

Προηγμένες Τεχνικές Προγραμματισμού

Εργασία Εργαστηρίων

Ονοματεπώνυμο: Στεφανιώρος Μιχαήλ

ΑΜ: 1072774

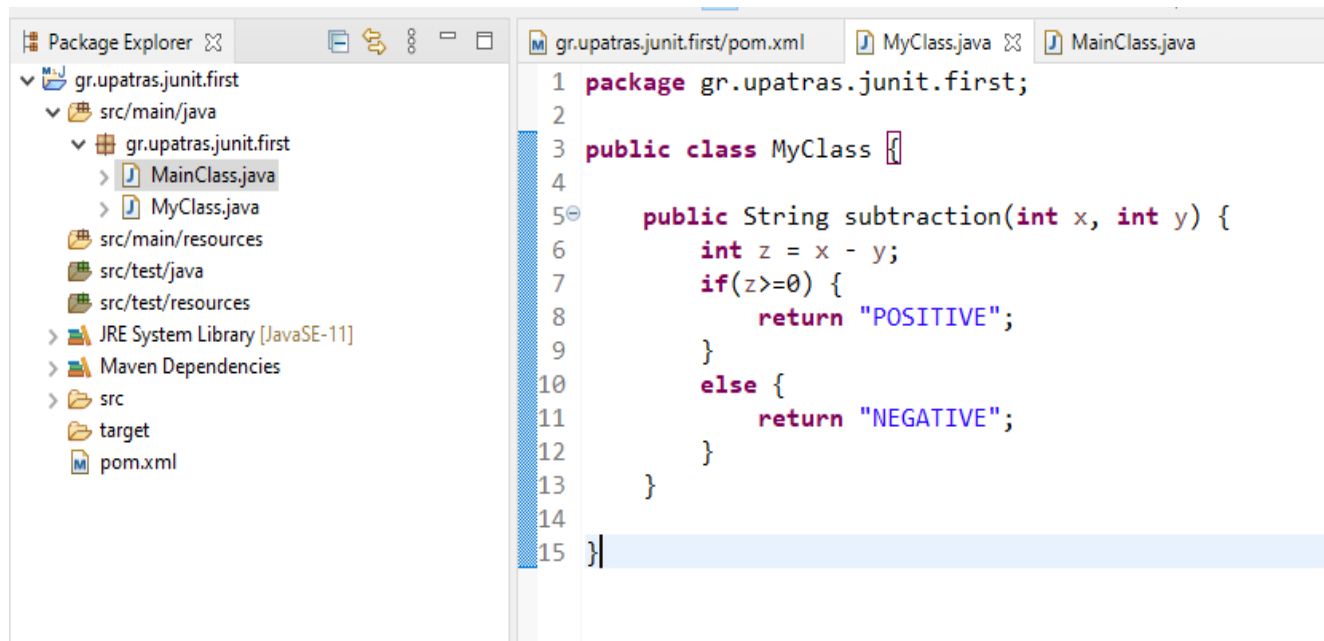
Κατεύθυνση: Υπολογιστών

Έτος: 4^ο

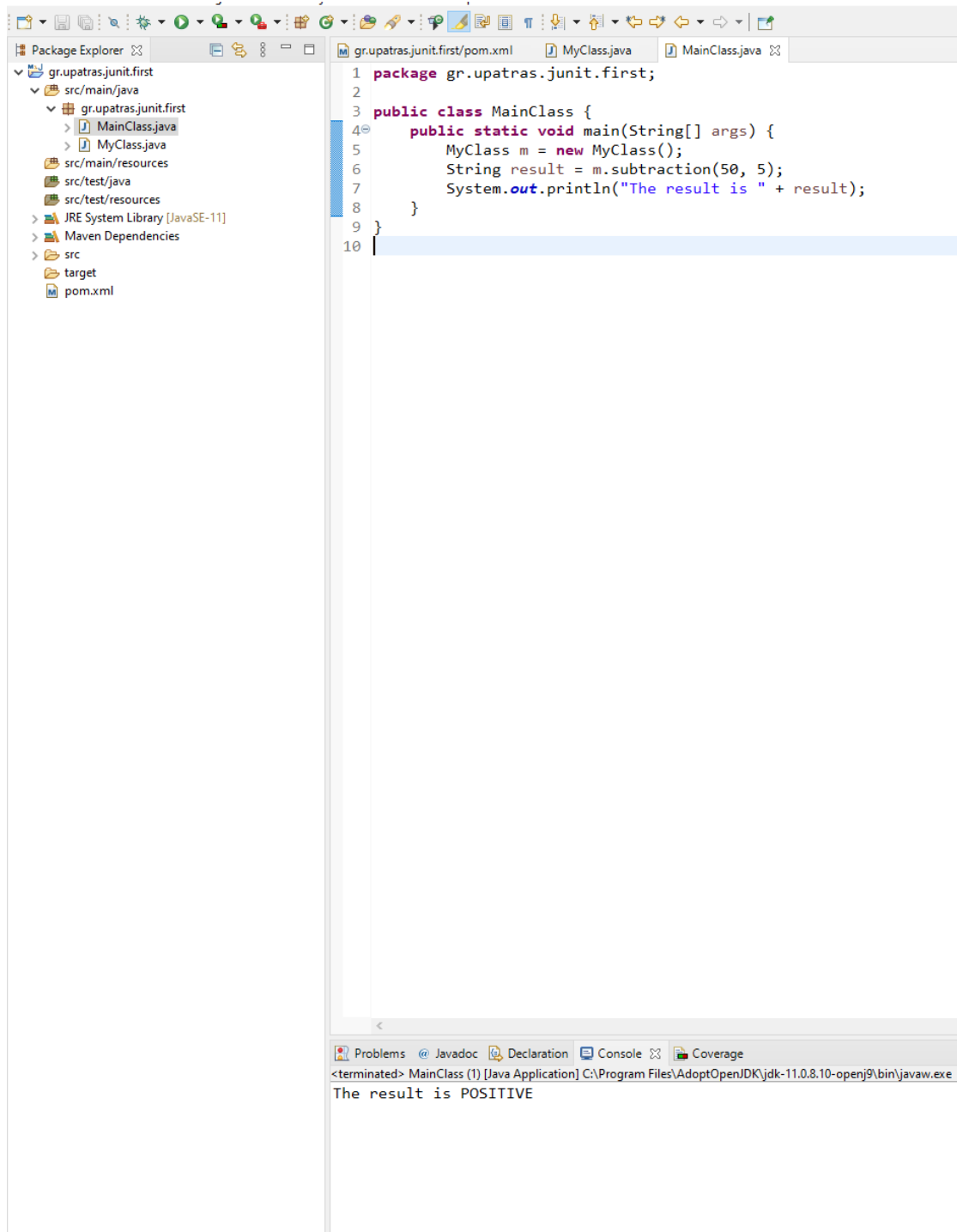
Η εκπόνηση της εργασίας έγινε με Java στο Eclipse

Άσκηση 1

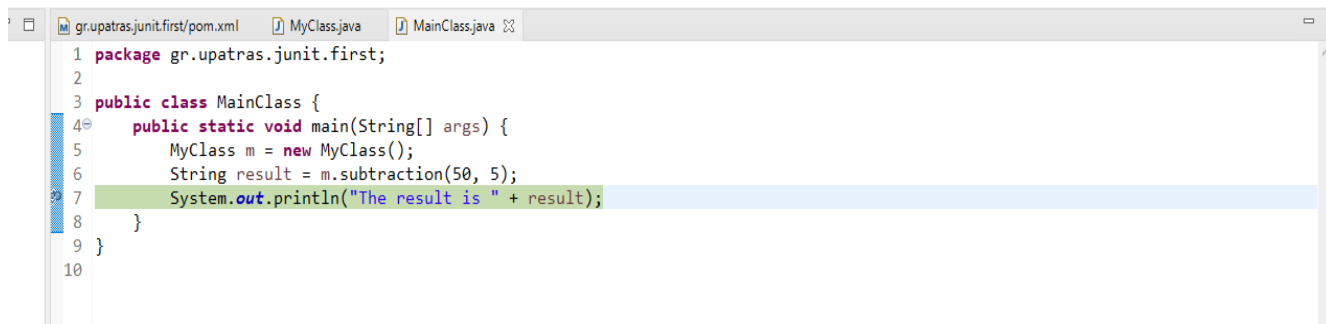
