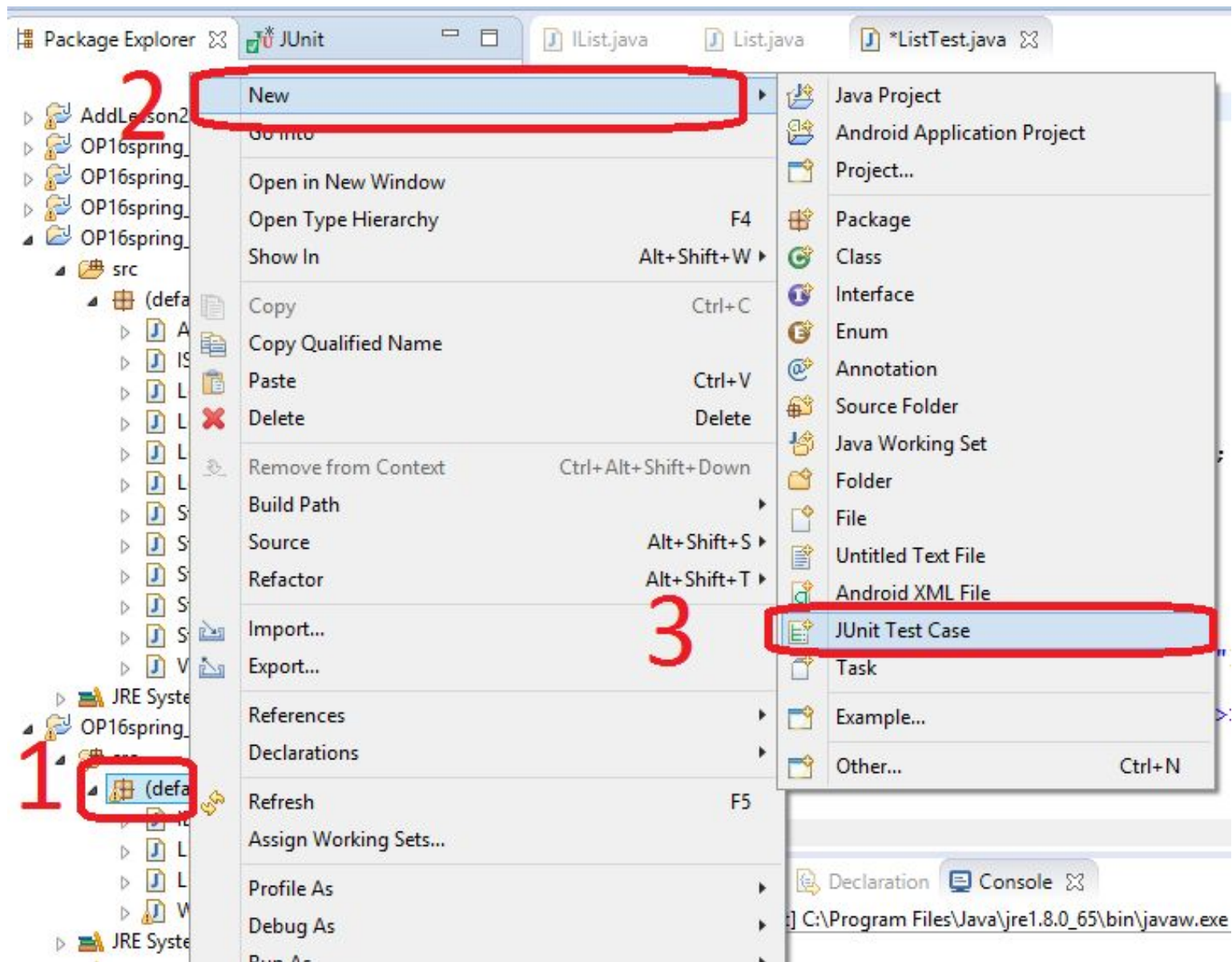


Основы программирования - Java ФИСТ 1 курс

Власенко Олег Федосович

**Лекция 7
Знакомство с unit тестами.
Списки – практика.**

Unit тесты



Создание test case в JUnit

JUnit Test Case

The use of the default package is discouraged.

New JUnit 3 test New JUnit 4 test

Source folder:

Package:

Name:

Superclass:

Which method stubs would you like to create?

setUpBeforeClass() tearDownAfterClass()
 setUp() tearDown()
 constructor

Do you want to add comments? (Configure templates and default value [here](#))

Generate comments

Class under test:

Создан test case в JUnit

```
import static org.junit.Assert.*;
import org.junit.Test;

public class Test1 {

    @Test
    public void test() {
        fail("Not yet implemented");
    }

}
```

Test case для List (1)

```
import static org.junit.Assert.*;
import org.junit.Test;

public class ListTest {

    // @Test
    // public void test() {
    //     fail("Not yet implemented");
    // }

    @Test
    public void testToString() {
        List list = new List();
        assertEquals("<<>>", list.toString());
    }
}
```

Test case для List (2)

@Test

public void testInsertToHead() {

List list = **new List();**

list.insertToHead(1);

assertEquals("<<1 >>", list.toString());

list.insertToHead(2);

assertEquals("<<2 1 >>", list.toString());

}

Test case для List (3)

@Test

```
public void testDeleteFromHead() {  
    List list = new List();  
    list.insertToHead(1);  
    list.insertToHead(2);  
    assertEquals("<<2 1 >>", list.toString() );  
    list.deleteFromHead();  
    assertEquals("<<1 >>", list.toString());  
}
```

Test case для List (4)

@Test

```
public void testGetHeadElement() {  
    List list = new List();  
    list.insertToHead(1);  
    list.insertToHead(2);  
    assertEquals("<<2 1 >>", list.toString());  
  
    assertEquals(2, list.getHeadElement());  
}
```


