

## P1 - Serie 02

Lukas Batschelet - 16-499-733

### (01) new -Operator

Der `new`-Operator zusammen mit dem Konstruktor einer Klasse erledigt zwei Dinge. Was genau?

- Reservierung von Speicher
- Initialisierung des Objekts zusammen mit dem Konstruktor

### (02) ArrayList in der Java API

Informieren Sie sich in der Java API Dokumentation über die Klasse `ArrayList` (welche eine Liste für Objekte repräsentiert).

- Wie instanziiert man eine solche Liste?
  - `ArrayList list = new ArrayList();`
  - Oder mit einer Zuordnung des Typs (hier mit dem Beispiel `String`)
    - `ArrayList<String> list = new ArrayList<>();`
- Wie fügt man ein Objekt zur Liste hinzu?
  - mit der Methode `.add()` wird ein Objekt am Ende der Liste eingefügt
    - Möglicher Parameter ist der Index vom Typ `int`, welcher angibt an welcher Position das Objekt in die Liste eingefügt werden soll
- Wie greift man auf ein Objekt an Position `i` zu?
  - mit der Methode `.get(i)`
- Wie löscht man den gesamten Inhalt der Liste?
  - mit der Methode `.clear()`
- Wie kann man überprüfen, ob ein bestimmtes Objekt in der Liste vorhanden ist?
  - mit der Methode `.contains()` wird ein `boolean` Wert zurückgegeben, ob das Objekt welches als Parameter "mitgegeben wurde" in der Liste ist

### (03) Unterschied zwischen Klassen und Objekten

Erläutern Sie anhand der Klasse `String` und des Objektes `"String"` den Unterschied zwischen Klasse und Objekt.

- Die Klasse `String` gibt den "Bauplan" für alle Objekte in ihr vor
  - Sie definiert die Methoden (`length()`, `substring()`, etc.) welche auf einem Objekt der Klasse ausgeführt werden können
- Das Objekt `"String"` wurde mit dem "Bauplan" der Klasse `String` "gebaut"
  - Methoden (in der Klasse definiert) die darauf angewendet werden

### (04) Ausgabe

Welche Ausgabe erzeugen folgende Anweisungen?

```
1  String testString = "Think different";
2  System.out.println(testString.length());
3  System.out.println(testString.substring(0, 4));
4  System.out.println(testString.toUpperCase());
5  System.out.println(testString.charAt(7));
6  System.out.println(testString);
```

```
//Ausgabe
15
Thin
THINK DIFFERENT
i
Think different
```

### (05) Random

Gegeben sei eine Objektvariable vom Typ `Random` mit Bezeichner `rand`. In welchen Intervallen suchen die folgenden Anweisungen eine Zufallszahl?

- `rand.nextInt(100) + 1;`
  - `[1, 100]`
- `rand.nextInt(51) + 100;`
  - `[100, 150]`
- `rand.nextInt(10) - 5;`
  - `[-5, 4]`

- `rand.nextInt(3) - 3;`
  - `[-3, -1]`

## (06) Aliase

*Was sind Aliase und weshalb können Aliase problematisch sein?*

- Aliase sind Variablen, die auf dasselbe Objekt zeigen. Bei primitiven Datentypen passiert das nicht, da der Wert kopiert wird.

```
int num1 = 17;
int num2 = num1;
num2 = 99; System.out.println(num1); // 17
System.out.println(num2); // 99
```

Bei Objektvariablen sieht es anders aus:

```
Integer num1 = new Integer(17);
Integer num2 = num1;
num2.setValue(99);
System.out.println(num1); // 99
System.out.println(num2); // 99
```

Hier greifen beide Variablen auf das selbe Objekt zu. Das ist nicht immer wünschenswert, weil:

- Es ist unklar, welche Variable für was zuständig ist.
- Änderungen an einer Variable beeinflussen die andere.
- Der Code wird unübersichtlicher.
- Es kann sogenannter *garbage* und damit einhergehender Datenverlust entstehen.
  - Im obigen Beispiel aus der Vorlesung ist dem Objekt `num1` selber kein Wert mehr zugewiesen, sondern nur noch die Verweisung auf `num2`. Der Wert `17` wird damit zu *garbage*

## Implementationsaufgaben

### (01) Rechteck

Schreiben Sie ein Programm, das nach der Länge und Breite eines Rechtecks fragt und danach die Fläche und den Umfang des Rechtecks berechnet und ausgibt. Zusätzlich soll Ihr Programm feststellen, ob es sich beim definierten Rechteck um ein Quadrat handelt oder nicht und eine entsprechende Ausgabe erzeugen.

```
import java.util.Scanner;

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

        System.out.println("Geben Sie die Länge des Rechtecks ein:");
        Scanner scan = new Scanner(System.in);
        double length = scan.nextDouble();
        System.out.println("Geben Sie die Breite des Rechtecks ein:");
        double width = scan.nextDouble();
        scan.close();

        double area = length * width;
        double perimeter = 2 * (length + width);

        System.out.println("Die Fläche des Rechtecks beträgt: " + area);
        System.out.println("Der Umfang des Rechtecks beträgt: " + perimeter);
        if (length == width) {
            System.out.println("Das Rechteck ist ein Quadrat.");
        } else {
            System.out.println("Das Rechteck ist kein Quadrat.");
        }
    }
}
```

