**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1

«Основные конструкции языка C#»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б: |  | преподаватель кафедры ИУ5 |
| Чиканчи В. А. |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 г.

Постановка задачи

Разработать программу для решения биквадратного уравнения.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и действительные корни уравнения (в зависимости от дискриминанта).
3. Если коэффициент А, В, С введен некорректно, то необходимо проигнорировать некорректное значение и вводить коэффициент повторно.
4. Корни уравнения выводятся зелёным цветом. Если корней нет, то сообщение выводится красным цветом.
5. Коэффициенты А, В, С могут быть заданы в виде параметров командной строки. Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. Проверка из пункта 3 в этом случае производится для параметров командной строки без повторного ввода с клавиатуры.

Текст программы

**Файл «main.py»:**

from python\_oop.shape import Rectangle, Circle, Square

import numpy as np

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

    N = 5

    rectangle = Rectangle(N, N, "blue")

    circle = Circle(N, "green")

    square = Square(N, "red")

**print**(rectangle)

**print**(circle)

**print**(square)

**print**("Сумма чисел от 1 до N:", np.**sum**(range(1, N + 1)))

**Файл «shape.py»:**

from abc import ABC, abstractmethod

import math

class Shape(ABC):

@abstractmethod

def area(self):

pass

class Color:

def \_\_init\_\_(self, color\_name):

self.color\_name = color\_name

class Rectangle(Shape):

def \_\_init\_\_(self, width, height, color):

self.width = width

self.height = height

self.color = Color(color)

def area(self):

return self.width \* self.height

def \_\_repr\_\_(self):

return "Rectangle(width={}, height={}, color={}, area={})".format(

self.width, self.height, self.color.color\_name, self.area()

)

class Circle(Shape):

def \_\_init\_\_(self, radius, color):

self.radius = radius

self.color = Color(color)

def area(self):

return math.pi \* (self.radius \*\* 2)

def \_\_repr\_\_(self):

return "Circle(radius={}, color={}, area={:.2f})".format(

self.radius, self.color.color\_name, self.area()

)

class Square(Rectangle):

def \_\_init\_\_(self, side, color):

super().\_\_init\_\_(side, side, color)

def \_\_repr\_\_(self):

return "Square(side={}, color={}, area={})".format(

self.width, self.color.color\_name, self.area()

)

Экранные формы с примерами выполнения программы

