

I. Введение

4. Unit тестирование

Unit тестирование



Модульное тестирование (Unit тестирование) – процесс позволяющий проверить на корректность отдельные модули исходного кода программы.



Цель модульного тестирования – изолировать отдельные части программы и показать, что по отдельности эти части работоспособны. Этот тип тестирования обычно выполняется разработчиками. Для проведения модульного тестирования существуют инструменты и библиотеки модульного тестирования.



Как и любая технология тестирования, модульное тестирование не позволяет отловить все ошибки программы.

Библиотека JUnit

JUnit

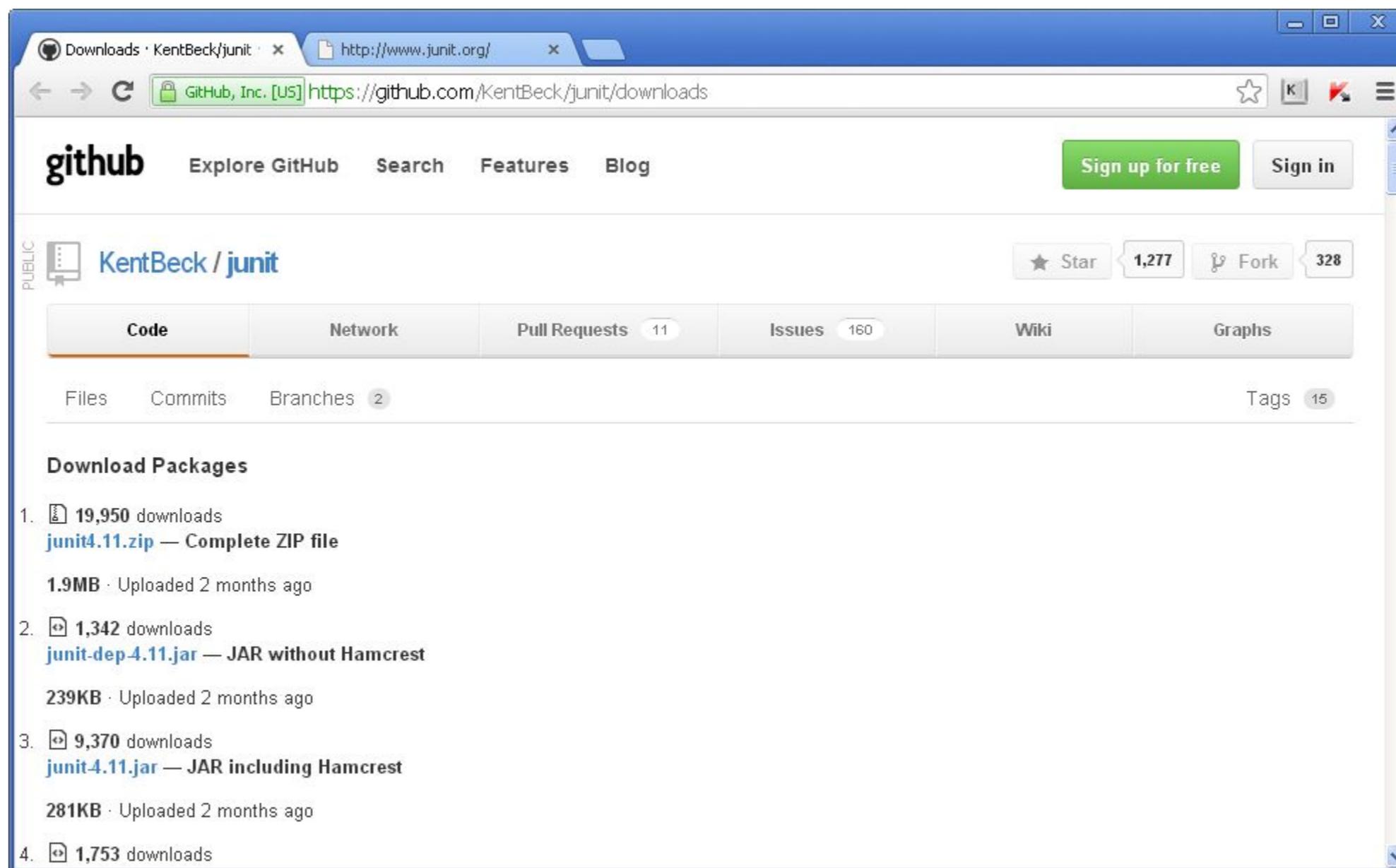


JUnit – библиотека для модульного тестирования программного обеспечения на языке Java.



