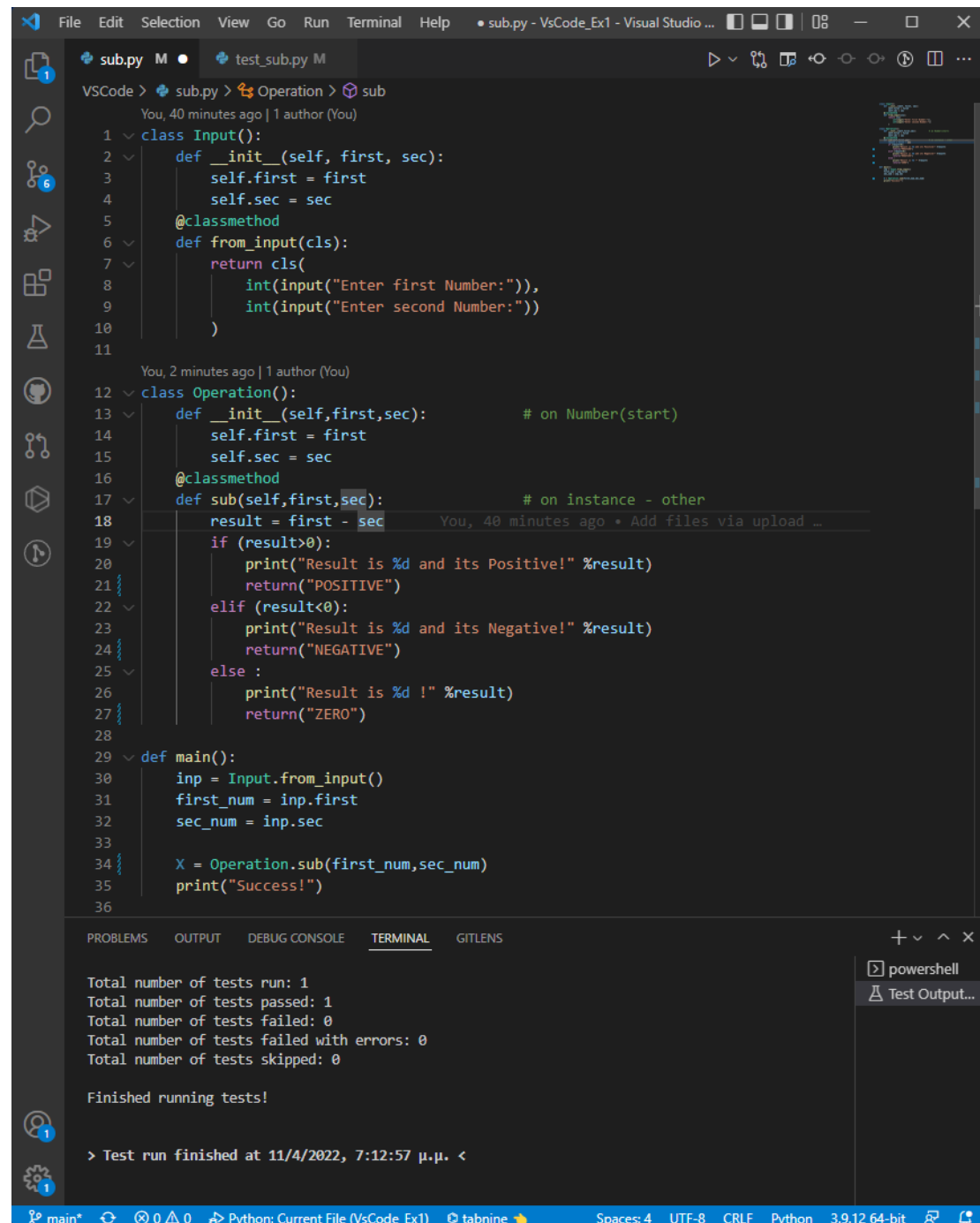


- Χρησιμοποιήστε κάποιο IDE (Eclipse, αλλά αν θέλετε vs code η κάποιο άλλο είναι αποδεκτό)

