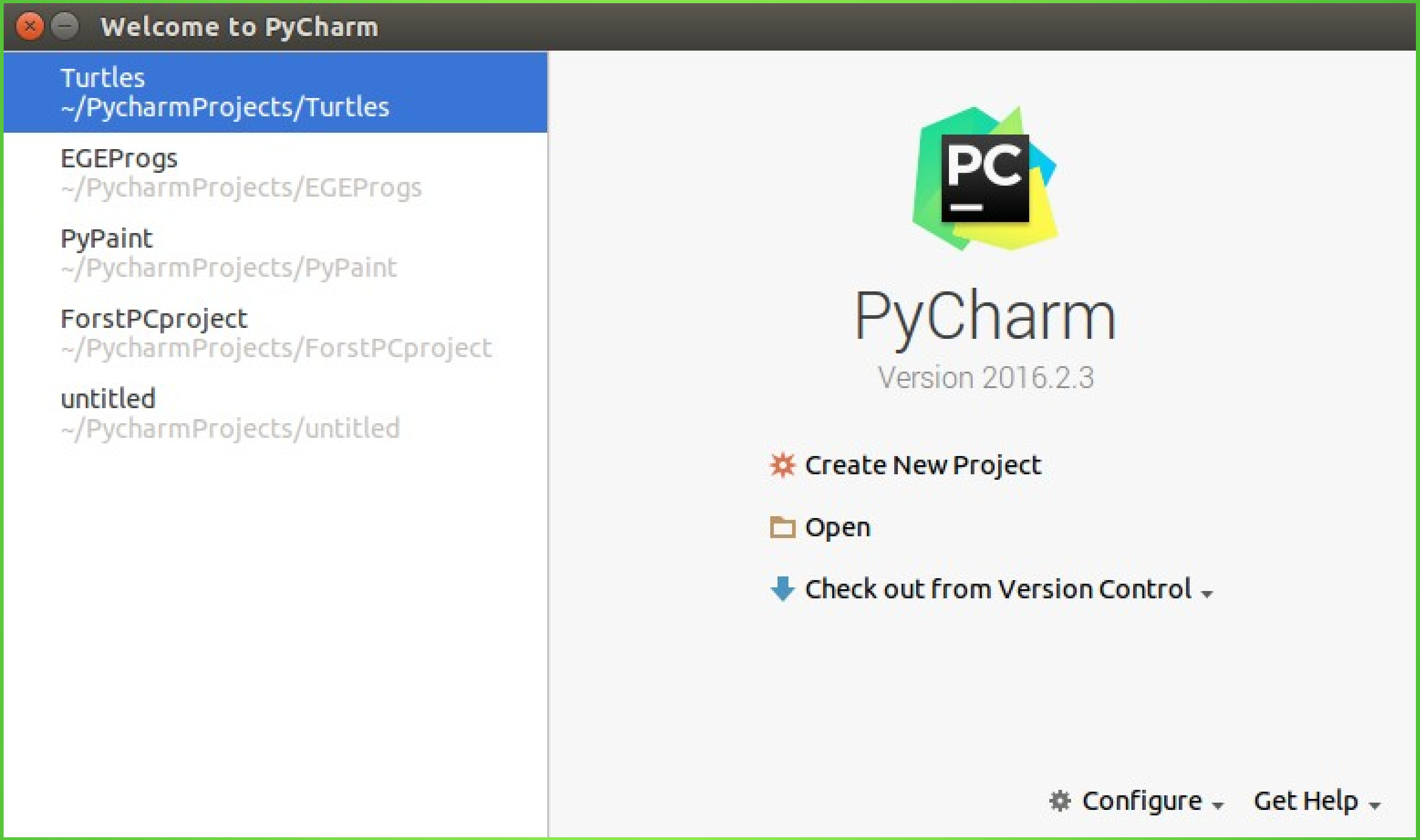
Среда разработки PyCharm

Урок 1

Стартовый экран



Welcome to PyCharm

Turtles
~/PycharmProjects/Turtles

EGEProgs
~/PycharmProjects/EGEProgs

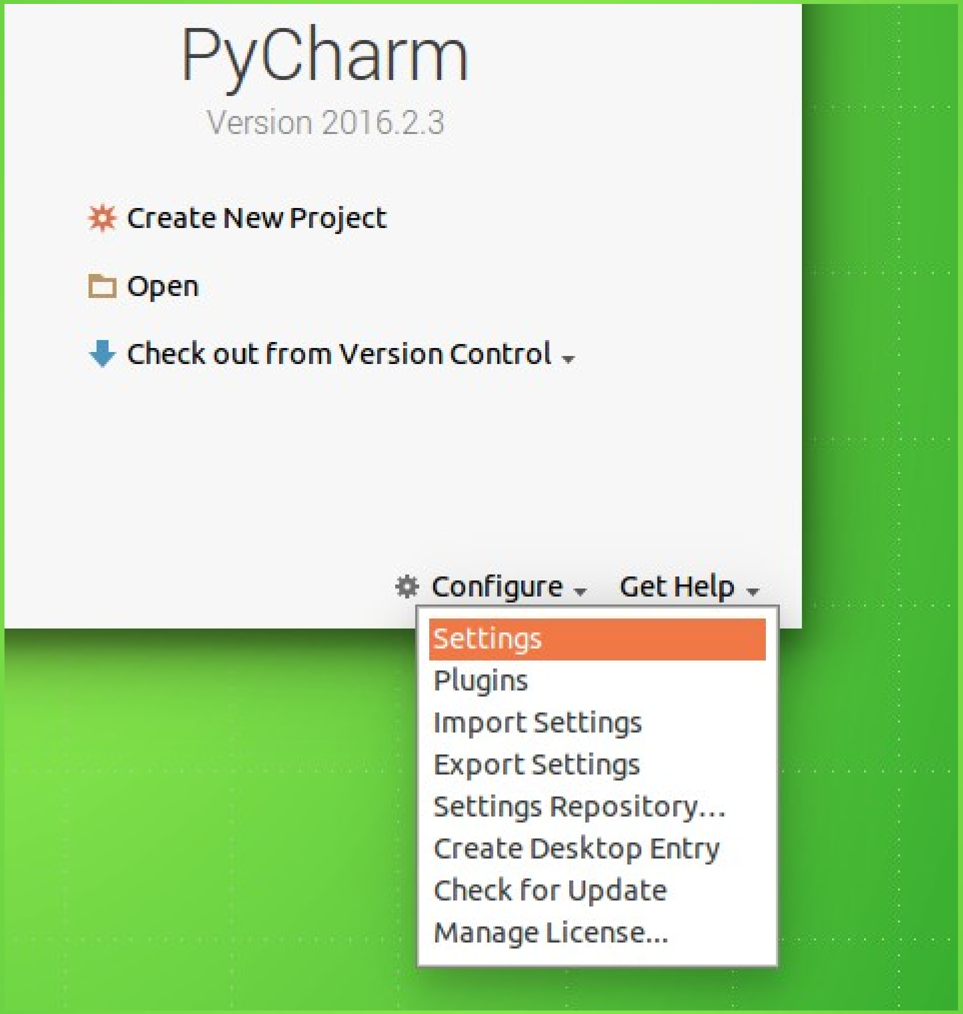
PyPaint
~/PycharmProjects/PyPaint

ForstPCproject
~/PycharmProjects/ForstPCproject

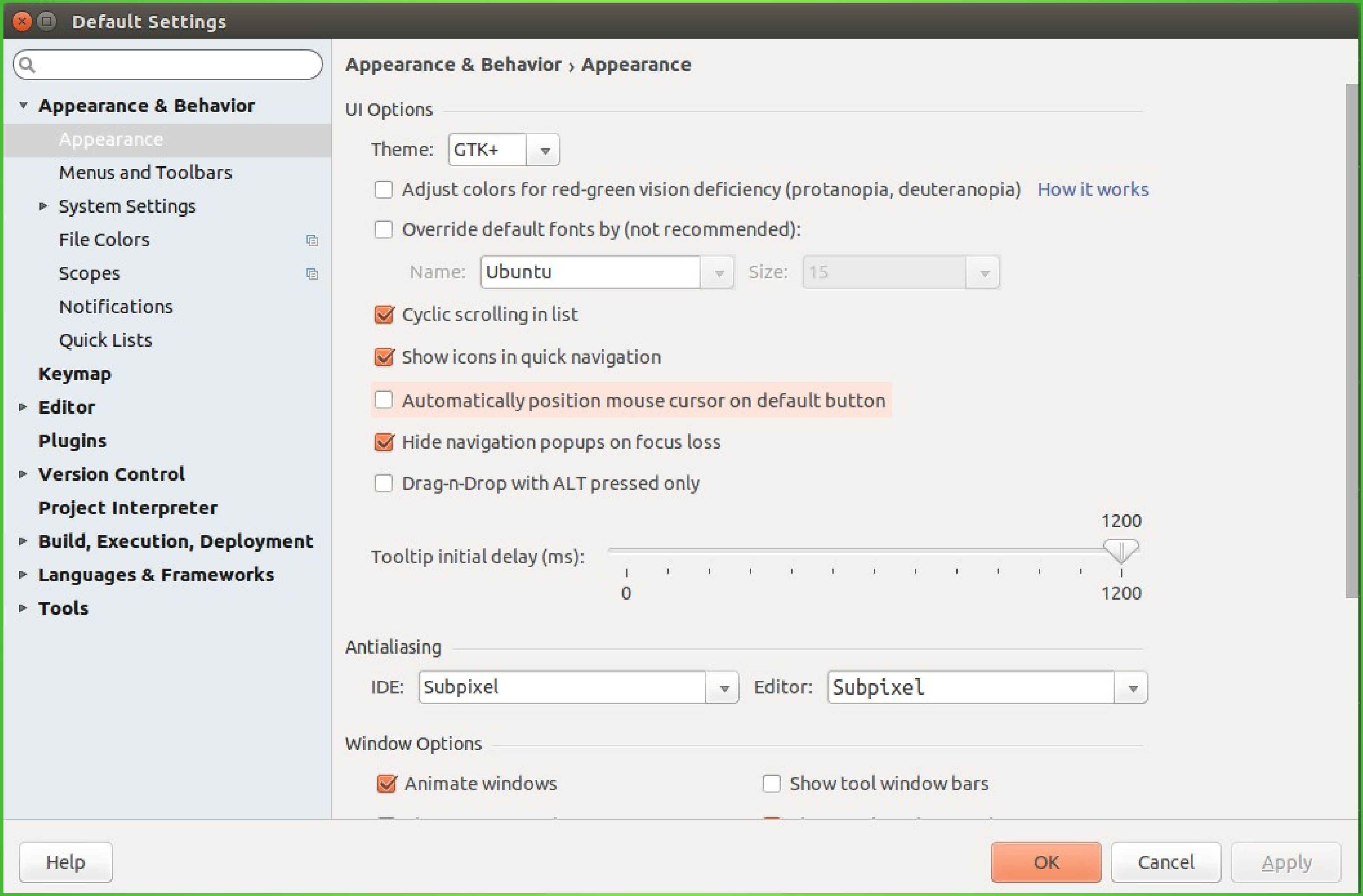
untitled
~/PycharmProjects/untitled

PyCharm
Version 2016.2.3

Установки / Settings



Окно установок



Default Settings

- ▾ Appearance & Behavior
  - Appearance
  - Menus and Toolbars
  - ▸ System Settings
  - File Colors
  - Scopes
  - Notifications
  - Quick Lists
- Keymap
- ▸ Editor
- Plugins
- ▸ Version Control
- Project Interpreter
- ▸ Build, Execution, Deployment
- ▸ Languages & Frameworks
- ▸ Tools

