



Εργασία 1

Προηγένες Τεχνικές

Prepared by

Νίκου Παύλος(up1062998)

up1062998@upnet.gr

12/4/2022

Code

ex1_lab1:

```
function subtract(num1, num2) {  
  if (!isNaN(num1) && !isNaN(num2)) {  
    isPositive = num1 >= num2  
    if (isPositive === true) {  
      return "POSITIVE";  
    }  
    return "NEGATIVE";  
  }  
  else return "incorrect input"  
}
```

ex1_lab1_Test:

```
describe("Testing ex1_lab1.js", () => {  
  it("Testing positive values for POSITIVE", () => {  
    expect(subtract(3,2)).toBe("POSITIVE")  
  })  
  it("Testing negative values for NEGATIVE", () => {  
    expect(subtract(0,3)).toBe("NEGATIVE")  
  })  
  it("Testing equal values", () => {  
    expect(subtract(0,0)).toBe("POSITIVE")  
  })  
  it("Testing negative values for POSITIVE", () => {  
    expect(subtract(-2,-3)).toBe("POSITIVE")  
  })  
  it("Testing negative values for NEGATIVE", () => {  
    expect(subtract(-3,-2)).toBe("NEGATIVE")  
  })  
  it("Testing string values that are numbers", () => {  
    expect(subtract("33","22")).toBe("POSITIVE")  
  })  
  it("Testing string values that arent numbers", () => {  
    expect(subtract("33er!!","help me")).toBe("incorrect input")  
  })  
  it("Testing float values for POSITIVE", () => {  
    expect(subtract(33.35,25.43)).toBe("POSITIVE")  
  })  
  it("Testing float values for NEGATIVE", () => {  
    expect(subtract(34,58.33)).toBe("NEGATIVE")  
  })  
})
```

Objectives

Χρησιμοποιήστε κάποιο IDE (Eclipse, αλλά αν θέλετε vs code η κάποιο άλλο είναι αποδεκτό)

1. Φτιάξτε μια μέθοδο που κάνει αφαίρεση ακεραίων σε όποια γλώσσα προγραμματισμού θέλετε και γυρίζει τη λέξη POSITIVE αν το αποτέλεσμα είναι θετικό αλλιώς NEGATIVE αν είναι αρνητικό(Δείξτε με screenshot)
2. Φτιάξτε μια άλλη μέθοδο που την καλεί. (πχ από main method από java) (Δείξτε με screenshot)
3. Στο περιβάλλον ανάπτυξης που θα επιλέξετε, βάλτε breakpoint στο αποτέλεσμα της μεθόδου. (Δείξτε με screenshot)
4. Δείξτε με screenshot πώς κάνετε debugging όταν καλείται η μέθοδος στο αποτέλεσμα
5. Δημιουργήστε ένα Unit test για τη μέθοδο που κάνει αφαίρεση ακεραίων (Δείξτε με screenshot)
6. Μπορείτε να κάνετε test coverage σε > 50% των γραμμών κώδικα; (Δείξτε με screenshot)

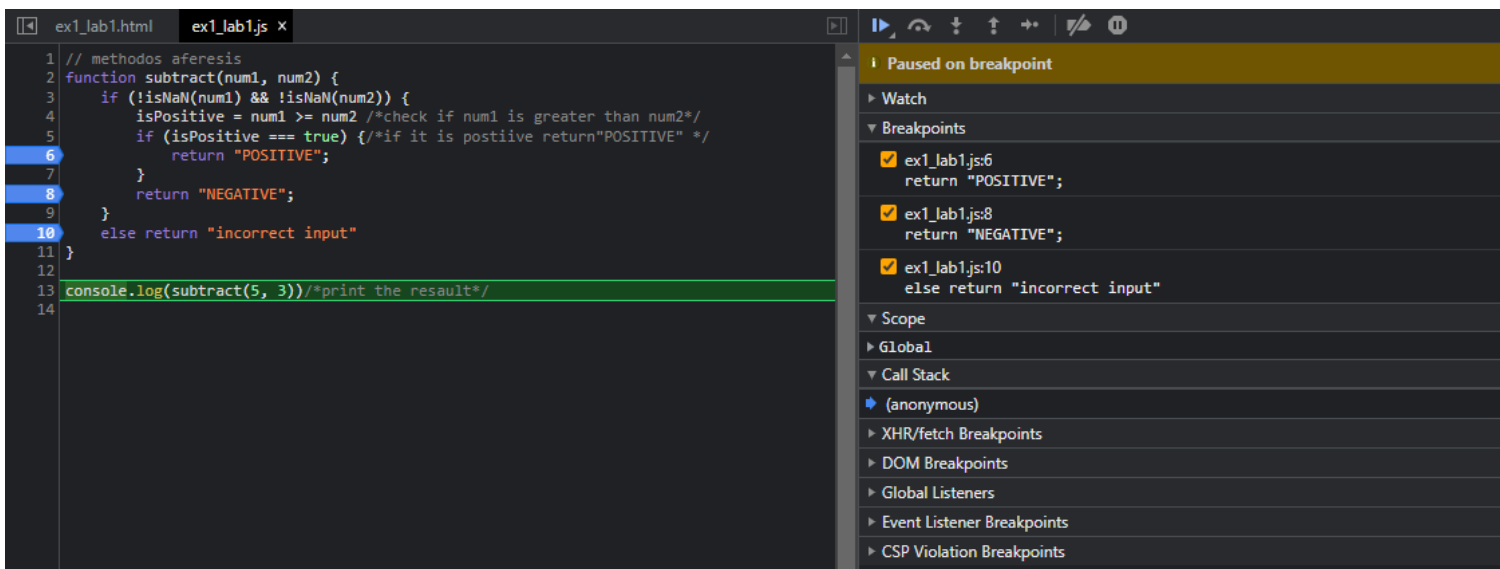
1.Φτιάξτε μια μέθοδο που κάνει αφαίρεση ακεραίων σε όποια γλώσσα προγραμματισμού θέλετε και γυρίζει τη λέξη POSITIVE αν το αποτέλεσμα είναι θετικό αλλιώς NEGATIVE αν είναι αρνητικό(Δείξτε με screenshot)

```
jasmine-standalone-4.1.0 > src > JS ex1_lab1.js > subtract
1  function subtract(num1, num2) {
2      if (!isNaN(num1) && !isNaN(num2)) { /*check if both args are numbers*/
3          isPositive = num1 >= num2 /*check if num1 is greater than num2*/
4          if (isPositive === true) { /*if it is postiiive return "POSITIVE" */
5              return "POSITIVE";
6          }
7          return "NEGATIVE";
8      }
9      else return "incorrect input"
10 }
```

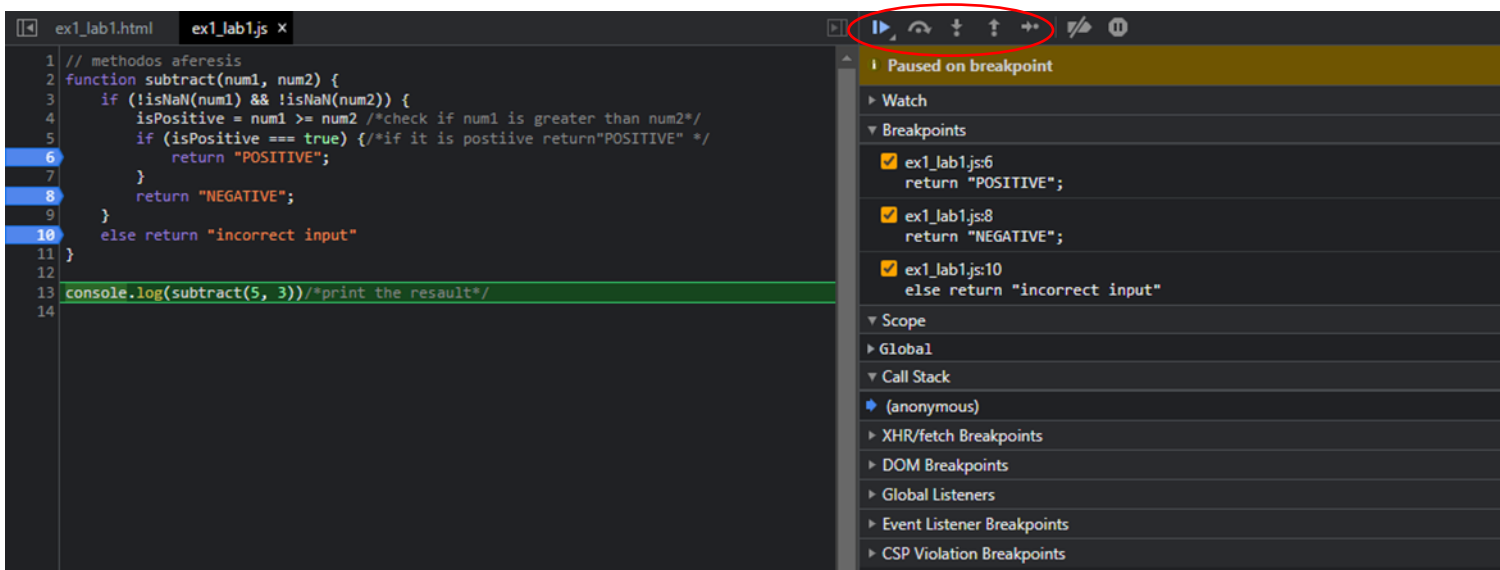
2.Φτιάξτε μια άλλη μέθοδο που την καλεί. (πχ από main method από java) (Δείξτε με screenshot)

```
console.log(subtract(5, 3))/*print the resault*/
```