Χρησιμοποιήθηκε το VS Code



```
File Edit Selection View Go Run Terminal Help • sub.py - VsCode_Ex1 - Visual Studio ...
sub.py M test_sub.py M
VSCode > sub.py > Operation > sub
You, 40 minutes ago | 1 author (You)
1 class Input():
2     def __init__(self, first, sec):
3         self.first = first
4         self.sec = sec
5     @classmethod
6     def from_input(cls):
7         return cls(
8             int(input("Enter first Number:")),
9             int(input("Enter second Number:"))
10        )
11
You, 2 minutes ago | 1 author (You)
12 class Operation():
13     def __init__(self,first,sec):          # on Number(start)
14         self.first = first
15         self.sec = sec
16     @classmethod
17     def sub(self,first,sec):              # on instance - other
18         result = first - sec
19         if (result>0):
20             print("Result is %d and its Positive!" %result)
21             return("POSITIVE")
22         elif (result<0):
23             print("Result is %d and its Negative!" %result)
24             return("NEGATIVE")
25         else :
26             print("Result is %d !" %result)
27             return("ZERO")
28
29 def main():
30     inp = Input.from_input()
31     first_num = inp.first
32     sec_num = inp.sec
33
34     X = Operation.sub(first_num,sec_num)
35     print("Success!")
36
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS

```
Total number of tests run: 1
Total number of tests passed: 1
Total number of tests failed: 0
Total number of tests failed with errors: 0
Total number of tests skipped: 0

Finished running tests!

> Test run finished at 11/4/2022, 7:12:57 μ.μ. <
```

main\* 0 0 Python: Current File (VsCode\_Ex1) tabnine Spaces: 4 UTF-8 CRLF Python 3.9.12 64-bit

- 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)

```
class Operation():
    def __init__(self,first,sec):          # on Number(start)
        self.first = first
        self.sec = sec
    @classmethod
    def sub(self,first,sec):              # on instance - other
        result = first - sec
        if (result>0):
            print("Result is %d and its Positive!" %result)
            return("POSITIVE")
        elif (result<0):
            print("Result is %d and its Negative!" %result)
            return("NEGATIVE")
        else :
            print("Result is %d !" %result)
            return("ZERO")
```

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

```
def main():
    inp = Input.from_input()
    first_num = inp.first
    sec_num = inp.sec

