

### Object-Oriented Programming in the Java language Part 2. TDD and JUnit



Yevhen Berkunskyi, NUoS eugeny.berkunsky@gmail.com <u>http://www.berkut.mk.ua</u>

**ava** 



# History

- Kent Beck developed the first xUnit automated test tool for Smalltalk in mid-90's
- Beck and Gamma (of design patterns Gang of Four) developed JUnit on a flight from Zurich to Washington, D.C.
- Martin Fowler: "Never in the field of software development was so much owed by so many to so few lines of code."
- JUnit has become the standard tool for Test-Driven Development in Java (see junit.org)
- JUnit test generators now part of many Java IDEs (IntelliJ IDEA, NetBeans, Eclipse, BlueJ, ...)



## Why create a test suite?

- Obviously you have to test your code—right?
  - You can do *ad hoc* testing (running whatever tests occur to you at the moment), or
  - You can build a test suite (a thorough set of tests that can be run at any time)
- Disadvantages of a test suite
  - It's a lot of extra programming
    - True, but use of a good test framework can help quite a bit
  - You don't have time to do all that extra work
    - False! Experiments repeatedly show that test suites reduce debugging time more than the amount spent building the test suite
- Advantages of a test suite
  - Reduces total number of bugs in delivered code
  - Makes code much more maintainable and refactorable



### **Architectural overview**

- JUnit test framework is a package of classes that lets you write tests for each method, then easily run those tests
- TestRunner runs tests and reports TestResults
- You test your class by extending abstract class *TestCase* (optional)
- To write test cases, you need to know and understand the Assert class





## Writing a TestCase

- To start using JUnit, create a subclass of *TestCase*, (optional in JUnit 4 and 5) to which you add test methods
- Name of class is important should be of the form MyClass *Test*
- This naming convention lets TestRunner automatically find your test classes

```
import org.junit.jupiter.api.BeforeEach;
```

```
import static org.junit.jupiter.api.Assertions.*;
```

```
class MainTest {
    @BeforeEach
    void setUp() {
```



### Writing methods in TestCase

- Pattern follows *programming by contract* paradigm:
  - Set up preconditions
  - Exercise functionality being tested
  - Check postconditions

```
Example:
```

```
public void testEmptyList() {
    Bowl emptyBowl = new Bowl();
    assertEquals("Size of an empty list should be zero.",
        0, emptyList.size());
    assertTrue("An empty bowl should report empty.",
        emptyBowl.isEmpty());
}
```

Things to notice:

- Specific method signature public void *test*Whatever()
  - Coding follows pattern
    - Notice the assert-type calls...



### Assert methods

- Each assert method has parameters like these: *message, expected-value, actual-value*
- Assert methods dealing with floating point numbers get an additional argument, a tolerance
- Each assert method has an equivalent version that does not take a message – however, this use is not recommended because:
  - messages helps documents the tests
  - messages provide additional information when reading failure logs



### **Assert methods**

- assertTrue(String *message*, Boolean *test*)
- assertFalse(String *message*, Boolean *test*)
- assertNull(String message, Object object)
- assertNotNull(String *message*, Object *object*)
- assertEquals(String message, Object expected, Object actual)
   // uses equals method
- assertSame(String message, Object expected, Object actual)
   // uses == operator
- assertNotSame(String message, Object expected, Object actual)



### More stuff in test classes

- Suppose you want to test a class Counter
- public class CounterTest {
  - This is the unit test for the Counter class
- public CounterTest() { } //Default constructor
- protected void setUp()
  - Test *fixture* creates and initializes instance variables, etc.
- protected void tearDown()
  - Releases any system resources used by the test fixture
- public void testIncrement(), public void testDecrement()
  - These methods contain tests for the Counter methods increment(), decrement(), etc.
  - Note capitalization convention



## JUnit tests for Counter

```
public class CounterTest {
```

```
Counter counter1;
@BeforeEach
protected void setUp() { // creates a test fixture
    counter1 = new Counter();
@Test
public void testIncrement() {
    assertTrue(counter1.increment() == 1);
    assertTrue(counter1.increment() == 2);
@Test
public void testDecrement() {
    assertTrue(counter1.decrement() == -1);
```