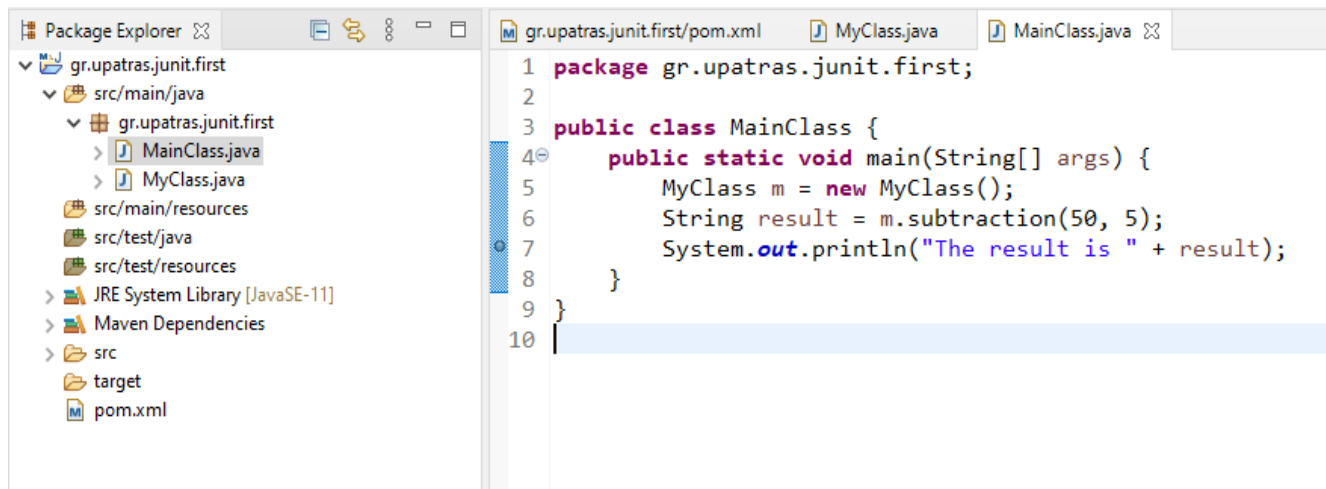
Μέθοδος που κάνει αφαίρεση ακεραίων και επιστρέφει την λέξη POSITIVE αν το αποτέλεσμα είναι θετικό, ενώ αντίθετα επιστρέφει NEGATIVE. Η μέθοδος υλοποιήθηκε σε μια public class MyClass. Στην συνέχεια, δημιουργήθηκε μια class MainClass στην οποία δημιουργείται αντικείμενο τύπου MyClass και εκτελεί την μέθοδο subtraction.



