

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Основы информатики»

Отчет по лабораторной работе №1

«Основные конструкции языка Python»

Выполнил:
студент группы ИУ5-35Б
Кулешова Ирина
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Нардид А. Н.
Подпись и дата:

Москва, 2022 г.

1. Описание задания:

Разработать программу для решения биквадратного уравнения.

1. Программа должна быть разработана в виде консольного приложения на языке Python.
2. Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
3. 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.
4. 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.

2. Текст программы:

```
import sys
import math

def get_coef(index, prompt):
    """
    Читаем коэффициент из командной строки или вводим с клавиатуры
    Args:
        index (int): Номер параметра в командной строке
        prompt (str): Приглашение для ввода коэффициента
    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()
    # Переводим строку в действительное число
    """
```

Проверяем можно ли преобразовать строку в число и если нельзя, то вводим коэффициент вновь.

"""

try:

coef = float(coef_str)

print(string_int)

except ValueError:

Handle the exception

print('Введено некорректное число.')

coef = get_coef(index, 'Введите коэффициент снова:')

return coef

def get_roots(a, b, c):

result = [] # Список корней

"""

Рассмотрим случаи, когда один из коэффициентов b или c равен 0 отдельно, так как их можно вычислить проще.

"""

if c == 0:

result.append(0)

Dc = - b / a

if Dc > 0:

root1 = math.sqrt(Dc)

root2 = - math.sqrt(Dc)

result.append(root1)

result.append(root2)

return result

elif b == 0:

Db = - c / a

if Db > 0:

root1 = math.sqrt(math.sqrt(Db))

root2 = - math.sqrt(math.sqrt(Db))

result.append(root1)

result.append(root2)

if Db == 0:

result.append(0)

return result

else:

D1 = b * b - 4 * a * c

if D1 < 0:

return result

elif D1 == 0:

D2 = - b / (2 * a)

if D2 < 0:

```

        return result
    # Если D2 = 0, то b = 0, а такой случай мы разобрали
    else:
        root1 = - math.sqrt(D2)
        root2 = math.sqrt(D2)
        result.append(root1)
        result.append(root2)
    return result
else:
    D3 = (- b - math.sqrt(D1)) / (2 * a)
    if D3 == 0:
        result.append(0)
    if D3 > 0:
        root1 = - math.sqrt(D3)
        root2 = math.sqrt(D3)
        result.append(root1)
        result.append(root2)
    D4 = (- b + math.sqrt(D1)) / (2 * a)
    if D4 == 0:
        result.append(0)
    if D4 > 0:
        root3 = - math.sqrt(D4)
        root4 = math.sqrt(D4)
        result.append(root3)
        result.append(root4)
    return result

```

```

def main():
    a = get_coef(1, 'Введите коэффициент A:')
    while a == 0:
        a = get_coef(1, 'Коэффициент A не может быть равен 0. Введите коэффициент A снова:')
    b = get_coef(2, 'Введите коэффициент B:')
    c = get_coef(3, 'Введите коэффициент C:')
    # Вычисление корней
    roots = get_roots(a,b,c)
    # Вывод корней
    len_roots = len(roots)
    if len_roots == 0:
        print('Нет корней')
    elif len_roots == 1:
        print('Один корень: {}'.format(roots[0]))
    elif len_roots == 2:
        print('Два корня: {} и {}'.format(roots[0], roots[1]))
    elif len_roots == 3:
        print('Три корня: {}, {} и {}'.format(roots[0], roots[1], roots[2]))
    elif len_roots == 4:

```

```
print('Четыре корня: {}, {}, {} и {}'.format(roots[0], roots[1], roots[2], roots[3]))
```

```
# Если сценарий запущен из командной строки
if __name__ == "__main__":
    main()
```

3. Экранные формы с примерами выполнения программы:

The image displays three sequential screenshots of a Python 3.9.0 shell interface, showing the execution of a program that calculates the roots of a cubic equation based on user input for coefficients A, B, and C.

Скриншот 1: Показывает запуск программы. Пользователь вводит коэффициенты A=0, B=-2, C=0. Программа выводит три корня: 0, 1.4142135623730951 и -1.4142135623730951.

```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license" for more information.
>>> [анализируем Lab1.py]
Введите коэффициент A:
0
Коэффициент A не может быть равен 0. Введите коэффициент A снова:
1
Введите коэффициент B:
-2
Введите коэффициент C:
0
Три корня: 0, 1.4142135623730951 и -1.4142135623730951
>>>
```

Скриншот 2: Показывает выполнение программы с другими входными данными: A=1, B=2, C=1. Программа выводит сообщение "Нет корней".



```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license" for more information.
>>> [анализируем Lab1.py]
Введите коэффициент A:
1
Введите коэффициент B:
2
Введите коэффициент C:
1
Нет корней
>>>
```

Скриншот 3: Показывает выполнение программы с еще другими входными данными: A=df, B=,ttt, C=-4. Программа выводит сообщение "Введено некорректное число." три раза и выводит два корня: 1.4142135623730951 и -1.4142135623730951.

```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license" for more information.
>>> [анализируем Lab1.py]
Введите коэффициент A:
df
Введено некорректное число.
Введите коэффициент снова:
1!f
Введено некорректное число.
Введите коэффициент снова:
1
Введите коэффициент B:
,ttt
Введено некорректное число.
Введите коэффициент снова:
0
Введите коэффициент C:
-4
Два корня: 1.4142135623730951 и -1.4142135623730951
>>>
```

I/O ОтладкиОболочка Python



Команды выполняются без отладки. Используйте клавиши со стрелками для истории.

Опции ▾

>>> Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license" for more information.
[анализируем Lab1.py]
Введите коэффициент A:
0
Коэффициент A не может быть равен 0. Введите коэффициент A снова:
ghj
Введено некорректное число.
Введите коэффициент снова:
1
Введите коэффициент B:
0
Введите коэффициент C:
0
Один корень: 0
>>>

I/O ОтладкиОболочка Python

Команды выполняются без отладки. Используйте клавиши со стрелками для истории.

Опции ▾

>>> Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license" for more information.
[анализируем Lab1.py]
Введите коэффициент A:
1
Введите коэффициент B:
-13
Введите коэффициент C:
36
Четыре корня: -2.0, 2.0, -3.0 и 3.0
>>> |