Note that each test begins with a *brand new* counter

This means you don't have to worry about the order in which the tests are run





- TestSuites collect a selection of tests to run them as a unit
- Collections automatically use TestSuites, however to specify the order in which tests are run, write your own:

```
public static Test suite() {
    suite.addTest(new TestBowl("testBowl"));
    suite.addTest(new TestBowl("testAdding"));
    return suite;
}
```

- Should seldom have to write your own TestSuites as each method in your TestCase should be independent of all others
- Can create TestSuites that test a whole package:

```
public static Test suite() {
   TestSuite suite = new TestSuite();
   suite.addTestSuite(TestBowl.class);
   suite.addTestSuite(TestFruit.class);
   return suite;
```



### JUnit in IntelliJ IDEA

### At first you have to create a directory for your tests

Iab4new -	[D:\ldeaProjects\lab4new] - [l	lab4new]\src\la	b4\Main.java - IntelliJ IDEA 201	163.4 — 🗆 🗙	
<u>File E</u> dit <u>\</u>	New	•	Module	/ <u>H</u> elp	
lab4nev	Add Framework Support		File	🕴 🔚 Main 🚽 🕨 🗰 🔍 🖪	
Iab4nev	Add trainework Support Cut Copy Copy Path Copy Relative Path Copy Relative Path Paste Find Usages Find Usages Find in Path Replace in Path Analyze Refactor Add to Favorites Show Image Thumbnails Reformat Code Optimize Imports Remove Module	Ctrl+X Ctrl+C Ctrl+Shift+C Ctrl+Alt+Shift+C Ctrl+V Alt+F7 Ctrl+Shift+F Ctrl+Shift+F Ctrl+Shift+T Ctrl+Shift+T Ctrl+Alt+L Ctrl+Alt+O Delete	<ul> <li>The</li> <li>Directory</li> <li>HTML File</li> <li>Stylesheet</li> <li>JavaScript File</li> <li>TypeScript File</li> <li>CFML/CFC file</li> <li>CoffeeScript File</li> <li>XSLT Stylesheet</li> <li>Edit File Templates</li> <li>Resource Bundle</li> <li>XML Configuration File </li> <li>Diagram </li> <li>Data Source</li> </ul>	ain() > lab4; class Main { plic static void main(String[] args) { X ATBuild	
satio Ref Cii ★ IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Build Module Tab4new Rebuild Module Tab4new' Local History Synchronize Tab4new' Show in Explorer Directory Path Compare With Open Module Settings Move Module to Group	Ctrl+Shift+F9 Ctrl+Alt+F12 Ctrl+D F4		New Directory  Enter new directory name:  test  OK Cancel	×