MainClass στην οποία καλείται η μέθοδος subtraction και το αποτέλεσμα:



Στην συνέχεια, εισάγεται breakpoint στο αποτέλεσμα της μεθόδου και γίνεται debugging.

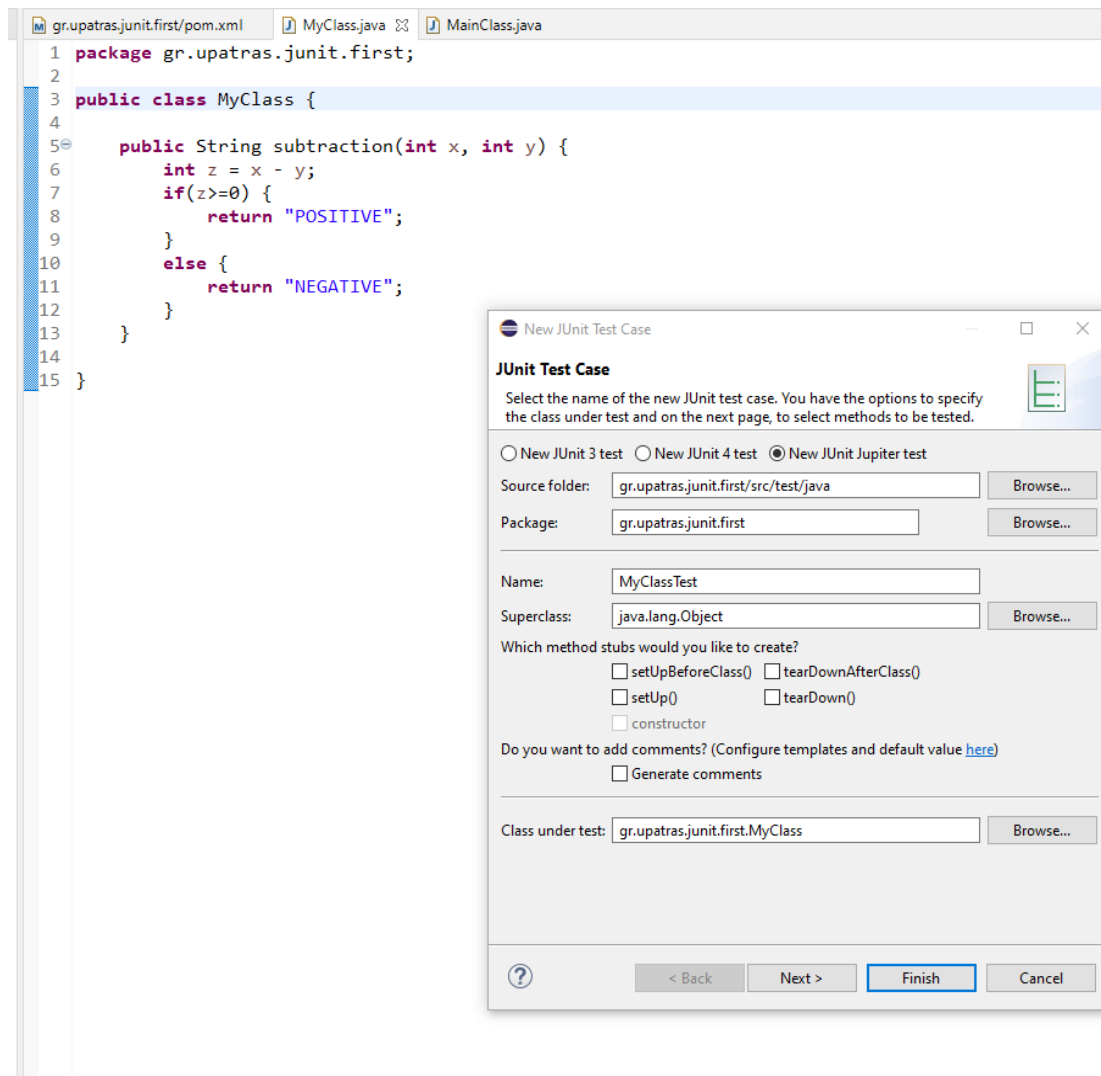


(x)= Variables Breakpoints Expressions	
Name	Value
no method return value	
args	String[0] (id=36)