При переходе от 3 к 4 версии в JUnit произошли большие изменения – были использованы новые возможности из Java 5. Тесты могут объявляться с помощью аннотаций.

Загрузка JUnit



Downloads · KentBeck/junit · x http://www.junit.org/ x

GitHub, Inc. [US] https://github.com/KentBeck/junit/downloads

github Explore GitHub Search Features Blog Sign up for free Sign in

PUBLIC KentBeck / junit Star 1,277 Fork 328

Code Network Pull Requests 11 Issues 160 Wiki Graphs

Files Commits Branches 2 Tags 15

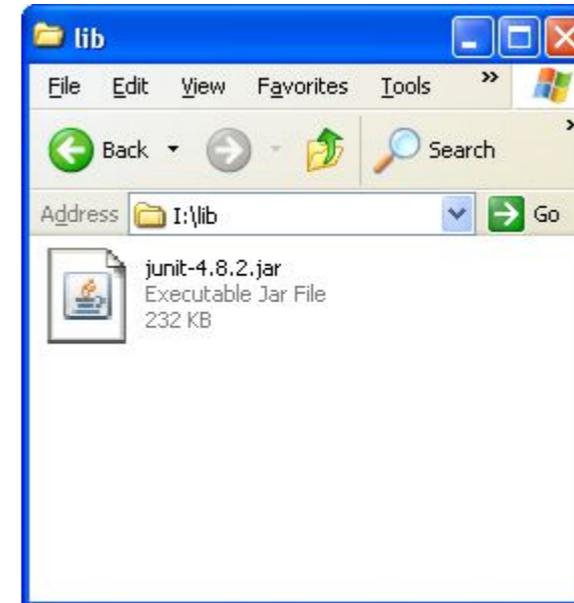
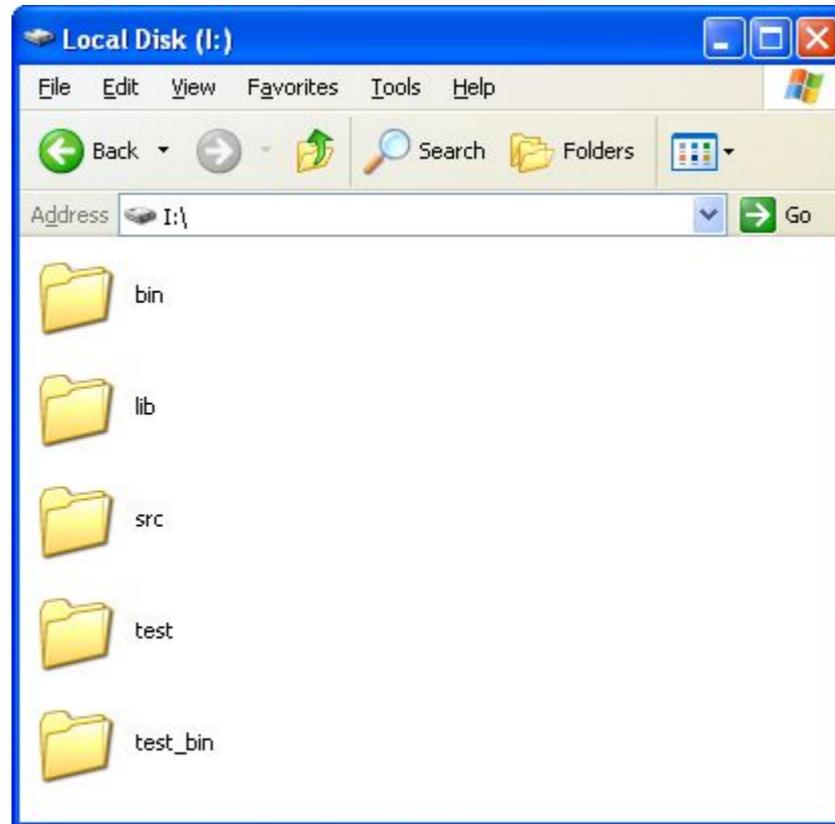
Download Packages