Appearance & Behavior › Appearance

UI Options

Theme: GTK+

□ Adjust colors for red-green vision deficiency (protanopia, deuteranopia) How it works

□ Override default fonts by (not recommended):

Name: Ubuntu Size: 15

☑ Cyclic scrolling in list

☑ Show icons in quick navigation

□ Automatically position mouse cursor on default button

☑ Hide navigation popups on focus loss

□ Drag-n-Drop with ALT pressed only

Tooltip initial delay (ms): 1200 (0 – 1200)

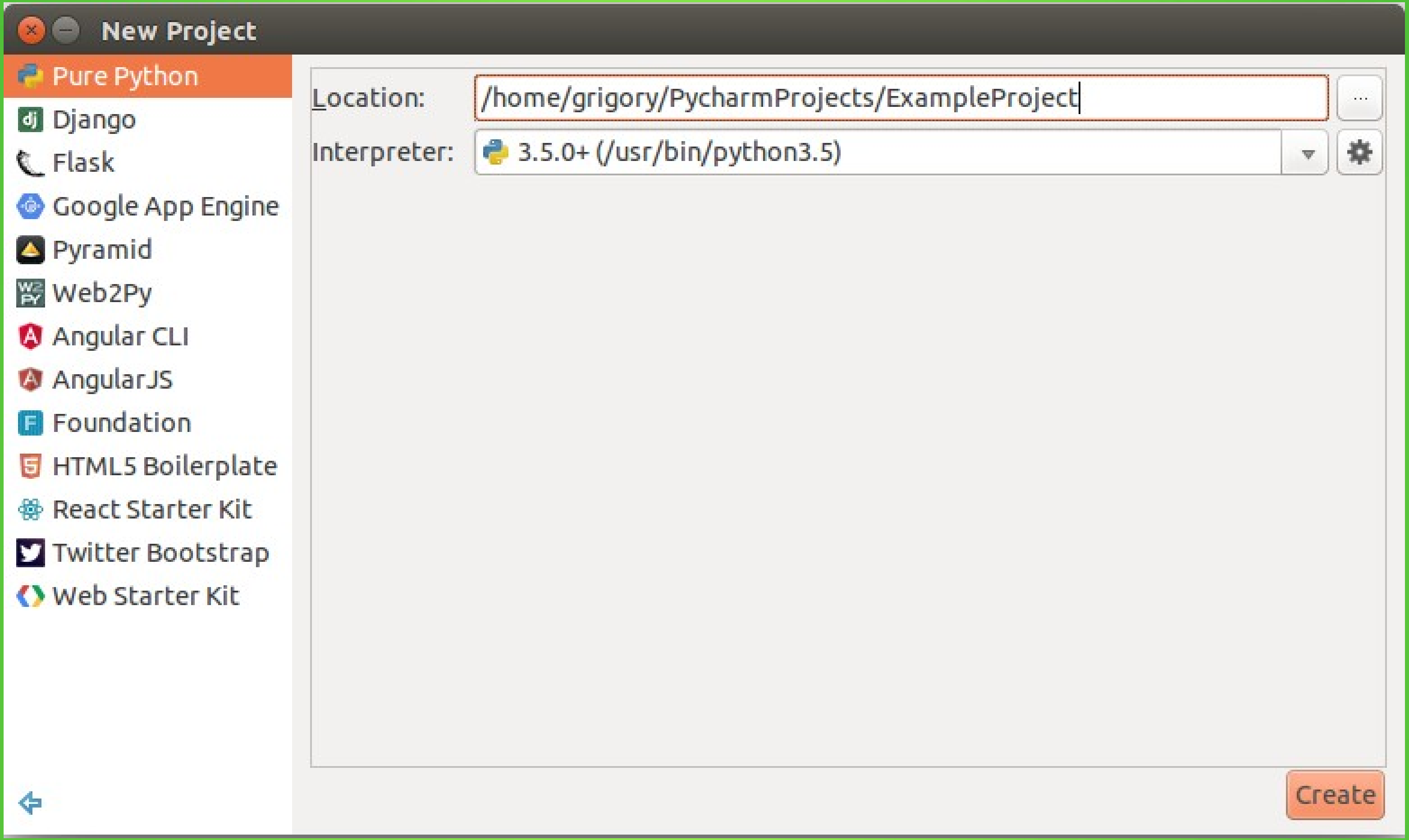
Antialiasing

IDE: Subpixel Editor: Subpixel

Window Options

☑ Animate windows □ Show tool window bars

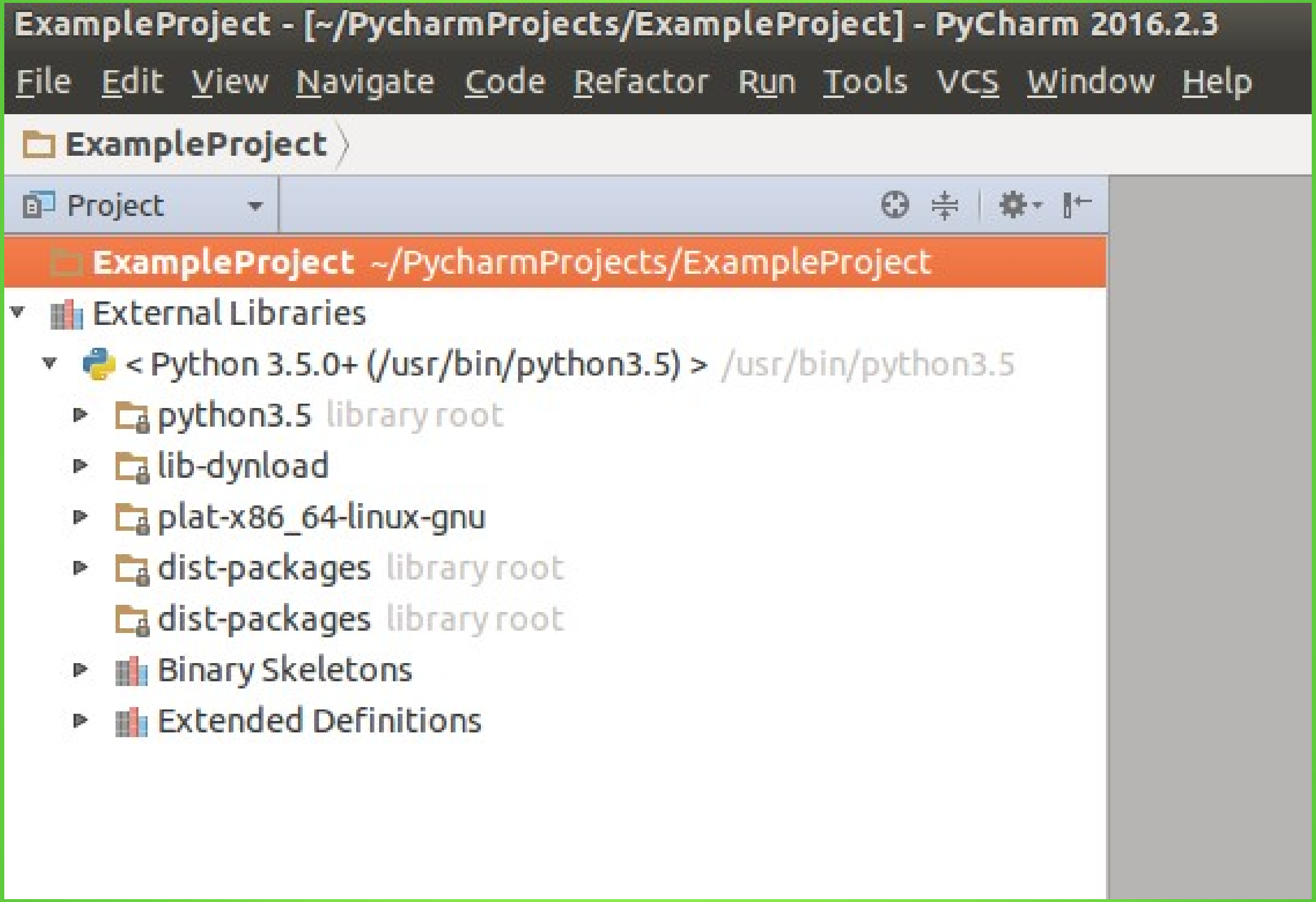
Новый проект



Location: /home/grigory/PycharmProjects/ExampleProject

Interpreter: 3.5.0+ (/usr/bin/python3.5)

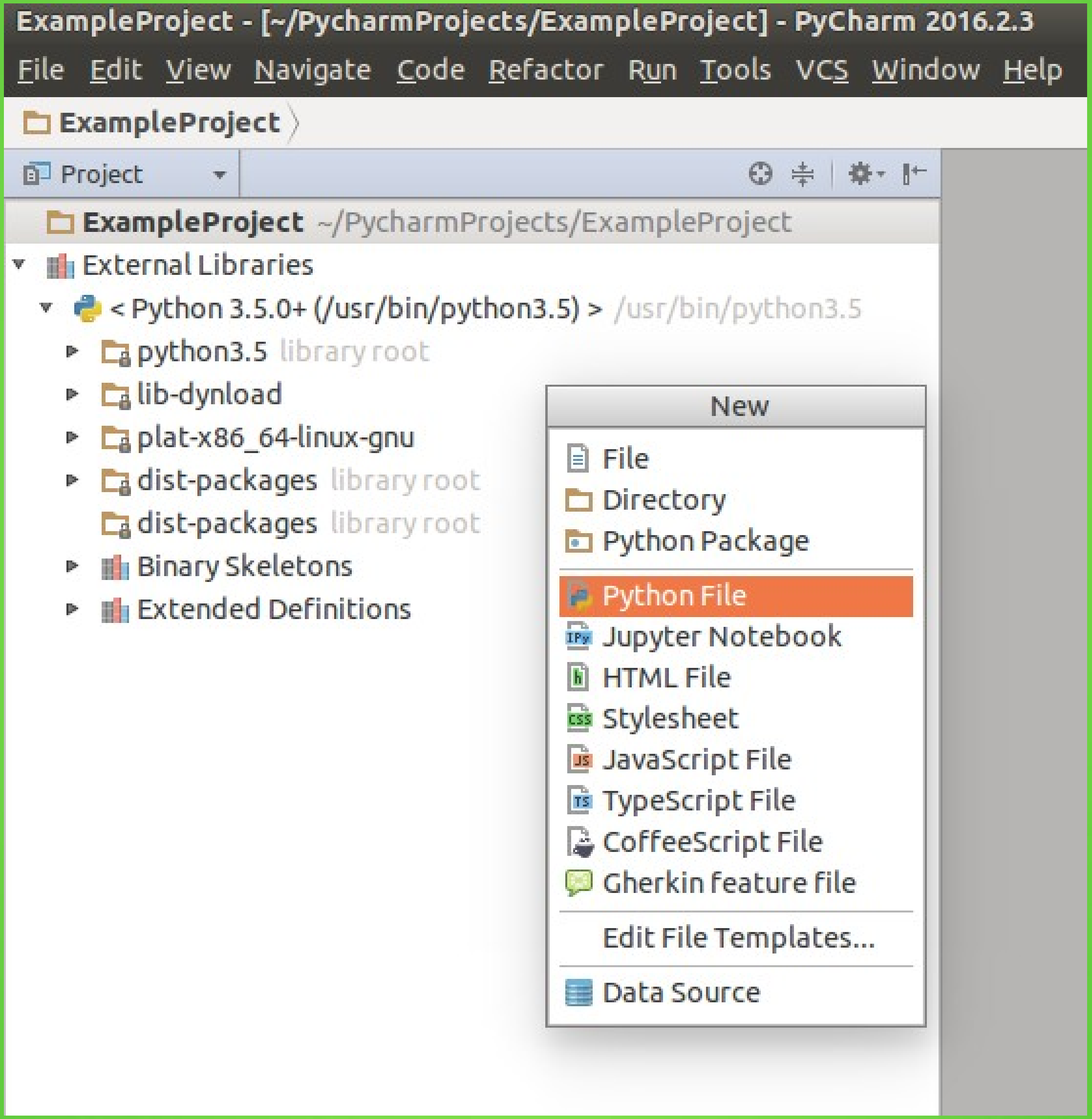
Структура проекта



ExampleProject - [~/PycharmProjects/ExampleProject] - PyCharm 2016.2.3
File Edit View Navigate Code Refactor Run Tools VCS Window Help
ExampleProject
Project

