

Hinweise für den Prüfling

Einlese- und Auswahlzeit: 30 Minuten

Bearbeitungszeit (insgesamt): 180 Minuten

Auswahlverfahren

Es gibt zwei Aufgabengruppen A und B, aus denen jeweils ein Vorschlag zu bearbeiten ist. Der vorliegende Vorschlag aus der Gruppe A (Objektorientierte Modellierung) ist ein Pflichtvorschlag.

Wählen Sie von den zwei vorliegenden Vorschlägen der Gruppe B (Datenbanken) einen zur Bearbeitung aus. Der nicht ausgewählte Vorschlag muss am Ende der Einlesezeit der Aufsicht führenden Lehrkraft zurückgegeben werden.

Erlaubte Hilfsmittel

1. ein Wörterbuch zur deutschen Rechtschreibung
2. eine Liste der fachspezifischen Operatoren

Sonstige Hinweise

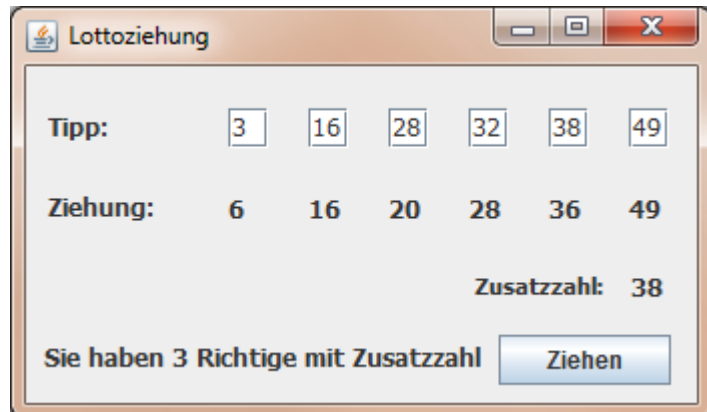
ohne PC-Nutzung
Java-Variante

In jedem Fall vom Prüfling auszufüllen

Name: _____	Vorname: _____
Prüferin / Prüfer: _____	Datum: _____

Lotto 6 aus 49

Mit einem Programm soll das Lottospiel simuliert werden. Ein Spieler gibt sechs Zahlen von 1 bis 49 als Tipp ab. Bei der Ziehung werden aus einer Trommel mit 49 durchnummerierten Kugeln nacheinander sechs Zahlen und eine Zusatzzahl ohne Zurücklegen gezogen. Anschließend wird durch Vergleich von Tipp und Ziehung die Anzahl der Richtigen ermittelt.



Aufgaben

1. Die Benutzeroberfläche soll entsprechend der obenstehenden Abbildung in der von *JFrame* abgeleiteten GUI-Klasse *Benutzeroberflaeche* gestaltet werden. Geben Sie alle dazu notwendigen GUI-Komponenten an und beschreiben Sie, welchem Zweck die GUI-Komponenten dienen.
(7 BE)
2. In der Materialvorlage ist das Klassendiagramm für das Programm „Lottoziehung“ gegeben. Erläutern Sie die Beziehungen und deren Implementierungen.
(7 BE)
3. Beschreiben Sie die in der Materialvorlage gegebene Implementierung der Klasse *Trommel*. Begründen Sie, dass eine Kugel (Zahl) nicht zweimal gezogen werden kann.
(16 BE)
4. Implementieren Sie die Methode *sortieren()* der Klasse *Ziehung* und erläutern Sie den von Ihnen zugrunde gelegten Sortieralgorithmus.
(10 BE)

5. Der Konstruktor der Klasse *Spiel* speichert die getippten Zahlen in dem Feld *Tipp*. Die Methode *gibRichtige()* der Klasse *Spiel* liefert einen String, mit dem die Anzahl der Richtigen ausgegeben werden soll. Dabei sollen folgende Fälle berücksichtigt werden:

0 Richtige
1 Richtige
2 Richtige
3 Richtige
3 Richtige mit Zusatzzahl
4 Richtige
4 Richtige mit Zusatzzahl
5 Richtige
5 Richtige mit Zusatzzahl
6 Richtige

