



**Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Московский государственный технический  
университет имени Н.Э.Баумана (национальный  
исследовательский университет)»**

**Факультет «Информатика и системы управления»**

**Лабораторная работа №2 по курсу "Базовые компоненты интернет-технологий"**

**Выполнил: студент группы ИУ5-33Б Николай Горкунов**

**Проверяющий: преподаватель Юрий Гапанюк**

**2022 г.**

## 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

main.py

```
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square

"""
Для приложения pyqt6
"""

import sys
from PyQt6.QtWidgets import QApplication, QWidget, QPushButton,
QLabel

"""
Моя кнопка
"""

class Button(QWidget):
    def __init__(self):
        super().__init__()
        self.btn = QPushButton("Выход", self)
```

```

        self.btn.setGeometry(100, 20, 100, 30)
        self.btn.clicked.connect(self.onClicked)
        self.setWindowTitle("Кнопка")
        self.setGeometry(10, 10, 300, 70)
        self.show()

    def onClicked(self):
        print("Пока!")
        QApplication.quit()

def main():
    r = Rectangle("синего", 3, 3)
    c = Circle("зеленого", 3)
    s = Square("красного", 3)
    print(r)
    print(c)
    print(s)
    """
    Приложение pyqt6
    """

    app = QApplication(sys.argv)
    button = Button()
    app.exec()

if __name__ == "__main__":
    main()

```

figure.py

```

from abc import ABC, abstractmethod

class Figure(ABC):
    """
    Абстрактный класс «Геометрическая фигура»
    """

    @abstractmethod
    def square(self):
        """
        содержит виртуальный метод для вычисления площади фигуры.
        """

        pass

```

circle.py

```
from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor
import math

class Circle(Figure):
    """
    Класс «Круг» наследуется от класса «Геометрическая фигура».
    """
    FIGURE_TYPE = "Круг"

    @classmethod
    def get_figure_type(cls):
        return cls.FIGURE_TYPE

    def __init__(self, color_param, r_param):
        """
        Класс должен содержать конструктор по параметрам «радиус» и
        «цвет». В конструкторе создается объект класса «Цвет фигуры» для
        хранения цвета.
        """
        self.r = r_param
        self.fc = FigureColor()
        self.fc.colorproperty = color_param

    def square(self):
        """
        Класс должен переопределять метод, вычисляющий площадь
        фигуры.
        """
        return math.pi*(self.r**2)

    def __repr__(self):
        return '{} {} цвета радиусом {} площадью {}'.format(
            Circle.get_figure_type(),
            self.fc.colorproperty,
            self.r,
            self.square()
        )
```

color.py

```
class FigureColor:
    """
    Класс «Цвет фигуры»
    """

    def __init__(self):
        self._color = None

    @property
    def colorproperty(self):
        """
        Get-аксесор
        """
        return self._color

    @colorproperty.setter
    def colorproperty(self, value):
        """
        Set-аксесор
        """
        self._color = value
```

rectangle.py

```
from lab_python_oop.figure import Figure
from lab_python_oop.color import FigureColor

class Rectangle(Figure):
    """
    Класс «Прямоугольник» наследуется от класса «Геометрическая
    фигура».
    """
    FIGURE_TYPE = "Прямоугольник"

    @classmethod
    def get_figure_type(cls):
        return cls.FIGURE_TYPE

    def __init__(self, color_param, width_param, height_param):
        """
```

Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета.

```
"""
    self.width = width_param
    self.height = height_param
    self.fc = FigureColor()
    self.fc.colorproperty = color_param

def square(self):
    """
    Класс должен переопределять метод, вычисляющий площадь
    фигуры.
    """
    return self.width*self.height

def __repr__(self):
    return '{} {} цвета шириной {} и высотой {} площадью
    {}.format(
        Rectangle.get_figure_type(),
        self.fc.colorproperty,
        self.width,
        self.height,
        self.square()
    )
```

square.py

```
from lab_python_oop.rectangle import Rectangle
```

```
class Square(Rectangle):
    """
    Класс «Квадрат» наследуется от класса «Прямоугольник».
    """
    FIGURE_TYPE = "Квадрат"

    @classmethod
    def get_figure_type(cls):
        return cls.FIGURE_TYPE

    def __init__(self, color_param, side_param):
        """
```

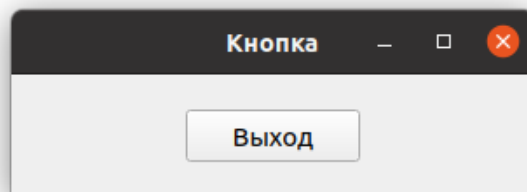
Класс должен содержать конструктор по параметрам «сторона» и «цвет».

```
"""
self.side = side_param
super().__init__(color_param, self.side, self.side)

def __repr__(self):
    return '{} {} цвета со стороной {} площадью {}'.format(
        Square.get_figure_type(),
        self.fc.colorproperty,
        self.side,
        self.square()
    )
```

## Тестирование

```
(lab_python_oop) nop@nopc:~/Projects/bmstu_3sem/BKIT_2022$ python3.8 main.py
Прямоугольник синего цвета шириной 3 и высотой 3 площадью 9.
Круг зеленого цвета радиусом 3 площадью 28.274333882308138.
Квадрат красного цвета со стороной 3 площадью 9.
```



```
(lab_python_oop) nop@nopc:~/Projects/bmstu_3sem/BKIT_2022$ python3.8 main.py
Прямоугольник синего цвета шириной 3 и высотой 3 площадью 9.
Круг зеленого цвета радиусом 3 площадью 28.274333882308138.
Квадрат красного цвета со стороной 3 площадью 9.
Пока!
(lab_python_oop) nop@nopc:~/Projects/bmstu_3sem/BKIT_2022$ █
```