  
Susi wuerfelt 2  
Fritz wuerfelt 5  
Egon wuerfelt 5  
Susi wuerfelt 3  
Fritz wuerfelt 1  
Egon wuerfelt 4  
Endergebnis Susi: 523  
Endergebnis Fritz: 451  
Endergebnis Egon: 546
```

(10 Punkte)

Die Klasse RingPlayer ist selbst zu programmieren. Hinweis: Speichern Sie den Nachfolger in einem Attribut des Typs RingPlayer.

**Abgabetermin:** Die Lösungen sind bis spätestens Donnerstag, den 11.1.2018 um 8:00 Uhr (strikt!) über das elektronische Abgabesystem einzureichen. Nachträglich eingereichte Lösungen zählen als nicht abgegeben.