

Aufgabe 1 (57 Punkte)

In dieser Aufgabe wird ein Getränkeautomat erstellt. Ein Getränkeautomat kann mehrere Fächer haben. Ein Fach enthält mehrere Getränke.

1 – A (7 Punkte)

```
public class Getraenk {  
  
    private String name;  
  
    public Getraenk(String name) {  
        this.name = name;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
}
```

1 – B (28 Punkte)

```
public class Fach {  
  
    private int preis;  
  
    private ArrayList<Getraenk> getraenke;  
  
    private final int MAX_GETRAENKE;  
  
    public Fach(int preis) {  
        this.preis = preis;  
        getraenke = new ArrayList<Getraenk>();  
        MAX_GETRAENKE = 10;  
    }  
  
    public int getPreis() {  
        return preis;  
    }  
  
    /**  
     * Prüft, ob die Variable "getraenke" leer ist.  
     * @return true, wenn die Variable "getraenke" leer ist, sonst false  
     */  
    public boolean istFachLeer() {  
        return getraenke.isEmpty();  
    }  
  
    /**  
     * Prüft, ob die Variable "getraenke" voll ist.  
     * @return true, wenn die Variable "getraenke" voll ist, sonst false  
     */  
    public boolean istFachVoll() {  
        return getraenke.size() >= MAX_GETRAENKE;  
    }  
  
    /**  
     * Fügt der Variable "getraenke" ein Getraenk hinzu,  
     * sofern noch Platz im Fach ist (MAX_GETRAENKE).  
     * Wenn das Fach bereits voll ist, wird eine  
     * entsprechende Meldung auf der Konsole ausgegeben.  
     * @param getraenk ein Getraenk  
     */  
    public void addGetraenk(Getraenk getraenk) {  
        if (!istFachVoll()) {  
            getraenke.add(getraenk);  
        } else {  
            System.out.println("Fach ist voll");  
        }  
    }  
  
}
```

```
    }  
}  
  
/**  
 * Entfernt das erste Getraenk in der Liste und  
 * gibt es zurueck. Wenn kein Getraenk mehr in  
 * der Liste ist, so wird null  
 * zurueckgegeben und eine Meldung auf der  
 * Konsole ausgegeben, dass das Fach leer ist.  
 * @return das naechste Getraenk oder null, wenn  
 * das Fach leer ist.  
 */  
public Getraenk getGetraenk() {  
    if (!istFachLeer()) {  
        return gettraenke.remove(0);  
    } else {  
        System.out.println("Fach ist leer");  
        return null;  
    }  
}  
  
public void printFachInfo() {  
    System.out.println("Der Preis fuer ein Getraenk in diesem Fach ist: " + getPreis());  
    System.out.println("Das Fach enthaelt noch folgende Getraenke:");  
    for (Getraenk getraenk : gettraenke) {  
        System.out.println(getraenk.getName());  
    }  
}
```

1 – C (12 Punkte)

```
public class Automat {

    private Fach fach1;

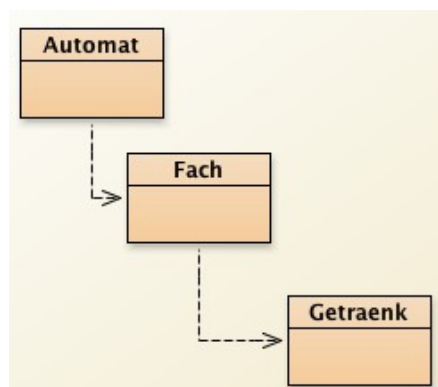
    public Automat() {
        fach1 = new Fach(100);
    }

    public Getraenk kaufeGetraenkFach1() {
        return fach1.getGetraenk();
    }

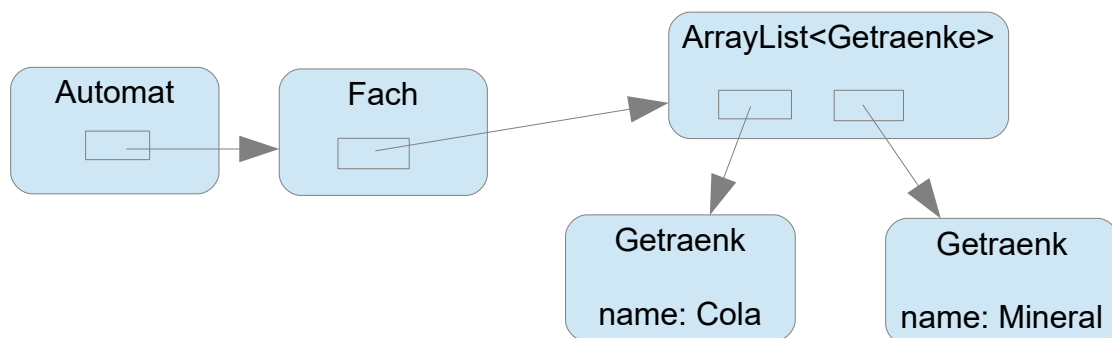
    /**
     * Füllt das Fach 1 mit Getraenken.
     * @param nameGetraenk der Name des Getraenks
     */
    public void fuelleFach1(String nameGetraenk) {
        while(!fach1.istFachVoll()) {
            fach1.addGetraenk(new Getraenk(nameGetraenk));
        }
    }
}
```

