

 Evgen959 / Advanced\_Backend Public[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)[Advanced\\_Backend](#) / [Lesen 032](#) / [code](#) / [hw\\_na\\_les031\\_2](#) / [src](#) / [Student.java](#) 

...



Evgen959 newHWTestMainLes32

43acba2 · 9 minutes ago



83 lines (69 loc) · 2.51 KB

[Code](#) [Blame](#)[Raw](#)

```
1  import java.util.ArrayList;
2  import java.util.List;
3  import java.util.Objects;
4
5  public class Student {
6      private static final int MAX_QUANTITY_GRADES = 10;
7      private static final int MAX_GRADE = 10;
8      private static final int MIN_GRADE = 1;
9      private String name;
10     private List<Integer> grades = new ArrayList<>();
11     private int sumOfGrades = 0;
12
13     public Student(String name) {
14         this.name = name;
15     }
16
17     public Student(String name, int ...grade) {
18         this.name = name;
19         addGrade(grade);
20     }
21
22     public void addGrade(int grade){
23         if (grades.size() < MAX_QUANTITY_GRADES && grade >= MIN_GRADE && grade <= MAX_GRADE){
24             grades.add(grade);
25             sumOfGrades = sumOfGrades + grade;
26         }
27     }
28     // 1 test добавляется ли в лист значение
29     // 2 test при добавлении одного значения меняется корректно сумма
30     // 3 test добавляются ли последующие значения оценок
31     // 4 test корректно ли сумма при добавлении значений оценок
32     // 5 test не добавляются не корректные оценки (<0 > 10 )
33     // 6 test больше 10 оценок нельзя добавить
34
35     public void addGrade(int ...grade){
36         for (int i = 0; i < grade.length; i++) {
37             addGrade(grade[i]);
38         }
39     }
40     // 1 test список корректно добавляется
41     // 2 test
42
43     public Integer getShortfall(){
44         return MAX_QUANTITY_GRADES * MAX_GRADE - sumOfGrades;
45     }
46
47
48
49     @Override
50     public String toString() {
51         return "Student{" +
52             "name='" + name + '\'' +
53             ", grades=" + grades +
54             ", sumOfGrades=" + sumOfGrades +
55             ", getShortfall=" + getShortfall() +
56             '}';
```

```
57     }
58
59     public String getName() {
60         return name;
61     }
62
63     public List<Integer> getGrades() {
64         return new ArrayList<>(grades);
65     }
66
67     public int getSumOfGrades() {
68         return sumOfGrades;
69     }
70
71     @Override
72     ✓ public boolean equals(Object o) {
73         if (this == o) return true;
74         if (o == null || getClass() != o.getClass()) return false;
75         Student student = (Student) o;
76         return sumOfGrades == student.sumOfGrades && Objects.equals(name, student.name) && Objects.equals(grades, student.grades);
77     }
78
79     @Override
80     public int hashCode() {
81         return Objects.hash(name, grades, sumOfGrades);
82     }
83 }
```