### JUnit in IntelliJ IDEA

#### Then, mark it as Test Sources Root

Project Proje	lab4new	test >					📲 🦳 Main 💌 🕨 👾 🕷	
Image: Sec     Main main ()     Image: Sec	Project	*	⊕ ≑ \$	F* I← 💽 N	lain.java ×			
Copy as Plain Text   Copy Relative Path   Ctrl+Alt+Shift+C   Citrl-Alt+Shift+C   Find Usages   Alt   Find usages   Alt   Repiace in Path   Ctrl+Shift+R   Repiace in Path   Analyze   Befactor   Add to Favorites   Show Image Thumbhails   Ctrl+Shift+R   Analyze   Befactor   Add to Favorites   Show Index Thumbhails   Ctrl+Alt+O   Optimize Imports   Ctrl+Alt+O   Delete.   Build Module TabAnew'   Local Histor   Show in Explorer   Sources Root   Test Sources Root	■ lab4r ■ laid ■ sr te a la ■ la ■ la ■ sr	eew D:\ldeaProjects\lab4new dea c Mew	Ctrl+X Ctrl+C Ctrl+Shift+C	1 2 3 4 5 6 7	Main packa publi	<pre>main() ge lab4; c class Main { wublic static void main(String[] args) { </pre>		
Refactor   Add to Favorites   Show Image Thumbnails   Ctrl+Alt+T   Reformat Code   Ctrl+Alt+O   Optimize Imports   Ctrl+Alt+O   Delete   Build Module Tab4new'   Local History   Sources Root   Test Sources Root   Directory Path   Ctrl+Alt+F12   Resources Root   Test Resources Root   Studied		Copy as Plain Text Copy Relative Path ( Paste Find Usages Find in Path Replace in Path Analyze	Ctrl+Alt+Shift+C Ctrl+V Alt+F7 Ctrl+Shift+F Ctrl+Shift+R	8	3			
Reformat Code       Ctrl+Alt+L         Optimize Imports       Ctrl+Alt+O         Delete       Delete         Build Module 'lab4new'       Image: Compare With Ctrl+Alt+F12         Show in Explorer       Image: Ctrl+Alt+F12         Directory Path       Ctrl+Alt+F12         Compare With       Ctrl+Alt+F12         Mark Directory as       Image: Ctrl+Alt+F12		<u>R</u> efactor Add to F <u>a</u> vorites Show Image Thumbnails	Ctrl+Shift+T					
Local History Synchronize 'test' Show in Explorer Directory Path Ctrl+Alt+F12 Compare With Mark Directory as Ctrl+D Excluded		Reformat Code Optimize Imports Delete	Ctrl+Alt+L Ctrl+Alt+O Delete					
Show in Explorer     Test Sources Root       Directory Bath     Ctrl+Alt+F12       Compare With     Ctrl+D       Mark Directory as     Excluded		Local <u>H</u> istory	•	Sources Ro	oot			
Mark Directory as		Show in Explorer Directory Path	Ctrl+Alt+F12	Resources Test Resou	Root rces Root			
		Mark Directory as	Ctrl+D	Excluded	Sources Ro	ot		



### JUnit in IntelliJ IDEA

### In your class choose "Go To ► Test

lab4new > = src > = lab4 > C Main >					1:
Project ▼ ③ ≑ ♣ ▪ I+	C Main.java ×				
lab4new D:\ldeaProjects\lab4ne     lidea     src     test	Main 1 packa	age lab4;			
lab4new.iml ► IIIII External Libraries	3 public 4 5 Public 5 Public 7 Public 8 } 9	Copy Reference Paste Paste from History Paste Simple Column Selection Mode Find Usages Refactor Folding Analyze	Ctrl+Alt+Shift+C Ctrl+V Ctrl+Shift+V Ctrl+Alt+Shift+V Alt+Shift+Insert Alt+F7	) {	
		Go To Generate Recompile 'Main.java' Run 'Main' Cebug 'Main' Run 'Main' with Coverage Save 'Main' Local <u>H</u> istory	Alt+Insert Ctrl+Shift+F9 Ctrl+Shift+F10	Jump to Navigation Bar Declaration Implementation(s) Type Declaration Super Class Test	Alt+Home Ctrl+B Ctrl+Alt+B Ctrl+Shift+B Ctrl+U Ctrl+Shift+T
		Compare with Clip <u>b</u> oard File Encoding			

01



### JUnit in IntelliJ IDEA

🖥 lab4new 👌 🖿 src 👌 🛅 lab4 👌 💕 Main	>	🛄 Create Test	
Project  Project  Project  Project  Point Content of the second	Main         1       package lab4;         2       3         3       public class Main {         4       5         5       Choose Test for Main (0 found) ≯ in (state)         6       Create New Test         7       }         8       }         9	Testing library:       JUnit5         JUnit5 library not found in the module         Class name:       MainTest         Superclass:       Image: Class name:         Destination package:       lab4         Generate:       Image: StetUp/@Before         Image: Class name:       Image: StetUp/@Before         Image: Class name:       Image: StetUp/@Before         Image: StetUp/@Before       Image: StetUp/@Before <tr< th=""><th>Fix • • •</th></tr<>	Fix • • •
As test is not ex Create New Tes	st, choose . <b>t…</b> Fill data in the Dialo	Dg	