1 – D (10 Punkte)

1. Zeichnen Sie das Klassendiagramm der Aufgaben 1 – A bis 1 – C (5 Punkte)



2. Zeichnen Sie das Objektdiagramm der Aufgaben 1 – A bis 1 – C unter der Annahme, dass sich 2 Getraenke im Automaten befinden. Ein Getraenk heisst „Cola“, ein Getraenk heisst „Mineral“ (5 Punkte)



Aufgabe 2 (49 Punkte)

2 – A (6 Punkte)

Gegeben ist folgende Klasse:

```
public class Kreis {  
  
    private int durchmesser;  
    private String farbe;  
  
    public Kreis() {  
        ausgeben();  
    }  
  
    public Kreis(String farbe) {  
        this.farbe = farbe;  
        durchmesser = 15;  
        ausgeben();  
    }  
  
    public Kreis(String farbe, int durchmesser) {  
        this.farbe = farbe;  
        durchmesser = durchmesser;  
        ausgeben();  
    }  
  
    private void ausgeben() {  
        System.out.println("Farbe: " + farbe + " Durchmesser: " + durchmesser);  
    }  
}
```

In einer Anderen Klasse wird folgender Code ausgeführt:

```
Kreis k1 = new Kreis();  
  
Kreis k2 = new Kreis("blau");  
  
Kreis k3 = new Kreis("rot", 7);
```

Was wird auf der Konsole ausgegeben?

```
Farbe: null Durchmesser: 0  
Farbe: blau Durchmesser: 15  
Farbe: rot Durchmesser: 0
```

2 – B (3 Punkte)

Gegeben ist folgende Klasse:

```
public class Ticketautomat {  
  
    private int preis;  
  
    public void kaufeTicket(int eingeworfenesGeld) {  
  
        if (eingeworfenesGeld <= 0) {  
            String meldung = "Ungueltiger Betrag";  
        } else {  
            preis -= eingeworfenesGeld;  
            String meldung = "Sie haben " + eingeworfenesGeld + " Rappen eingeworfen";  
        }  
        System.out.println(meldung);  
    }  
}
```

In einer Anderen Klasse wird folgender Code ausgeführt:

```
Ticketautomat t = new Ticketautomat();  
t.kaufeTicket(70);
```

Was wird auf der Konsole ausgegeben? (3 Punkte, eine Antwort ist richtig)

- ☐ Ungueltiger Betrag
- ☐ Sie haben 0 Rappen eingeworfen
- ☐ null
- ☐ Sie haben 70 Rappen eingeworfen
- ☒ Das kompiliert nicht

2 – C (3 Punkte)

1. Was ist der Inhalt / Wert von Variable a? (1 Punkt)

```
String a = "Hallo" + "Welt" + 40 + 2;
```

HalloWelt402

2. Was ist der Inhalt / Wert von Variable b? (1 Punkt)

```
String b = 40 + 2 + "Hallo" + "Welt";
```

42HalloWelt

3. Was ist der Inhalt / Wert von Variable c? (1 Punkt)

```
String c = "Hallo" + 40 + 2 + "Welt";
```

Hallo402Welt

2 – D (11 Punkte)

