Aufgabe: Münzenspiel

Spielregeln

1. Zu Beginn legen die Spieler einige Münzen (10 - 30) auf den Tisch.
2. Zwei Spieler nehmen nun abwechslungsweise eine, zwei oder drei Münzen vom Tisch.
3. Gewonnen hat der Spieler, der zuletzt Münzen vom Tisch genommen hat.

Auftrag

Schreiben Sie ein Programm, um das Spiel zu spielen.

Ablauf in der Konsole

```
Willkommen zum Münzenspiel
Es sind noch 11 Münzen auf dem Stapel
Spieler 1 nimm 1-3 Münzen > 3

Es sind noch 8 Münzen auf dem Stapel
Spieler 2 nimm 1-3 Münzen > 7
Du darst höchstens 3 Münzen nehmen
Es sind noch 8 Münzen auf dem Stapel
Spieler 2 nimm 1-3 Münzen > 2

Es sind noch 6 Münzen auf dem Stapel
Spieler 1 nimm 1-3 Münzen > 0
Du musst mindestens 1 Münze nehmen
Es sind noch 6 Münzen auf dem Stapel
Spieler 1 nimm 1-3 Münzen > 3

Es sind noch 3 Münzen auf dem Stapel
Spieler 2 nimm 1-3 Münzen > 3

Spieler 1 hat verloren
Bis bald
```

Programmgerüst mit Logik

Als Vorbereitung haben Sie im Programmgerüst die Programmlogik als Kommentare umgesetzt.

```
import java.util.Scanner;

/**
```

```
 * play a game with coins
 *
 * @author Marcel Suter
 * @since 2018-09-28
 * @version 1.0
 */
public class CoinGame{

    static Scanner scanner;

    /**
     * constructor
     */
    public CoinGame() {
        scanner = new Scanner(System.in);
    }

    /**
     * starts the execution
     *
     * @param args
     *            command line arguments
     */
    public static void main(String[] args) {
        CoinGame program = new CoinGame();

        System.out.println("Willkommen zum Münzenspiel");
        program.play();
        System.out.println("Bis bald");
        scanner.close();
    }

    /**
     * play the coin game
     */
    private void play() {
        // TODO Variablen deklarieren
        // TODO aktueller Spieler gleich 1

        /* generate random number of coins
           between 10 and 30 */
        TODO = (int)(Math.random() * 20 + 10);

        // TODO Solange Restliche Münzen grösser Null
            // TODO Ausgabe: Restliche Münzen
            // TODO Ausgabe: aktueller Spieler nimm 1-3 Münzen
            // TODO Eingabe: Anzahl Münzen

            // TODO Falls Anzahl Münzen kleiner 1
```

```
                System.out.println("Du musst mindestens 1 Münze nehmen");
            // TODO Sonst
                // TODO Falls Anzahl Münzen grösser 3
                    // TODO Ausgabe: Du kannst höchstens 3 Münzen nehme
                // TODO Sonst
                    // TODO Reduziere Restliche Münzen um Anzahl
                    // TODO Aktueller Spieler = 3 - Aktueller Spieler
                // Ende der inneren Selektion
            // Ende der äusseren Selektion
        // Ende der Iteration
      // TODO Ausgabe: aktueller Spieler hat verloren
    }
}
```

Hinweise

Zufällige Anzahl Münzen

Zu Beginn des Programms wird eine zufällige Anzahl Münzen generiert. Dies geschieht mit dem Befehl `… (int)(Math.random() * 20 + 10);`

1. `Math.random()` erzeugt eine Zufallszahl zwischen 0 und 1.
2. Multipliziert man diese Zufallszahl mit 20, so erhält man eine Zahl zwischen 0 und 20.
3. Nun addiert man 10, so erhält man eine Zahl zwischen 10 und 30.

Sie brauchen das Ergebnis dieses Befehls nur in einer Variable zu speichern.

Aktueller Spieler

Die Spieler 1 und 2 ziehen abwechslungsweise. Für die Anzeige des jeweils aktuellen Spielers, benötigen wir eine Variable. Diese Variable erhält am Anfang den Wert 1, also ist Spieler **1** am Zug.

Hat der Spieler eine gültige Anzahl eingegeben, so wechselt der Spieler. Dazu rechnen wir `Aktueller Spieler = 3 - Aktueller Spieler`.

- Ist der aktuelle Spieler = 1, so ist das Resultat der Rechnung 2.
- Ist der aktuelle Spieler = 2, so ist das Resultat der Rechnung 1.

Das Resultat wird in der Variable für den aktuellen Spieler gespeichert.

Vorgehen

Stellen Sie nach jedem Schritt sicher, dass das Programm korrekt kompiliert wird. Fahren Sie erst fort, nachdem alle Fehler korrigiert sind.

1. Programmgerüst mit Logik

1. Erstellen Sie einen neues Modul in Ihrem Projekt.
2. Kopieren Sie das Programmgerüst in ihre IDE

2. Variablen

1. Bestimmen Sie alle benötigten Variablen mit Bezeichnung (Name) und Datentyp (int, char, float, boolean, ...).
2. Deklarieren Sie die Variablen in der Methode `play()`.

3. Ein- / Ausgaben

Realisieren Sie alle Ein- und Ausgaben

4. Selektionen und Iterationen

Realisieren Sie schrittweise die Selektionen und Iterationen.

5. Testen

Spielen Sie das Spiel gegen sich selber oder einen Kollegen. Prüfen Sie auch, wie das Programm auf ungültige Züge (z.B. 0 Münzen, 5 Münzen) reagiert.

Zeigt das Programm den Sieger korrekt an?

Erweiterung

Erstellen Sie ein neues Programm (CoinsSolo), bei dem ein Spieler gegen den Computer spielen kann. Für den Zug des Computers können Sie verschiedene Varianten programmieren:

- Der Computer nimmt immer genau 1 Münze
- Der Computer nimmt eine zufällige Anzahl Münzen
  `coinTake = (int)(Math.random() * 3 + 1);`
- Der Computer nimmt genau so viele Münzen, dass er am Ende gewinnt.

© Marcel Suter

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/modul/archiv/m319/learningunits/lu04/aufgaben/muenzenspiel

Last update: **2024/03/28 14:07**