1. 19,950 downloads
[junit4.11.zip](#) — Complete ZIP file
1.9MB · Uploaded 2 months ago
2. 1,342 downloads
[junit-dep-4.11.jar](#) — JAR without Hamcrest
239KB · Uploaded 2 months ago
3. 9,370 downloads
[junit-4.11.jar](#) — JAR including Hamcrest
281KB · Uploaded 2 months ago
4. 1,753 downloads



Домашняя страница JUnit – www.junit.org на момент подготовки презентации не работала. JUnit версии 4.8.2 был загружен с сайта <https://github.com/KentBeck/junit/downloads>.

Структура директорий

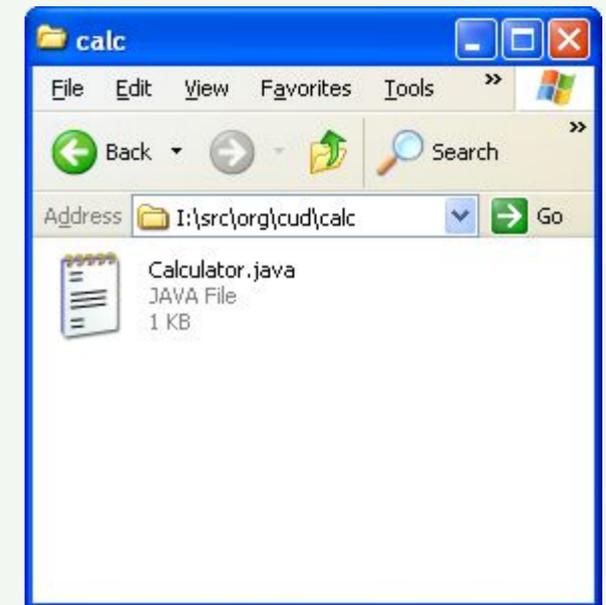


JUnit

Проверка	Описание
<code>fail(String)</code>	Метод не проходит тест. Может использоваться чтобы проверить что некоторая часть кода недостижима. Или чтобы получить падающий тест до того как тест реализован.
<code>assertTrue(true) / assertTrue(false)</code>	Всегда true / false. Может использоваться для того чтобы predetermined результат теста до того как тест реализован.
<code>assertTrue([message], boolean condition)</code>	Проверяет что булевское условие true.
<code>assertEquals([String message], expected, actual)</code>	Проверяет что два значения одинаковы. Для массивов проверяет ссылки одинаковы а не содержимое массивов.
<code>assertEquals([String message], expected, actual, tolerance)</code>	Проверяет что значения float или double совпадают с заданной точностью. tolerance число разрядов которые должны быть одинаковыми.
<code>assertNull([message], object)</code>	Проверяет что ссылка null.
<code>assertNotNull([message], object)</code>	Проверяет что ссылка не null.
<code>assertSame([String], expected, actual)</code>	Проверяет что две переменные ссылаются на один объект.
<code>assertNotSame([String], expected, actual)</code>	Проверяет что две переменные ссылаются на различные объекты.