### (02) Zufällige Addition

Schreiben Sie ein Programm, das eine zufällige Additionsaufgabe mit zwei positiven Zahlen anzeigt. Die Summe der beiden Zahlen darf maximal 20 betragen. Der Benutzer soll dann ein Ergebnis eingeben können und das Programm soll überprüfen, ob die Eingabe korrekt war oder nicht und eine entsprechende Rückmeldung ausgeben.

```
import java.util.Random;
import java.util.Scanner;

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

        Random random = new Random();
        final int MAX = 21;
        int number1 = random.nextInt(MAX);
        // Die Summe der beiden Zahlen darf maximal 20 betragen.
        int number2 = random.nextInt(MAX - number1);
        int sum = number1 + number2;

        Scanner scan = new Scanner(System.in);

        System.out.println("Die Aufgabe lautet: " + number1 + " + " + number2);
        System.out.println("Geben Sie das Ergebnis ein:");
        int guess = scan.nextInt();

        scan.close();

        if (guess == sum) {
            System.out.println("Das Ergebnis ist korrekt!");
        } else {
            System.out.println("Das Ergebnis ist falsch!");
        }
    }
}
```

### (03) Username and Password

Schreiben Sie ein Programm, das einen Benutzer separat nach seinem Vor- und Nachnamen fragt und diese einliest. Danach soll ein Benutzername nach folgendem Muster erzeugt werden:

$$FL_1L_2L_3L_4L_5D_1D_2D_3$$

wobei

- $F$  dem ersten Buchstaben des Vornamens entspricht.

- $L_i$  dem  $i$ -ten Buchstaben des Nachnamens entspricht.
- $D_1D_2D_3$  einer zufälligen Zahl zwischen 000 und 999 entspricht.

Falls der eingegebene Nachname kürzer als 5 Zeichen sein sollte, werden entsprechend weniger Zeichen verwendet.

Zusätzlich erzeugen Sie ein zufälliges Passwort für den Benutzer. Das Passwort soll mit einer 7 oder 8 oder 9 starten, gefolgt von 5 zufälligen ganzen Zahlen von 0 bis 9, gefolgt von einem Bindestrich -, gefolgt von drei zufälligen Grossbuchstaben.

Geben Sie Benutzernamen und Passwort aus.

Tipp: Die Zahlen 65 bis 90 repräsentieren die Grossbuchstaben A bis Z in Unicode und Sie können bspw. die ganze Zahl 77 folgendermassen in einen Buchstaben umwandeln:

```
char d1 = (char) 77;
```

```
import java.util.Scanner;
import java.util.Random;

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

        Scanner scan = new Scanner(System.in);

        System.out.println("Geben Sie Ihren Vornamen ein:");
        String firstName = scan.nextLine();
        System.out.println("Geben Sie Ihren Nachnamen ein:");
        String lastName = scan.nextLine();
        scan.close();

        int lastNameLength = lastName.length();
        final int MAX_LETTERS_OF_LASTNAME = 5;

        String firstLetter = firstName.substring(0, 1);
        // Math.min(a, b) sorgt dafür, dass auch Nachnamen mit weniger
        // als MAX_LETTERS_OF_LASTNAME Buchstaben funktionieren
        String lastNameArray = lastName.substring(0, Math.min(MAX_LETTERS_OF_LASTNAME,
            lastNameLength));

        Random random = new Random();
        int randomNumber = random.nextInt(1000);
        // String.format("%03d", randomNumber) fügt führende Nullen hinzu,
        // falls randomNumber < 100
        String username = firstLetter.toUpperCase() + lastNameArray.toUpperCase() +
            String.format("%03d", randomNumber);

        System.out.println("Der Benutzername lautet: " + username);

        final int UNICODE_LOWER_BOUND = 65;
        final int UNICODE_UPPER_BOUND = 91;
        // ermöglicht die zufällige Ausgabe des ganzen Zahlenteils des Passworts
        int randomPasswordNumber = random.nextInt(700000, 1000000);
        char randomPasswordLetter1 = (char) random.nextInt(UNICODE_LOWER_BOUND,
UNICODE_UPPER_BOUND);
        char randomPasswordLetter2 = (char) random.nextInt(UNICODE_LOWER_BOUND,
UNICODE_UPPER_BOUND);
        char randomPasswordLetter3 = (char) random.nextInt(UNICODE_LOWER_BOUND,
UNICODE_UPPER_BOUND);

        String password = randomPasswordNumber + "-" + randomPasswordLetter1 +
            randomPasswordLetter2 + randomPasswordLetter3;

        System.out.println("Das Passwort lautet: " + password);
    }
}
```