

Nur zum Selbststudium!

0.13 Aufgabe

Geben Sie die Lösungsworte der Quizze aus der Lernnotiz an.

40. Aufgabe (Set, equals, hashCode, VL 24)

Gegeben Seien folgende Klassen.

```
public class Studierend {
    private String name;
    private int matnr;

    public Studierend(String name, int matnr) {
        this.name = name;
        this.matnr = matnr;
    }

    @Override
    public String toString() {
        return "Studierend [name=" + name + ", matnr=" + matnr + "]";
    }
}
```

```
import java.util.HashSet;
import java.util.Set;

public class Main {

    public static void main(String[] args) {
        Set<Studierend> studis = new HashSet<Studierend>();
        Studierend s = new Studierend ("Olga", 100);
        studis.add(s);
        studis.add(new Studierend("Sandra", 101));
        studis.add(s);
        studis.add(new Studierend("Olga", 100));
        System.out.println(studis);
    }
}
```

Kopieren Sie die Klassen in vier verschiedene Projekte und ergänzen Sie die Klasse Studierend jeweils um eine equals- und eine hashCode-Methode, so dass folgende Ausgaben (eventuell mit anderer Reihenfolge der Objekte) ausgegeben werden.

Hinweis: Nur die letzte Ausgabe gehört zur vollständig korrekten Implementierung der Methoden, für die anderen Ausgaben sind die Methoden bewusst geschickt falsch zu programmieren.

- a) [Studierend [name=Olga, matnr=100], Studierend [name=Sandra, matnr=101], Studierend [name=Olga, matnr=100]]

- b) [Studierend [name=Olga, matnr=100], Studierend [name=Sandra, matnr=101], Studierend [name=Olga, matnr=100], Studierend [name=Olga, matnr=100]]
c) [Studierend [name=Olga, matnr=100]]
d) [Studierend [name=Olga, matnr=100], Studierend [name=Sandra, matnr=101]]

41. Aufgabe (Bearbeitung von Listen, VL 16)

Zu entwickeln ist ein Programm zur Analyse der Wahrscheinlichkeiten von Poker-Blättern.

- a) Schreiben Sie eine Klasse Karte mit den Objektvariablen Farbe mit den möglichen Werten „KARO“, „HERZ“, „KREUZ“, „PIK“ und Wert mit den möglichen Werten (in aufsteigender Reihenfolge) „SIEBEN“, „ACHT“, „NEUN“, „ZEHN“, „BUBE“, „DAME“, „KOENIG“, „AS“.
b) Ergänzen Sie eine toString()-Methode, so dass z. B. die Farbe als „Karo“ und der Wert als „7“ ausgegeben werden.
c) Schreiben Sie eine Klasse Kartenstapel, mit einer Klassenmethode gibAlleKarten(), die einen Kartenstapel mit den 32 Karten zurückgibt und folgende Objektmethoden anbietet. (Wieder: Methode entwickeln, dann ausprobieren/testen)

Methode	Funktionalität
toString()	schöne Ausgabe
mischen()	mischt den Stapel, dabei soll nicht die Methode shuffle() der Klasse ArrayList genutzt werden
fuenfGeben()	gibt ein neues Kartenstapel-Objekt mit den ersten fünf Karten zurück
istPaar()	gibt es zwei Karten im Kartenstapel mit gleichem Wert?
istZweiPaar()	gibt es zweimal zwei Karten mit gleichem Wert?
istDrilling()	gibt es drei Karten mit gleichem Wert?
istStrasse()	handelt es sich um fünf Karten mit direkt aufsteigenden Werten?
istFlush()	haben alle Karten die gleiche Farbe?
istFullHouse()	gibt es einen Drilling und ein zusätzliches Paar?
istVierling()	gibt es vier Karten mit gleichem Wert?
istStraightFlush()	ist es ein Flush in Straßenform?
istRoyalFlush()	ist es ein Straight-Flush, der ein As enthält?

Hinweis: Überlegen Sie, warum es geschickt sein könnte, die Karten vor der Analyse zu sortieren.

- d) Schreiben Sie ein Programm, das 100000-mal neue gemischte Kartenstapel erzeugt, die ersten fünf Karten mit fuenfGeben() als Kartenstapel-Objekt abhebt, nach den Kriterien Paar bis Royal Flush untersucht und jeweils deren Auftreten zählt (ein Drilling soll dabei z. B. auch als Paar gezählt werden). Geben Sie weiterhin alle Straight Flushes aus. Abschließend sollen die Ergebnisse möglichst genau wie im folgenden Beispiel formatiert ausgegeben werden.

[Kreuz 8, Kreuz 9, Kreuz 10, Kreuz Bube, Kreuz Dame]

[Herz 8, Herz 9, Herz 10, Herz Bube, Herz Dame]

[Herz 8, Herz 9, Herz 10, Herz Bube, Herz Dame]

[Pik 10, Pik Bube, Pik Dame, Pik Koenig, Pik As]

PAAR ZWEIPAAR DRILLING STRAIGHT FLUSH FULLHOUSE VIERLING STRAIGHTFLUSH ROYALFLUSH

71661 12783 6292 1949 95 692 119 4 1

- e) Beim Pokern gibt es Varianten mit der Möglichkeit, Karten zu tauschen. Schreiben Sie eine Methode, die für gegebene fünf Karten berechnet, welche Karten gegen Karten aus dem Kartenstapel getauscht werden sollen. Lassen Sie wieder ein Simulationsprogramm mit 100000 Läufen laufen, das die Verbesserungen durch Ihr Tauschverfahren zählt. Beispiele: Verbesserung von PAAR auf ZWEIPAAR bringt einen Punkt, von PAAR auf DRILLING bringt zwei Punkte. Versuchen Sie ihr Tauschverfahren zu optimieren (Ziel ca. 37000 Punkte).