## JUnit in IntelliJ IDEA

	Iab4new - [D:\IdeaProjects\Iab4new] - [	[ab4new]\test\lab4\Mai	inTest.java - IntelliJ IDEA 2016.3.4
	<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode Analy <u>a</u>	<u>z</u> e <u>R</u> efactor <u>B</u> uild R <u>u</u> n	<u>T</u> ools VC <u>S</u> <u>W</u> indow <u>H</u> elp
	lab4new > intest > intest > C Mai	inTest >	
	to Project ▼ 😯 🚔 🏕 🗜	C Main.java × C Ma	inTest.java ×
	ab4new D:\IdeaProjects\Iab4ne	MainTest	setUp()
	src	1 package la	ab4;
	e 🕨 🖿 test	3 <b>+import</b>	1
	lab4new.iml	6	
	Similar External Libraries	7 8 class Mair	nTest {
	8	9 @Befor	reEach
		10 <b>void</b> s	setUp() {
		12 - }	
class MainTest {		13	
Main main;		{ P1	
(BeforeEach woid setUp() ( main = new Main(); )			
void scop() [ main - new Main(); ]			
QTest			
void testCalculateY() {			
double x = 2.3;			
double expected = 5.29;			
assertEquals (expected, result, delta: 0,001, mess	age: "y must be a x*x"):		
- }			
}			



### JUnit in IntelliJ IDEA

class MainTest {	
Main main;	
@BeforeEach	
<pre>void setUp() { main = new Main(); }</pre>	
@Test	
void testCalculateY() {	
double $x = 2.3;$	
double expected = 5.29;	
<pre>double result = main.calculateY(x);</pre>	
assertEquals (expected, result, delta: 0.001, message: "y	must be a x*x");
A lab4new - [D:\IdeaProjects\Iab4new] - [Iab4new]\src	\lab4\Main.java - IntelliJ IDEA 2016.3.4 —
File Edit View Navigate Code Applyze Defector P	uild Pun Tools VCS Window Help
}	
lab4new > src > lab4 > G Main >	🖣 MainTest.testCalculateY 🚽 🕨 🏾
v 🗊 Project 🔹 😳 ‡ 🕸 - I+	C Main.java × C MainTest.java ×
👔 🔻 🖿 lab4new D:\IdeaProjects\Iab4new	Main calculateY()
🗝 🕨 🖿 .idea	1 package lab4:
📱 🕨 🖿 out	2
😫 🕨 🖿 src	3 <b>b</b> public class Main {
털 🕨 🖿 test	4
र्दे 🔚 lab4new.iml	5 > O public static void main(String[] args) {
VI External Libraries	6
	7 🕒 }
L	10 return 0:
, ener	
	12 }
	13



