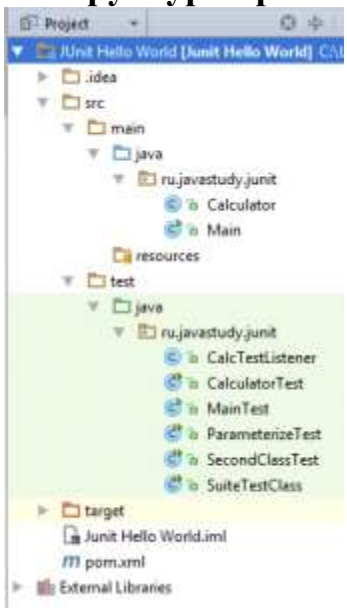
Пример 1 Тестирование Калькулятора

! Используемые технологии и библиотеки JUnit 4.12, для других – подключение библиотеки и методы могут отличаться

1. Описание задачи

Создать юнит тесты для приложения «калькулятор». Показать применение базовых аннотаций JUnit,

2. Структура проекта



Класс Calculator описывает простые арифметические операции. Этот класс послужит основой для написания юнит тестов. Тестирующие классы находятся в пакете test.

3. Проверьте файл проекта pom.xml

!Для подключения библиотеки JUnit используется maven.

```
1                   <?xml version="1.0" encoding="UTF-8"?>
2                   <project xmlns="http://maven.apache.org/POM/4.0.0"
3                            xmlns:xsi="http://www.w3.org/2001/XMLSchema-
4               instance"
5                            xsi:schemaLocation="http://maven.apache.org/PO
6               M/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
7                       <modelVersion>4.0.0</modelVersion>
8
9                       <groupId>ru.javastudy</groupId>
10                      <artifactId>junitStudy</artifactId>
11                      <version>1.0-SNAPSHOT</version>
12
13
14                      <dependencies>
15
16                          <dependency>
17                              <groupId>junit</groupId>
18                              <artifactId>junit</artifactId>
```

```
19                      <version>4.12</version>
20                      <scope>test</scope>
21                  </dependency>
22                  <dependency>
23                      <groupId>junit</groupId>
24                      <artifactId>junit</artifactId>
25                      <version>4.12</version>
26                  </dependency>
27
28              </dependencies>
29


          </project>
```

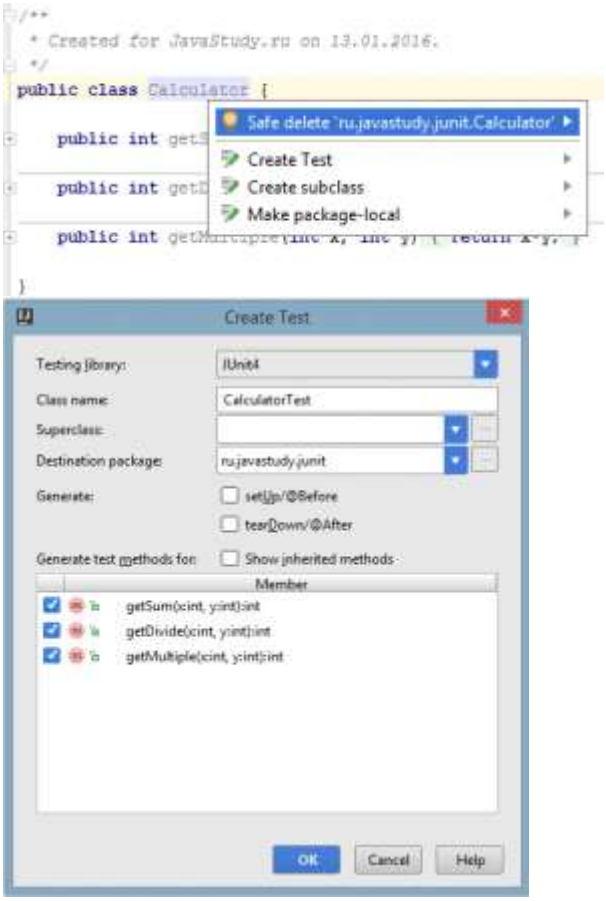
Других зависимостей подключать не нужно.

4. Calculator

```
1                   package ru.javastudy.junit;
2
3                   /**
4                    * Created for JavaStudy.ru on 13.01.2016.
5                    */
6                   public class Calculator {
7
8                       public int getSum(int x, int y) {
9                           return x+y;
10                      }
11
12                      public int getDivide(int x, int y) {
13                          return x/y;
14                      }
15
16                      public int getMultiple(int x, int y) {
17                          return x*y;
18                      }
19
20                  }
```

5. Создание тестирующих классов

В IntelliJ IDEA можно создать тестирующих класс автоматически. Для этого можно нажать alt + enter на классе и выбрать «Create test». Далее выбрать методы, которые нужно будет протестировать. В результате будет создан класс CalculatorTest с тремя выбранными методами. Эти методы необходимо реализовать самостоятельно.



6. CalculatorTest

После создания тестирующего класса нам необходимо реализовать методы, которые мы хотим проверить. Так же были добавлены другие методы, которые будут демонстрировать работу базовых JUnit аннотаций.

```
1                   package ru.javastudy.junit;
2
3                   import org.junit.*;
4
5                   import static org.junit.Assert.*;
6
7                   /**
8                    * Created by retinka on 13.01.2016.
9                    */
10                  public class CalculatorTest {
11
12                      private Calculator calculator;
13
14                      @BeforeClass
15                      public static void beforeClass() {
16                          System.out.println("Before CalculatorTest.class");
17                      }
18
19                      @AfterClass
20                      public  static void afterClass() {
```

```
21                     System.out.println("After CalculatorTest.class");
22                 }
23
24                 @Before
25                 public void initTest() {
26                     calculator = new Calculator();
27                 }
28
29                 @After
30                 public void afterTest() {
31                     calculator = null;
32                 }
33
34                 @Test
35                 public void testGetSum() throws Exception {
36                     assertEquals(15, calculator.getSum(7,8));
37                 }
38
39                 @Test
40                 public void testGetDivide() throws Exception {
41                     assertEquals(20, calculator.getDivide(100,5));
42                 }
43
44                 @Test
45                 public void testGetMultiple() throws Exception {
46
47                 }
48
49                 @Test(expected = ArithmeticException.class)
50                 public void divisionWithException() {
51                     calculator.getDivide(15,0);
52                 }
53
54                 @Ignore("Message for ignored test")
55                 @Test
56                 public void ignoredTest() {
57                     System.out.println("will not print it");
58                 }
59
60                 @Test(timeout = 500)
61                 public void timeStampTest() {
62                     while (true);
63                 }
64             }
```

В результате запуска тестов получим следующую картину: