

Java03 – Lotto (1/3) – Gewinnzahlen ziehen und tippen sowie Ausgabe der Zahlen

Projekt-Beschreibung: Lotto „6 aus 16“

Wir wollen als BlueJ-Projekt ein Lotto-Spiel „6 aus 16“ programmieren.

Man kann bei diesem Lotto-Spiel 6 verschiedene Zahlen von 1 bis 16 als Tipp abgeben.

Anschließend werden die 6 Gewinnzahlen als Zufallszahlen von 1 bis 16 „gezogen“.

Abschließend sollen die 6 gezogenen Zahlen und die 6 getippten Zahlen ausgegeben werden und eine Auswertung erfolgen, wie viele „Richtige“ unter den getippten Zahlen sind.

Überblick:

Lotto (1/3): Gewinnzahlen „naiv“ ziehen, Tipp abgeben, Gewinnzahlen und Tipp-Zahlen ausgeben

Lotto (2/3): Gewinnzahlen „korrekt“ ziehen, ggf. auch Kontrolle, ob der Tipp „korrekt“ ist

Lotto (3/3): Auswertung mit Ausgabe der Anzahl der „Richtigen“, ggf. Sortierung der beiden Felder

Schritt 1 – Neues BlueJ-Projekt erstellen und „Quellcode-Gerüst“ einfügen

Öffne zunächst BlueJ und erstelle mit „Projekt“ → „Neues Projekt“ ein neues BlueJ-Projekt mit dem Namen „**Java_03_Lotto**“ und speichere es in Deinem Home-Laufwerk („Location“ → „Choose“).

Lege in diesem BlueJ-Projekt eine neue Klasse „Lotto“ an, öffne den Quelltext und lösche den gesamten „automatisch“ erzeugten Quelltext.

Kopiere in den (nun leeren) Quellcode den folgenden Code:

```
import java.util.Random;
public class Lotto {
    private int[] getippt; // Feld "getippt" zum Speichern der getippten Z.
    private int[] gezogen; // Feld "gezogen" zum Speichern der gezogenen Z.
    private Random zufallszahl = new Random(); // "zufallszahl" Typ "Random"

    public Lotto() { // Konstruktor
        getippt = new int[6]; // Initialisierung als Feld mit 6 Zahlen
        gezogen = new int[6]; // Initialisierung als Feld mit 6 Zahlen
    }

    public void ziehenGewinnzahlen1() { // „naives“ Ziehen der Gewinnzahlen
        for(int i = 0; i < gezogen.length; i++) {
            gezogen[i] = zufallszahl.nextInt(16)+1;
        }
    }

    public void ausgebenGewinnzahlen() { // Ausgabe der gezog. Gewinnzahlen
        System.out.print("Gezogene Gewinnzahlen: ");
        for(int i = 0; i < gezogen.length; i++) {
            System.out.print(gezogen[i] + " - ");
        }
        System.out.println(); // sonst nächste Ausgabe nicht in neuer Zeile
    }
}
```

Schritt 2 – Test von „ziehenGewinnzahlen1()“ und „ausgebenGewinnzahlen()“

Übersetze die Klasse, erzeuge ein neues Objekt „lotto1“ der Klasse „Lotto“.

Führe „ziehenGewinnzahlen1()“ und „ausgebenGewinnzahlen()“ dann mehrmals (jeweils direkt hintereinander) aus.

Was fällt Dir auf?

Warum handelt es sich bei „ziehenGewinnzahlen1()“ um ein „naives“ Ziehen der Gewinnzahlen?

Was muss verbessert werden?

Schritt 3 – Methode „tippen()“ zur Abgabe des Tipps

Mit der Methode „tippen()“ soll der Tipp der 6 Zahlen (von 1 bis 16) abgegeben werden können.

Füge den folgenden Code ganz unten in Deinen Quellcode (aber noch oberhalb der schließenden „Klassen-Klammer“) ein und ergänze ihn so, dass 6 Tippzahlen abgeben werden können und im Feld „getippt[]“ abgespeichert werden.

```
public void tippen(int z1, ...) {  
    getippt[0] = z1;  
    ...;  
}
```

Schritt 4 – Methode „ausgebenTippzahlen()“ zum Ausgeben der getippten Zahlen

Füge eine Methode „ausgebenTippzahlen()“ ganz unten in Deinen Quellcode (aber noch oberhalb der schließenden „Klassen-Klammer“) ein.

Orientiere Dich an der Methode „ausgebenGewinnzahlen()“ (siehe oben).

ZUSATZ-AUFGABE

Wenn Du schnell bist, kannst Du Dich (noch einmal) mit der Methode „findeMinimum()“ (08.12.2023) beschäftigen. Wesentliche Teile des Codes werden für eine Methode zum Sortieren der Feld-Elemente eines Arrays (der Größe nach) benötigt.