

## PR1 – Formular für Lesenotizen

### WS2020/21

Nachname Abdel Kader	Vorname Schehat	Matrikelnummer 1630110	Abgabedatum: 17.12.2020
-------------------------	--------------------	---------------------------	----------------------------

## Maps, Rekursion & Zeichenketten-Vertiefung

### Lernzielfragen:

- a) Betrachten Sie die folgende Datenstruktur:

```
TreeMap<String, ArrayList<Integer>> c= new TreeMap<String, ArrayList<Integer>>();
```

Der Benutzer gibt ein Wort und eine ganze Zahl auf der Console ein:

```
String wort= console.next();
int zahl= console.nextInt();
```

```
Scanner console = new Scanner(System.in);
TreeMap<String, ArrayList<Integer>> c= new TreeMap<>();
int i = 0;
while (i < 5) {
    System.out.print("Geben Sie ein Wort und eine Zahl ein: ");
    String wort= console.next();
    int zahl= console.nextInt();
    if (!c.containsKey(wort)) {
        ArrayList<Integer> arr = new ArrayList<>();
        arr.add(zahl);
        c.put(wort, arr);
    } else {
        c.get(wort).add(zahl);
    }
    i++;
}
System.out.println(c);
```

- d) Double Zahl mit 4 Zahlen vor dem Komma und 2 Zahlen nach dem Komma formatieren. Soll gegebenenfalls mit 0'en gefüllt werden

```
Scanner console = new Scanner(System.in);
System.out.print("bitte eine Zahl eingeben: ");
double d = console.nextDouble();
String num = Double.toString(d);
String[] spl = num.split("\\.");
String pre = "";
String post = "";
if (spl[0].length() > 4) {
    pre = spl[0].substring(spl[0].length() - 4);
} else {
    pre = "0000".substring(0, 4-spl[0].length());
}
if (spl[1].length() > 2 ) {
    post = spl[1].substring(0, 2);
} else {
    post = "00".substring(0, 2-spl[1].length());
}
num = pre + "." + post;
System.out.println(num);
```

**Backtracking c)**

```

if (y < 0 || y >= lab.length || x < 0 || x >= lab.length) return "Draußen";
System.out.println();
if (lab[x][y] == "2") return "Ziel";
if (lab[x][y] == "#") return "Wand";
lab[x][y] = ".";
String weg = suche(x, y + 1, lab); // Rechts
if (weg.endsWith("Ziel")) return "R " + weg;
weg = suche(x + 1, y, lab); // Unten
if (weg.endsWith("Ziel")) return "U " + weg;
lab[x][y] = " ";
return "Sackgasse";

```

**Rekursion**

- Es gibt keine Endlosrekursion, sondern wenn der stack frame die Grenze des verfügbaren RAM erreicht

**Backtracking****Umwandlung Zahlen zu Strings**

- Integer.toString(<zahl>) => geht auch für Double usw.

**Formatierung von Zahlen in Strings**

```

int i= 78973;
String is= Integer.toString(i);
int ziffern= is.length();
String prefix= "00000000".substring(0, 8-ziffern);
is= prefix + is;

```

**Umwandlung von Strings in Zahlen**

- Integer.parseInt(<zahl>) => geht auch für Double usw.

**StringBuilder**

```

StringBuilder sb= new StringBuilder()
StringBuilder sb= new StringBuilder("Initialer Inhalt")

```

- Klasse für veränderbare Zeichenketten

- Anhängen ans Ende:  
sb.append("mehr Text");
- Einfügen:  
sb.insert(pos, "einzufügender Text");
- Ersetzen:  
sb.setChar(pos, 'X');
- Konvertierung in eine konstante Zeichenkette  
String s = sb.toString();
- Mehr Details finden Sie hier [2].

**String.split**

Methode	Beschreibung	Beispiel
split( <i>begrenzer</i> )	Separiert den String anhand des gegebenen Begrenzers in ein Array von Teilstrings	s.split(" ") liefert {"long", "book"} s.split("o") liefert {"l", "ng b", "", "k"}

- Reguläre Ausdrücke erwartet String.split
  - ⇒ "[a-z]" beschreibt die kleinen Buchstaben von a-z
  - ⇒ "\d+" beschreibt Anzahl der Ziffern
  - ⇒ ".{5}" beschreibt Menge, die aus 5 Zeichen bestehen. Punkt bedeutet beliebige Zeichen
  - ⇒ "\\." Beschreibt Punkte