

## 4.1 Vorstellung der Aufgaben der Praktika

In den Praktika zur Veranstaltung Objektorientierte Programmiertechniken / SA 2 führen Sie Schritte zur Erstellung eines Softwareprodukts unter Berücksichtigung von Aspekten der Objektmodellierung durch. Diese sind unter anderem deshalb notwendig, um die Wartbarkeit und Erweiterbarkeit eines umfangreichen Softwareprodukts zu gewährleisten.

Heute behandeln Sie Erweiterungen der Programmiersprache Java aus Java 5 und Java 8 und Erweiterungen des Testframeworks JUnit aus JUnit5.

## 4.2 Erweiterungen aus Java 5: Generics und for each

Grundlage ist das Ergebnis des letzten Praktikumstermins. Mit Hilfe von Subversion erhalten Sie den aktuellen Stand Ihrer Anwendung.

Ändern Sie die Anwendung aus dem dritten Praktikumstermin ab. Es sollen jetzt mehrere Freizeitbäder zwischengespeichert werden können.

Tauschen Sie daher das Attribut *freizeitbad* vom Typ *Freizeitbad* aus gegen ein Attribut *freizeitbaeder* vom Typ *ArrayList<Freizeitbad>*. Dieses kann sofort mit *ArrayList<>()* initialisiert werden.

Die get-Methode muss angepasst werden.

Die set-Methode soll ausgetauscht werden gegen eine Methode *public void addFreizeitbad(Freizeitbad freizeitbad)*. Es soll zu den bereits vorhandenen Freizeitbädern ein weiteres hinzugefügt werden.

Es sollen in den Views alle im ArrayList vorhandenen Freizeitbäder angezeigt werden. Benutzen Sie dazu eine for each – Schleife.

```
public void zeigeFreizeitbaederAn(){
    if(freizeitbaederModel.getFreizeitbaeder().size() > 0){
        StringBuffer text = new StringBuffer();
        // Ergaenzen: for each - Schleife ueber ArrayList

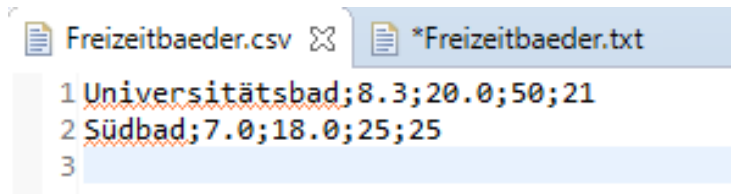
        {
            text.append(fzb.gibFreizeitbadZurueck(' ') + "\n");
        }
        this.txtAnzeige.setText(text.toString());
    }
    else{
        zeigeInformationsfensterAn(
            "Bisher wurde kein Freizeitbad aufgenommen!");
    }
}
```

Anstatt ein Freizeitbad in die csv-Datei bzw. txt-Datei zu schreiben, sollen die entsprechenden Methoden alle Freizeitbäder des ArrayList in die csv-Datei bzw. txt-Datei schreiben. Benutzen Sie dazu eine for each – Schleife.

```

1 Daten des Freizeitbades
2 Name des Freizeitbads: Universitätsbad
3 Öffnungszeit des Freizeitbads: 8.3 - 20.0
4 Beckenlänge des Freizeitbads: 50
5 Wassertemperatur des Freizeitbads: 21
6
7 Daten des Freizeitbades
8 Name des Freizeitbads: Südbad
9 Öffnungszeit des Freizeitbads: 7.0 - 18.0
10 Beckenlänge des Freizeitbads: 25
11 Wassertemperatur des Freizeitbads: 25

```

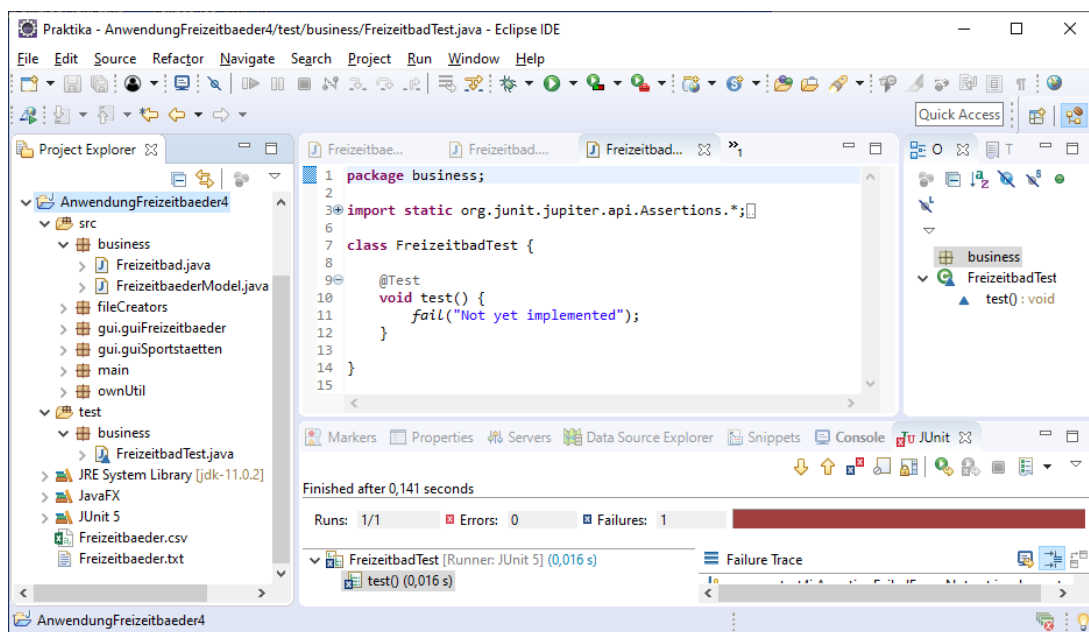


### 4.3 Erweiterungen aus Java 8: Lambda - Ausdruck

Benutzen Sie für die Listener, die den Menüpunkten *csv-Export* und *txt-Export* des *FreizeitbaederView* hinzugefügt werden, Lambda – Ausdrücke.

### 4.4 Test mit JUnit 5

Erstellen Sie ein neues source folder *test* (, im Kontextmenü des Projekts wählen Sie dazu *New/SourceFolder*, ) auf derselben Ordner Ebene wie das source folder *src*. Legen Sie in dem neuen source folder ein package *business* an und in diesem eine JUnit Testklasse *FreizeitbadTest* (). (Im Kontextmenü des package *business* im source folders *test* wählen Sie dazu *New/Other.../Java/JUnit/JUnit Test Case*.)



Wenn Sie den Test ausführen, wird dieser noch nicht erfolgreich sein, da er noch nicht implementiert ist. (Im Kontextmenü der Klasse *FreizeitbadTest* wählen Sie zum Ausführen des Tests *Run As/JUnit Test*.)

Erstellen Sie einen Test, welcher den Konstruktor der Klasse *Freizeitbad* mit dem folgenden Testfall testet. Der Name sei *Stadtbad*, die Öffnungszeit von 7.00 bis 17.00 Uhr, die Beckenlänge 25 Meter und die Temperatur 24 Grad Celsius. Benutzen Sie *assertTrue* und einen *booleanSupplier*, welcher überprüft, ob die Beckenlänge im Konstruktor auf 25 gesetzt wurde. Der Test sollte erfolgreich durchlaufen.