1. Was wird auf der Konsole ausgegeben? (1 Punkt)

```
System.out.println(26 % 5);
```

1

2. Welchen Wert (true/false) liefern folgende Ausdrücke? (6 Punkte; Jede richtige Antwort gibt 1 Punkt, jede falsche Antwort gibt 1 Punkt Abzug, keine Antwort gibt keine Punkte)

true false

☒ ☐ (25 % 5) != 0 || 2 <= 3

☐ ☒ "Hans Muster".length() == 10

☒ ☐ true && (!false || !true)

☒ ☐ 20 % 7 == 3 || 20 % 7 == 6

☒ ☐ (new String("Hallo")).equals(new String("Hallo"))

☐ ☒ new String("Hallo") == new String("Hallo")

3. Was wird auf der Konsole ausgegeben? (4 Punkte)

```
int a = 0;
int b = 20;
while (a < 4 || b >= 15) {
    a++;
    b = b - 3;
}
System.out.println("Wert von a: " + a);
System.out.println("Wert von b: " + b);
```

Wert von a: 4

Wert von b: 8

2 – E (7 Punkte)

Was wird auf der Konsole ausgegeben? (7 Punkte)

```
String[] strArr = new String[7];
strArr[1] = "Haus";
strArr[3] = "Rot";
strArr[4] = "Sugus";
strArr[5] = strArr[4].substring(1, 4);
strArr[6] = "Blau";

for (String string : strArr) {
    System.out.println(string);
}
```

```
null
Haus
null
Rot
Sugus
ugu
Blau
```

2 – F (5 Punkte)

Gegeben:

```
int[] array = new int[20];
```

Schreiben Sie einen For-Loop, um alle Elemente im Array mit der Zahl ihres Indexes zu füllen. An der Position 0 soll also eine 0 stehen, an der Position 1 eine 1 und so weiter.

```
for(int i = 0; i < array.length; i++) {
    array[i] = i;
}
```

2 – G (5 Punkte)

Gegeben sei eine `ArrayList<String>` mit dem Namen „stringList“. Schreiben Sie einen While-Loop mit einem Iterator, um alle Strings in der Liste auf die Konsole auszugeben.

```
Iterator<String> iter = stringList.iterator();  
while (iter.hasNext()) {  
    System.out.println(iter.next());  
}
```

2 – H (3 Punkte)

Gegeben ist folgende Klasse:

```
public class Zaehler {  
  
    private static int zaehler1;  
    private int zaehler2;  
  
    public Zaehler() {  
        zaehler1 += 5;  
        zaehler2 += 5;  
    }  
}
```

Welchen Wert haben `zaehler1` und `zaehler2` nach der Ausführung von folgendem Code in den Objekten `z1`, `z2` und `z3`?

```
Zaehler z1 = new Zaehler();  
Zaehler z2 = new Zaehler();  
Zaehler z3 = new Zaehler();
```

```
zaehler1: 15  
zaehler2 in z1: 5  
zaehler2 in z2: 5  
zaehler2 in z3: 5
```


Aufgabe 2 – I (3 Punkte)

Gegeben ist folgender Code:

```
short s = 250;
byte b = (byte) s;
System.out.println(b);
```

Was wird auf der Konsole ausgegeben?

-6

Aufgabe 2 – J (3 Punkte)

Gegeben ist folgende Klasse:

```
public class Person {

    private String name;

    public Person(String name) {
        this.name = name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }
}
```

In einer Anderen Klasse wird folgender Code ausgeführt:

```
Person p1 = new Person("Michael");
Person p2 = new Person("Julia");
Person p3 = new Person("Max");
p2 = p3;
p1 = p2;
p1.setName("Katrin");
System.out.println(p1.getName());
System.out.println(p2.getName());
System.out.println(p3.getName());
```

Was wird auf der Konsole ausgegeben?

Katrin
Katrin
Katrin