m	MyClass (id=37)
result	"POSITIVE" (id=39)

Παρακάτω, φαίνεται η διαδικασία δημιουργίας ενός junit test για την μέθοδο αφαίρεσης ακεραίων.

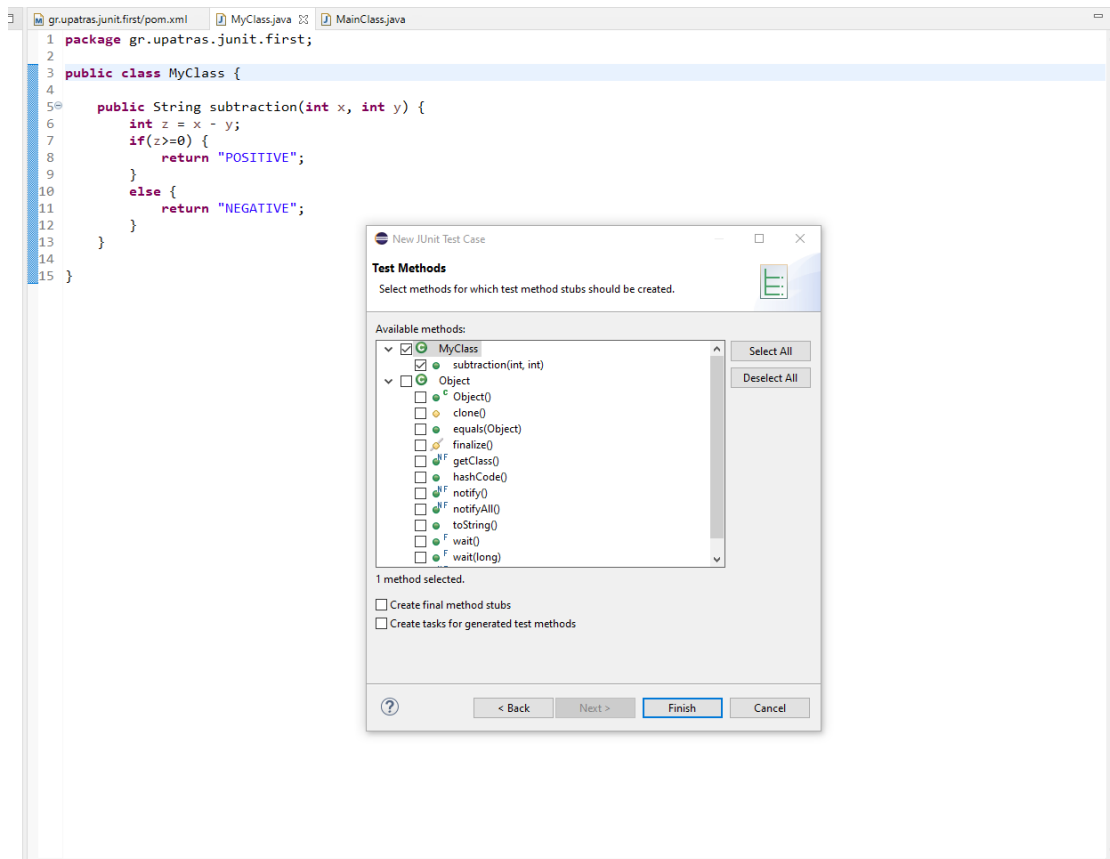
Βήμα 1

Right click στην κλάση που βρίσκεται η μέθοδος για την οποία θέλουμε να κάνουμε test (εδώ MyClass) -> New -> Junit Test Case