    X = Operation.sub(first_num,sec_num)
    print("Success!")
```

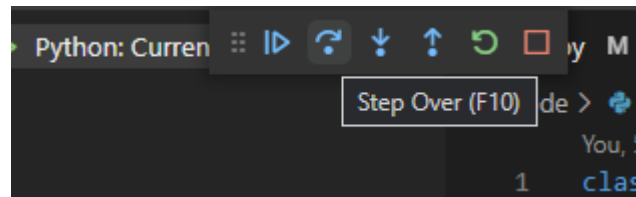
- Στο περιβάλλον ανάπτυξης που θα επιλέξετε, βάλτε breakpoint στο αποτέλεσμα της μεθόδου. (Δείξτε με screenshot)

You, 3 minutes ago | 1 author (You)

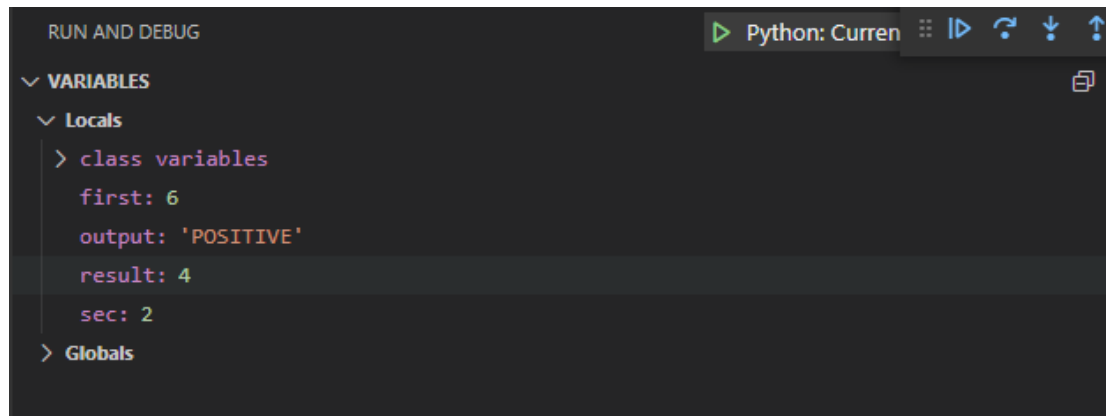
```
12  class Operation():
13      def __init__(self,first,sec):
14          self.first = first
15          self.sec = sec
16      @classmethod
17      def sub(self,first,sec):
18
19          result = first - sec
20          if (result>0):
21              output = "POSITIVE"
22          elif (result<0):
23              output = "NEGATIVE"
24          else :
25              output = "ZERO"
26
27          return(output)
28
```

- Δείξτε με screenshot πώς κάνετε debugging όταν καλείται η μέθοδος στο αποτέλεσμα

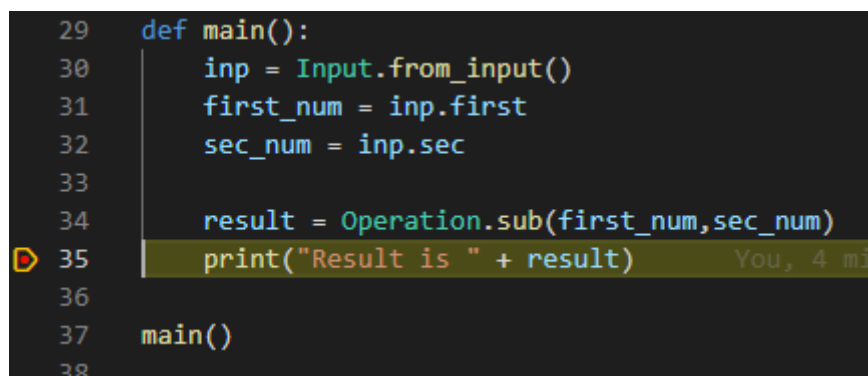