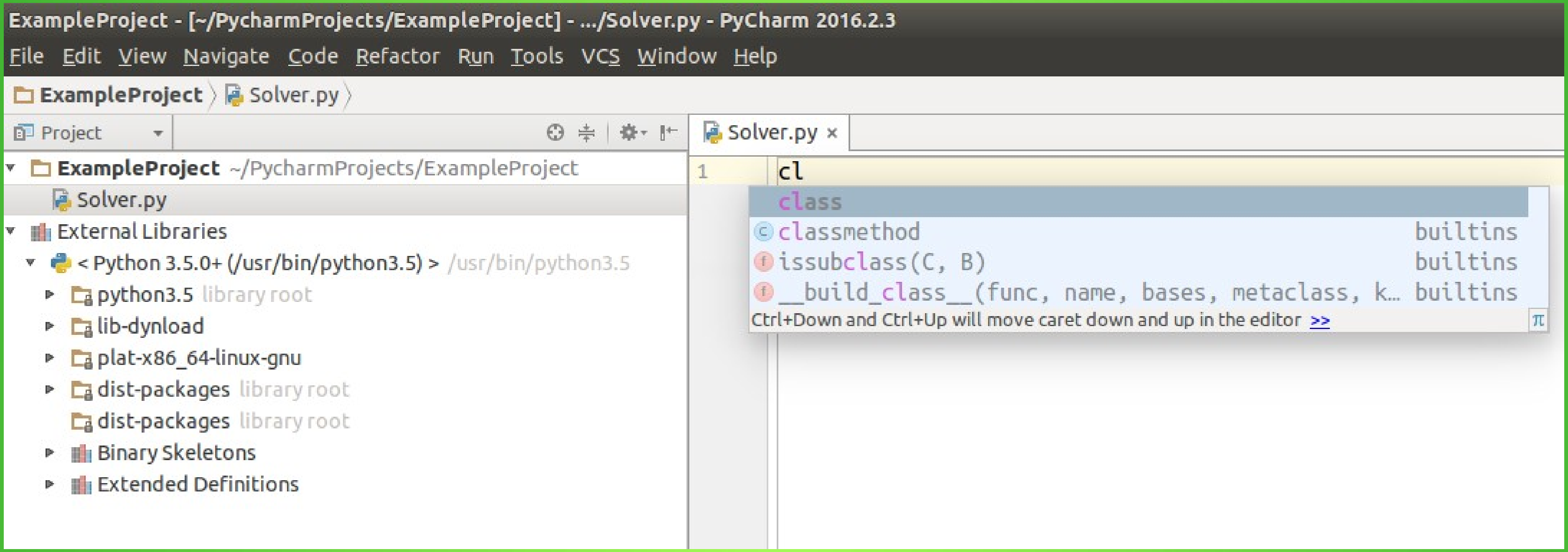
```
ExampleProject ~/PycharmProjects/ExampleProject
▾ External Libraries
  ▾ < Python 3.5.0+ (/usr/bin/python3.5) > /usr/bin/python3.5
    ▸ python3.5 library root
    ▸ lib-dynload
    ▸ plat-x86_64-linux-gnu
    ▸ dist-packages library root
      dist-packages library root
    ▸ Binary Skeletons
    ▸ Extended Definitions
```

Новый файл проекта



ExampleProject - [~/PycharmProjects/ExampleProject] - PyCharm 2016.2.3

Создание класса



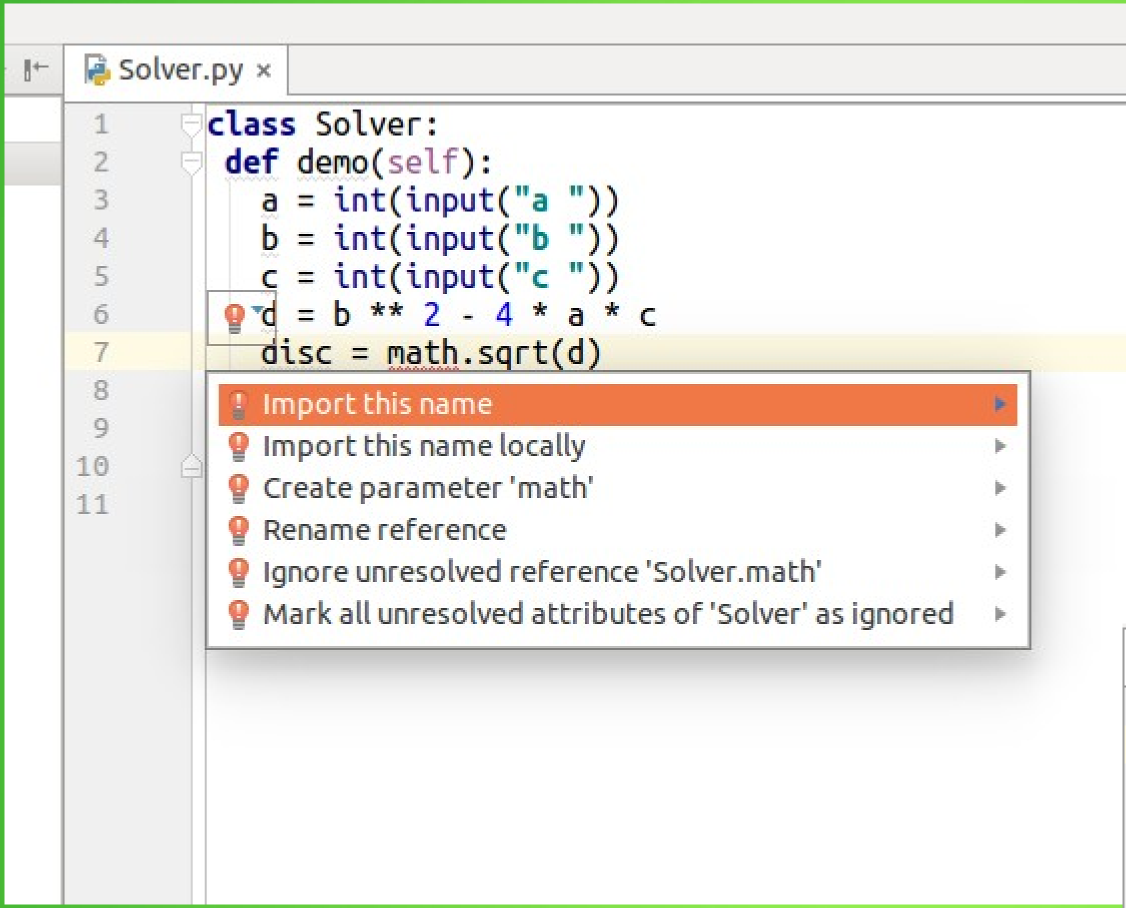
ExampleProject - [~/PycharmProjects/ExampleProject] - .../Solver.py - PyCharm 2016.2.3

File Edit View Navigate Code Refactor Run Tools VCS Window Help
ExampleProject 〉 Solver.py 〉
Project
ExampleProject ~/PycharmProjects/ExampleProject
Solver.py
External Libraries
< Python 3.5.0+ (/usr/bin/python3.5) > /usr/bin/python3.5
python3.5 library root
lib-dynload
plat-x86_64-linux-gnu
dist-packages library root
dist-packages library root
Binary Skeletons
Extended Definitions
Solver.py ×

```
cl
```

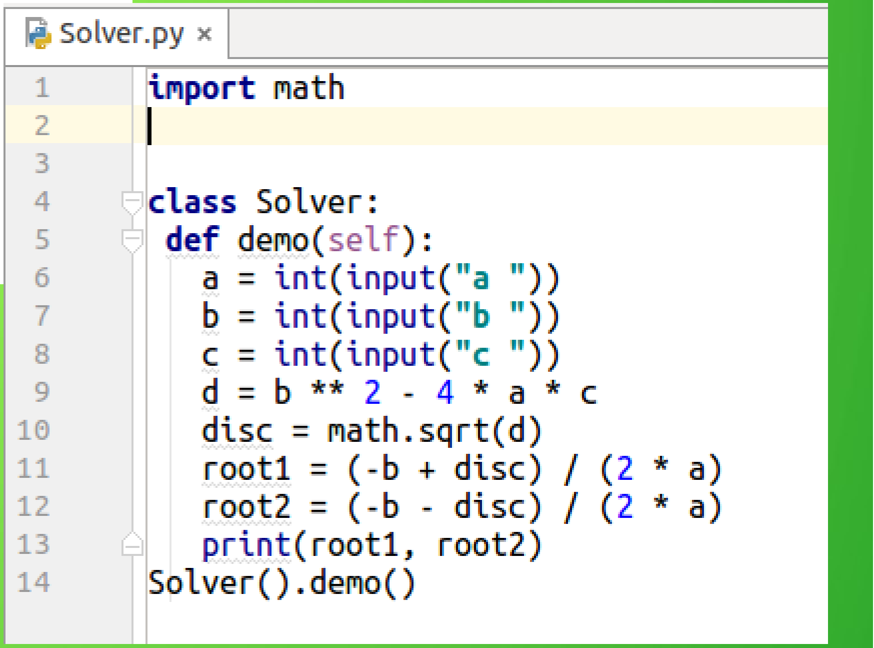
```
class
classmethod                                            builtins
issubclass(C, B)                                       builtins
__build_class__(func, name, bases, metaclass, k…       builtins
Ctrl+Down and Ctrl+Up will move caret down and up in the editor >>
```

Как исправить ошибку?


