

Bitte lösen Sie die Übungsaufgaben bis zum 03/04 Dezember. Wir besprechen die Lösungen in der nächsten Übung (sowohl Aufgabenblatt 8 als auch Aufgabenblatt 7). Lösen Sie die Aufgaben am besten in Pair Programming. Optionale Aufgaben könnten ein wenig länger dauern und müssen nicht gelöst werden. Das ist eine total subjektive Einschätzung, es könnte sein, dass Sie sehr schnell die Lösung finden. Daher meine Bitte: IMMER als Richtwert nehmen.

Aufgabe 8.1 (Überschreiben) – Gegeben sei das untere Programm. Was sind jeweils die Ausgaben? Machen Sie sich eine Skizze.

```
public class A {
    int myVar = 10;
    public void m1() {
        System.out.println("A m1");
    }
    public void m2() {
        System.out.println("A m2");
    }
    public void m3() {
        System.out.println("A m3");
    }
}

public class B extends A{
    public void m1() {
        System.out.println("B m1");
    }
}

public class C extends B {
    public void m3() {
        System.out.println("C m3");
    }
}

public class UeberschreibenTest {
    public static void main(String[] args) {

        A a = new A();
        B b = new B();
        C c = new C();
        A a1 = new C();

        a.m1(); // 1
        b.m2(); // 2
        c.m3(); // 3

        a.m2(); // 4
        b.m1(); // 5
        c.m2(); // 6

        a.m3(); // 7
        b.m3(); // 8
        c.m1(); // 9

        a1.m1(); // 10
        a1.m2(); // 11
        a1.m3(); // 12
    }
}
```

Aufgabe 8.2 (Recap Vorlesung) - Gehen Sie die Folien nochmals durch und vollziehen Sie die Beispiele nach. Wenn Sie noch unsicher sind, implementieren Sie die Beispiele nach.

Aufgabe 8.3 (String-Array-Liste) – Erweitern Sie die auf Übungsblatt 7 bereits implementierten Methoden um folgende Methoden, **ohne Java-Bibliotheken zu verwenden**. (Exception handling soll nicht implementiert werden).

Folgende Spezifikation ist angegeben:

```
// Erzeugt ein String-Array mit der Standardkapazität 10
public StringArrayList ()

// Erzeugt ein String-Array mit der angegebenen „Kapazität“
public StringArrayList (int capacity)

// Gibt die Anzahl der Elemente im Array zurück
public int getSize ()

// NEU: Gibt true zurück wenn die StringArrayListe leer ist, false sonst
public boolean isEmpty ()

// NEU: Gibt true zurück wenn s Element der StringArrayListe ist, false sonst
public boolean contains(String s)

// Fügt den übergebenen String zum String-Array hinzu
// Ist das Array „voll“, wird die Größe angepasst
public void add(String aktuellerString)

// Gibt den String an der Position index zurück
public String get(int index)

// Löscht den String an der Position index
public void remove(int index)
```

Aufgabe 8.4 (Klassen - OPTIONAL) – Überarbeiten Sie die Klasse Cocktail derart, dass die Schnittstelle nach außen NICHT verändert wird. Ersetzen Sie die bisherige interne unflexible Speicherung der Zutatenliste über ein Array durch die StringArrayListe, die Sie implementiert haben. Überarbeiten Sie Ihre Methoden entsprechend. Testen Sie Ihre neuen Methoden. Die bisherigen Tests dürften weiterhin laufen (wenn Sie die Schnittstellen nicht verändert haben).

- Name // Name des Cocktails
 - Zutaten // String-Array mit Zutaten → NEU: Zutatenliste intern als StringArrayList
 - IstGeheimrezept // gibt an ob das Rezept „gedruckt“ werden kann oder nicht
1. Schreiben Sie einen Konstruktor, der als Parameter den Namen des Cocktails bekommt.
 2. Schreiben Sie einen Konstruktor, der als Parameter den Namen sowie die Zutatenliste des Cocktails bekommt.
 3. Schreiben Sie getter und setter für die Instanzvariablen
 4. Schreiben Sie eine Methode, die abfragt ob es ein Geheimrezept ist
 5. Schreiben Sie eine Methode, die den Namen und die Zutatenliste ausgibt (wenn es kein Geheimrezept ist, ansonsten pssst...)

6. Angenommen, Sie experimentieren mit den Cocktail-Rezepten und würden gerne ab und zu die Zutatenliste ergänzen. → NEU: Schreiben Sie eine Methode, die eine neue Zutat hinzufügt.
7. Was ist Regression? Was sind Regressionstests?

Aufgabe 8.5 (Klassen, Vererbung) – Angenommen Sie führen eine Anforderungsanalyse für ein Mietportal für Wohnungen und Häuser durch. Was könnten Eigenschaften der Objekte „Wohnung“ und „Haus“, die im Portal verwaltet und angezeigt werden, sein? Was könnte in einer gemeinsamen Oberklasse „Immobilie“ definiert werden? Hier geht es nicht um eine syntaktisch korrekte Klassendefinition oder UML-Darstellung. Nennen Sie einfach Eigenschaften und Methoden, die in einer der jeweiligen Klassen Sinn machen.