Test case для List (5)

@Test

public void testContains() {

List list = new List();

list.insertToHead(1);

list.insertToHead(2);

assertEquals("<<2 1 >>", list.toString());

assertEquals(true, list.contains(1));

assertEquals(true, list.contains(2));

assertEquals(false, list.contains(3));

}

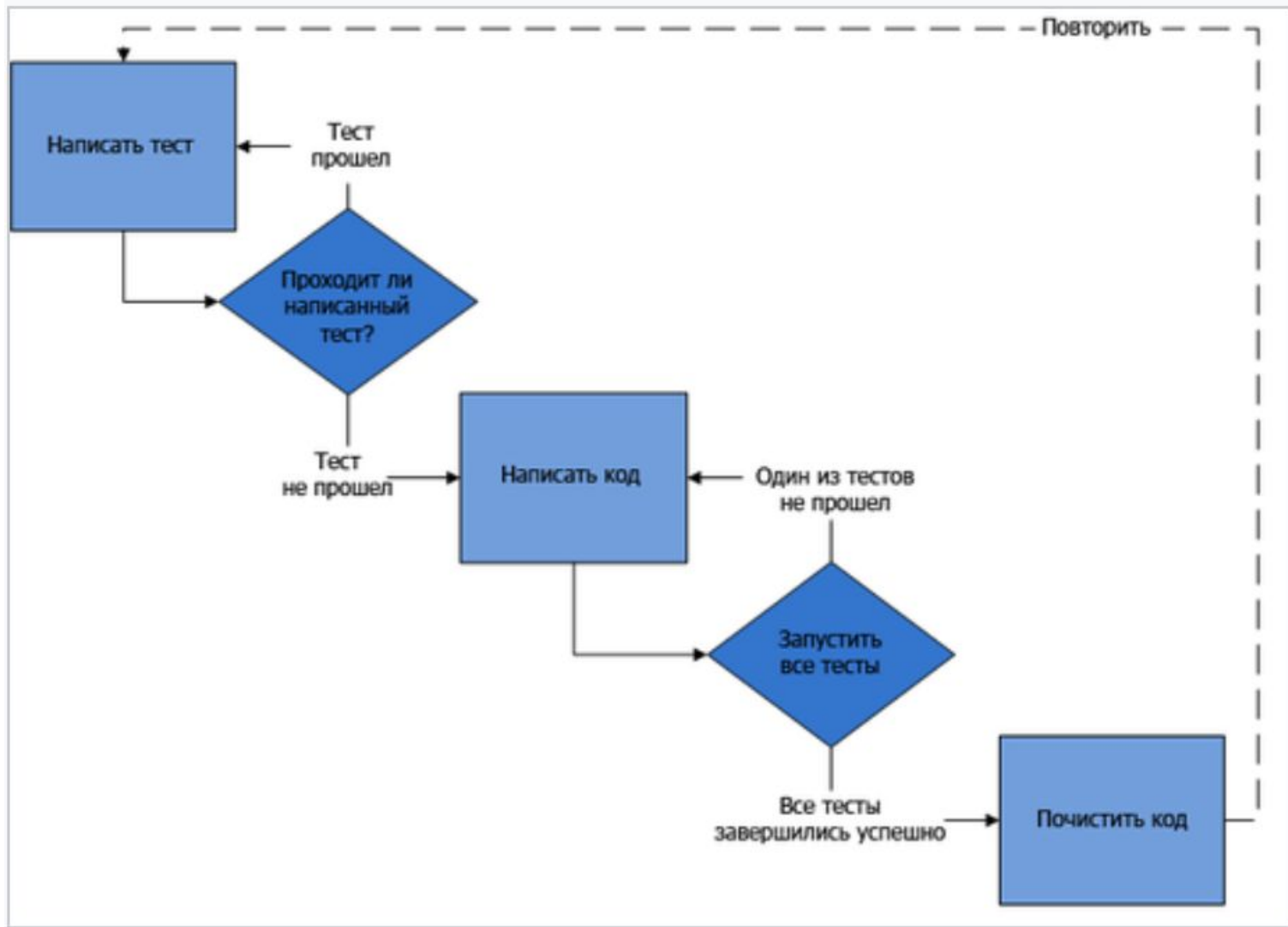
}

TDD - test-driven development

https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D1%87%D0%B5%D1%80%D0%B5%D0%B7_%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5

Разработка через тестирование (англ. *test-driven development*, *TDD*) — техника разработки программного обеспечения, которая основывается на повторении очень коротких циклов разработки: сначала пишется тест, покрывающий желаемое изменение, затем пишется код, который позволит пройти тест, и под конец проводится рефакторинг нового кода к соответствующим стандартам.

TDD - test-driven development



Ну а теперь задачи

- 1) Удалить последний элемент в списке
- 2) Найти максимальный элемент в списке и увеличить его в 100 раз
- 3) Подсчитать сумму всех четных элементов списка
- 4) Переставить первый и последний элемент списка

Ну а теперь задачи

- 5) Переставить местами минимальный и максимальный элементы списка (не стоящие рядом)
- 6) Переставить местами минимальный и максимальный элементы списка (стоящие рядом)
- 7) Переставить местами 2 любых элемента списка, заданные ссылками (первые и не первые, рядом и не рядом, порядок - произвольный)
- 8) Отсортировать (по возрастанию) элементы списка. Значения не изменять, можно менять только ссылки.

Спасибо за внимание!

Власенко Олег Федосович

E-mail: vlasenko.oleg@gmail.com

Vk: vk.com/oleg.f.vlasenko

Телефон: +7 902 246 05 47