**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»

Отчёт по лабораторной работе №2

«Объектно-ориентированные возможности языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Васькин М.И. |  | Гапанюк Ю.Е. |
| дата: 18.11.2022 |  |  |

**Задание:**

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»:

o Определите метод "repr", который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Используйте метод format

-<https://pyformat.info/>o Название фигуры («Прямоугольник», «Квадрат», «Круг») должно задаваться в виде поля данных класса и возвращаться методом класса.

1. В корневом каталоге проекта создайте файл main.py для тестирования Ваших классов (используйте следующую конструкцию

- [https://docs.python.org/3/library/\_\_main\_\_.html)](https://docs.python.org/3/library/__main__.html). Создайте следующие объекты и выведите о них информацию в консоль (N=11 - номер Вашего варианта по списку группы):

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

1. **Дополнительное задание.** Протестируйте корректность работы Вашей программы с помощью модульного теста.

**Создание виртуального окружения:**

PS C:\BKIT > python3 -m venv venv

PS C:\BKIT > venv\Scripts\activate

(venv) PS C:\BKIT> pip freeze

(venv) PS C:\BKIT> pip install faker

…

(venv) PS C:\BKIT> pip freeze

#Ниже результат выполнения команды

Faker==15.1.3 numpy==1.23.4 python-dateutil==2.8.2 pytz==2022.6 six==1.16.0

(venv) PS C:\BKIT> pip freeze > requirements.txt

**Текст программы: main.py:**

|  |
| --- |
| *from* lab\_python\_oop.rectangle *import* Rectangle *from* lab\_python\_oop.circle *import* Circle *from* lab\_python\_oop.square *import* Square *from* faker *import* Faker  *def* *main*():  r *=* Rectangle("синего", 11, 11) c *=* Circle("зеленого", 11) s *=* Square("красного", 11) print(r) print(c) print(s)  fake *=* Faker() |
| print(fake.name(), fake.address(), fake.date\_of\_birth(minimum\_age *=* 18), sep*=*"\n")  *if* \_\_name\_\_ *==* "\_\_main\_\_":  main() |

**lab\_python\_oop/circle.py:**

|  |
| --- |
| *from* lab\_python\_oop.figure *import* Figure *from* lab\_python\_oop.color *import* FigureColor *import* math    *class* 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()  ) |

**lab\_python\_oop/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 |

**lab\_python\_oop/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* |

**lab\_python\_oop/rectangle.py:**

*from* lab\_python\_oop.figure *import* Figure *from* lab\_python\_oop.color *import* FigureColor

|  |
| --- |
| *class* 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()  ) |

**lab\_python\_oop/square.py:**

|  |
| --- |
| *from* lab\_python\_oop.rectangle *import* Rectangle    *class* 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()  ) |

**Результаты выполнения программы:**

