**URL to GitHub Repository: https://github.com/lixy1979/java_week_12**

**URL to Public Link of your Video: https://youtu.be/QZxfvv3YEPM**

**TestDemo.java**

```
import java.util.Random;

public class TestDemo {

public int addPositive(int a, int b) {

if (a <= 0 || b <= 0) {

throw new IllegalArgumentException("Both parameters must be
positive!");

}

return a + b;

}

public int randomNumberSquared() {

int a = getRandomInt();

int squA = a*a;

System.out.printf("randomNumberSquared %d", squA);

return squA;

}

int getRandomInt() {

Random random = new Random();

return random.nextInt(10) + 1;

}

}
```

**TestDemoTest.java**

```
import static org.assertj.core.api.Assertions.assertThat;

import static org.assertj.core.api.Assertions.assertThatThrownBy;

import static org.junit.jupiter.params.provider.Arguments.arguments;

import static org.mockito.Mockito.doReturn;
import static org.mockito.Mockito.spy;

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 {

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) {

assertThat(testDemo.addPositive(a, b)).isEqualTo(expected);

}

else assertThatThrownBy(() ->

testDemo.addPositive(a, b))

.isInstanceOf(IllegalArgumentException.class);

}

static Stream<Arguments> argumentsForAddPositive(){

return Stream.of(

arguments(2, 4, 6, false),
arguments(0, 4, 0, true),

arguments(2, 0, 0, true),

arguments(-2, 1, 0, true),

arguments(1, -4, 0, true)

);

}

@Test

void assertThatNumberSquaredIsCorrect(){

testDemo.randomNumberSquared();

TestDemo mockDemo = spy(testDemo);

doReturn(5).when(mockDemo).getRandomInt();

int fiveSquared = mockDemo.randomNumberSquared();

assertThat(fiveSquared).isEqualTo(25);

}

}
```