3. Στο περιβάλλον ανάπτυξης που θα επιλέξετε, βάλτε breakpoint στο αποτέλεσμα της μεθόδου. (Δείτε με screenshot)



4. Δείτε με screenshot πώς κάνετε debugging όταν καλείται η μέθοδος στο αποτέλεσμα.



Ανάλογα με το κουμπί που θα πατήσουμε (βλέπουμε κόκκινο κύκλο) μπορούμε είτε να προχωρήσουμε στην επόμενη γραμμή, στην προηγούμενη, ή να προσπεράσουμε μια συνάρτηση ώστε να μην μπει το debugger μέσα αν ξέρουμε ότι η συνάρτηση αυτή δεν έχει κάποιο πρόβλημα.

5. Δημιουργήστε ένα Unit test για τη μέθοδο που κάνει αφαίρεση ακεραίων
(Δείξτε με screenshot)

Για την δημιουργία του unitTest χρησιμοποιούμε το Jasmine.

```
jasmine-standalone-4.1.0 > spec > JS ex1_lab1_test.js > ...
1  describe("Testing ex1_lab1.js", () => {
2      it("Testing positive values for POSITIVE", () => {
3          expect(subtract(3,2)).toBe("POSITIVE")
4      })
5      it("Testing negative values for NEGATIVE", () => {
6          expect(subtract(0,3)).toBe("NEGATIVE")
7      })
8      it("Testing equal values", () => {
9          expect(subtract(0,0)).toBe("POSITIVE")
10     })
11     it("Testing negative values for POSITIVE", () => {
12         expect(subtract(-2,-3)).toBe("POSITIVE")
13     })
14     it("Testing negative values for NEGATIVE", () => {
15         expect(subtract(-3,-2)).toBe("NEGATIVE")
16     })
17     it("Testing string values that are numbers", () => {
18         expect(subtract("33","22")).toBe("POSITIVE")
19     })
20     it("Testing string values that arent numbers", () => {
21         expect(subtract("33er!!","help me")).toBe("incorrect input")
22     })
23     it("Testing float values for POSITIVE", () => {
24         expect(subtract(33.35,25.43)).toBe("POSITIVE")
25     })
26     it("Testing float values for NEGATIVE", () => {
27         expect(subtract(34,58.33)).toBe("NEGATIVE")
28     })
29 })
30 |
```

το αποτέλεσμα του τεστ αυτού εμφανίζεται στο jasmine Spec Runner

upatras eclass | Προηγμένες Τεχν... X | https://eclass.upatras.gr/module... X | Document X | Github - jasmine/jasmine: Simpl... X | Jasmine Spec Runner v4.1.0 X | +

127.0.0.1:5500/jasmine-standalone-4.1.0/SpecRunner.html

python exess | Μάθημα 2018-201... | Προπτυχιακές Στο... | upatras eclass | UPnet Webmail - K... | ΝΕΑ ΠΡΟΤΙΑ | YES

Jasmine 4.1.0 options

9 specs, 0 failures, randomized with seed 71588 finished in 0.023s

Testing ex1_lab1.js

- Testing negative values for NEGATIVE
- Testing positive values for POSITIVE
- Testing string values that are numbers
- Testing float values for NEGATIVE
- Testing string values that arent numbers
- Testing negative values for POSITIVE
- Testing negative values for NEGATIVE
- Testing float values for POSITIVE
- Testing equal values