### JUnit in IntelliJ IDEA

#### Run test of the generated method. It fails

IJ	🛛 lab4new - [D:\IdeaProjects\Iab4new] - [lab4new]\test\Iab4\MainTest.java - IntelliJ IDEA 2016.3.4 - 🗆 🗙							
<u>F</u> ile	e <u>E</u> di	lit <u>V</u> iew <u>N</u> avigate <u>C</u> ode Analy <u>z</u> e	e <u>R</u> efactor <u>B</u> uild R <u>u</u> n <u>T</u> ools VC <u>S</u> <u>W</u> indow <u>H</u> elp					
	lab4	4new 🔪 🖿 test 🔪 🖿 Iab4 🔪 💣 Main <sup>-</sup>	Test > 🕴 💽 MainTest.testCalculateY 🗸 🕨 🕷 🔲 👔 🔍					
sct	E P	Project 🔹 😴	Ð ≑ 🗱 C Main.java × C MainTest.java ×					
Proje	•	lab4new D:\ldeaProjects\lab4new	MainTest					
÷	•	idea						
		out						
e	•	src 🖿	20 🕼 🖯 void testCalculateY() {					
uctu	•	test	21 double x = 2.3;					
: Str		ab4new.iml	double expected = 5.29;					
	► II	External Libraries	23 double result = main.calculateY(x);					
			24 assertEquals(expected, result, delta: 0.001, message: "y must be a x*x"					
	Run	MainTest.testCalculateY	\$~ <u>↓</u>					
	$\blacktriangleright$ $\blacksquare$ $\downarrow_2^3$ $\downarrow_2^2$ $\downarrow_2^$							
	9	V U Test Results 41ms	фев 19, 2017 10:39:17 PM org.junit.platform.launcher.core.ServiceLoaderTestEngineRegist 🔥					
	MainTest 41ms INFO: Discovered TestEngines with IDs: [junit-jupiter]							
	U testCalculate 41ms							
es	org.opentest4j.AssertionFailedError: y must be a x*x ==> expected: <5.29> but was: <0.0							
vorit								
Ear	-17		<pre>Actual :0.0 </pre>					
*	764							
	**							
	4	4: Run 🔄 <u>6</u> : TODO 📧 Terminal	■ <u>U</u> : Messages Q Event Log					
	Defa	fault File template	9:35 CRLF\$ UTF-8\$ @ 费					



### JUnit in IntelliJ IDEA

#### Write correct method body. Run test of the generated method. It should be OK

īī	lab4new - [D:\ldeaProjects\lab4new] - [lab4new]\test\la	ab4\MainTest.java - IntelliJ IDEA 2016.3.4 —	×				
<u>F</u> ile	le <u>E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode Analy <u>z</u> e <u>R</u> efactor <u>B</u> uild	d R <u>u</u> n <u>T</u> ools VC <u>S</u> <u>W</u> indow <u>H</u> elp					
1	🖥 lab4new 👌 🖿 test 👌 🖿 lab4 👌 😅 MainTest 👌	👫 🕡 MainTest.testCalculateY 🚽 🕨 🗮 👔	Q				
ect	🖻 Project 💌 😲 😤 👫 🚺	C Main.java × C MainTest.java ×	*				
Proj	▼ ■ab4new D:\IdeaProjects\Iab4new	MainTest	Ant				
÷.	idea 1	.8	Build				
-	out 1	.9 @Test	-				
an	▶ src 2	vo 🗣 🖻 🛛 void testCalculateY() {	2				
truct	test 2	double $x = 2.3;$	Data				
Z: S	lab4new.iml	double expected = 5.29;	bas				
8	External Libraries	assertEquals(expected, result, delta; 0,001, message; "y must be a x*)	х II				
	2		m				
	2	6	May				
	2	7 }	S P				
			roje				
	Pup AninTecttectCalculateV	and a second	C s				
			-				
	$\blacktriangleright$ $\bigcirc$ 1 test passed – 31ms						
	Files\Java\jdk1.8.0_112\bin\java"						
👔 🔻 🖲 MainTest 31ms фев 19, 2017 10:45:09 РМ org.junit.platform.launcher.core.ServiceLoaderTestEngineRegi							
	Image: Second state sta						
rites							
Favo							
Ň							
×	>>>	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	>				
-	🕨 🛓 : Run 🛛 🔄 : TODO 🛛 🖾 Terminal 📕 0: Messages	C Event Log	J				
	Tests Passed: 1 passed (moments ago)	6:1 CRLF\$ UTF-8\$ 🚡	⇔				



### **More Information**

- http://www.junit.org
  - Download of JUnit
  - Lots of information on using JUnit
- <u>http://sourceforge.net/projects/cppunit</u>
   C++ port of Junit
- <u>http://www.thecoadletter.com</u>
  - Information on Test-Driven Development















### Object-Oriented Programming in the Java language Part 2. TDD and JUnit



Yevhen Berkunskyi, NUoS eugeny.berkunsky@gmail.com <u>http://www.berkut.mk.ua</u>

**ava**