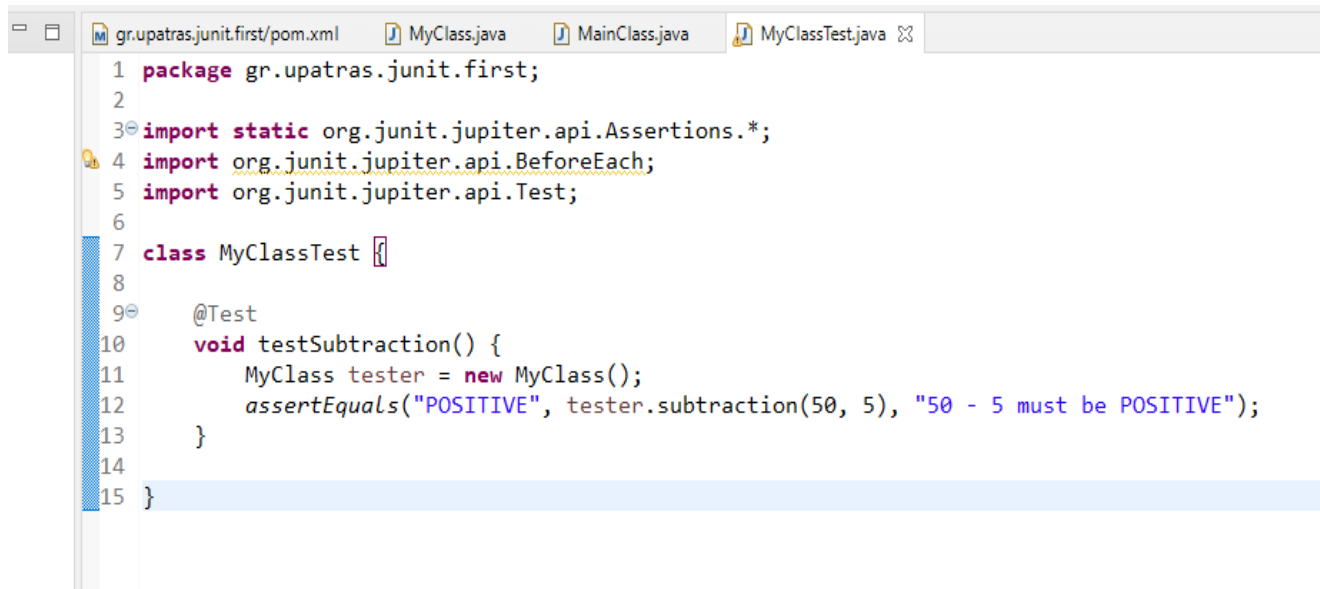
Βήμα 2

Πατάμε next και στο επόμενο παράθυρο επιλέγουμε την κλάση και την μέθοδο που θέλουμε να τεστάρουμε



