**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-34Б |  |  |
| Свечникова Анна |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2021г.

**Постановка задачи**

Цель лабораторной работы: изучение объектно-ориентированных возможностей языка Python.

**Задание:**

Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием pip.

Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.

Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab\_python\_oop.

Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab\_python\_oop.

Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать здесь.

Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать здесь.

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.

Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля math.

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.

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:**

**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

**Color.py:**

class FigureColor: *"""* *Класс «Цвет фигуры»* *"""*def \_\_init\_\_(self): self.\_color = None @property def colorproperty(self): *"""* *Get-аксессор* *"""*return self.\_color @colorproperty.setter def colorproperty(self, value): *"""* *Set-аксессор* *"""*self.\_color = value

**Square.py:**

from lab\_python\_oop.rectangle import Rectangleclass 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() )

**Circle.py:**

from lab\_python\_oop.figure import Figurefrom lab\_python\_oop.color import FigureColorimport mathclass 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() )

**Rectangle.py:**

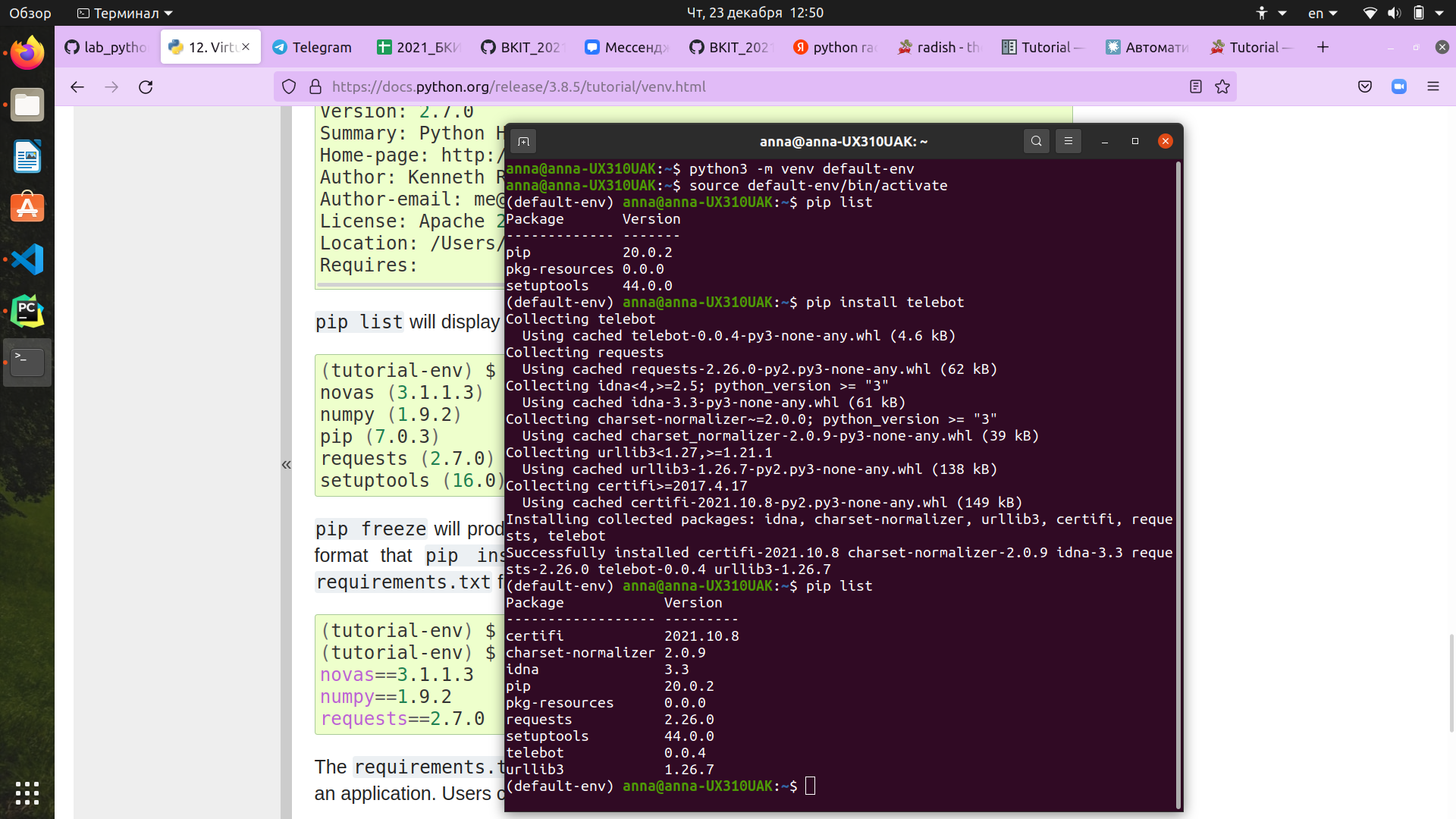
from lab\_python\_oop.figure import Figurefrom lab\_python\_oop.color import FigureColorclass Rectangle(Figure): *"""* *Класс «Прямоугольник» наследуется от класса «Геометрическая фигура».* *"""*FIGURE\_TYPE = "Прямоугольник" @classmethod def get\_figure\_type(cls): return cls.FIGURE\_TYPE def \_\_init\_\_(self, 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() )