Библиотека “Калькулятор!”

```
package org.cud.calc;  
  
public class Calculator {  
    public int sum(int... numbers) {  
        int total = 0;  
        for (int i : numbers) {  
            total += i;  
        }  
        return total;  
    }  
}
```



Компиляция библиотеки

```
I:\>dir src\org\cud\calc
Volume in drive I has no label.
Volume Serial Number is 8009-8B63

Directory of I:\src\org\cud\calc

01/13/2013  12:56 PM    <DIR>          .
01/13/2013  12:56 PM    <DIR>          ..
11/14/2012  05:00 PM                185 Calculator.java
               1 File(s)                185 bytes
               2 Dir(s)  237,193,134,080 bytes free

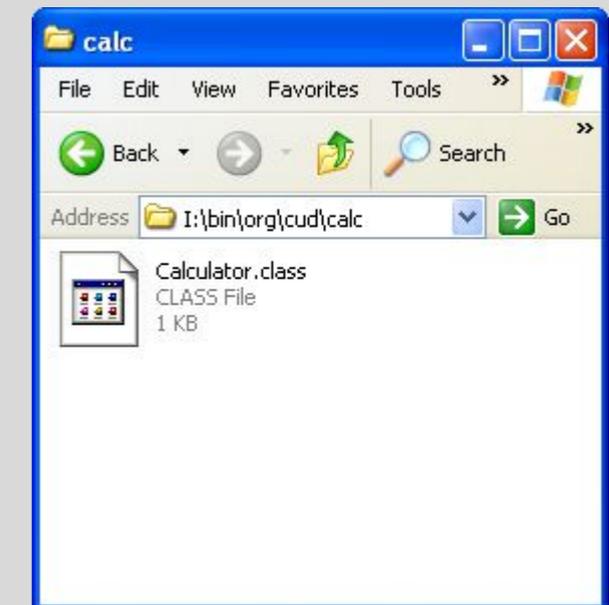
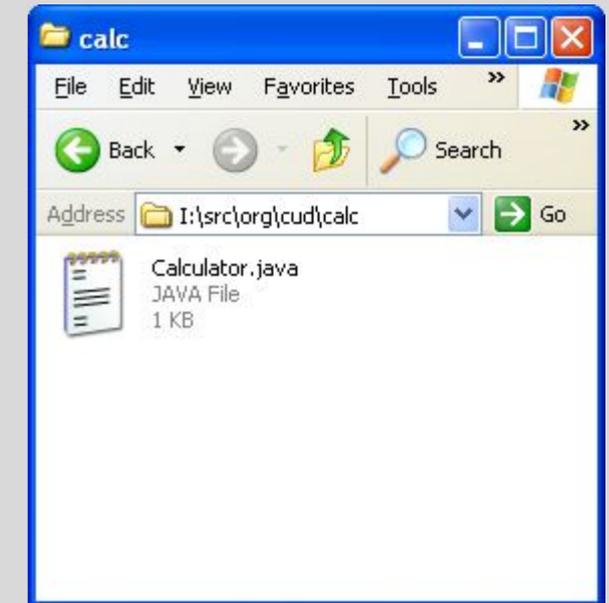
I:\>javac -d bin src\org\cud\calc\Calculator.java

I:\>dir bin\org\cud\calc
Volume in drive I has no label.
Volume Serial Number is 8009-8B63

Directory of I:\bin\org\cud\calc

01/13/2013  12:56 PM    <DIR>          .
01/13/2013  12:56 PM    <DIR>          ..
01/13/2013  05:40 PM                369 Calculator.class
               1 File(s)                369 bytes
               2 Dir(s)  237,193,134,080 bytes free

I:\>
```



Тест

```
package org.cud.calc;

import static org.junit.Assert.assertEquals;

import org.junit.Test;

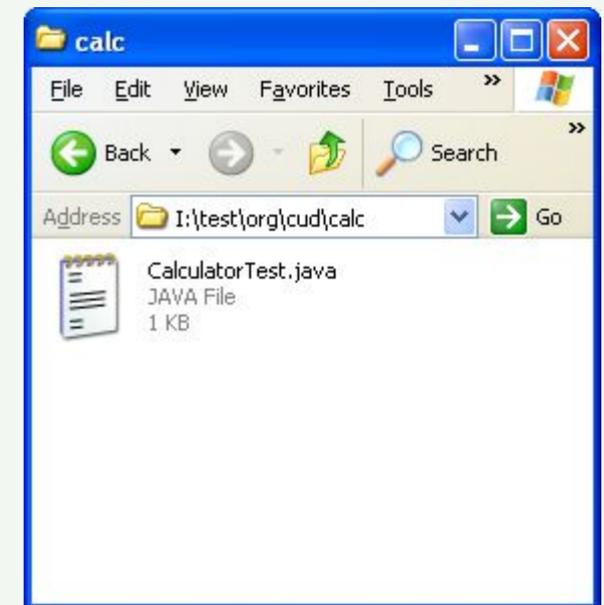
public class CalculatorTest {

    @Test
    public void test() {

        Calculator calculator = new Calculator();

        assertEquals(9, calculator.sum(2, 3, 4));

    }
}
```



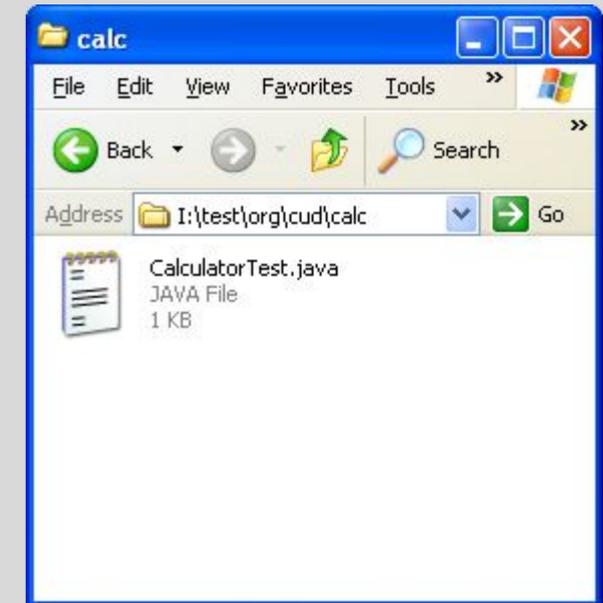
Используется статическое импортирование. В данном примере это позволяет вместо вызова `org.junit.Assert.assertEquals` использовать просто `assertEquals`.

Компиляция теста

```
I:\>dir test\org\cud\calc
Volume in drive I has no label.
Volume Serial Number is 8009-8B63

Directory of I:\test\org\cud\calc

01/13/2013  05:50 PM    <DIR>          .
01/13/2013  05:50 PM    <DIR>          ..
11/14/2012  05:21 PM                256 CalculatorTest.java
               1 File(s)                256 bytes
               2 Dir(s)  237,193,125,888 bytes free
```



```
I:\>javac -d test_bin -cp lib/junit-4.8.2.jar;bin test\org\cud\calc\CalculatorTest.java
```

```
I:\>dir test_bin\org\cud\calc
Volume in drive I has no label.
Volume Serial Number is 8009-8B63

Directory of I:\test_bin\org\cud\calc

01/13/2013  06:42 PM    <DIR>          .
01/13/2013  06:42 PM    <DIR>          ..
01/13/2013  06:42 PM                484 CalculatorTest.class
               1 File(s)                484 bytes
               2 Dir(s)  237,193,125,888 bytes free
```



```
I:\>
```

Запуск теста

```
I:\>java -cp lib/junit-4.8.2.jar;test_bin;bin org.junit.runner.JUnitCore org.cud.calc.CalculatorTest
JUnit version 4.8.2
.
Time: 0
OK (1 test)

I:\>
```

