

Факультет Радиотехнический

Кафедра ИУ5 Системы обработки информации и управления

**Отчет по лабораторной работе № 2 по курсу
Базовые компоненты**

"Объектно-ориентированные возможности языка Python"

7

(количество листов)

Исполнитель

Студент группы РТ5-31Б _____

Татаев С.А.

«__»_____ 2022 г.

Проверил

Доцент кафедры ИУ5 _____

Гапанюк Ю.Е.

«__»_____ 2022 г.

Оглавление

1 Цель выполнения лабораторной работы	3
2 Листинг основной программы для проверки классов (main.py)	3
3 Листинг программы с абстрактным классом «Геометрическая фигура» (figure.py)	3
4 Листинг программы с классом «Прямоугольник» (rectangle.py)	3
5 Листинг программы с классом «Круг» (circle.py)	4
6 Листинг программы с классом «Квадрат» (square.py)	5
7 Листинг программы с классом «Цвет фигуры» (color.py)	6
8 Результаты работы программы	7
9 Выводы по результатам работы	7

Цель выполнения лабораторной работы

Изучение объектно-ориентированных возможностей языка Python путём разработки нескольких наследующихся классов.

Листинг основной программы для проверки классов (main.py)

```
from lab_python_oop.rectangle import Rectangle
from lab_python_oop.circle import Circle
from lab_python_oop.square import Square
import pymorphy2

def main():
    morph = pymorphy2.MorphAnalyzer()
    blue = morph.parse("синий")[0].inflect({'gent'})
    green = morph.parse("зелёный")[0].inflect({'gent'})
    red = morph.parse("красный")[0].inflect({'gent'})
    rect = Rectangle(blue.word, 20, 19)
    circle = Circle(green.word, 20)
    square = Square(red.word, 20)
    print(rect)
    print(circle)
    print(square)

if __name__ == "__main__":
    main()
```

Листинг программы с абстрактным классом «Геометрическая фигура» (figure.py)

"абстрактный класс "Геометрическая фигура", от которого наследуются другие"

```
from abc import ABC, abstractmethod
```

```
class Figure(ABC):
    @abstractmethod
    # вычисление площади
    def square(self):
        pass
```

Листинг программы с классом «Прямоугольник» (rectangle.py)

"класс "Прямоугольник""

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

```
class Rectangle(Figure):
```

```
FIGURE_TYPE = "Прямоугольник"
```

```
@classmethod
```

```
# get-method
```

```
def get_figure_type(cls):
```

```
    return cls.FIGURE_TYPE
```

```
def __init__(self, color, width, height):
```

```
    self.width = width
```

```
    self.height = height
```

```
    self.fc = FigureColor()
```

```
    self.fc.colorproperty = color
```

```
# вычисление площади
```

```
def square(self):
```

```
    return self.width * self.height
```

```
# вывод данных фигуры
```

```
def __repr__(self):
```

```
    return '{} {} цвета с шириной {} и высотой {} имеет площадь {}'.format(
```

```
        Rectangle.get_figure_type(),
```

```
        self.fc.colorproperty,
```

```
        self.width,
```

```
        self.height,
```

```
        self.square()
```

```
)
```

Листинг программы с классом «Круг» (circle.py)

```
"""класс "Круг"""
```

```
from lab_python_oop.figure import Figure
```

```
from lab_python_oop.color import FigureColor
```

```
from math import pi
```

```
class Circle(Figure):
```

```
    FIGURE_TYPE = "Круг"
```

```

@classmethod
# get-method
def get_figure_type(cls):
    return cls.FIGURE_TYPE

def __init__(self, color, r):
    self.radius = r
    self.fc = FigureColor()
    self.fc.colorproperty = color

# вычисление площади
def square(self):
    return round(pi * (self.radius ** 2), 2)

def __repr__(self):
    return '{} {} цвета радиусом {} имеет площадь {}'.format(
        Circle.get_figure_type(),
        self.fc.colorproperty,
        self.radius,
        self.square()
    )

```

Листинг программы с классом «Квадрат» (square.py)

```

"""класс "Квадрат"""
from lab_python_oop.rectangle import Rectangle

class Square(Rectangle):
    FIGURE_TYPE = "Квадрат"

    @classmethod
    # get-method
    def get_figure_type(cls):
        return cls.FIGURE_TYPE

    def __init__(self, color, side):
        self.side = side

```

```

super().__init__(color, self.side, self.side)

# вывод данных фигуры
def __repr__(self):
    return '{} {} цвета со стороной {} имеет площадь {}'.format(
        Square.get_figure_type(),
        self.fc.colorproperty,
        self.side,
        self.square()
    )

```

Листинг программы с классом «Цвет фигуры» (color.py)

```

"""класс "Цвет фигуры"""

```

```

class FigureColor:
    def __init__(self):
        self._color = None



    @property
    # get-метод
    def colorproperty(self):
        return self._color

    @colorproperty.setter
    # set-метод
    def colorproperty(self, value):
        self._color = value

```

Результаты работы программы

```
D:\documents\Basic components\lab 2>python main.py
Прямоугольник синего цвета с шириной 20 и высотой 19 имеет площадь 380
Круг зелёного цвета радиусом 20 имеет площадь 1256.64
Квадрат красного цвета со стороной 20 имеет площадь 400
```

Команды выполняются без отладки. Используйте клавиши со стрелками для   Оpciones ▾

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 b
Type "help", "copyright", "credits" or "license" for more information.
>>> [анализируем main.py]
Прямоугольник синего цвета с шириной 20 и высотой 19 имеет площадь 380
Круг зелёного цвета радиусом 20 имеет площадь 1256.64
Квадрат красного цвета со стороной 20 имеет площадь 400
>>>
```

Выводы по результатам работы

При выполнении работы я изучил объектно-ориентированные возможности языка Python и разработал нескольких наследующихся классов.