

Aufgabenblatt 2

— Veröffentlicht am 05.05.2010 —

Abgabe bis 11.05.2010, 13:00 per SVN

Aufgabe 1 (Wahr oder falsch? - 2 Punkte)

Überlegen Sie sich für jede Aussage ein Argument, mit dem Sie eindeutig bestimmen können, ob die Aussage wahr oder falsch ist. Für jede richtig angekreuzte Antwort werden 0,5 Punkte angerechnet, für jede falsch angekreuzte Antwort 0,5 Punkte abgezogen. Nicht angekreuzte Antworten geben keinen Punktabzug.

wahr	falsch	
		Eine Zuweisung wird in jeder Programmiersprache durch einen Ausdruck in der Form „X := 123;“ formuliert.
		Die Werte einer als <i>static</i> deklarierten Variablen können sich während der Laufzeit nicht ändern.
		Alle diese Zeichen dürfen in Java-Variablenamen verwendet werden: ü, ß, Ø, α, ξ, Λ, €, £
		Dies ist ein gültiger Java Ausdruck: <code>int 123 = ١٢٣;</code>

Die Zeichen bei der letzten Multiple-Choice Frage sind keine Zahlen, sondern einfach Buchstaben in einer bestimmten Sprache.

Aufgabe 2 (Pascal nach Java - 4 Punkte)

Schreiben Sie das folgende PASCAL-Programm in Java. Verwenden Sie dazu die Bibliothek `input.jar` aus Aufgabe 2 des Aufgabenblatts 1. Was berechnet das Programm ?

```
PROGRAM was_mache_ich(input, output);
VAR i, n: integer;
BEGIN
  write('Bitte geben Sie eine positive, ganze Zahl ein: ');
  readln(n);
  i := 2;
  WHILE ((n MOD i) <> 0) AND (i < n)
    DO i := i + 1;
  IF i = n THEN write(n + ' ist ein(e) was_mache_ich')
    ELSE write(n + ' ist kein(e) was_mache_ich')
END.
```

Benutzen Sie dabei folgende Vorlage:

```
public class WasMacheIch {

  public static void main(String[] args) {
    System.out.print("Bitte geben Sie eine positive, ganze Zahl ein: ");
    int n = Input.readInt();

    ...
  }
}
```

Bemerkung: Es wird hier nochmal darauf hingewiesen, dass bei dieser Aufgabe die Benutzung von input.jar ausdrücklich verlangt ist.

Aufgabe 3 (Statisch-Funktionale Programmierung mit Klassen - 10 Punkte)

Ziel dieser Aufgabe ist es, ein System zur Verwaltung von Kraftfahrzeugen zu entwickeln.

a) Erstellen Sie eine Klasse *TransportAuthority*, die bis zu 1000 Autos registrieren kann.¹ Für jedes Auto werden das Kennzeichen, der Besitzer und dessen Adresse gespeichert. Verwenden Sie dabei folgende Vorlage:

```
public class TransportAuthority {  
  
    private static String[] licensePlateList = new String[1000];  
    private static String[] ownerList = new String[1000];  
    private static String[] addressList = new String[1000];  
    ...  
  
    public static void registerCar(String licensePlate, String owner, String address) {  
        ...  
    }  
  
}
```

b) Erweitern Sie Ihre Klasse um die beiden Funktionen `getOwner(String licensePlate)` und `getAddress(String licensePlate)` die für ein bekanntes Kennzeichen den Besitzer bzw. dessen Adresse zurück geben.

Hinweis: Unter Java vergleichen Sie zwei Strings `s1` und `s2` mit: `s1.equals(s2)`.

c) Die Klasse *TransportAuthority* soll nun in der Lage sein, pro Fahrzeug bis zu 500 Verkehrsdelikte aufzuzeichnen. Fügen Sie dazu die Funktion `registerDelict(String licensePlate, double fee, int points)` hinzu und implementieren Sie die Funktion, indem Sie die Gebühren und Punkte für jedes Auto jeweils in einem Array speichern.

Implementieren Sie die Funktionen `getFees(String licensePlate)` und `getPoints(String licensePlate)`, welche jeweils ein Array mit den Gebühren und Punkten für das angegebene Kennzeichen zurückliefern.

Aufgabe 4 (Statisch-Funktionale Programmierung mit Klassen - Teil 2 - 9 Punkte)

Das System aus Aufgabe 3 wird hier erweitert!

a) Eine neue Klasse *MeterMaid* soll die Arbeit einer Mitarbeiterin des Ordnungsamtes simulieren. Die Klasse soll zwei Funktionen anbieten: Mit `checkCar(String licensePlate)` wird das zu überprüfende Fahrzeug festgelegt und mit der Funktion `fine()` wird für das Fahrzeug eine zufällige Strafe zwischen 10 und 100 EUR sowie 0 bis 3 Punkten registriert.

Hinweis: `new java.util.Random().nextInt(22)` erzeugt eine Zufallszahl zwischen 0 und 22.

b) Fügen Sie der Klasse *MeterMaid* die Funktion `printTicket()` hinzu. Diese soll für das zuvor überprüfte Fahrzeug einen Strafzettel ausstellen, der neben dem Namen und der Adresse des Besitzers alle registrierten Strafen auflistet und die Summe der Gebühren und Punkte aufführt.

c) Schreiben Sie ein Programm, das unter Zuhilfenahme der zuvor entwickelten Klassen zunächst

¹Es werden niemals mehr als 1000 Autos registriert.

ein Auto bei der *TransportAuthority* registriert, dann dieses Auto mittels der *MeterMaid* überprüft und drei Mal eine Strafe registriert. Danach soll ein Strafzettel ausgestellt werden.