URL to GitHup Repository: https://github.com/aliustunyer/Unit-Tests.git

URL to Youtube Video : https://youtu.be/JfXblXPc02o

```java
import java.util.Random;
public class TestDemo {
public int addPositive(int a, int b) {
if (a>0 && b>0) {
return a+b;
}
else throw new IllegalArgumentException ("Both parameters must be
positive!" );
}
public int randomNumberSquared() {
int randomInt = getRandomInt();
return randomInt * randomInt;
}
int getRandomInt() {
Random random = new Random();
return random.nextInt(10) + 1;
}
}
import static org.assertj.core.api.Assertions.assertThat;
import static org.assertj.core.api.Assertions.assertThatThrownBy;
import static org.junit.jupiter.api.Assertions.*;
import static org.mockito.Mockito.doReturn;
import static org.mockito.Mockito.spy;
import java.util.Random;
import java.util.stream.Stream;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.Arguments;
import org.junit.jupiter.params.provider.MethodSource;

class TestDemoTest {
```

```java
private TestDemo testDemo;
@BeforeEach
void setUp() throws Exception {
testDemo = new TestDemo();
}
@ParameterizedTest
@MethodSource("TestDemoTest#argumentsForAddPositive")
void assertThatTwoPositiveNumbersAreAddedCorrectly(int a , int b, int
expected, boolean expectException) {
 if (!expectException) {
 int result = testDemo.addPositive(a, b);
 assertThat(result).isEqualTo(expected);
 } else {
 assertThatThrownBy(() -> testDemo.addPositive(a, b))
 .isInstanceOf(IllegalArgumentException.class)
 .hasMessage("Both parameters must be positive!");
 }
}
public static Stream<Arguments> argumentsForAddPositive() {
 return Stream.of(
 Arguments.arguments(2, 3, 5, false),
 Arguments.arguments(5, 3, 8, false),
 Arguments.arguments(2, 0, 0, true),
 Arguments.arguments(-2, 3, 0, true),
 Arguments.arguments(2, -3, 0, true)
 );
}
@Test
void assertThatNumberSquaredIsCorrect() {
TestDemo mockDemo = spy(new TestDemo());
doReturn(5).when(mockDemo).getRandomInt();
int fiveSquared = mockDemo.randomNumberSquared();
assertThat(fiveSquared).isEqualTo(25);
}
}
```