Solver.py ×
Import this name
Import this name locally
Create parameter 'math'
Rename reference
Ignore unresolved reference 'Solver.math'
Mark all unresolved attributes of 'Solver' as ignored

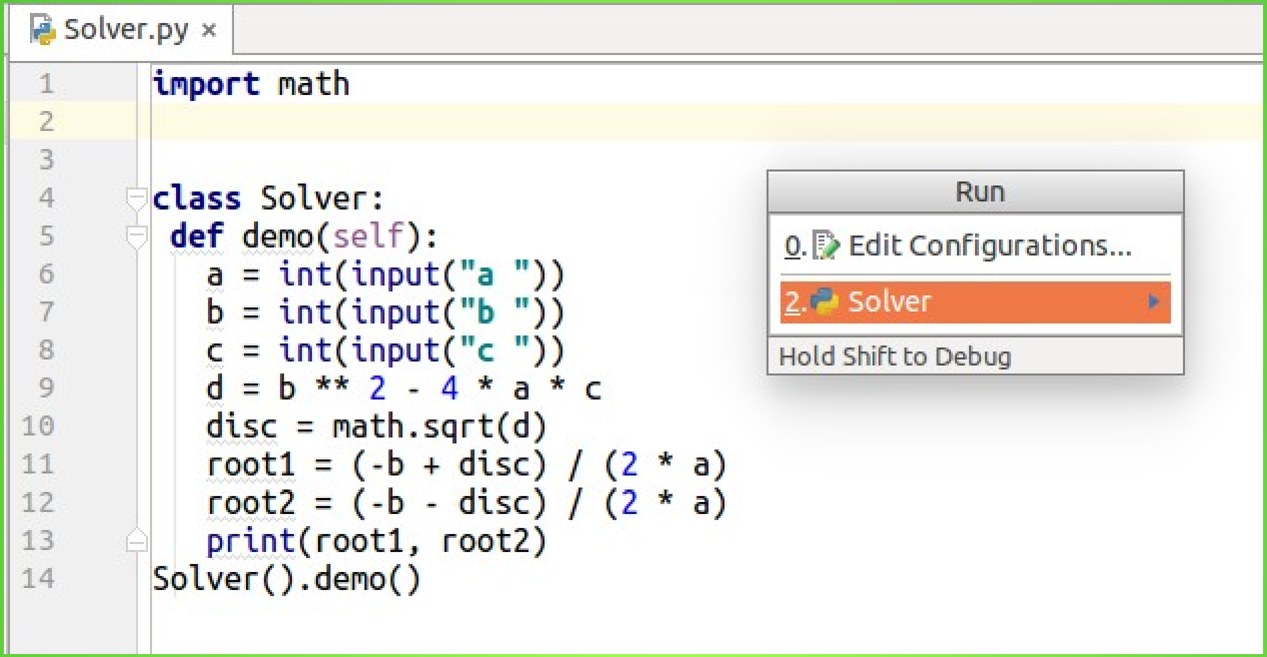
```
class Solver:
 def demo(self):
   a = int(input("a "))
   b = int(input("b "))
   c = int(input("c "))
   d = b ** 2 - 4 * a * c
   disc = math.sqrt(d)
```

```
class Solver:
 def Calculate(self):
  while True:
    a = int(input("a "))
    b = int(input("b "))
    c = int(input("c "))
    d = b ** 2 - 4 * a * c
    if d >= 0:
      disc = math.sqrt(d)
      root1 = (-b + disc) / (2 * a)
      root2 = (-b - disc) / (2 * a)
      print(root1, root2)
    else:
      print('Error')

Solver().Calculate()
```



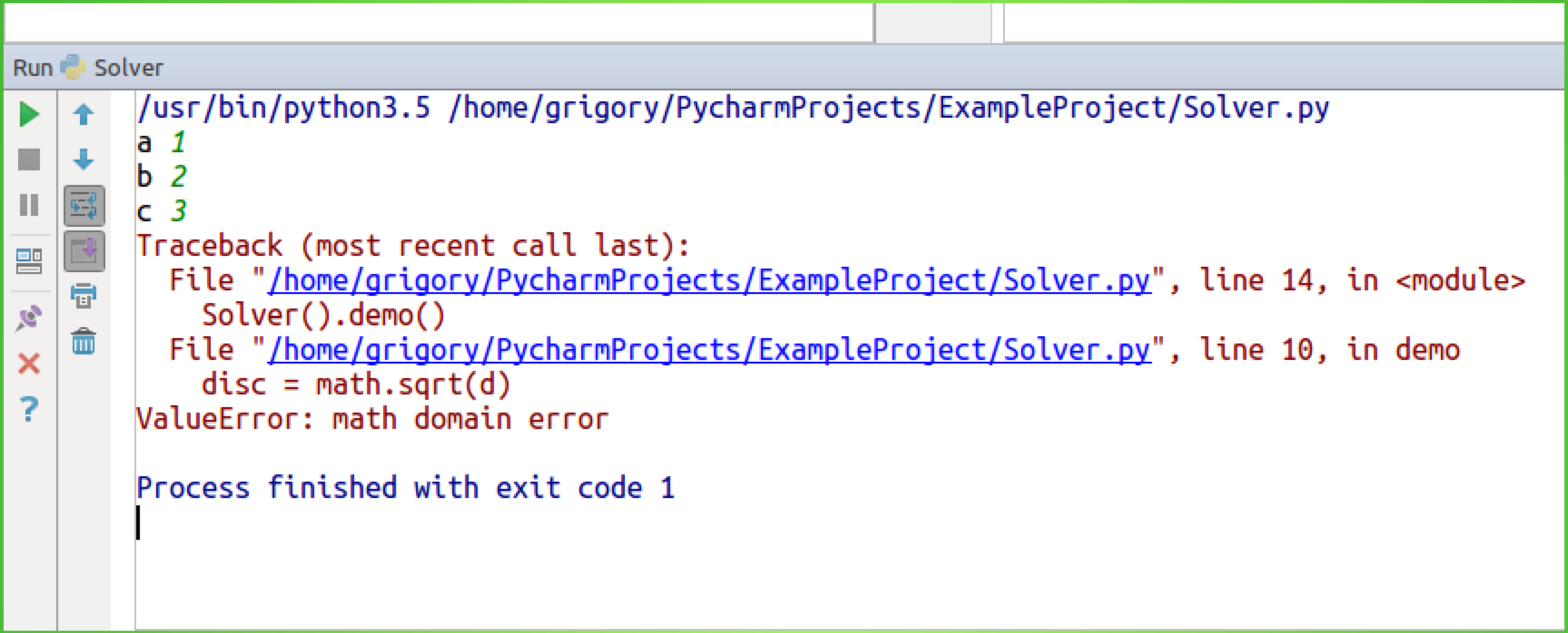
```
import math


class Solver:
 def demo(self):
   a = int(input("a "))
   b = int(input("b "))
   c = int(input("c "))
   d = b ** 2 - 4 * a * c
   disc = math.sqrt(d)
   root1 = (-b + disc) / (2 * a)
   root2 = (-b - disc) / (2 * a)
   print(root1, root2)
Solver().demo()
```

Запуск...



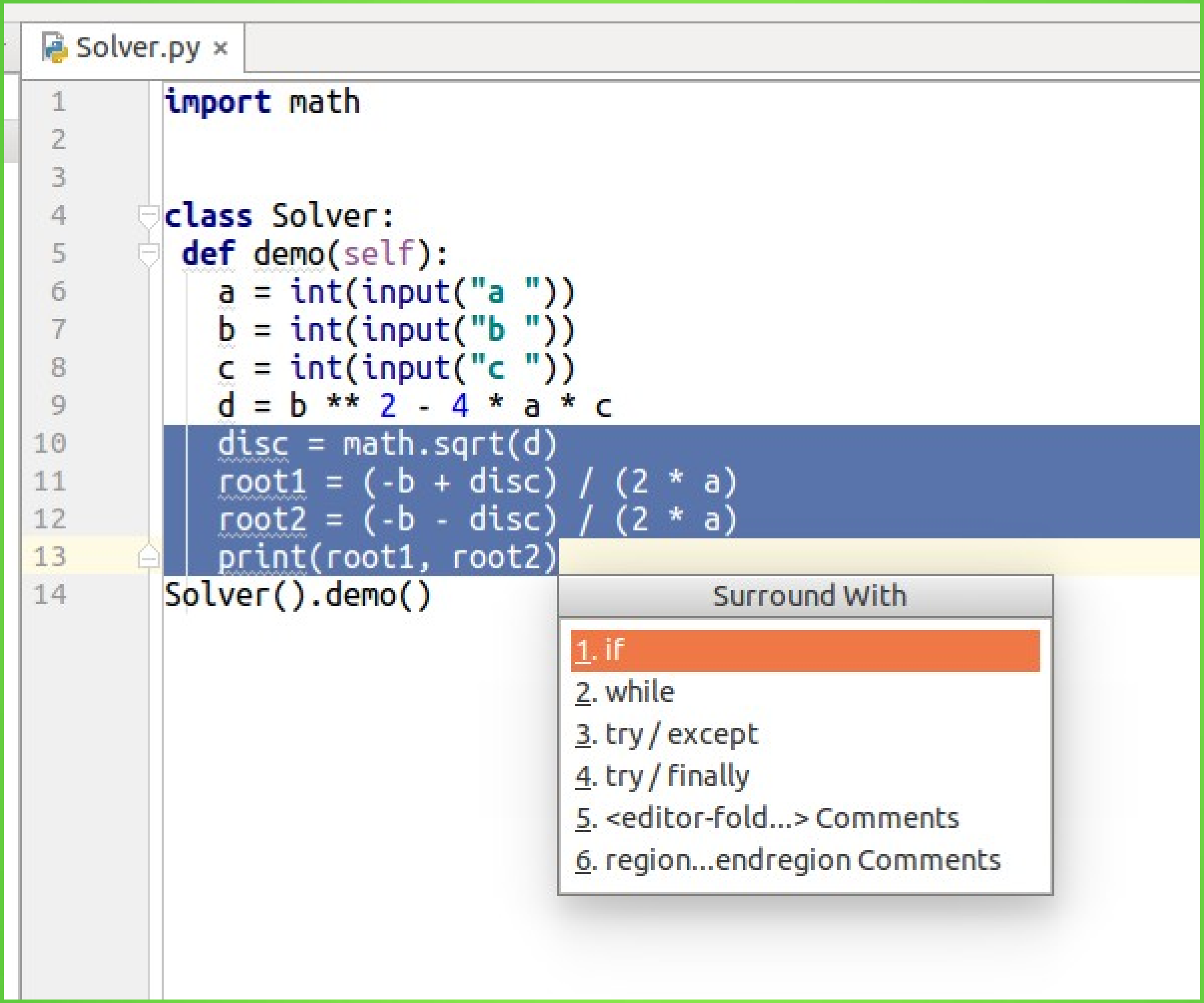
```
import math


class Solver:
 def demo(self):
   a = int(input("a "))
   b = int(input("b "))
   c = int(input("c "))
   d = b ** 2 - 4 * a * c
   disc = math.sqrt(d)
   root1 = (-b + disc) / (2 * a)
   root2 = (-b - disc) / (2 * a)
   print(root1, root2)
Solver().demo()
```

Ошибка... Корень из отрицательного...