Параметризованный тест

```
package org.cud.calc;

import static org.junit.Assert.assertEquals;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized.Parameters;

import java.util.Arrays;
import java.util.Collection;

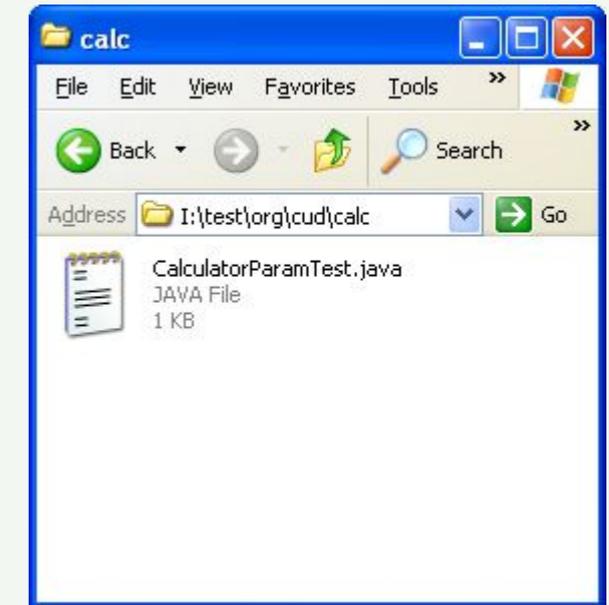
@RunWith(value = org.junit.runners.Parameterized.class)
public class CalculatorParamTest {
    int result;
    int[] numbers;

    @Parameters
    public static Collection<int[][]> parameters() {

        return Arrays.asList(new int[][][] { { { 2 }, { 1, 1 } },
            { { -2 }, { -1, -1 } }, { { 9 }, { 2, 3, 4 } }, { { 0 }, { } },
            { { 0 }, { 0, 0, 0, 0 } } });
    }

    public CalculatorParamTest(int[] result, int[] numbers) {
        this.result = result[0];
        this.numbers = numbers;
    }

    @Test
    public void testSum() {
        Calculator calculator = new Calculator();
        assertEquals(result, calculator.sum(numbers));
    }
}
```

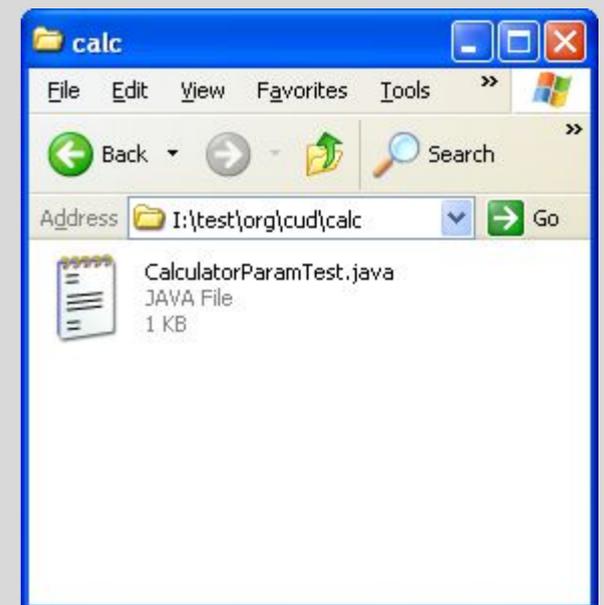


Компиляция параметризованного теста

```
I:\>dir test\org\cud\calc
Volume in drive I has no label.
Volume Serial Number is 8009-8B63

Directory of I:\test\org\cud\calc

01/13/2013  07:16 PM    <DIR>          .
01/13/2013  07:16 PM    <DIR>          ..
11/14/2012  05:32 PM                881 CalculatorParamTest.java
           1 File(s)                881 bytes
           2 Dir(s)  237,193,134,080 bytes free
```



```
I:\>javac -d test_bin -classpath lib/junit-4.8.2.jar;bin test\org\cud\calc\CalculatorParamTest.java
```

```
I:\>dir test_bin\org\cud\calc
Volume in drive I has no label.
Volume Serial Number is 8009-8B63

Directory of I:\test_bin\org\cud\calc

01/13/2013  07:46 PM    <DIR>          .
01/13/2013  07:46 PM    <DIR>          ..
01/13/2013  07:46 PM                1,247 CalculatorParamTest.class
           1 File(s)                1,247 bytes
           2 Dir(s)  237,193,113,600 bytes free
```



```
I:\>
```

Запуск параметризованного теста

```
I:\>java -cp lib/junit-4.8.2.jar;test_bin;bin org.junit.runner.JUnitCore org.cud.calc.CalculatorParamTest
JUnit version 4.8.2
.....
Time: 0.015

OK (5 tests)

I:\>
```