**Московский государственный технический**

**университет им. Н.Э. Баумана**

**Факультет ИУ**

**Кафедра ИУ5**

**Курс «Основы информатики»**

**Отчет лабораторной работе №4**

Выполнил студент группы ИУ5-36Б:

Турланов В. Е.

Подпись и дата:

Проверил преподаватель каф.:

Нардид А. Н.

Подпись и дата:

Москва, 2024 г

**Описание задания**

Разработать программу для решения биквадратного уравнения.

1. Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием pip.
2. Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.
3. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab\_python\_oop.
4. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab\_python\_oop.
5. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать здесь.
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать здесь.
7. Класс «Прямоугольник» наследуется от класса «Геометрическая фигура». Класс должен содержать конструктор по параметрам «ширина», «высота» и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета. Класс должен переопределять метод, вычисляющий площадь фигуры.
8. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля math.
9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:
10. Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - https://pyformat.info/
11. Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
12. В корневом каталоге проекта создайте файл main.py для тестирования Ваших классов (используйте следующую конструкцию - https://docs.python.org/3/library/\_\_main\_\_.html). Создайте следующие объекты и выведите о них информацию в консоль (N - номер Вашего варианта по списку группы):

* Прямоугольник синего цвета шириной N и высотой N.
* Круг зеленого цвета радиусом N.
* Квадрат красного цвета со стороной N.
* Также вызовите один из методов внешнего пакета, установленного с использованием pip.

**Текст программы**

**main.py**

from lab\_python\_oop.rectangale import Rectangle

from lab\_python\_oop.circle import circle

from lab\_python\_oop.quadrate import quadrate

num = True

while(num):

    print("Введите название фигуры:")

    name = input()

    match(name):

        case "Прямоугольник":

            print("Введите ширину:")

            width = int(input())

            print("Введите высоту:")

            height = int(input())

            print("Введите цвет:")

            color = input()

            item1 = Rectangle(name, height, width, color)

            print(item1.reper())

        case "Круг":

            print("Введите радиус:")

            radius = int(input())

            print("Введите цвет:")

            color = input()

            item2 = circle(name, radius, color)

            print(item2.reper())

        case "Квадрат":

            print("Введите длину стороны:")

            side = int(input())

            print("Введите цвет:")

            color = str(input())

            item3 = quadrate(name, side, color)

            print(item3.reper())

    print("Хотите продолжить? (Да/Нет)")

    procces = input()

    if procces == "Нет":

        num = False

    else:

        num = True

**circle.py**

from lab\_python\_oop.geometric\_shape import geometricShape

from lab\_python\_oop.geometric\_shape import figureColor

import math

from colorama import init, Fore, Back, Style

class circle (geometricShape):

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

        self.name = name

        self.radius = radius

        self.circleColor = figureColor(color)

    def square(self):

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

    init()

    def reper(self):

        print(f"{Fore.GREEN}Фигура: {self.name} | радиус: {self.radius} | площадь: {self.square()} | цвет: {self.circleColor.color}{Fore.RESET}")

**geometric\_shape.py**

import math

from abc import ABC, abstractmethod

class geometricShape(ABC):

    @abstractmethod

    def square(self): pass

class figureColor:

    def \_\_init\_\_(self, color):

        self.\_\_color = color

    @property

    def color(self):

        return self.\_\_color

    @color.setter

    def color(self, value):

        self.\_\_color = value

**quadrate.py**

from lab\_python\_oop.geometric\_shape import figureColor

from lab\_python\_oop.rectangale import Rectangle

from colorama import init, Fore, Back, Style

class quadrate(Rectangle):

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

        self.name = name

        self.side = side

        self.quadrateColor = figureColor(color)

    def square(self):

        return self.side\*\*2

    init()

    def reper(self):

        print(f"{Fore.GREEN}Фигура: {self.name} | длина стороны: {self.side} | площадь: {self.square()} | цвет: {self.quadrateColor.color}{Fore.RESET}")

**rectangale.py**

from lab\_python\_oop.geometric\_shape import geometricShape

from lab\_python\_oop.geometric\_shape import figureColor

from colorama import init, Fore, Back, Style

class Rectangle(geometricShape):

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

        self.name = name

        self.width = width

        self.height = height

        self.rectangleColor = figureColor(color)

    def square(self):

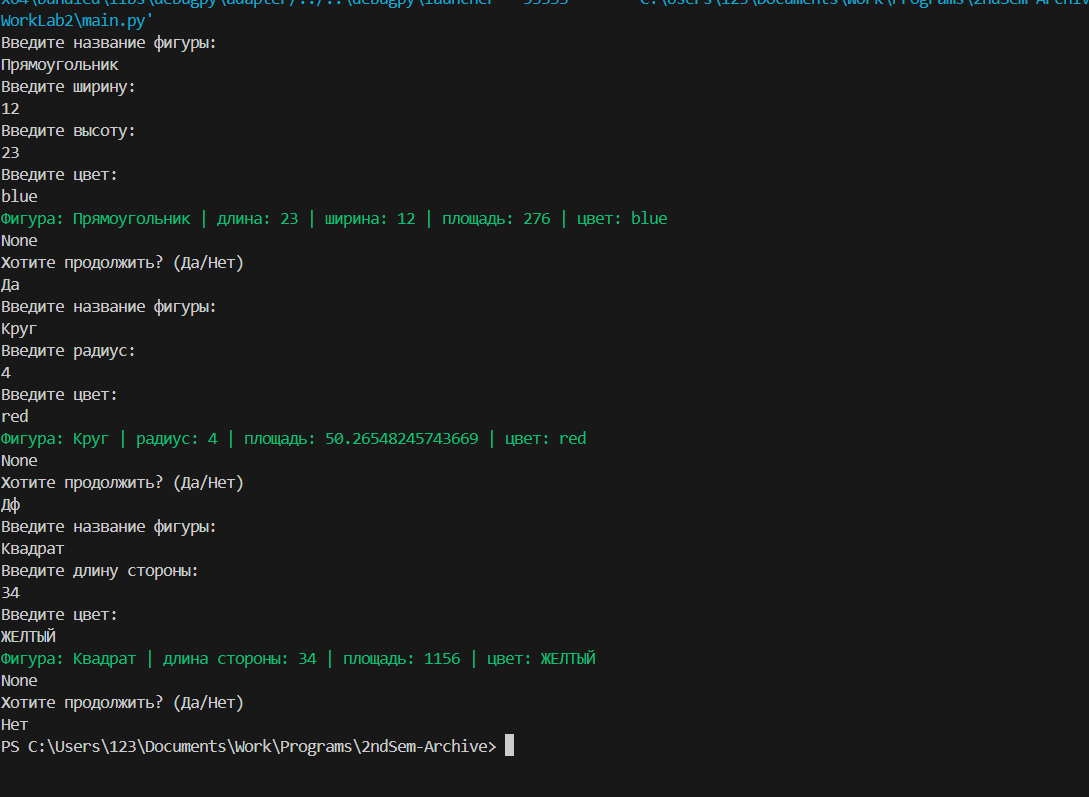
        return self.width \* self.height

    init()

    def reper(self):

        print(f"{Fore.GREEN}Фигура: {self.name} | длина: {self.height} | ширина: {self.width} | площадь: {self.square()} | цвет: {self.rectangleColor.color}{Fore.RESET}")

**Выполнение программы**

****