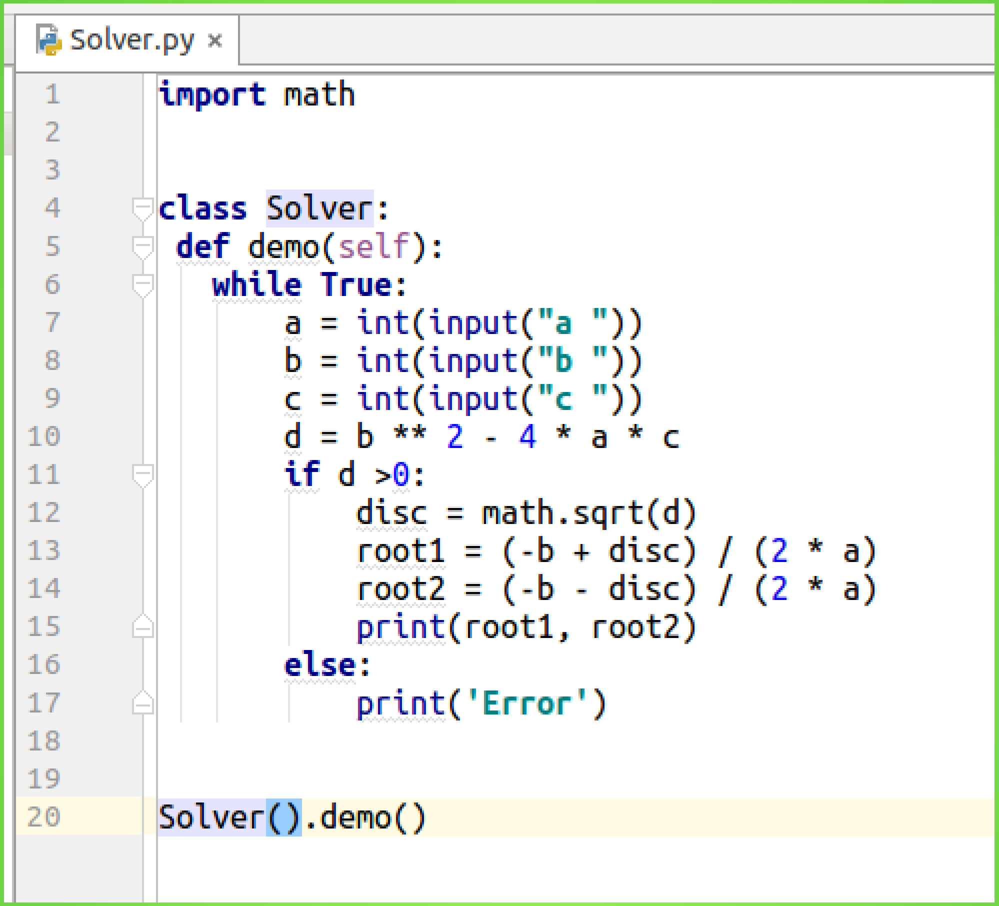
```
/usr/bin/python3.5 /home/grigory/PycharmProjects/ExampleProject/Solver.py
a 1
b 2
c 3
Traceback (most recent call last):
  File "/home/grigory/PycharmProjects/ExampleProject/Solver.py", line 14, in <module>
    Solver().demo()
  File "/home/grigory/PycharmProjects/ExampleProject/Solver.py", line 10, in demo
    disc = math.sqrt(d)
ValueError: math domain error

Process finished with exit code 1
```

Code → Surround with



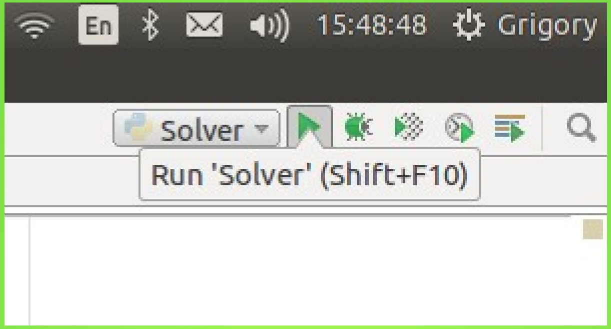
```
import math


class Solver:
 def demo(self):
   a = int(input("a "))
   b = int(input("b "))
   c = int(input("c "))
   d = b ** 2 - 4 * a * c
   disc = math.sqrt(d)
   root1 = (-b + disc) / (2 * a)
   root2 = (-b - disc) / (2 * a)
   print(root1, root2)
Solver().demo()
```

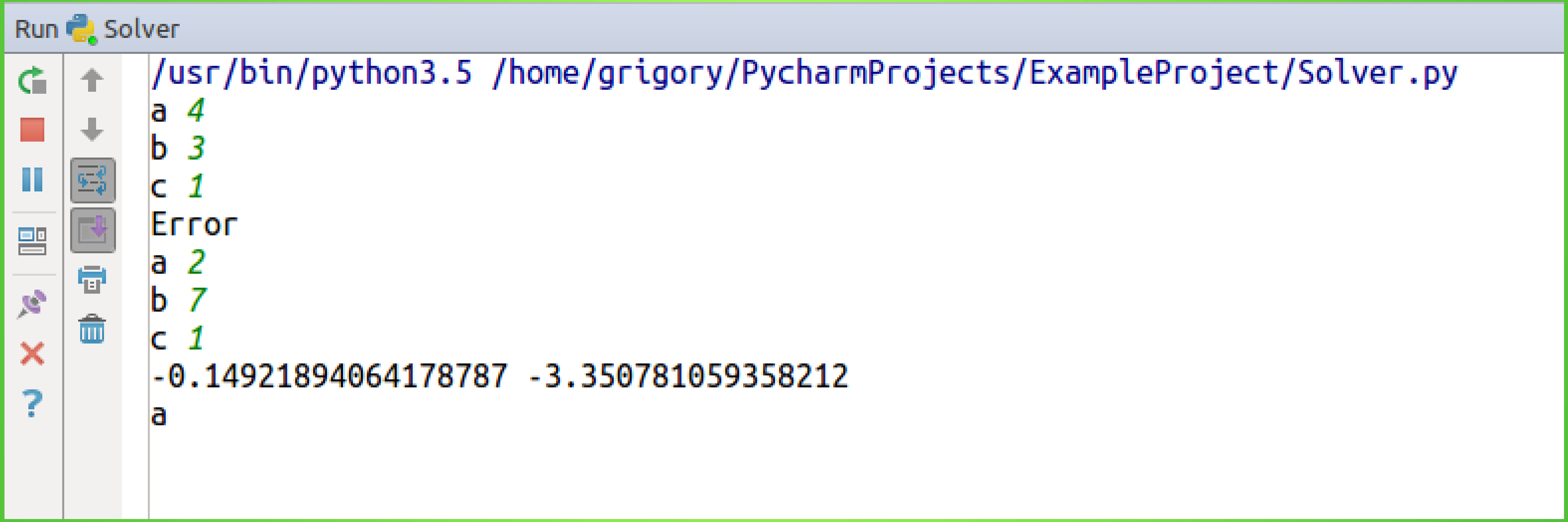
Surround… И снова Surround



```
import math


class Solver:
 def demo(self):
   while True:
       a = int(input("a "))
       b = int(input("b "))
       c = int(input("c "))
       d = b ** 2 - 4 * a * c
       if d >0:
           disc = math.sqrt(d)
           root1 = (-b + disc) / (2 * a)
           root2 = (-b - disc) / (2 * a)
           print(root1, root2)
       else:
           print('Error')


Solver().demo()
```

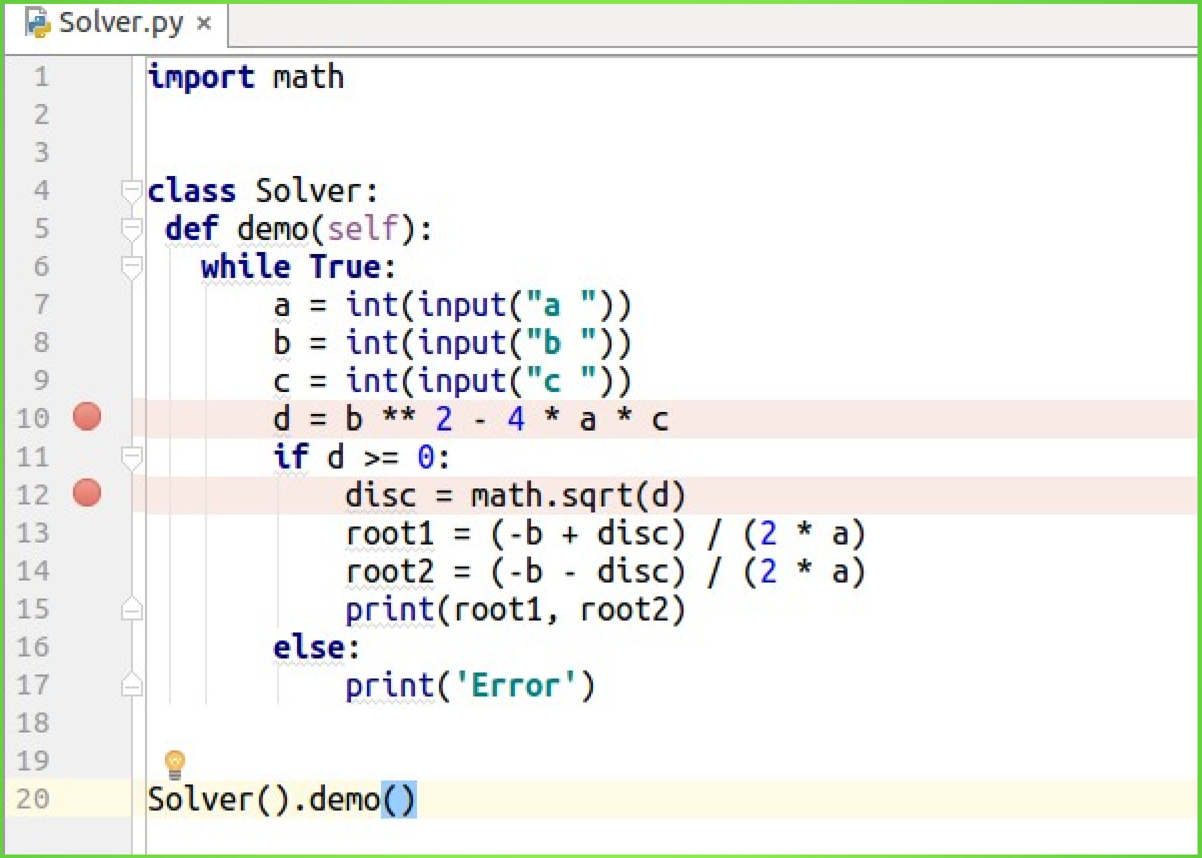
Запуск





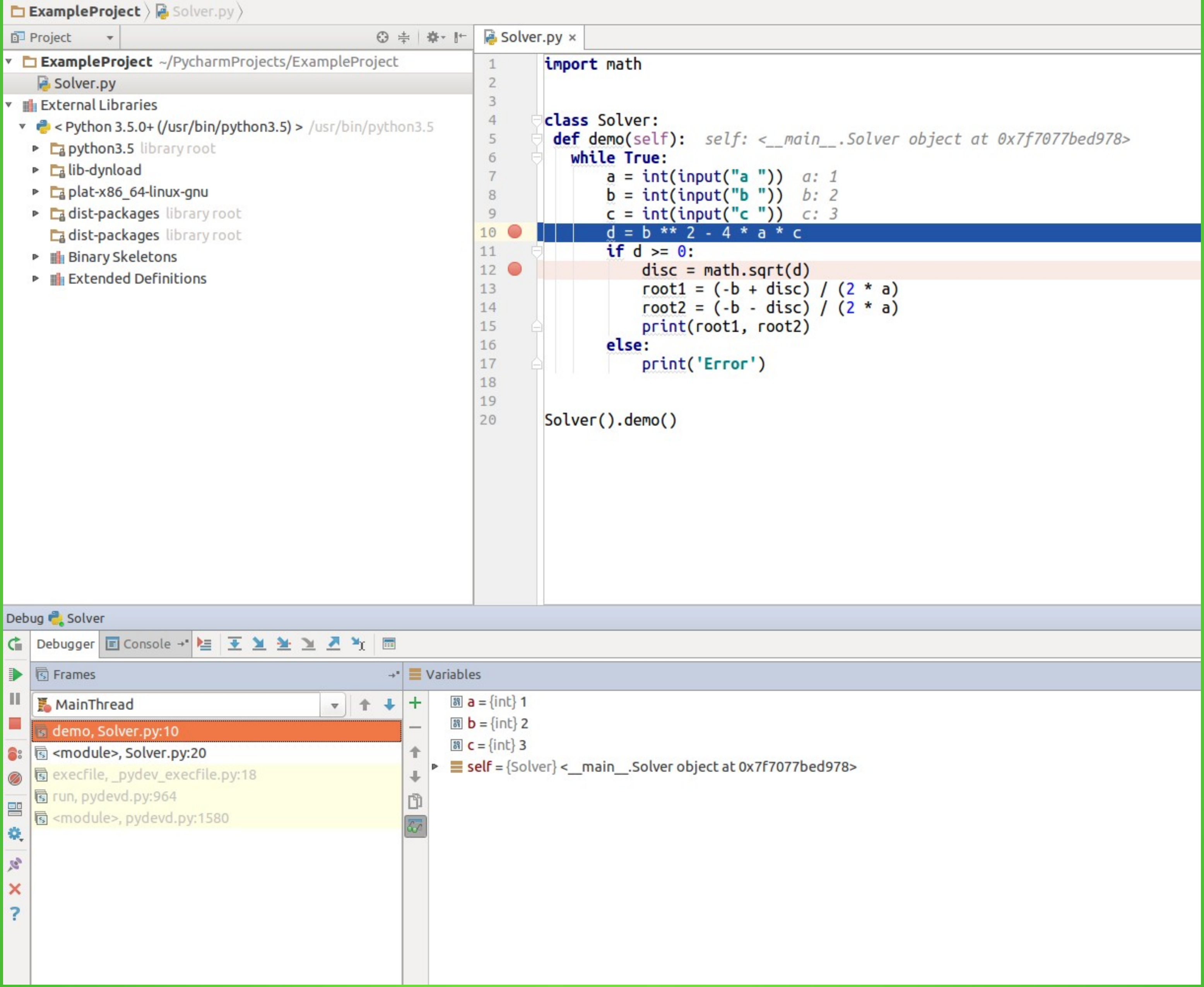
```
/usr/bin/python3.5 /home/grigory/PycharmProjects/ExampleProject/Solver.py
a 4
b 3
c 1
Error
a 2
b 7
c 1
-0.14921894064178787 -3.350781059358212
a
```

