

### **Задание**

Разработать программу для решения биквадратного уравнения.

* Программа должна быть разработана в виде консольного приложения на языке Python.
* Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).
* Коэффициенты А, В, С могут быть заданы в виде параметров командной строки ( вариант задания параметров приведен в конце файла с примером кода ). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. Описание работы с параметрами командной строки.
* Если коэффициент А, В, С введен или задан в командной строке некорректно, то необходимо проигнорировать некорректное значение и вводить коэффициент повторно пока коэффициент не будет введен корректно. Корректно заданный коэффициент - это коэффициент, значение которого может быть без ошибок преобразовано в действительное число.

### **Листинг**

**import** sys

**import** math

**class** SquareRoots:

**def** \_\_init\_\_(self):

'''

Конструктор класса

'''

*# Объявление коэффициентов*

self.coef\_A **=** 0.0

self.coef\_B **=** 0.0

self.coef\_C **=** 0.0

*# Количество корней*

self.num\_roots **=** 0

*# Список корней*

self.roots\_list **=** []

**def** get\_coef(self, index, prompt):

'''

Читаем коэффициент из командной строки или вводим с клавиатуры

Args:

index (int): Номер параметра в командной строке

prompt (str): Приглашение для ввода коэффицента

Returns:

float: Коэффициент квадратного уравнения

'''

**try**:

*# Пробуем прочитать коэффициент из командной строки*

coef\_str **=** sys.argv[index]

**except**:

*# Вводим с клавиатуры*

print(prompt)

coef\_str **=** input()

*# Заставляем пользователя совершить повторный ввод, если у него не получилось*

coef **=** 0.0

try\_again **=** **True**

**while** try\_again:

**try**:

*# Пробуем перевести строку в действительное число*

coef **=** float(coef\_str)

try\_again **=** **False**

**except**:

*# Вводим с клавиатуры*

print(prompt)

coef\_str **=** input()

**return** coef

**def** get\_coefs(self):

'''

Чтение трех коэффициентов

'''

self.coef\_A **=** self.get\_coef(1, 'Введите коэффициент А:')

self.coef\_B **=** self.get\_coef(2, 'Введите коэффициент B:')

self.coef\_C **=** self.get\_coef(3, 'Введите коэффициент C:')

**def** bigD\_to\_roots(self, bigD):

**if** bigD **==** 0.0:

root **=** math.sqrt(bigD)

self.num\_roots **+=** 1

self.roots\_list.append(root)

**elif** bigD **>** 0.0:

root **=** math.sqrt(bigD)

self.num\_roots **+=** 2

self.roots\_list.append(root)

self.roots\_list.append(**-**root)

**def** calculate\_roots(self):

'''

Вычисление корней квадратного уравнения

'''

a **=** self.coef\_A

b **=** self.coef\_B

c **=** self.coef\_C

*# Вычисление дискриминанта и корней*

D **=** b**\***b **-** 4**\***a**\***c

**if** D **==** 0.0:

bigD **=** **-**b **/** (2.0**\***a)

self.bigD\_to\_roots(bigD)

**elif** D **>** 0.0:

sqD **=** math.sqrt(D)

bigD **=** (**-**b **+** sqD) **/** (2.0**\***a)

self.bigD\_to\_roots(bigD)

bigD **=** (**-**b **-** sqD) **/** (2.0**\***a)

self.bigD\_to\_roots(bigD)

**def** print\_roots(self):

*# Проверка отсутствия ошибок при вычислении корней*

**if** self.num\_roots **!=** len(self.roots\_list):

print(('Ошибка. Уравнение содержит {} действительных корней, ' **+**\

'но было вычислено {} корней.').format(self.num\_roots, len(self.roots\_list)))

**else**:

texts **=** ['Нет корней', 'Один корень: ', 'Два корня: ', 'Три корня: ', 'Четыре корня: ']

print(texts[self.num\_roots], end **=** '')

**for** i **in** self.roots\_list:

print(i, end **=** ' ')

print('')

**def** main():

'''

Основная функция

'''

*# Создание объекта класса*

r **=** SquareRoots()

*# Последовательный вызов необходимых методов*

r.get\_coefs()

r.calculate\_roots()

r.print\_roots()

*# Если сценарий запущен из командной строки*

**if** \_\_name\_\_ **==** "\_\_main\_\_":

main()

*# Пример запуска*

*# lab1.py 1 0 -4*

### **Тестирование**

