**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет ИУ

Кафедра ИУ5

Курс «Основы информатики»

Отчет по лабораторной работе №\_2\_

«Объектно-ориентированные возможности языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-23Б: |  | преподаватель каф. \_\_\_\_ |
| Селиханович Д.С. |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 г.

**Постановка задачи**

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. Последний класс «Геометрическая фигура» содержит абстрактный метод расчета площадных фигур. Подробнее об абстрактных классах и методах Вы можете прочитать здесь.
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойства Вы можете прочитать здесь.
7. Класс «Прямоугольник» наследуется от класса «Геометрическая фигура». Класс должен соблюдать конструктор по параметрам «ширина», «высота» и «цвет». В конструкторе создан объект класса «Цвет фигуры» для хранения цвета. Класс должен переопределить метод, поддерживающий площадь фигуры.
8. Класс «Круг» стал классом «Прямоугольник», задается параметр «радиус». Для расчета площади используется константа math.pi из модуля math.
9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен сохранять конструктор по длине сторон. Для классов «Прямоугольник», «Квадрат», «Круг»:

* Определите метод «repr», который возвращает в виде основные параметры фигуры, ее цвет и площадь. Используйте формат метода — https://pyformat.info/
* Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и приходить методом класса.

1. В корневом каталоге проекта создайте файл main.py для внедрения данных классов (используйте этот источник света — https://docs.python.org/3/library/\_\_main\_\_.html ). Создайте следующие объекты и выведите о них информацию в консоль (N — номер вашего представителя по списку группы):

* Прямоугольник синего цвета N и высота N.
* Круг зеленого цвета в радиусе N.
* Квадрат красного цвета со стороны N.
* Также вызовите один из методов внешнего пакета, установленного с помощью pip.

**Текст программы**

**\lab\_02\main.py**

from lab\_python\_oop.rectangle import Rectangle

from lab\_python\_oop.circle import Circle

from lab\_python\_oop.square import Square

from colorama import init, Fore, Back, Style

def main():

    init()

    r = Rectangle(3, 2, "синего")

    c = Circle(5, "зеленого")

    s = Square(5, "красного")

    print(Fore.BLUE)

    print(r)

    print(Fore.GREEN)

    print(c)

    print(Fore.RED)

    print(s)

    print(Style.RESET\_ALL)

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

    main()

**\lab\_02\lab\_python\_oop\\_\_init\_\_**

**\lab\_02\lab\_python\_oop\circle.py**

from lab\_python\_oop.figure import GeometricFigure

from lab\_python\_oop.color import Color

import math

class Circle(GeometricFigure):

    type = "Круг"

    @classmethod

    def get\_type(arg):

        return arg.type

    def \_\_init\_\_(self, Rad\_param, color\_param):

        self.radius = Rad\_param

        self.color = Color()

        self.color.colorproperty = color\_param

    def area(self):

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

    def \_\_repr\_\_(self):

        return '{} {} цвета радиусом {} площадью {}.'.format(

            Circle.get\_type(),

            self.color.colorproperty,

            self.radius,

            self.area()

        )

**\lab\_02\lab\_python\_oop\color.py**

class Color:

    def \_\_init\_\_(self):

        self.color = None

    @property

    def colorproperty(self):

        return self.color

    @colorproperty.setter

    def colorproperty(self, value):

        self.color = value

**\lab\_02\lab\_python\_oop\figure.py**

from abc import ABC, abstractmethod

class GeometricFigure(ABC):

    @abstractmethod

    def area(self):

        """Вычисляет площадь фигуры"""

        pass

**\lab\_02\lab\_python\_oop\rectangle.py**

from lab\_python\_oop.figure import GeometricFigure

from lab\_python\_oop.color import Color

class Rectangle(GeometricFigure):

    type = "Прямоуголник"

    @classmethod

    def get\_type(arg):

        return arg.type

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

        self.width = width\_param

        self.height = height\_param

        self.color = Color()

        self.color.colorproperty = color\_param

    def area(self):

        return self.height\*self.width

    def \_\_repr\_\_(self):

        return '{} {} цвета шириной {} и высотой {} площадью {}.'.format(

            Rectangle.get\_type(),

            self.color.colorproperty,

            self.width,

            self.height,

            self.area()

        )

**\lab\_02\lab\_python\_oop\square.py**

from lab\_python\_oop.rectangle import Rectangle

class Square(Rectangle):

    type = "Квадрат"

    @classmethod

    def get\_type(arg):

        return arg.type

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

        self.side = side\_param

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

    def \_\_repr\_\_(self):

            return '{} {} цвета со стороной {} площадью {}.'.format(

                Square.get\_type(),

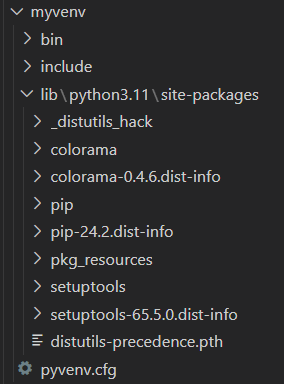
                self.color.colorproperty,

                self.side,

                self.area()

            )

**\lab\_02\my\_venv\**

****

**Анализ результатов**