Отладка. Точки останова



```
import math


class Solver:
 def demo(self):
   while True:
       a = int(input("a "))
       b = int(input("b "))
       c = int(input("c "))
       d = b ** 2 - 4 * a * c
       if d >= 0:
           disc = math.sqrt(d)
           root1 = (-b + disc) / (2 * a)
           root2 = (-b - disc) / (2 * a)
           print(root1, root2)
       else:
           print('Error')


Solver().demo()
```

Отладка. Debugging...

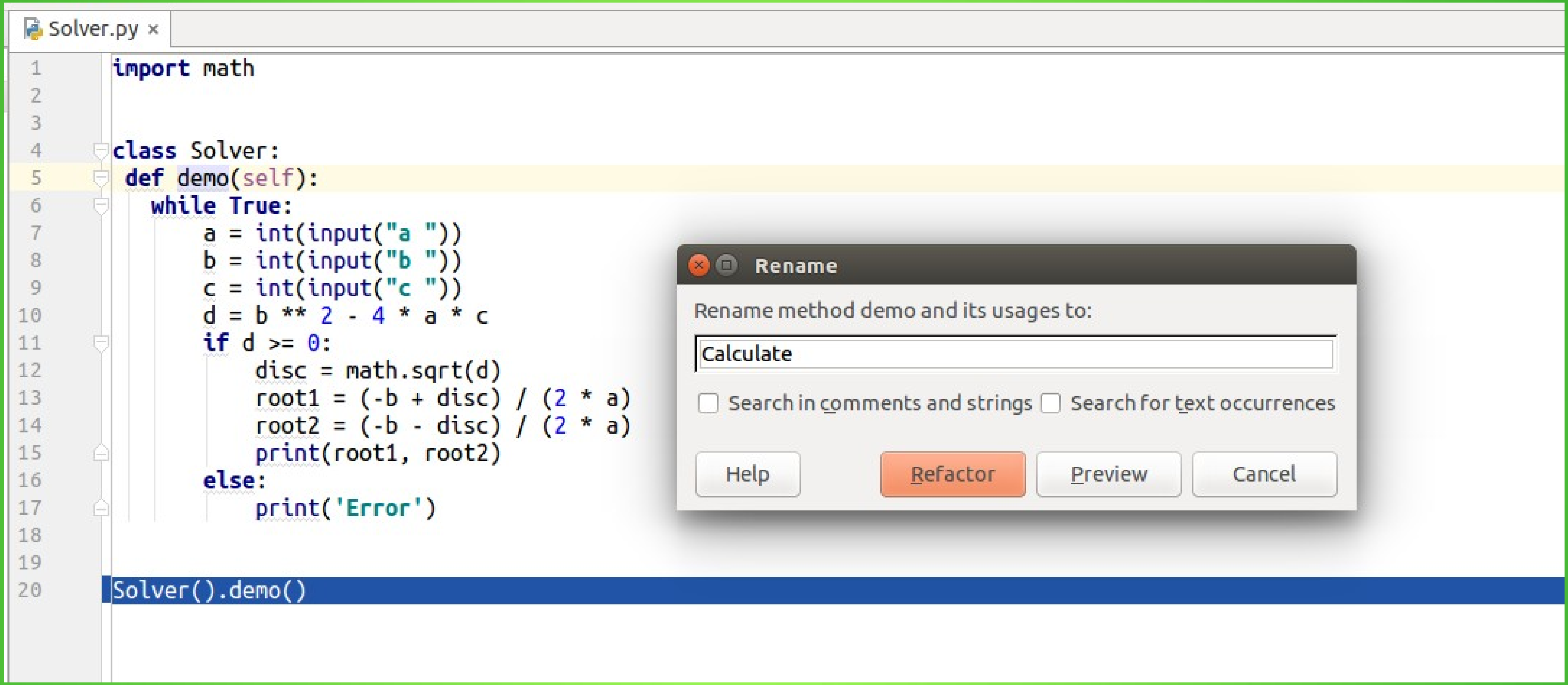


```
import math


class Solver:
 def demo(self):  self: <__main__.Solver object at 0x7f7077bed978>
   while True:
       a = int(input("a "))  a: 1
       b = int(input("b "))  b: 2
       c = int(input("c "))  c: 3
       d = b ** 2 - 4 * a * c
       if d >= 0:
           disc = math.sqrt(d)
           root1 = (-b + disc) / (2 * a)
           root2 = (-b - disc) / (2 * a)
           print(root1, root2)
       else:
           print('Error')


Solver().demo()
```

```
demo, Solver.py:10
<module>, Solver.py:20
execfile, _pydev_execfile.py:18
run, pydevd.py:964
<module>, pydevd.py:1580
```

```
a = {int} 1
b = {int} 2
c = {int} 3
self = {Solver} <__main__.Solver object at 0x7f7077bed978>
```

Refactor → Rename



```
import math


class Solver:
 def demo(self):
   while True:
       a = int(input("a "))
       b = int(input("b "))
       c = int(input("c "))
       d = b ** 2 - 4 * a * c
       if d >= 0:
           disc = math.sqrt(d)
           root1 = (-b + disc) / (2 * a)
           root2 = (-b - disc) / (2 * a)
           print(root1, root2)
       else:
           print('Error')


Solver().demo()
```

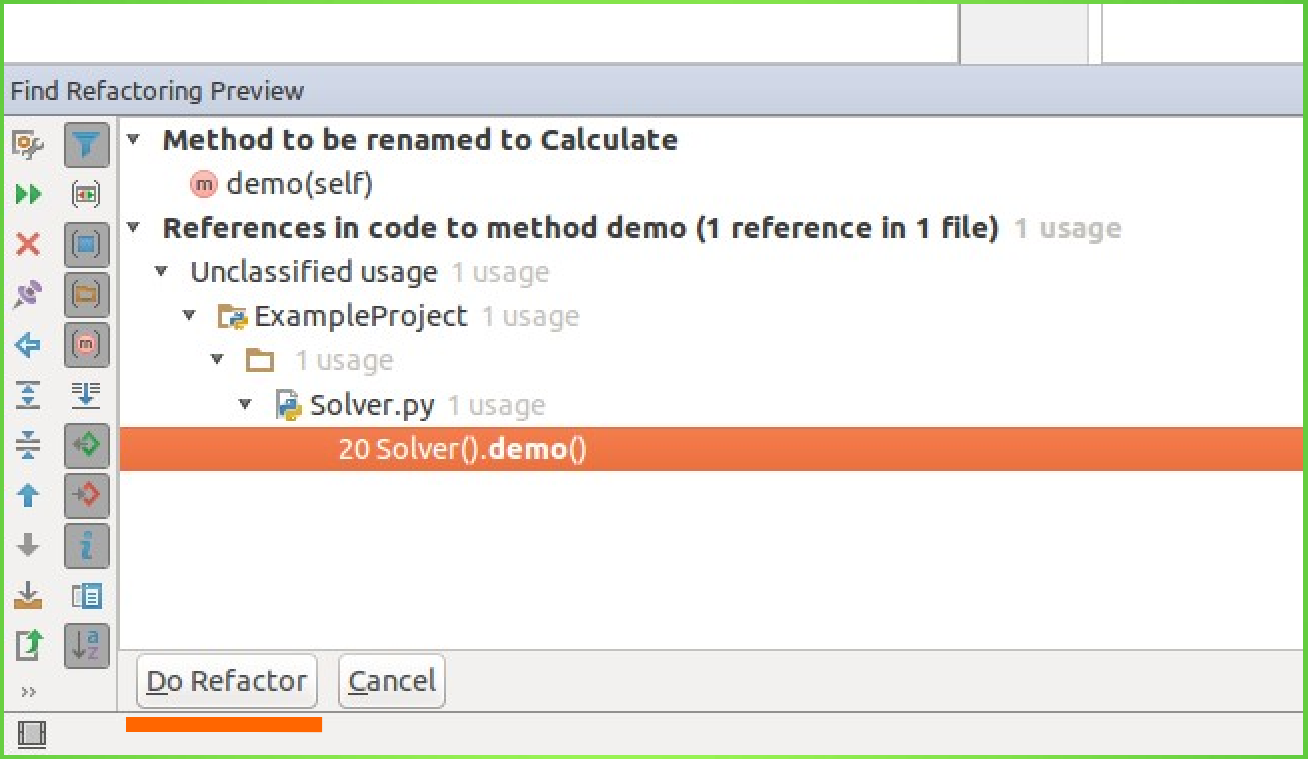
Rename

Rename method demo and its usages to:

Calculate

□ Search in comments and strings □ Search for text occurrences

Refactor → Rename



Find Refactoring Preview

- Method to be renamed to Calculate
  - demo(self)
- References in code to method demo (1 reference in 1 file) 1 usage
  - Unclassified usage 1 usage
    - ExampleProject 1 usage
      - 1 usage
        - Solver.py 1 usage
          - 20 Solver().demo()

