

Sommersemester 2019  
**Übungen zur Vorlesung**  
**Objektorientierte Softwareentwicklung (BA-INF-024)**  
**Aufgabenblatt 1 (20 Punkte)**

Zu bearbeiten bis: 19.04.2019

Hinweis: Auf der Vorlesungswebseite gibt es ein Präsenzblatt, welches in der nächsten Woche (15.4 - 18.4) besprochen wird.

**Aufgabe 1** (*Übersetzung von Java-Quellcode und Kommandozeilenparameter - 6 Punkte*)

a) Öffnen Sie einen beliebigen Text-Editor und fügen Sie den folgenden Quellcode ein:

```
public class Name {  
    public static void main(String[] args) {  
        System.out.println("Hallo " + args[0] + " !");  
    }  
}
```

Speichern Sie die Datei unter dem Namen `Name.java`. Öffnen Sie nun eine Kommandozeile und wechseln Sie in das Verzeichnis, in dem Sie die Datei abgespeichert haben. Übersetzen Sie dann das Programm mit `javac Name.java`<sup>1</sup>. Rufen Sie anschließend das Programm nach dem Muster `java Name Erik Mustermann` auf. Nachfolgend können bei jedem Programmaufruf andere Werte angegeben werden, ohne dass ein erneutes Übersetzen notwendig ist.

b) Erweitern Sie das Programm, in dem drei Integervariablen mit Eingabewerten aus der Kommandozeile belegt werden<sup>2</sup> und anschließend entweder die Summe, Differenz, Produkt oder Quotient der eingegebenen Werte ausgegeben wird. Dabei soll die Rechenoperation durch die Dritte eingelesene Zahl bestimmt werden.

**Aufgabe 2** (*Java-Projekt mit Eclipse und Bibliotheksverwaltung - 4 Punkte*)

a) Auf Übungszettel 0 haben Sie die Entwicklungsumgebung Eclipse eingerichtet. Diese Aufgabe soll Sie vertraut mit der Nutzung von Eclipse und dem Einbinden von Paketen machen. Starten Sie Ihre Eclipse-Installation und erzeugen Sie mit *File* → *New* ein neues Java-Projekt. Klicken Sie mit der rechten Maustaste auf das gerade erstellte Projekt und wählen Sie diesmal: *New* → *Class*. Anschließend geben Sie noch den Klassennamen ein (z.B.: `Name`) und wählen die Option für die Erstellung von `public static void main(String[] args)` an. Ergänzen Sie die `main`-Methode mit folgenden Eingaben:

```
System.out.print("Wie ist dein Name? ");  
String name = Input.readString();  
System.out.println("Hallo " + name + " !");
```

Ein rotes Kreuz signalisiert Ihnen, dass die Datei `Name.java` einen Fehler enthält. Sie sehen, dass in der Zeile `String name = Input.readString();` das `Input` bemängelt wird. Damit will der Compiler Ihnen mitteilen, dass ihm eine Benutzer-Bibliothek zur Übersetzung des Quelltextes fehlt. Die fehlende Bibliothek `input.jar` finden Sie auf der Webseite der Vorlesung. Importieren Sie (z.B. via Drag'n'Drop) die Datei `input.jar` in das neu erstellte Projekt. Eclipse verwaltet den *Classpath* unabhängig von Parameterangaben in den Projekteinstellungen. Fügen Sie die Datei `input.jar` dem *Classpath* hinzu, indem Sie diese mit der rechten Maustaste anklicken und aus dem Kontextmenü den Befehl *Build Path* → *Add to Build Path* auswählen. Das rote Fehlerkreuz sollte verschwinden. Starten Sie nun das Programm, indem Sie aus dem Eclipse-Kontextmenü der Datei `Name.java` den Befehl *Run As* → *Java Application* auswählen.

<sup>1</sup>Hinweis: Falls Sie ein MS Windows-Betriebssystem verwenden, stellen Sie sicher, dass Sie den vollständigen Pfad zu `javac` angegeben oder den Pfad zum bin-Ordner der Umgebungsvariable „PATH“ hinzugefügt haben.

<sup>2</sup>Hinweis: verwenden Sie die Hüllklasse `Integer` für die Konvertierung der als String übergebenen Kommandozeilenparametern

### Aufgabe 3 (Java-Code-Fragment - 3 Punkte)

Legen Sie ein neues Java-Projekt an. Erstellen Sie wie in Aufgabe 2 ein neues Java-Projekt und eine Java-Class-File. Fügen Sie das folgende Codefragment in die main-Methode ein:

```
java.util.Scanner sc = new java.util.Scanner(System.in);
System.out.println("Eingabe: ");
int k = sc.nextInt(); // liest eine Ganzzahl (int) ein
System.out.println(++k); // Ausgabe a)
System.out.println(k); // Ausgabe b)
System.out.println(k++); // Ausgabe c)

int k2 = 5;
double result = k2 / 2; // Stelle 1
System.out.println("5 / 2 = " + result); // Ausgabe d)

boolean b = false;
if(b = false) {
    System.out.println("b ist falsch."); // Ausgabe e)
} else {
    System.out.println("b ist wahr."); // Ausgabe f)
}
```

- a) Erklären Sie die Ausgaben a), b) und c) des Programms. Wieso wird bei allen dieselbe Zahl ausgegeben?
- b) Erklären Sie, was an Stelle 1 verändert werden muss, um eine Gleitkomma-Division durchzuführen.
- c) Erklären Sie, warum Ausgabe f) zustande kommt. Welcher Fehler wurde gemacht? Nutzen Sie einen bedingten Ausdruck, um die gesamte if-else-Anweisung in einer einzelnen Zeile mit derselben Funktion zu ersetzen.

### Aufgabe 4 (Operatoren in Java - 7 Punkte)

a) gegeben sei folgender Quelltext:

```
public class Operatoren {
    public static void main(String [] args){
        int a=15;
        int b=34;
        int c=1;
        System.out.println("1: "+ (a==b));
        System.out.println("2: "+ (a==b-19));
        System.out.println("3: "+ (a^b));
        System.out.println("4: "+ (a++^b));
        System.out.println("5: "+ (7<<++c));
        System.out.println("6: "+ (a=b=c=0x10));
        System.out.println("7: "+ (1e1));
    }
}
```

Was sind die Ergebnisse des Programms? Erklären Sie wie die Ergebnisse zustande kommen.