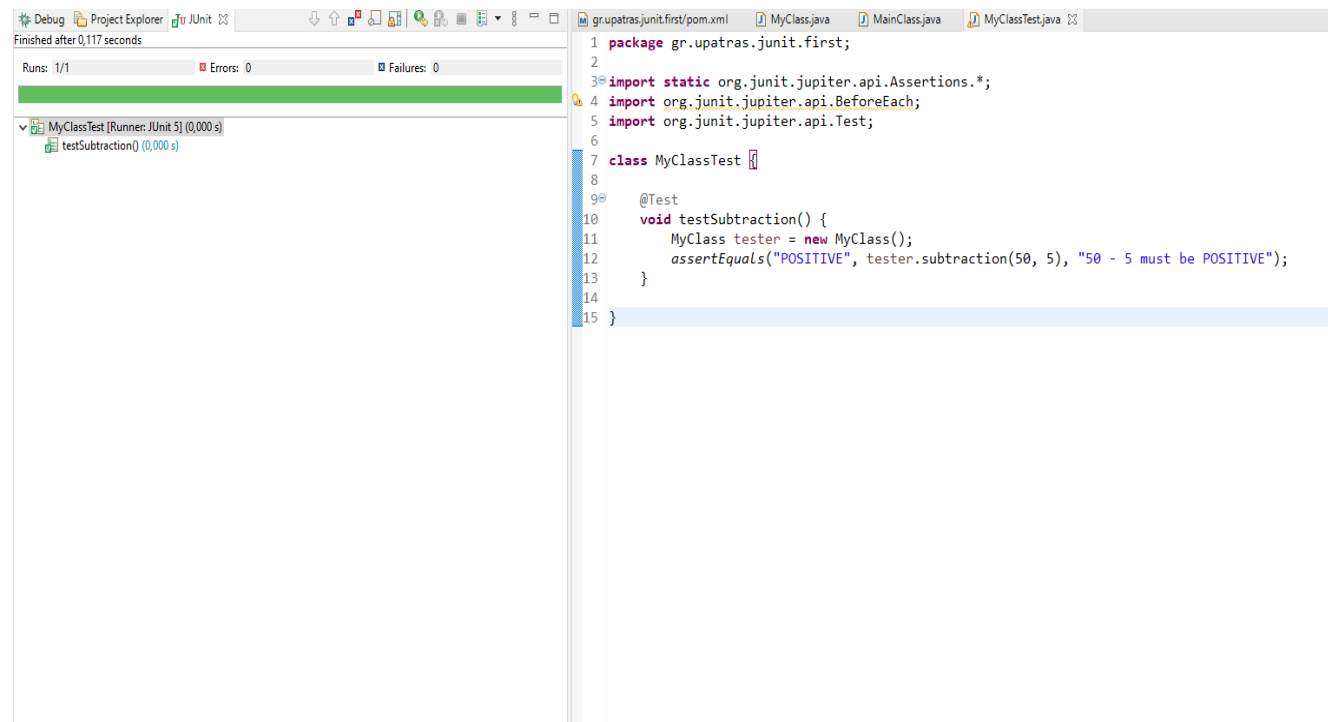
Βήμα 3

Γράφουμε τον παρακάτω κώδικα ή συμπληρώνουμε αναλόγως με τα test cases που θέλουμε να εξετάσουμε.