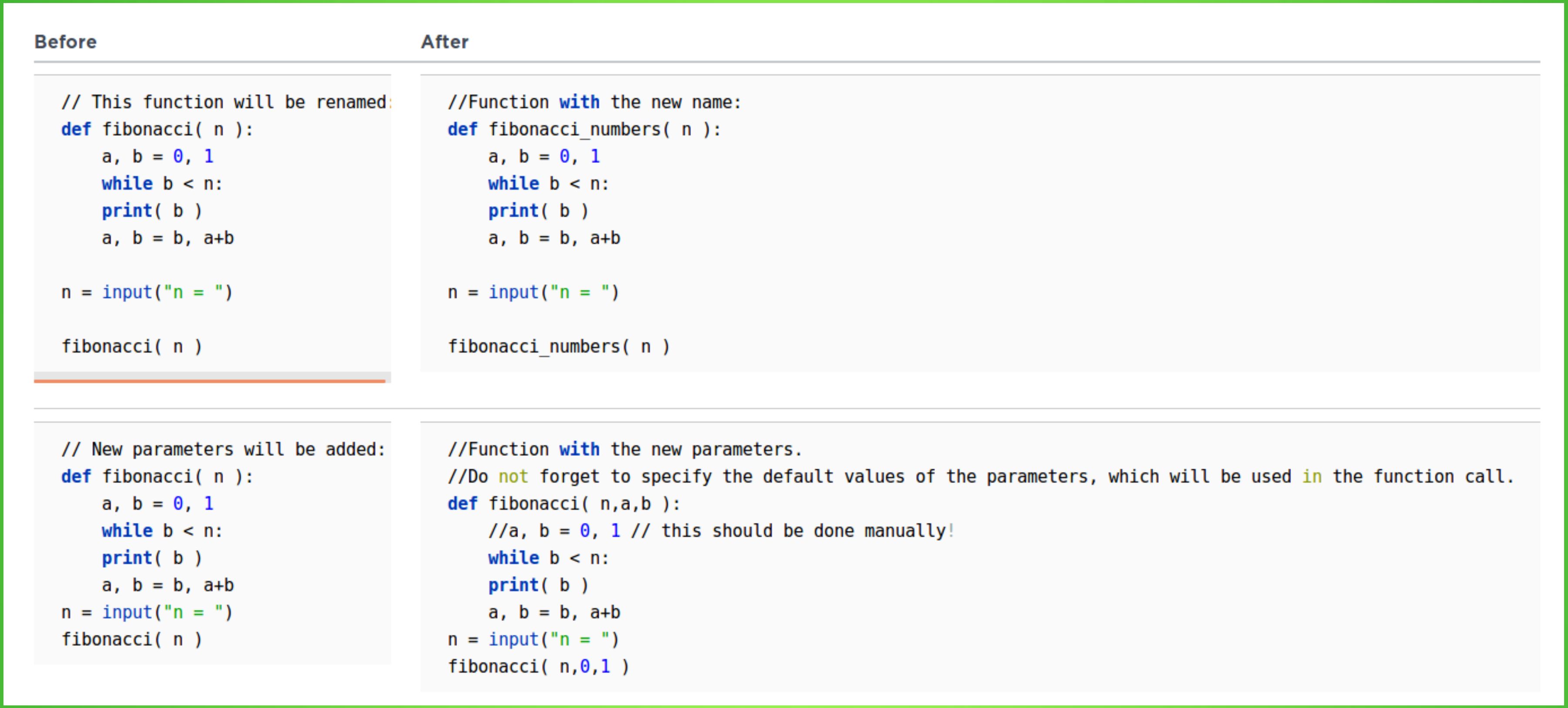
Изменить заголовок функции

Команда Change Signature (Refactoring) объединяет несколько различных модификаций, которые применяются к заголовку функции. Инструмент можно использоваться для:

- Изменения имени функции
- Добавления новых параметров и удаления существующих
- Присвоения параметрам значений по умолчанию
- Изменения порядка следования параметров

При изменении заголовка функции PyCharm ищет все обращения к данной функции и переформатирует их.

Изменить заголовок функции



Before

```
// This function will be renamed:
def fibonacci( n ):
    a, b = 0, 1
    while b < n:
    print( b )
    a, b = b, a+b

n = input("n = ")

fibonacci( n )
```

After

```
//Function with the new name:
def fibonacci_numbers( n ):
    a, b = 0, 1
    while b < n:
    print( b )
    a, b = b, a+b

n = input("n = ")

fibonacci_numbers( n )
```

Before

```
// New parameters will be added:
def fibonacci( n ):
    a, b = 0, 1
    while b < n:
    print( b )
    a, b = b, a+b
n = input("n = ")
fibonacci( n )
```

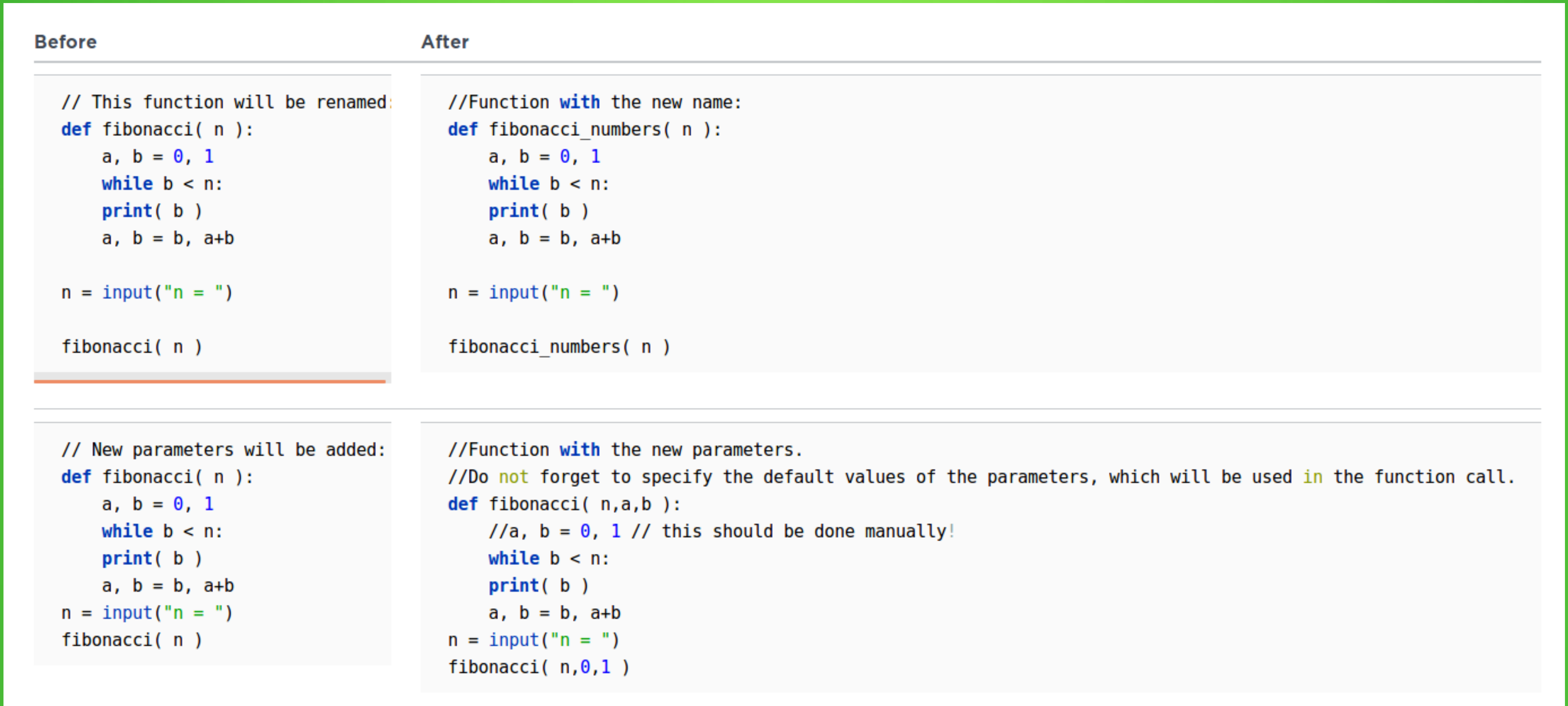
After

```
//Function with the new parameters.
//Do not forget to specify the default values of the parameters, which will be used in the function call.
def fibonacci( n,a,b ):
    //a, b = 0, 1 // this should be done manually!
    while b < n:
    print( b )
    a, b = b, a+b
n = input("n = ")
fibonacci( n,0,1 )
```

```
// This function will be renamed:
def fibonacci( n ):
    a, b = 0, 1
    while b < n:
    print( b )
    a, b = b, a+b

n = input("n = ")

fibonacci( n )
```

```
//Function with the new name:
def fibonacci_numbers( n ):
    a, b = 0, 1
    while b < n:
    print( b )
    a, b = b, a+b

n = input("n = ")

fibonacci_numbers( n )
```

Изменить заголовок функции



Before **After**

```
// This function will be renamed:
def fibonacci( n ):
    a, b = 0, 1
    while b < n:
    print( b )
    a, b = b, a+b

n = input("n = ")

fibonacci( n )
```

```
//Function with the new name:
def fibonacci_numbers( n ):
    a, b = 0, 1
    while b < n:
    print( b )
    a, b = b, a+b

n = input("n = ")

fibonacci_numbers( n )
```

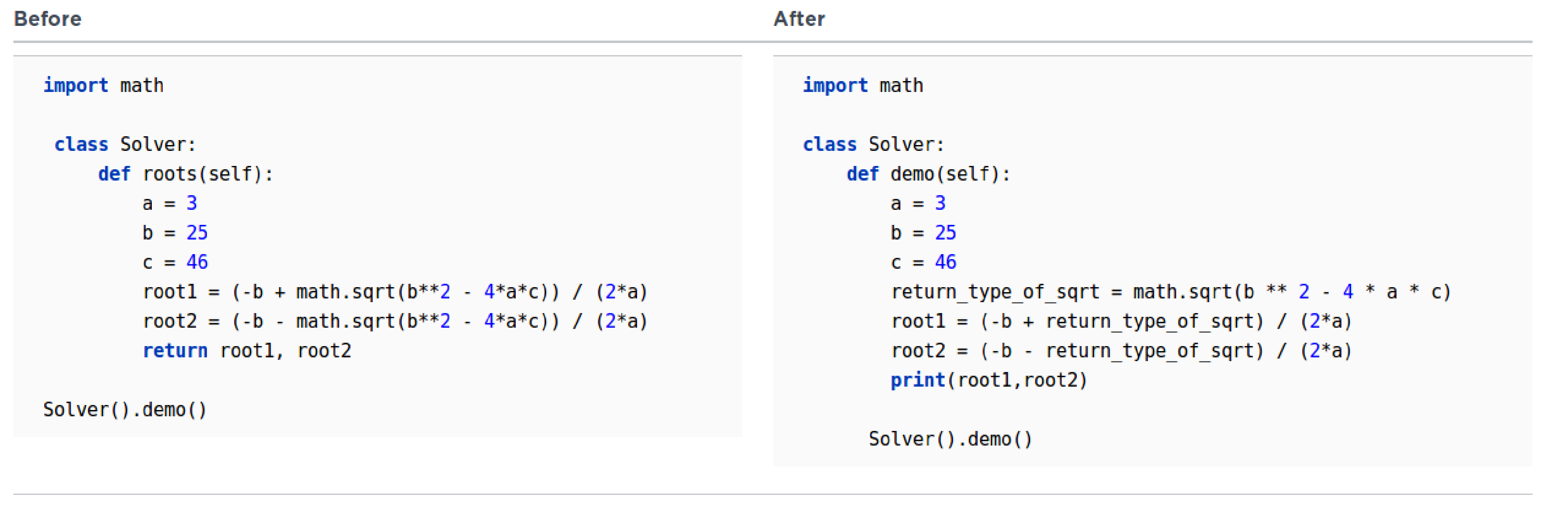
```
// New parameters will be added:
def fibonacci( n ):
    a, b = 0, 1
    while b < n:
    print( b )
    a, b = b, a+b
n = input("n = ")
fibonacci( n )
```

```
//Function with the new parameters.
//Do not forget to specify the default values of the parameters, which will be used in the function call.
def fibonacci( n,a,b ):
    //a, b = 0, 1 // this should be done manually!
    while b < n:
    print( b )
    a, b = b, a+b
n = input("n = ")
fibonacci( n,0,1 )
```

```
// New parameters will be added:
def fibonacci( n ):
    a, b = 0, 1
    while b < n:
    print( b )
    a, b = b, a+b
n = input("n = ")
fibonacci( n )
```

```
//Function with the new parameters.
//Do not forget to specify the default values of the parameters, which will be
used in the function call.
def fibonacci( n,a,b ):
    //a, b = 0, 1 // this should be done manually!
    while b < n:
    print( b )
    a, b = b, a+b
n = input("n = ")
fibonacci( n,0,1 )
```