**main.py:**

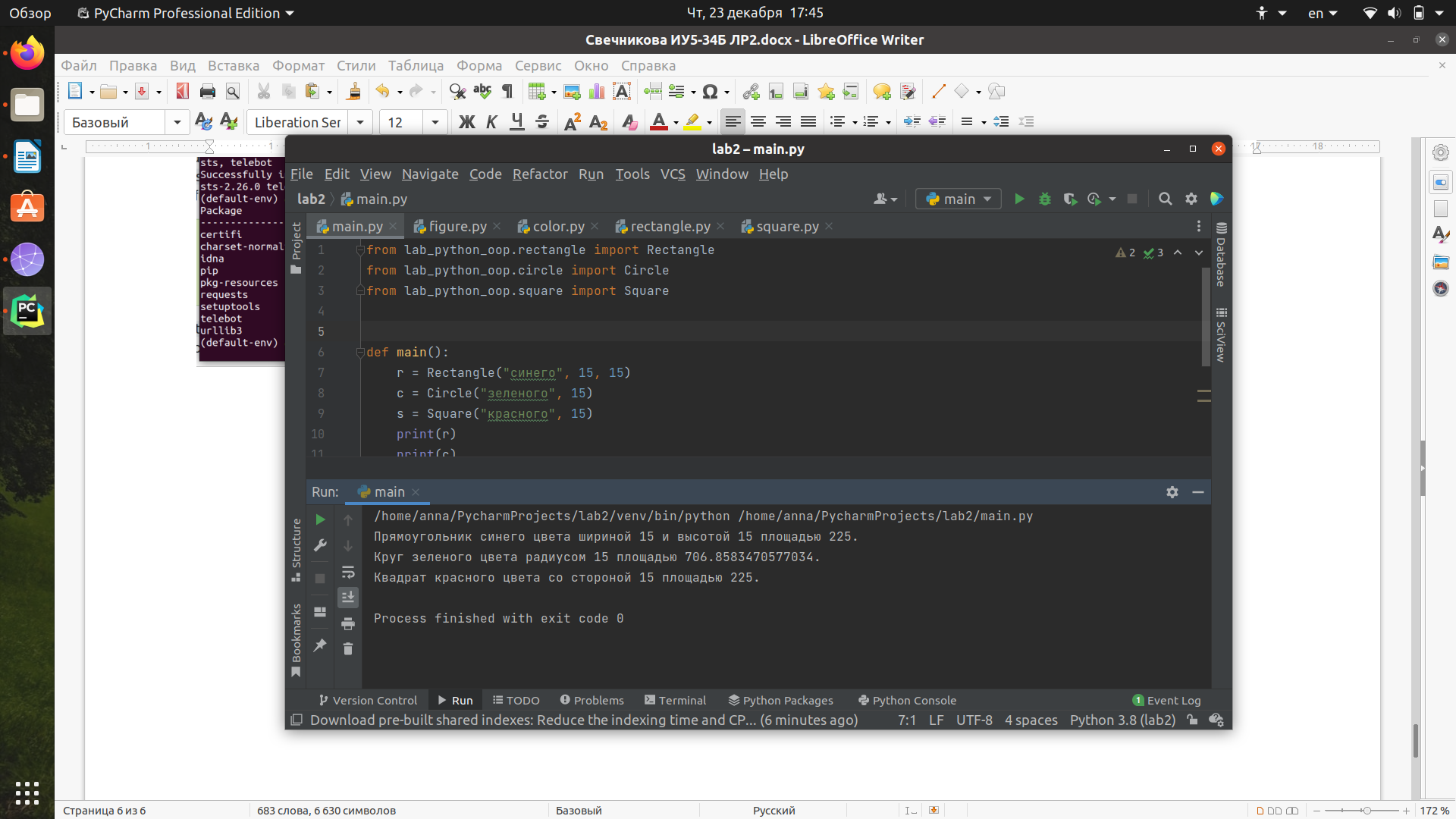
from lab\_python\_oop.rectangle import Rectanglefrom lab\_python\_oop.circle import Circlefrom lab\_python\_oop.square import Squaredef main(): r = Rectangle("синего", 15, 15) c = Circle("зеленого", 15) s = Square("красного", 15) print(r) print(c) print(s)if \_\_name\_\_ == "\_\_main\_\_": main()

**Примеры выполнения**

Создание виртуального окружения и установка в него пакета:



Работа с классами:



Структура проекта:

