**Московский государственный технический университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

**Отчёт по Лабораторной работе №2**

**“Объектно-ориентированные возможности языка Python”**

Отчёт

(вид документа)

Листы А4

(вид носителя)

6

(количество листов)

|  |  |
| --- | --- |
| Исполнитель: | Студент группы ИУ5-54Б  Савельев Алексей Александрович |

Москва - 2020

**Цель работы**: Изучения объектно-ориентированных возможностей языка Python

**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.](https://docs.python.org/3/library/abc.html)
6. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать [здесь.](https://docs.python.org/3/library/functions.html#property)
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.](https://docs.python.org/3/library/math.html)
9. Класс «Квадрат» наследуется от класса «Прямоугольник». Класс должен содержать конструктор по длине стороны. Для классов «Прямоугольник», «Квадрат», «Круг»:
   * Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format - <https://pyformat.info/>
   * Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.
10. 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:**

модуль main.py

**import** requests

**from** LR2.lab\_python\_oop.rectangle **import** Rectangle

**from** LR2.lab\_python\_oop.squad **import** Squad

**from** LR2.lab\_python\_oop.circle **import** Circle

**def** main():

print(requests.\_\_license\_\_ + "\n")

rect = Rectangle(10,15, "Blue")

sqd = Squad(10,"Black")

circ = Circle(5, "Red")

rect.about()

sqd.about()

circ.about()

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

main()

модуль geomFigure.py

**from** abc **import** ABC

**class** GeomFigure(ABC):

**def** square(self):

**pass**

модуль rectangle.py

**from** LR2.lab\_python\_oop.geomFigure **import** GeomFigure

**from** LR2.lab\_python\_oop.figureColor **import** FigureColor

**class** Rectangle(GeomFigure):

fig = "Прямоугольник"

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

self.h = height

self.w = width

self.figureColor = FigureColor()

self.figureColor.set(color)

**def** square(self):

**return** self.h \* self.w

**def** about(self):

print("Фигура: \033[31m{}\033[0m, Ширина: {}, Высота: {}, Площадь: {}, Цвет: {}".format(self.fig, self.w, self.h, self.square(), self.figureColor.get()))

модуль squad.py

**from** LR2.lab\_python\_oop.rectangle **import** Rectangle

**from** LR2.lab\_python\_oop.figureColor **import** FigureColor

**class** Squad(Rectangle):

fig = "Квадрат"

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

self.side = side\_length

self.figureColor = FigureColor()

self.figureColor.set(color)

**def** square(self):

**return** self.side \* self.side

**def** about(self):

print("Фигура: \033[31m{}\033[0m, Сторона: {}, Площадь: {}, Цвет: {}".format(self.fig, self.side, self.square(), self.figureColor.get()))

модуль circle.py

**from** LR2.lab\_python\_oop.geomFigure **import** GeomFigure

**from** LR2.lab\_python\_oop.figureColor **import** FigureColor

**import** math

**class** Circle(GeomFigure):

fig = "Круг"

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

self.r = radius

self.figureColor = FigureColor()

self.figureColor.set(color)

**def** square(self):

**return** math.pi\*pow(self.r, 2)

**def** about(self):

print("Фигура: \033[31m{}\033[0m, Радиус: {}, Площадь: {}, Цвет: {}".format(self.fig, self.r, self.square(), self.figureColor.get()))

модуль figureColor.py

**class** FigureColor:

**def** \_\_init\_\_(self):

self.\_color = **None**

**def** get(self):

**return** self.\_color

**def** set(self, value):

self.\_color = value

**Окно терминала:**