Извлечь переменную



Before

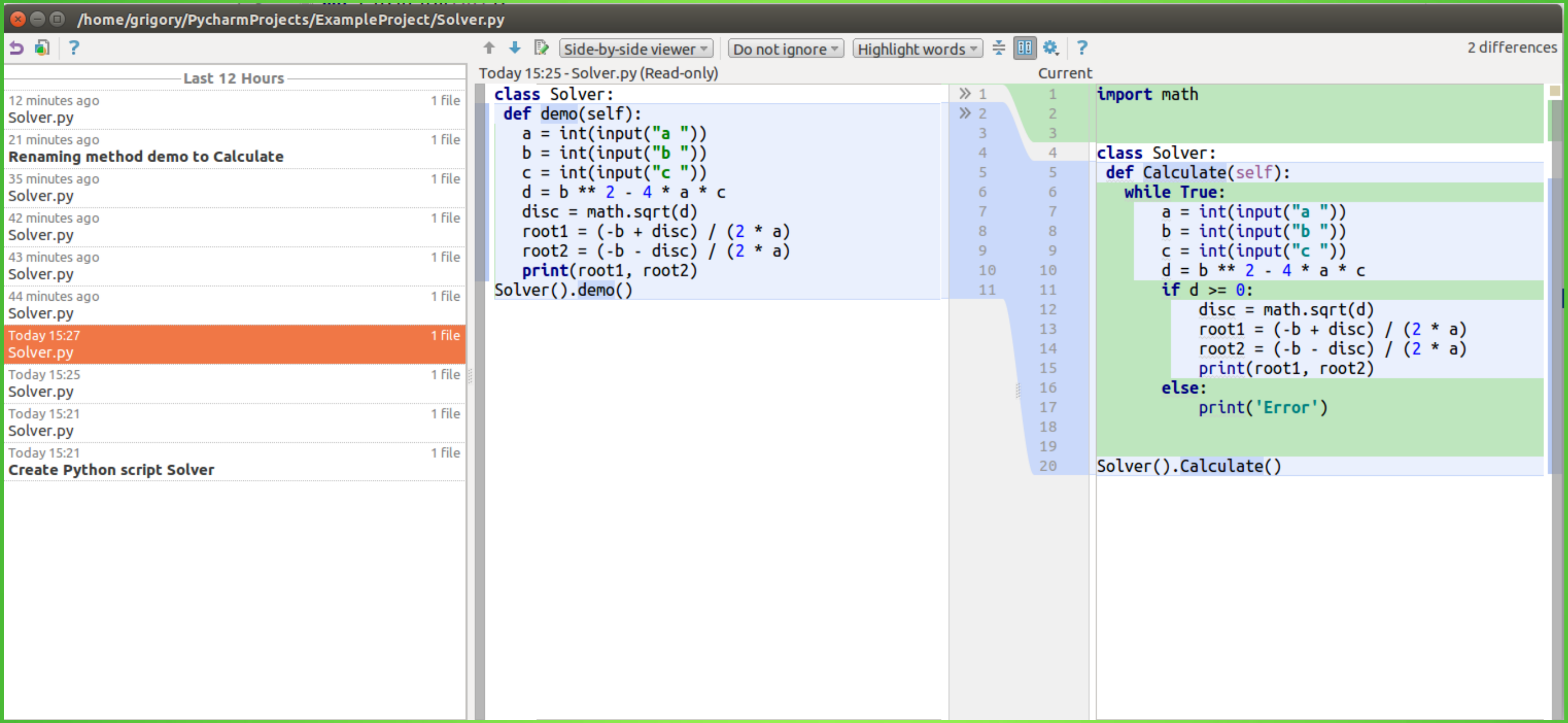
```
import math

 class Solver:
     def roots(self):
         a = 3
         b = 25
         c = 46
         root1 = (-b + math.sqrt(b**2 - 4*a*c)) / (2*a)
         root2 = (-b - math.sqrt(b**2 - 4*a*c)) / (2*a)
         return root1, root2

Solver().demo()
```

After

```
import math

class Solver:
    def demo(self):
        a = 3
        b = 25
        c = 46
        return_type_of_sqrt = math.sqrt(b ** 2 - 4 * a * c)
        root1 = (-b + return_type_of_sqrt) / (2*a)
        root2 = (-b - return_type_of_sqrt) / (2*a)
        print(root1,root2)

      Solver().demo()
```

Refactor → Extract → Variable

VCS → Local history → Show history



/home/grigory/PycharmProjects/ExampleProject/Solver.py

Side-by-side viewer | Do not ignore | Highlight words

2 differences

Last 12 Hours

12 minutes ago — 1 file
Solver.py

21 minutes ago — 1 file
Renaming method demo to Calculate

35 minutes ago — 1 file
Solver.py

42 minutes ago — 1 file
Solver.py

43 minutes ago — 1 file
Solver.py

44 minutes ago — 1 file
Solver.py

Today 15:27 — 1 file
Solver.py

Today 15:25 — 1 file
Solver.py

Today 15:21 — 1 file
Solver.py

Today 15:21 — 1 file
Create Python script Solver

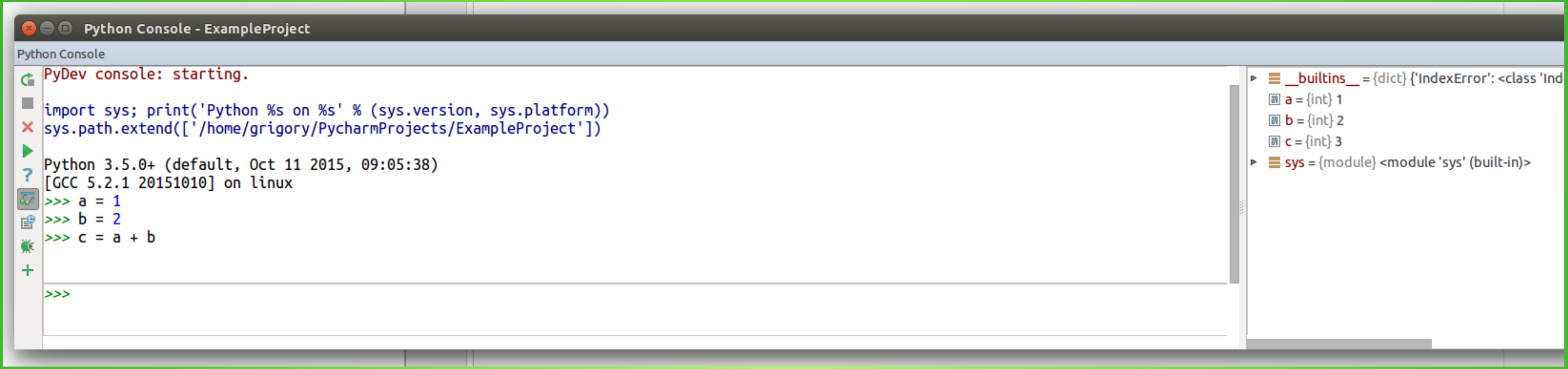
Today 15:25 - Solver.py (Read-only)

```
class Solver:
 def demo(self):
   a = int(input("a "))
   b = int(input("b "))
   c = int(input("c "))
   d = b ** 2 - 4 * a * c
   disc = math.sqrt(d)
   root1 = (-b + disc) / (2 * a)
   root2 = (-b - disc) / (2 * a)
   print(root1, root2)
Solver().demo()
```

Current

```
import math


class Solver:
 def Calculate(self):
    while True:
        a = int(input("a "))
        b = int(input("b "))
        c = int(input("c "))
        d = b ** 2 - 4 * a * c
        if d >= 0:
            disc = math.sqrt(d)
            root1 = (-b + disc) / (2 * a)
            root2 = (-b - disc) / (2 * a)
            print(root1, root2)
        else:
            print('Error')


Solver().Calculate()
```

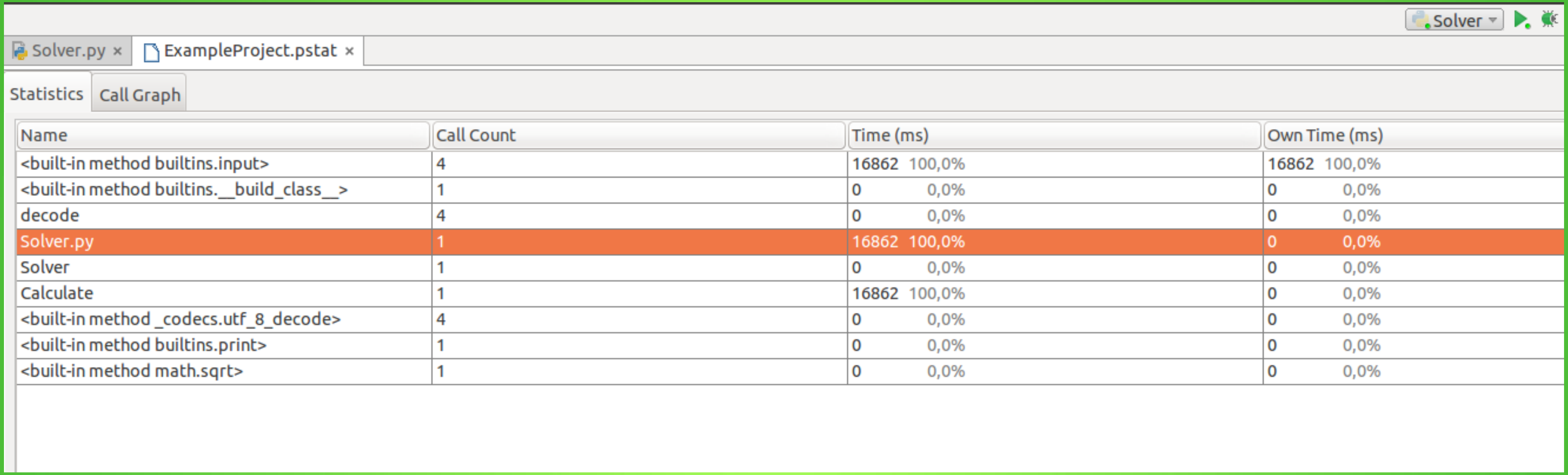
Tools → Python console



```
PyDev console: starting.

import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['/home/grigory/PycharmProjects/ExampleProject'])

Python 3.5.0+ (default, Oct 11 2015, 09:05:38)
[GCC 5.2.1 20151010] on linux
>>> a = 1
>>> b = 2
>>> c = a + b

>>>
```

```
__builtins__ = {dict} {'IndexError': <class 'Ind
a = {int} 1
b = {int} 2
c = {int} 3
sys = {module} <module 'sys' (built-in)>
```

Run → Profile 'solver'



Name	Call Count	Time (ms)	Own Time (ms)
<built-in method builtins.input>	4	16862 100,0%	16862 100,0%
<built-in method builtins.__build_class__>	1	0 0,0%	0 0,0%
decode	4	0 0,0%	0 0,0%
Solver.py	1	16862 100,0%	0 0,0%
Solver	1	0 0,0%	0 0,0%
Calculate	1	16862 100,0%	0 0,0%
<built-in method _codecs.utf_8_decode>	4	0 0,0%	0 0,0%
<built-in method builtins.print>	1	0 0,0%	0 0,0%
<built-in method math.sqrt>	1	0 0,0%	0 0,0%