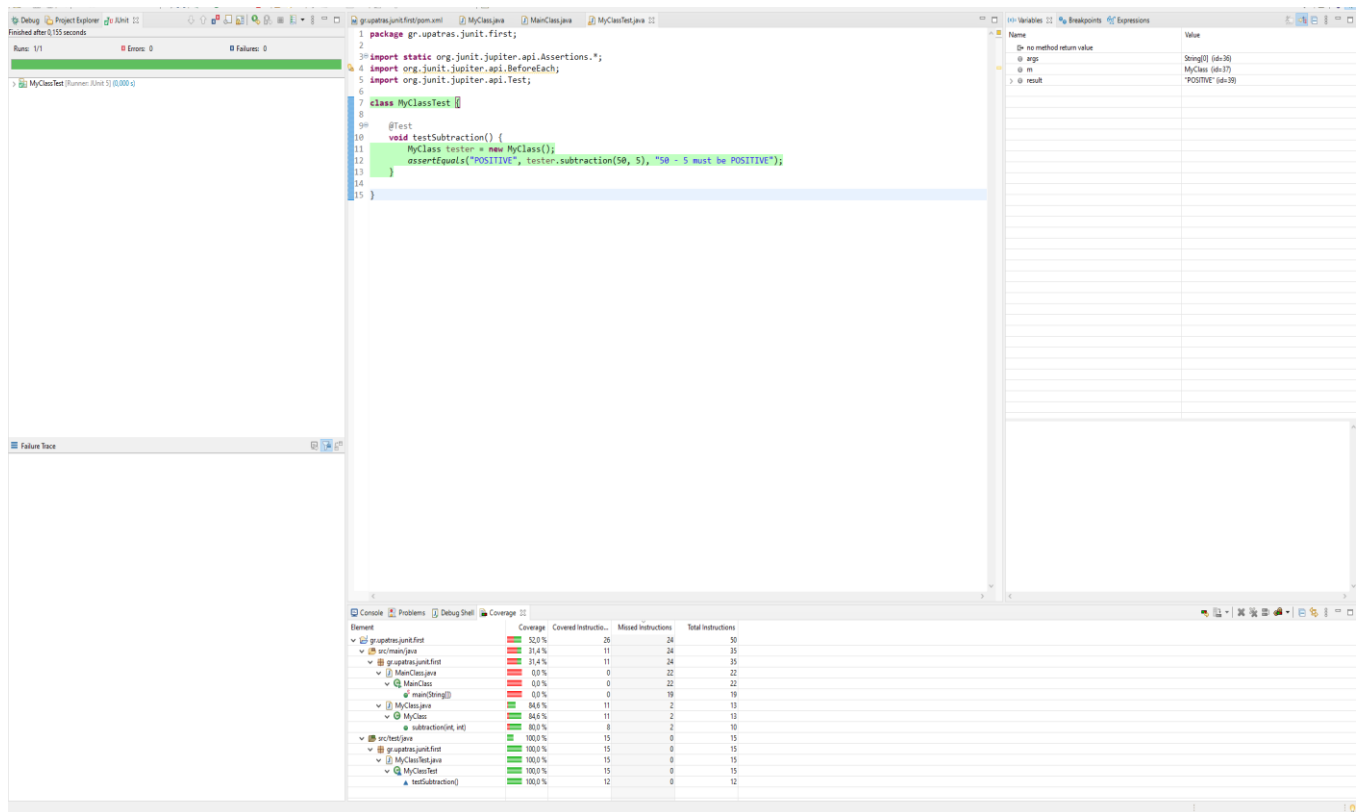
Βήμα 4

Right click στην καινούρια test class -> Run As -> JUnit Test



Παραπάνω, είναι φανερό ότι το test case είναι σωστό και η μέθοδος λειτουργεί σωστά.

Τελικά, βλέπουμε ότι μπορούμε να κάνουμε test coverage στο 52% των γραμμών κώδικα του project και συγκεκριμένα για την κλάση MyClass, στο 84.6%.



Κλάσεις:

MyClass:

```
package gr.upatras.junit.first;
```

```
public class MyClass {

    public String subtraction(int x, int y) {
        int z = x - y;
        if(z >= 0) {
            return "POSITIVE";
        }
        else {
            return "NEGATIVE";
        }
    }

}
```

MainClass:

```
package gr.upatras.junit.first;
```



```

public class MainClass {
    public static void main(String[] args) {
        MyClass m = new MyClass();
        String result = m.subtraction(50, 5);
        System.out.println("The result is " + result);
    }
}

```

MyClassTest:

```

package gr.upatras.junit.first;

import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

class MyClassTest {

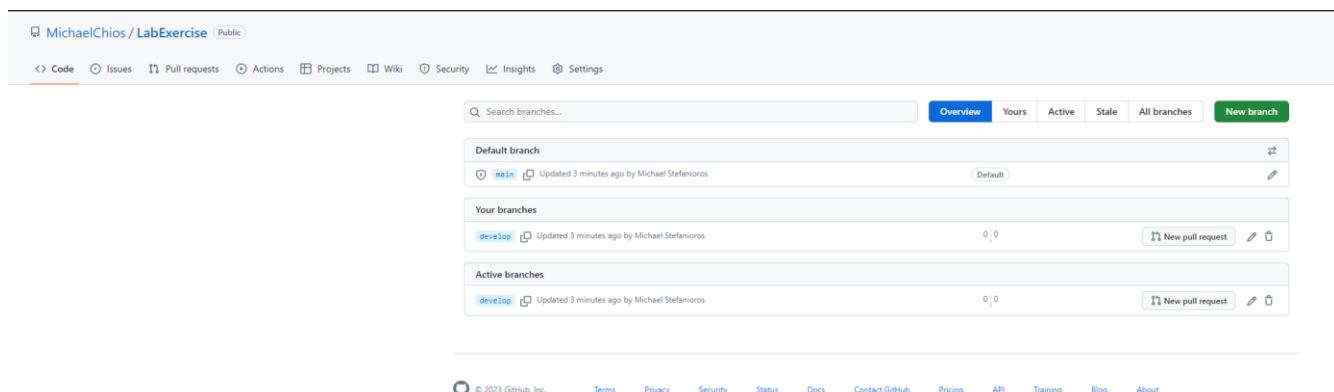
    @Test
    void testSubtraction() {
        MyClass tester = new MyClass();
        assertEquals("POSITIVE", tester.subtraction(50, 5),
"50 - 5 must be POSITIVE");
    }

}


```


Άσκηση 2


Έχοντας ανεβάσει το project σε ένα github repository, φτιάχνουμε δύο branches, ένα main και ένα develop.