Μπάρα ενεργειών



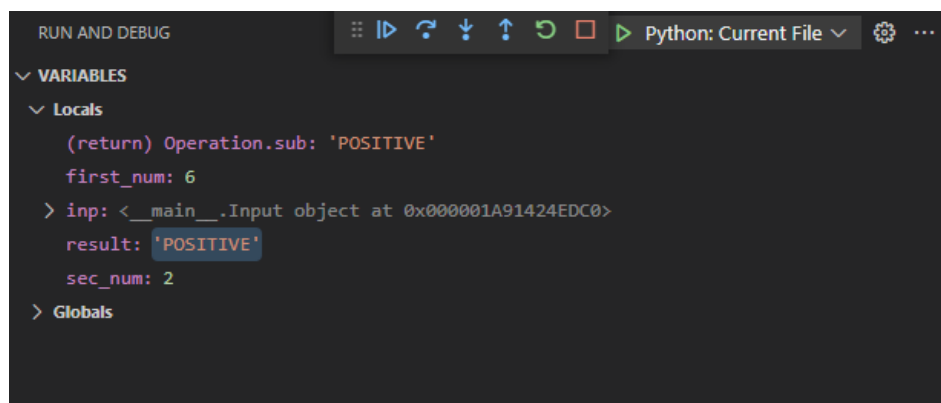
Τιμές Μεταβλητών πριν το breakpoint της εξόδου.



Breakpoint πριν την ολοκλήρωση της μεθόδου

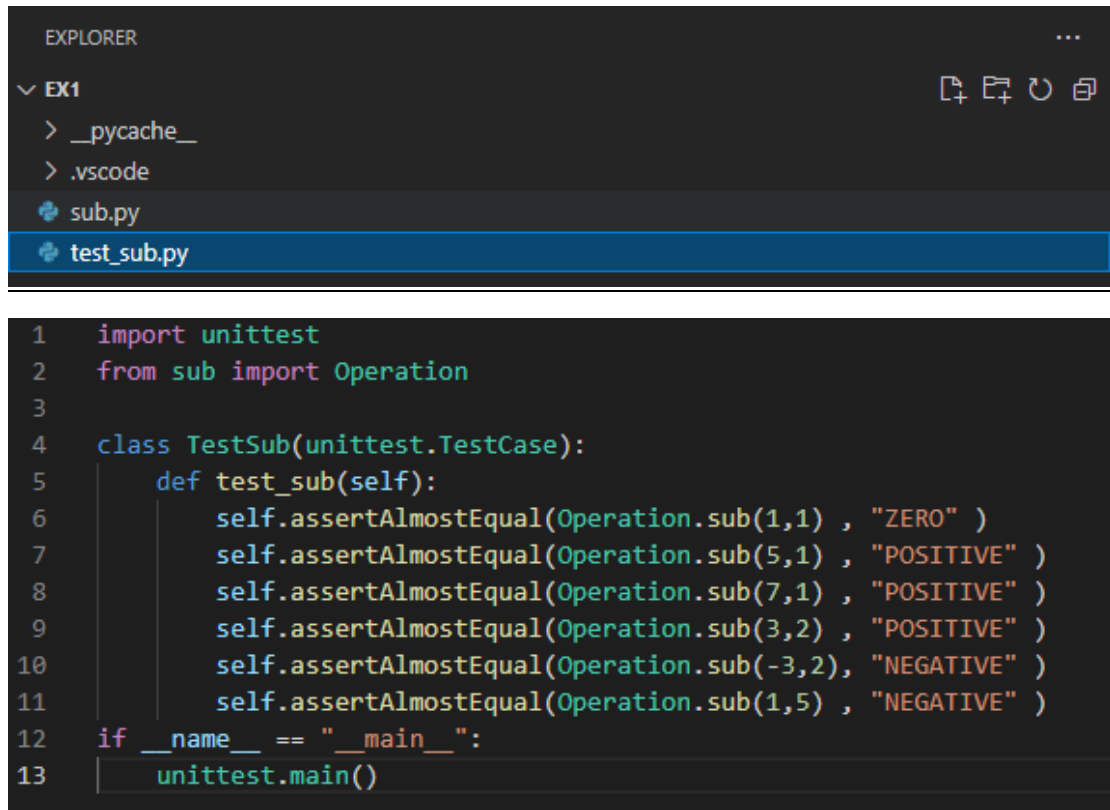


Τιμές Μεταβλητών αφού ολοκληρώθηκε η μέθοδος



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

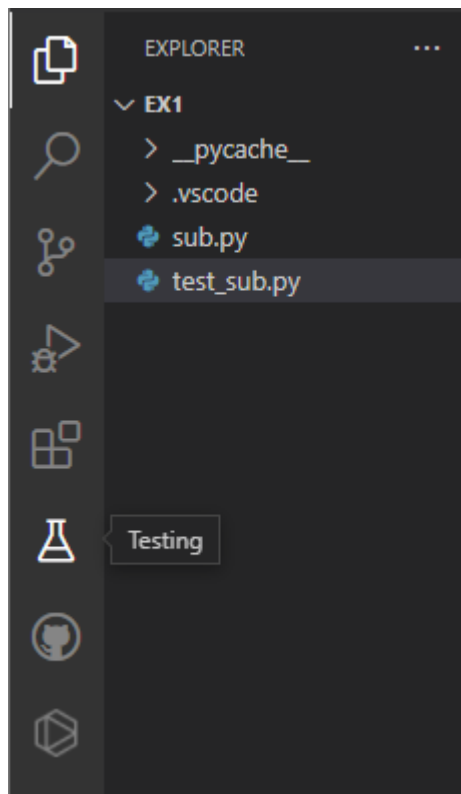
Αφού έχουμε δημιουργήσει ένα αρχείο test\_sub.py



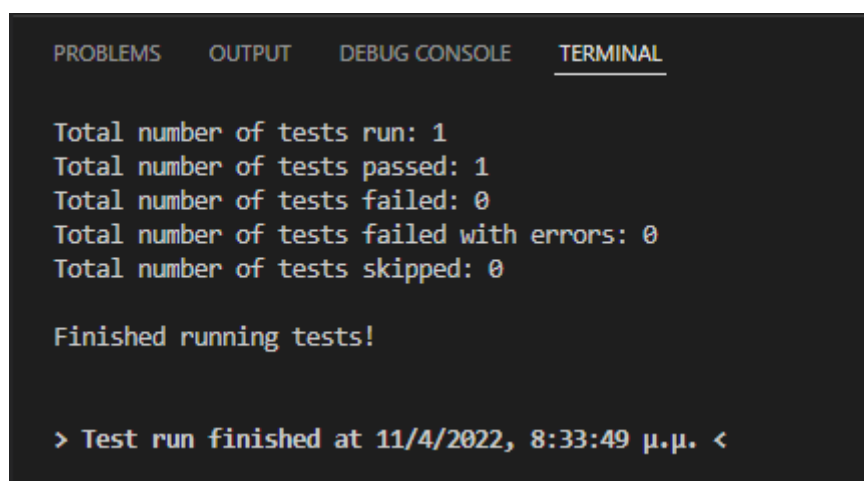
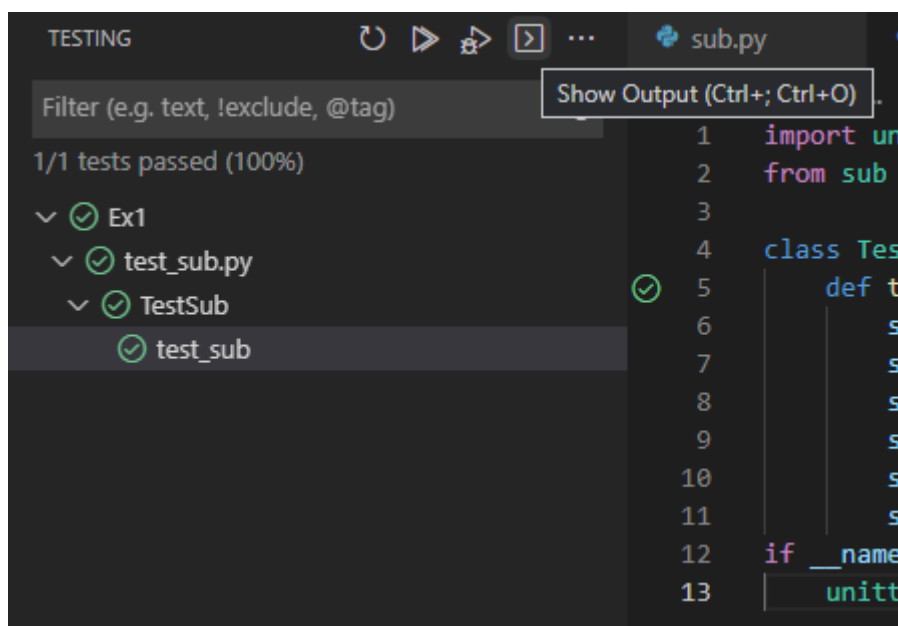
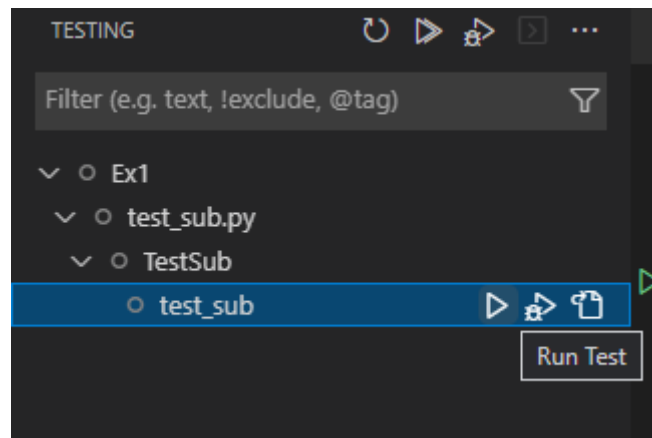
The image shows two screenshots from Visual Studio Code. The top screenshot shows the Explorer sidebar with a project named 'EX1'. It contains a folder with sub-files and two Python files: 'sub.py' and 'test\_sub.py'. 'test\_sub.py' is selected. The bottom screenshot shows the code editor with the content of 'test\_sub.py'.

