

## 8. Übung

**Abgabe: bis 10.01.2023, 6 Uhr**

**Abnahme DFHI: 10.01.2023**

**Abnahme PI: 11.01.2023**

Die Aufgaben in dieser Übung können in ähnlicher Form typischerweise in der Klausur auftauchen.

### Aufgabe 1 - Patientenwarteschlange

Programmieren Sie eine Patientenwarteschlange. Patienten können an die Warteschlange angehängt werden. Die Eigenschaften der dazu benötigten Klasse Patient können Sie dem untenstehenden Testprogramm entnehmen. Patienten können aus der Warteschlange wieder entfernt werden, d.h. die Patienten verlassen die Warteschlange ohne eine Behandlung. Der Patient, der an erster Stelle steht, kann zum Arzt gerufen werden. Dabei kann kein Patient in der Warteschlange übersprungen werden, d.h. wer vorne steht, wird zuerst bedient. Wenn ein Patient der Warteschlange entnommen wird, rücken alle anderen nach. Außerdem soll der Inhalt der Warteschlange ausgegeben werden können.

Das folgende Programm soll lauffähig sein:

```
public class PatientenwarteschlangeDialog {
    public static void main(String[] args) {
        try {
            // Warteschlange fuer maximal 10 Patienten anlegen
            PatientenWarteschlange patientenwarteschlange = new
                PatientenWarteschlange(10);
            patientenwarteschlange.neuerPatient(4711, "Tom", "Ate");
            // Neue Patienten anhaengen
            patientenwarteschlange.neuerPatient(1234, "Elle", "Fant");
            patientenwarteschlange.neuerPatient(1111, "Moni", "Tor");
            patientenwarteschlange.neuerPatient(2222, "Phil", "Harmonie");
            System.out.println(patientenwarteschlange); // Warteschlange ausgeben
            Patient entfernterPatient = patientenwarteschlange.entfernePatient(1234);
            System.out.println(entfernterPatient); // Patient entfernen
            // naechsten Patient herausholen
            Patient naechsterPatient = patientenwarteschlange.derNaechsteBitte();
            System.out.println("Naechster Patient: " + naechsterPatient);
            System.out.println(patientenwarteschlange);
        } catch (RuntimeException e) {
            System.out.println(e);
        }
    }
}
```

Beispielausgabe:

Warteliste

Patientennummer	Vorname	Nachname
4711	Tom	Ate
1234	Elle	Fant
1111	Moni	Tor
2222	Phil	Harmonie

Patient gelöscht: 1234, Elle Fant

Nächster Patient: 4711, Tom Ate

Warteliste

Patientennummer	Vorname	Nachname
1111	Moni	Tor
2222	Phil	Harmonie

## Aufgabe 2 - Messwerte

Schreiben Sie eine Klasse `ArrayFunctions` einer Klassenmethode, *Mittelwert berechne*`Mittelwert(double[] messwerte)`, die zu dem übergebenen Feld das arithmetische Mittel berechnet und zusätzlich den Messwert bestimmt, der am nächsten dran liegt und den, der am weitesten davon entfernt ist. Da die Methode drei Werte berechnet und eine Methode nur maximal einen Wert zurückgeben kann, benötigen Sie noch eine Klasse `Mittelwert`.

Diese Klasse ist einfach nur ein Datencontainer für die Rückgabe der Werte. Sie enthält drei Attribute und entsprechende get-Methoden:

- `mittelwert`
- `nahesterWert`
- `entferntesterWert`

## Aufgabe 3 - Strings auswerten

Erweitern Sie die Klasse `ArrayFunctions` um die Klassenmethode *int stringsAuswerten*`(String[] strings)`. Die Methode zählt, wie viele Strings im Array ausschließlich aus Groß- oder Kleinbuchstaben bestehen. Dieser Wert wird als Ergebnis zurückgegeben.

Beispiele:

`strings = [HTW. saarland, Saarbruecken]` `ergebnis = 2`

`strings = [HTW. saarland, Saarbruecken, MERZIG, UNIversitaet]` `ergebnis = 3`

## Aufgabe 4 - Testen

Testen Sie die Aufgaben 2 und 3 mithilfe von JUnit.