Beispiele:

3 Richtige: Drei der sechs Lottozahlen wurden richtig getippt.

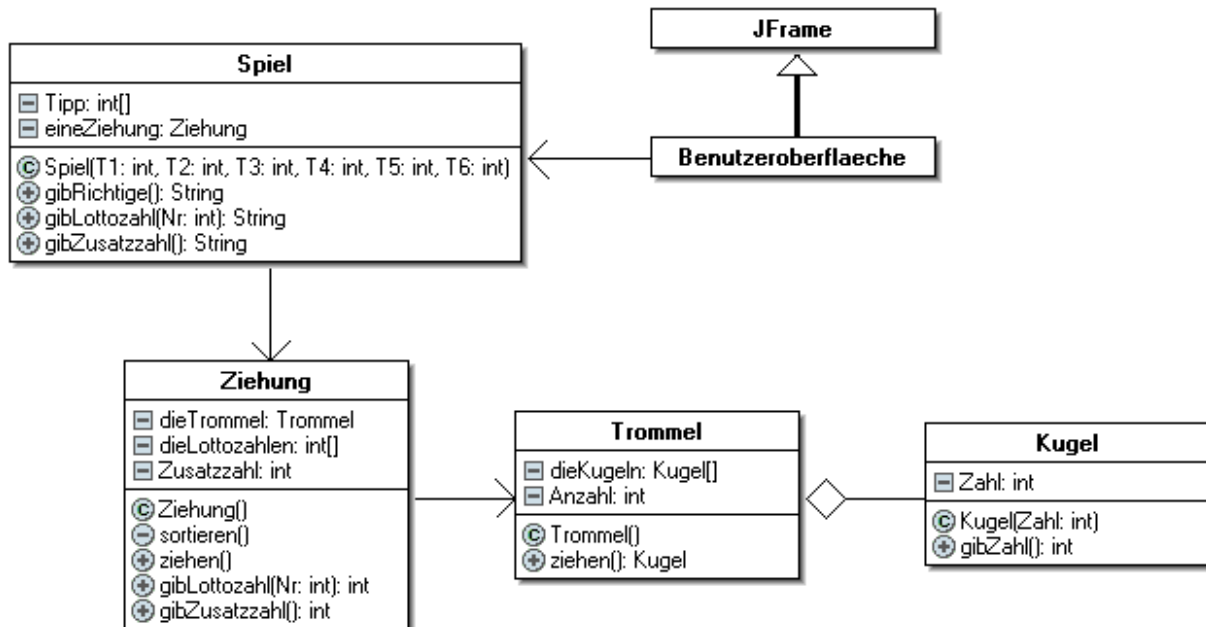
4 Richtige mit Zusatzzahl: Vier der sechs Lottozahlen wurden richtig getippt und die Zusatzzahl ist eine der beiden übrigen getippten Zahlen.

- 5.1 Entwerfen Sie den Algorithmus der Methode *gibRichtige()* passend zu den Vorgaben.
- 5.2 Implementieren Sie die Methode *gibRichtige()* passend zu den Vorgaben im UML-Diagramm.

(20 BE)

Material

UML-Klassendiagramm „Lottoziehung“

Implementierung der Klasse *Trommel*

```

public class Trommel {
    private Kugel[] dieKugeln;
    private int Anzahl;

    public Trommel() {
        Anzahl = 49;
        dieKugeln = new Kugel[Anzahl];
        for (int i = 0; i < Anzahl; i++) {
            dieKugeln[i] = new Kugel(i + 1);
        }
    }

    public Kugel ziehen() {
        int pos = (int) (Anzahl * Math.random());
        Kugel eineKugel = dieKugeln[pos];
        for (int i = pos; i < Anzahl - 1; i++) {
            dieKugeln[i] = dieKugeln[i + 1];
        }
        Anzahl = Anzahl - 1;
        return eineKugel;
    }
}

```

Hinweis:

`Math.random()` liefert eine Zufallszahl größer gleich 0 und kleiner 1 vom Datentyp `double`.