```
1 import unittest
2 from sub import Operation
3
4 class TestSub(unittest.TestCase):
5     def test_sub(self):
6         self.assertEqual(Operation.sub(1,1) , "ZERO" )
7         self.assertEqual(Operation.sub(5,1) , "POSITIVE" )
8         self.assertEqual(Operation.sub(7,1) , "POSITIVE" )
9         self.assertEqual(Operation.sub(3,2) , "POSITIVE" )
10        self.assertEqual(Operation.sub(-3,2), "NEGATIVE" )
11        self.assertEqual(Operation.sub(1,5) , "NEGATIVE" )
12
13 if __name__ == "__main__":
14     unittest.main()
```

Στο extension του testing της python



Τρέχουμε το test και βλέπουμε τα αποτελέσματα του



- Μπορείτε να κάνετε test coverage σε > 50% των γραμμών κώδικα; (Δείξτε με screenshot)

Αφού εγκαταστήσουμε το module coverage

```
pip install coverage
```

Τρέχουμε το test\_sub.py με το coverage

```
coverage run test_sub.py
```

```
Enter first Number:6
Enter second Number:2
Result is POSITIVE
.
-----
Ran 1 test in 0.000s
OK
```

```
Enter first Number:6
Enter second Number:2
Result is POSITIVE
```

Αυτό δημιουργεί ένα report

```
coverage report
```

Name	Stmts	Miss	Cover
-----			
sub.py	27	2	93%
test_sub.py	12	0	100%
-----			
TOTAL	39	2	95%

Και όπως βλέπουμε στο unit test πετύχαμε coverage 100%