Advanced_Backend / Lesen 039 / code / hw_na_les038_1 / test / MainTest.java

Evgen959 newLes39 82fdf81 · 8 minutes ago

124 lines (104 loc) · 4.17 KB

Code Blame Raw

```java
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.Test;

import java.util.HashSet;
import java.util.List;

import static org.junit.jupiter.api.Assertions.*;

class MainTest {

    @Test
    void getAllStudents() {
        List<Student> list1 = List.of(
            new Student("Jan", "Jackobson"),
            new Student("Peter", "Peterson"),
            new Student("Mark", "Karlson"),
            new Student("Maria", "Simpson"),
            new Student("Poul", "Worker")
        );

        List<Student> list2 = List.of(
            new Student("Garry", "Potter"),
            new Student("Peter", "Peterson"),
            new Student("John", "Karlson"),
            new Student("Maria", "Simpson"),
            new Student("Mike", "Tayson"),
            new Student("Lena", "Smith")
        );

        List<Student> list3 = List.of(
            new Student("Maria", "Simpson"),
            new Student("Mike", "Tayson"),
            new Student("Lena", "Smith")
        );
```

```
36
37              List<Student> expectedList = List.of(
38                      new Student("Jan", "Jackobson"),
39                      new Student("Peter", "Peterson"),
40                      new Student("Mark", "Karlson"),
41                      new Student("Maria", "Simpson"),
42                      new Student("Poul", "Worker"),
43                      new Student("Garry", "Potter"),
44                      new Student("John", "Karlson"),
45                      new Student("Mike", "Tayson"),
46                      new Student("Lena", "Smith")
47              );
48
49              List<Student> actualResult = Main.getAllStudents(list1, list2, list3);
50
51              Assertions.assertEquals(expectedList.size(), actualResult.size());
52              Assertions.assertEquals(new HashSet<>(expectedList), new HashSet<>(actualResult)
53          }
54
55          @Test
56          @DisplayName("getAllCoursesStudents: regular")
57          void getAllCoursesStudents() {
58              List<Student> back = List.of(
59                      new Student("Jan", "Jackobson"),
60                      new Student("Peter", "Peterson"),
61                      new Student("Mike", "Tayson"),
62                      new Student("Mark", "Karlson"),
63                      new Student("Maria", "Simpson"),
64                      new Student("Poul", "Worker")
65              );
66
67              List<Student> front = List.of(
68                      new Student("Garry", "Potter"),
69                      new Student("Peter", "Peterson"),
70                      new Student("John", "Karlson"),
71                      new Student("Maria", "Simpson"),
72                      new Student("Mike", "Tayson"),
73                      new Student("Lena", "Smith")
74              );
75
76              List<Student> qa = List.of(
77                      new Student("Sasha", "Gromova"),
78                      new Student("Mike", "Tayson"),
79                      new Student("Peter", "Peterson"),
80                      new Student("Lena", "Smith")
81              );
82
83              List<Student> expectedList = List.of(
84                      new Student("Mike", "Tayson"),
85                      new Student("Peter", "Peterson")
86              );
87
```

```
88              List<Student> actualResult = Main.getAllCoursesStudents(back, front, qa);
89
90              Assertions.assertEquals(expectedList.size(), actualResult.size());
91              Assertions.assertEquals(new HashSet<>(expectedList), new HashSet<>(actualResult)
92          }
93
94          @Test
95          @DisplayName("getAllCoursesStudents: regular")
96          void getAllCoursesStudents1() {
97              List<Student> back = List.of(
98                      new Student("Jan", "Jackobson"),
99                      new Student("Peter", "Peterson"),
100                     new Student("Mike", "Tayson"),
101                     new Student("Mark", "Karlson"),
102                     new Student("Maria", "Simpson"),
103                     new Student("Poul", "Worker")
104             );
105
106             List<Student> front = List.of(
107                     new Student("Garry", "Potter"),
108                     new Student("Peter", "Peterson"),
109                     new Student("John", "Karlson"),
110                     new Student("Maria", "Simpson"),
111                     new Student("Lena", "Smith")
112             );
113
114             List<Student> qa = List.of(
115                     new Student("Sasha", "Gromova"),
116                     new Student("Mike", "Tayson"),
117                     new Student("Lena", "Smith")
118             );
119
120             List<Student> actualResult = Main.getAllCoursesStudents(back, front, qa);
121
122             Assertions.assertTrue(actualResult.isEmpty());
123         }
124     }
```