|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

Факультет «Информатика и системы управления»

ДИСЦИПЛИНА:

«БКИТ»

**Лабораторная работа № 1**

Студент Бабаян А.А. ИУ5Ц-52Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(И.О. Фамилия) (Группа) (Подпись, дата)

Преподаватель Гапанюк Ю.Е.  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(И.О. Фамилия) (Подпись, дата)

**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): |
|  | try: |
|  | # Пробуем прочитать коэффициент из командной строки |
|  | coef\_str = sys.argv[index] |
|  | except: |
|  | # Вводим с клавиатуры |
|  | tester = False |
|  | coef = 0.0 |
|  | # проверка ввода корректности |
|  | while (tester != True): |
|  | print(prompt) |
|  | coef\_str = input() |
|  | try: |
|  | coef = float(coef\_str) |
|  | tester = True |
|  | except ValueError: |
|  | tester = False |
|  | print("Вы неправильно ввели значение!! Пожалуйста, введите значение еще раз!\n") |
|  | return coef |
|  |  |
|  |  |
|  | def get\_roots(a, b, c): |
|  |  |
|  | result = [] |
|  | D = b \* b - 4 \* a \* c |
|  |  |
|  | if D == 0.0: |
|  | root = -b / (2.0 \* a) |
|  |  |
|  | if root >= 0.0: |
|  |  |
|  | x11 = math.sqrt(root) |
|  |  |
|  | x21 = -1 \* math.sqrt(root) |
|  |  |
|  | result.append(x11) |
|  | result.append(x21) |
|  |  |
|  | else: |
|  | result.append('Данного корня нет!') |
|  | result.append('Данного корня нет!') |
|  |  |
|  | elif D > 0.0: |
|  |  |
|  | sqD = math.sqrt(D) |
|  | root1 = (-b + sqD) / (2.0 \*a) |
|  | root2 = (-b - sqD) / (2.0 \* a) |
|  |  |
|  | if root1 > 0 and root2 > 0: |
|  |  |
|  | x1= math.sqrt(root1) |
|  |  |
|  | x2= -1 \* math.sqrt(root1) |
|  |  |
|  | x3= math.sqrt(root2) |
|  |  |
|  | x4= -1 \* math.sqrt(root2) |
|  |  |
|  | result.append(x1) |
|  | result.append(x2) |
|  | result.append(x3) |
|  | result.append(x4) |
|  |  |
|  | elif root1 > 0 and root2 < 0: |
|  |  |
|  | x1= math.sqrt(root1) |
|  |  |
|  | x2= -1 \* math.sqrt(root1) |
|  |  |
|  | result.append(x1) |
|  | result.append(x2) |
|  | result.append('Данного корня нет!') |
|  | result.append('Данного корня нет!') |
|  |  |
|  | elif root1 < 0 and root2 > 0: |
|  |  |
|  | x3 = math.sqrt(root2) |
|  | x4 = -1 \* math.sqrt(root2) |
|  |  |
|  | result.append('Данного корня нет!') |
|  | result.append('Данного корня нет!') |
|  | result.append(x3) |
|  | result.append(x4) |
|  |  |
|  | elif root1 == 0 and root2 > 0: |
|  |  |
|  | x1 = math.sqrt(root1) |
|  | x2 = math.sqrt(root2) |
|  | x3= -1 \* math.sqrt(root2) |
|  |  |
|  | result.append(x1) |
|  | result.append(x2) |
|  | result.append(x3) |
|  |  |
|  | elif root1 > 0 and root2 == 0: |
|  | x1 = math.sqrt(root1) |
|  | x2 = -1 \* math.sqrt(root1) |
|  | x3 = math.sqrt(root2) |
|  |  |
|  | result.append(x1) |
|  | result.append(x2) |
|  | result.append(x3) |
|  |  |
|  | else: |
|  | result.append('Данного корня нет!') |
|  | result.append('Данного корня нет!') |
|  | result.append('Данного корня нет!') |
|  | result.append('Данного корня нет!') |
|  |  |
|  | return result |
|  |  |
|  | def lin(b, c): |
|  |  |
|  | result = [] |
|  | root = 0.0 |
|  |  |
|  | root = -1 \* c / b |
|  |  |
|  | if root > 0: |
|  |  |
|  | root1 = math.sqrt(root) |
|  | root2 = -1 \* math.sqrt(root) |
|  |  |
|  | result.append(root1) |
|  | result.append(root2) |
|  |  |
|  | elif root == 0: |
|  |  |
|  | result.append(root) |
|  |  |
|  | return result |
|  |  |
|  |  |
|  |  |
|  | def main(): |
|  |  |
|  | a = get\_coef(1, 'Введите коэффициент А:') |
|  | b = get\_coef(2, 'Введите коэффициент B:') |
|  | c = get\_coef(3, 'Введите коэффициент C:') |
|  |  |
|  | roots =[] |
|  | if a == 0.0: |
|  | if c == 0.0: |
|  | if b == 0.0: |
|  | roots = [1,1,1,1,1] |
|  | else: |
|  | roots = [1] |
|  | elif b == 0.0: |
|  | roots = [] |
|  | else: |
|  | roots = lin(b, c) |
|  | # Вычисление корней |
|  |  |
|  | else: |
|  | roots = get\_roots(a,b,c) |
|  | # Вывод корней |
|  | len\_roots = len(roots) |
|  | if len\_roots == 0: |
|  | print('Корней не существует!!') |
|  |  |
|  | elif len\_roots == 1: |
|  | print('x = 0') |
|  |  |
|  | elif len\_roots == 2: |
|  | print('Один корень : x1 = {}; x2 = {}'.format(roots[0],roots[1])) |
|  |  |
|  | elif len\_roots == 3: |
|  | print('Два корня : x1 = {}; x2 = {}; x3 = {}'.format(roots[0],roots[1],roots[2])) |
|  |  |
|  | elif len\_roots == 4: |
|  | print('Два корня : x1 = {}; x2 = {}; x3 = {}; x4 = {}'.format(roots[0], roots[1], roots[2], |
|  | roots[3])) |
|  | elif len\_roots == 5: |
|  |  |
|  | print('Может быть любое число по x') |
|  |  |
|  |  |
|  | # Если сценарий запущен из командной строки |
|  | if \_\_name\_\_ == "\_\_main\_\_": |
|  | main() |

**3. Экранные формы с примерами выполнения программы**





