



# Informatik I – Eprog HS11

## Übung 2

### 1 Aufgabe: Primitive- & Referenztypen

#### 1.1 Lernziele

1. Unterschied zwischen primitiven Datentypen und Objekten erklären können.
2. Verschiedene Funktionsweise der primitiven Datentypen und Referenztypen verstehen.

#### 1.2 Aufgabenstellung

##### a) Verständnis

1. Was verstehen Sie unter einem „primitiven Datentyp“?
2. Inwiefern unterscheiden sich primitive Datentypen von Referenztypen?

##### b) Funktionsweise

Wie lautet die Ausgabe der beiden Code-Snippets?<sup>1</sup> Begründen Sie Ihre Antwort.

```
1      int a = 5, b = 7;
2
3      a = b;
4      a++;
5
6      System.out.println(a);
7      System.out.println(b);
```

Listing 1: Snippet 1

<sup>1</sup>Gehen Sie davon aus, dass Number mit `setNumber(int newNumber)` eine Zahl korrekt speichern und mit `printNumber()` ausgeben kann?

```
1     Number x = new Number();
2     Number y = new Number();
3     x.setNumber(5);
4     y.setNumber(7);
5
6     x = y;
7     x.setNumber(8);
8
9     x.printNumber();
10    y.printNumber();
```

**Listing 2:** Snippet 2

## **2 Aufgabe: Deklarationen, Zuweisungen, Verschicken von Nachrichten**

### **2.1 Lernziele**

1. Elementare Arten von Statements kennen.

### **2.2 Aufgabenstellung**

Geben Sie je zwei Beispiele für die folgenden Arten von Statements:

- a) Deklaration**
- b) Initialisierung**
- c) Verschicken einer Nachricht**

### 3 Aufgabe: Cascading, Composition

#### 3.1 Lernziele

1. Cascading und Composition von Methodenaufrufen anhand von String-Operationen verstehen und anwenden können.

#### 3.2 Aufgabenstellung

Gegeben seien die folgenden Deklarationen:

```
1 String s1 = "butterfly";  
2 String s2 = "tiger";
```

Welche Werte liefern die folgenden Nachrichten-Aufrufe?

##### a) Aufrufe

1. `s1.substring(6).concat(s2.substring(2));`
2. `s1.substring(6).concat(" ").concat(s2).concat("!");`
3. `s1.substring(0, 7).concat(s2.substring(1, 2)).concat("\n").concat(s2.substring(2));`
4. `s2.toUpperCase().concat("WOODS".toLowerCase()).toLowerCase();`

## 4 Aufgabe: Codeverständnis

### 4.1 Lernziele

1. Eine Klasse und deren Verhalten mit Hilfe der [Java API](#) am Beispiel der `String`-Klasse kennen lernen.
2. Codelesen und Codeverständnis trainieren.

### 4.2 Aufgabenstellung

#### a) Code Snippets

Welche der nachfolgenden vierzehn Code-Fragmente sind syntaktisch korrekt und wie lautet in diesem Fall die Ausgabe? Begründen Sie Ihre Antwort. Benutzen Sie dazu folgende Hilfsmittel:

- Walter Savitch, Java: An Introduction to Problem Solving & Programming
- [Java API](#)

```
1 String s;  
2 s = "Hallo Welt";  
3 System.out.println(s);
```

**Listing 3:** Snippet 1

```
1 String s;  
2 "Hallo Welt" = s;  
3 System.out.println(s);
```

**Listing 4:** Snippet 2

```
1 String s = "Hallo Welt";  
2 s.toUpperCase();  
3 System.out.println(s);
```

**Listing 5:** Snippet 3

```
1 System.out.println("Hallo Welt".toUpperCase());
```

**Listing 6:** Snippet 4

```
1 String s = "Hallo Welt";  
2 PrintStream t = "Hallo Universitaet";  
3 t = s;  
4 System.out.println(t);
```

**Listing 7:** Snippet 5

```
1 String s = System.out.println("Hallo Welt");
```

**Listing 8:** Snippet 6

```
1 String s = "Hello World".toUpperCase();
2 System.out.println(s);
```

**Listing 9:** Snippet 7

```
1 String s = "w" + "ooo".toUpperCase() + "t";
2 System.out.println(s);
```

**Listing 10:** Snippet 8

```
1 String s;
2 String t = "Hallo Universitaet";
3 s = "Hallo Welt";
4 t = s;
5 s = t;
6 System.out.println(s);
7 System.out.println(t);
```

**Listing 11:** Snippet 9

```
1 String s = new String(int i = 0);
2 System.out.println(s);
```

**Listing 12:** Snippet 10

```
1 String s = new String("Hallo Welt").toUpperCase();
2 System.out.println(s);
```

**Listing 13:** Snippet 11

```
1 String s = new String("Hallo Welt".toUpperCase());
2 System.out.println(s);
```

**Listing 14:** Snippet 12

```
1 System.out.println(new String("Hallo ").concat(new String("schoene ").
    concat("Welt".toUpperCase())));
```

**Listing 15:** Snippet 13

```
1 String s = new "Hallo Welt";
2 System.out.println(s);
```

**Listing 16:** Snippet 14

## 5 Aufgabe: OOP

### 5.1 Lernziele

1. Den Unterschied zwischen Klassen und Objekten kennen.
2. Gängige Begriffe der objektorientierten Programmierung definieren und zuordnen können.

### 5.2 Aufgabenstellung

1. Stellen Sie fest, bei welchen Begriffen aus Table 1 es sich um Klassen, Objekte, Methoden oder Eigenschaften handeln könnte. Es können verschiedene Lösungen möglich sein, abhängig von der Implementierung.

Begriff	Klasse	Objekt	Methode	Eigenschaft
Hund				
bell				
Hund Barney				
Fellfarbe				
Auto				
Autonummer				
bremsen				

**Table 1:** OO-Begriffe

2. Erklären Sie die nachfolgenden Begriffe im Kontext der objektorientierten Programmierung:
  - (a) Klasse
  - (b) Instanz
  - (c) Nachricht
  - (d) Referenz
  - (e) Überladen (Overloading)

## 6 Aufgabe: Klassen

### 6.1 Lernziele

1. Eine Klasse mit Zustand (Instanzvariablen) und Methoden mit Übergabewerten implementieren können.
2. Das Verhalten der erstellten Klasse mittels eines TestDrivers überprüfen.

### 6.2 Aufgabenstellung

#### a) Adder

Schreiben Sie eine Klasse mit dem Namen `Adder` und den Methoden `reset()`, `add()` und `sum()`.

- Die Methode `add()` soll einen einzigen Parameter vom Typ `long` erwarten und diesen zur Summe aller bisher erhaltenen Werte hinzuaddieren.
- `sum()` soll die Summe der bisher erhaltenen Werte auf dem Bildschirm ausgeben.
- `reset()` setzt das Objekt zurück, sprich die Summe der bisher erhaltenen Werte werden wieder auf 0 gesetzt.

Erzeugen Sie in der `main()`-Methode eines TestDrivers einige Objekte der Klasse `Adder` und testen Sie die `reset()`-, `add()`- und `sum()`-Methoden.

```
1 public class Adder {
2
3     . . .
4
5     public void add( . . . ) {
6         . . .
7     }
8
9     public void reset() {
10        . . .
11    }
12
13    public void sum() {
14        . . .
15    }
16 }
```