Έπειτα πραγματοποιούμε μια μικρή αλλαγή στο develop και κάνουμε commit.


 develop ▾


 2 branches


 0 tags


Go to file


Add file ▾


 Code ▾

This branch is 1 commit ahead of main.  Contribute ▾

 MichaelChios This is a small change in develop branch c61c9ee 1 minute ago ⌚ 3 commits

 gr.upatras.junit.first This is a small change in develop branch 1 minute ago


 README.md Initial commit 12 minutes ago


README.md 


LabExercise

Εργασία στις Προηγμένες Τεχνικές Προγραμματισμού

Τέλος, κάνουμε merge το develop στο main.


 main ▾


 2 branches


 0 tags


Go to file


Add file ▾

 Code ▾

 MichaelChios This is a small change in develop branch c61c9ee 5 minutes ago ⌚ 3 commits

 gr.upatras.junit.first This is a small change in develop branch 5 minutes ago

 README.md Initial commit 16 minutes ago

README.md 

LabExercise

Εργασία στις Προηγμένες Τεχνικές Προγραμματισμού

Link for Github repository:

<https://github.com/MichaelChios/LabExercise>

Άσκηση 3

Αφού κατεβάσουμε το rest.example από το github, τρέχουμε το Application.java ως java application. Η εφαρμογή ακούει στο port 8888.

```
Application [Java Application] C:\Program Files\AdoptOpenJDK\jdk-11.0.8.10-open\bin\javaw.exe (31 Mai 2023, 9:53:54 μ.μ.)

:: Spring Boot ::
(v2.4.0)

2023-05-31 21:54:00.416 INFO 8784 --- [main] gr.upatras.rest.example.Application : Starting Application using Java 11.0.8 on LAPTOP-LA8K6TQP with PID 8784 (C:\Users\User\OneDrive - University
2023-05-31 21:54:00.426 INFO 8784 --- [main] gr.upatras.rest.example.Application : No active profile set, falling back to default profiles: default
2023-05-31 21:54:03.660 INFO 8784 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8888 (http)
2023-05-31 21:54:03.680 INFO 8784 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-05-31 21:54:03.680 INFO 8784 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.39]
2023-05-31 21:54:03.818 INFO 8784 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-05-31 21:54:03.818 INFO 8784 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 3261 ms
2023-05-31 21:54:04.566 INFO 8784 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2023-05-31 21:54:04.804 INFO 8784 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8888 (http) with context path ''
2023-05-31 21:54:05.391 INFO 8784 --- [main] gr.upatras.rest.example.Application : Started Application in 5.646 seconds (JVM running for 6.13)
2023-05-31 21:54:29.478 INFO 8784 --- [nio-8888-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-05-31 21:54:29.479 INFO 8784 --- [nio-8888-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2023-05-31 21:54:29.482 INFO 8784 --- [nio-8888-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 3 ms
```

Κατευθυνόμαστε στο link:

<http://localhost:8888/swagger-ui/>

Έπειτα, επιλέγουμε product-controller,

product-controllerProduct Controller

POST /productCreates a Product

GET /product/Retrieves all Products

This operation retrieves all Product entities.

Parameters

No parameters

Try it out

Responses

Code	Description	Links
200		No links

try it out

product-controller

Product Controller

POST

/product

Creates a Product

GET

/product/

Retrieves all Products

This operation retrieves all Product entities.

Parameters

No parameters

Cancel

Execute

Responses

και τελικά, execute. Το αποτέλεσμα που βλέπουμε είναι:

Responses

Curl

curl -X GET "http://localhost:8888/product/" -H "accept: application/json;charset=utf-8"

Request URL

http://localhost:8888/product/

Server response

Code

Details

200

Response body

```
{
  "id": 100,
  "pname": "Mobile",
  "batchno": "CLK98123",
  "price": 9000,
  "noofproduct": 6
},
{
  "id": 101,
  "pname": "Smart TV",
  "batchno": "LGS109167",
  "price": 60000,
  "noofproduct": 3
},
{
  "id": 102,
  "pname": "Washing Machine",
  "batchno": "387538K9",
  "price": 9000,
  "noofproduct": 7
},
{
  "id": 103,
  "pname": "Laptop",
  "batchno": "LHP290CP",
  "price": 24000,
  "noofproduct": 1
}
```

Download

Response headers

connection: keep-alive
content-type: application/json;charset=utf-8
date: Wed31 May 2023 19:03:27 GMT
keep-alive: timeout=60
transfer-encoding: chunked

Responses

Code

Description

Links

200

Success

No links

Media type

application/json;charset=utf-8

Controls Accept header.

Example Value | Schema

```
{
  "batchno": "string"
}
```

Σε τελευταίο στάδιο, έγινε προσπάθεια να μπει η εφαρμογή σε docker container. Τα απαραίτητα αρχεία βρίσκονται στον φάκελο του project.

Link for Github repository:

<https://github.com/MichaelChios/LabExercise>