

B Binary Search




Project ID: 5802

Forked from an inaccessible project.



Added text to remember to use intellijs help.

Mads Würtz Pedersen authored 4 weeks ago

Name	Last commit	Last update
 src/main/java	Exercise	1 year ago
 README.md	Added text to remember to use intellijs help.	4 weeks ago
 pom.xml	Updated JUnit version to the version used in...	4 weeks ago

[README.md](#)

Instructor Hours:

You are handed over a Java project, comprising of 2 packages, `employee_details` & `binary_search`

For the thrill seekers:

Creations of the methoeds for tests for classes can be quite easily be generated by IntelliJ. If you feel like it, exploring what intellij can help you with/do for you will help alot in the long run.

Task a

Package `employee_details` contains 2 classes, i.e., `Employee` and `EmployeeBusinessLogic`.

- Create a Test Class, `EmployeeTest` (should be placed in `test->java->employee_details` package)
- Write JUnit tests for the following methods in the test class:
 - `testCalculateAppraisal()` // Test to check appraisal
 - `testCalculateYearlySalary()` // Test to check yearly salary
- Use this Employee as test input:
`Employee employee=new Employee("John", 8000, 25);`

Task b

Package `binary_search` contains 2 classes for `BinarySearch` in both a recursive- and an iterative- version.

- Create 2 Test Classes, one for IterativeBinSearch `IterativeBinSearchTest` and the other for RecursiveBinSearch `RecursiveBinSearchTest` (should be placed in `test->java->binary_search` package)
- Write JUnit tests for the following methods in both test classes:
 - `shouldFindIndexOfNumber()` // Find present number
 - `shouldReturnNegativeInsertionPointWhenNotFound()` // Find not-present number
- Use this Array as test input:
`private static final int[] FIBOS = {1, 1, 2, 3, 5, 8, 13, 21, 34, 55};`

Task c

Create a package `JUnitTestSuite` under `test->java`, and create a class `TestSuite`, to run all the test methods in all the test classes placed in the `employee_details` and `binary_search` packages

Hint: Add the required annotations to run the Test Suite


```
1 package employee_details;
2
3 import org.junit.jupiter.api.AfterEach;
4 import org.junit.jupiter.api.BeforeEach;
5 import org.junit.jupiter.api.Test;
6
7 import static org.junit.jupiter.api.Assertions.*;
8
9 //Vi har løst den første Opgave A her.
10 class EmployeeTest {
11     Employee employee;
12
13     @BeforeEach
14     void setUp() {
15         employee = new Employee("John", 8000000, 25);
16     }
17
18     @AfterEach
19     void tearDown() {
20     }
21
22     @Test
23     void getName() {
24     }
25
26     @Test
27     void setName() {
28     }
29
30     @Test
31     void getMonthlySalary() {
32     }
33
34     @Test
35     void setMonthlySalary() {
36     }
37
38     @Test
39     void getAge() {
40     }
41
```

```
42     @Test
43     void setAge() {
44     }
45
46     @Test
47     void testCalculatorAppraisal(){
48         assertEquals(1000,EmployeeBusinessLogic.
calculateAppraisal(employee));
49
50     }
51
52     @Test
53     void testCalculateYearlySalary(){
54
55     }
56
57 }
```

```

1 package binary_search;
2
3 import org.junit.jupiter.api.BeforeEach;
4 import org.junit.jupiter.api.Test;
5
6 import static org.junit.jupiter.api.Assertions.*;
7
8 //Den første del af Opgave B er løst her.
9 class IterativeBinSearchTest {
10     private static final int[] FIBOS = {1, 1, 2, 3, 5
11     , 8, 13, 21, 34, 55};
12     IterativeBinSearch iterativeBinSearch;
13
14     @BeforeEach
15     void setUp() {
16         iterativeBinSearch = new IterativeBinSearch
17         ();
18     }
19     @Test
20     void shouldFindIndexOfNumber(){
21         assertEquals(3,iterativeBinSearch.find(FIBOS,
22         3));
23     }
24
25     @Test
26     void
27     shouldReturnNegativeInsertionPointWhenNotFound(){
28         assertTrue(iterativeBinSearch.find(FIBOS,28
29         ) <= -1);
30     }
31 }

```

```
1 package binary_search;
2
3 import org.junit.jupiter.api.BeforeEach;
4 import org.junit.jupiter.api.DisplayName;
5 import org.junit.jupiter.api.Test;
6
7 import static org.junit.jupiter.api.Assertions.*;
8
9 //Den anden del af Opgave B er løst her.
10 class RecursiveBinSearchTest {
11     private static final int[] FIBOS = {1, 1, 2, 3, 5
12     , 8, 13, 21, 34, 55};
13     RecursiveBinSearch recursiveBinSearch;
14
15     @BeforeEach
16     void setUp(){
17         recursiveBinSearch = new RecursiveBinSearch
18         ();
19     }
20
21     @Test
22     void RecursiveFindTest(){
23         assertEquals(3, recursiveBinSearch.
24         recursiveFind(FIBOS, 3));
25     }
26
27     @Test
28     void shouldFindIndexOfNumber(){
29
30     }
31
32     @Test
33     void
34     shouldReturnNegativeInsertionPointWhenNotFound(){
35
36     }
37 }
```

```
1 package JUnitTestSuite;
2
3 import org.junit.platform.suite.api.SelectPackages;
4 import org.junit.platform.suite.api.Suite;
5
6 @Suite
7 @SelectPackages({"employee", "binary_search"})
8 public class TestSuite {
9     //Her er Opgave C løst.
10    //OBS: Det skal understreges, at man under
       opgaven skal bruge Generator til at danne test-mapper
       under JUnit.
11 }
12
```