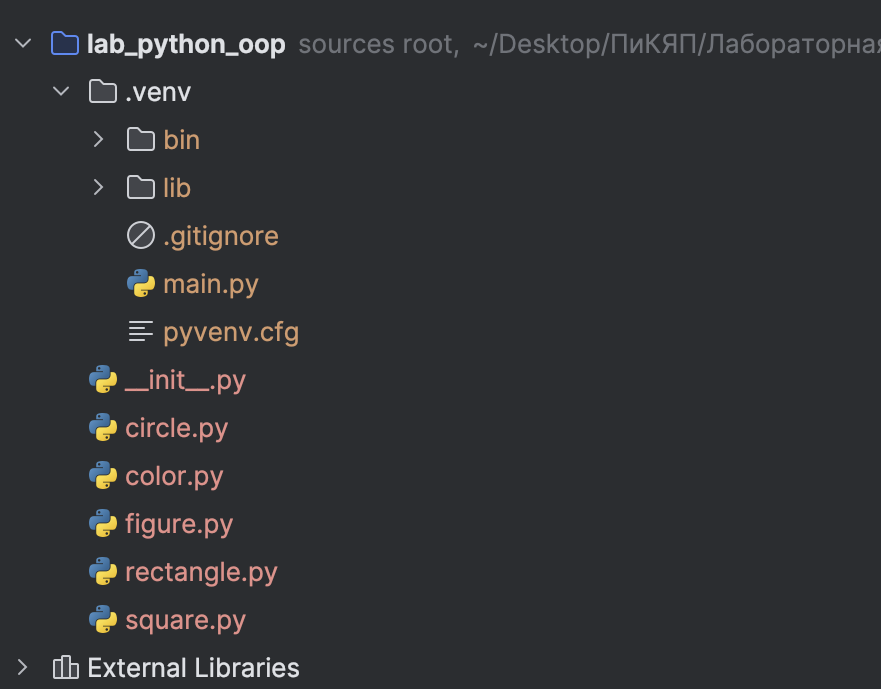
|  |  |
| --- | --- |
|  | Министерство науки и высшего образования Российской Федерации  Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»  (МГТУ им. Н.Э. Баумана) |
| ФАКУЛЬТЕТ «ИНЖЕНЕРНЫЙ БИЗНЕС И МЕНЕДЖМЕНТ»  КАФЕДРА «ПРОМЫШЛЕННАЯ ЛОГИСТИКА» (ИБМ-3)  Лабораторная работа № 2  «Парадигмы и конструкции языков программирования»  38.03.05 «Бизнес-информатика» (бакалавриат)  Студент ИБМ3-24Б А. А. Лютикова  (Подпись, дата)    2024 г. | |

В ходе выполнения лабораторной работы изучили объектно-ориентированные возможности языка Python, разработали программу, реализующую работу с классами геометрических фигур.

Расположение классов в отдельном файле пакета lab\_python\_oop:



**Класс Figure:**

|  |
| --- |
| from abc import ABC, abstractmethod  class Figure(ABC):  @abstractmethod  def square(self):  pass |

**Класс FigureColor:**

|  |
| --- |
| class FigureColor:  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 |

**Класс Rectangle:**

|  |
| --- |
| from figure import Figure from 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):   self.width = width\_param  self.height = height\_param  self.fc = FigureColor()  self.fc.colorproperty = color\_param   def square(self):   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()  ) |

**Класс Circle:**

|  |
| --- |
| from figure import Figure from color import FigureColor import math   class Circle(Figure):   FIGURE\_TYPE = "Круг"   @classmethod  def get\_figure\_type(cls):  return cls.FIGURE\_TYPE   def \_\_init\_\_(self, color\_param, r\_param):   self.r = r\_param  self.fc = FigureColor()  self.fc.colorproperty = color\_param   def square(self):  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()  ) |

**Класс Square:**

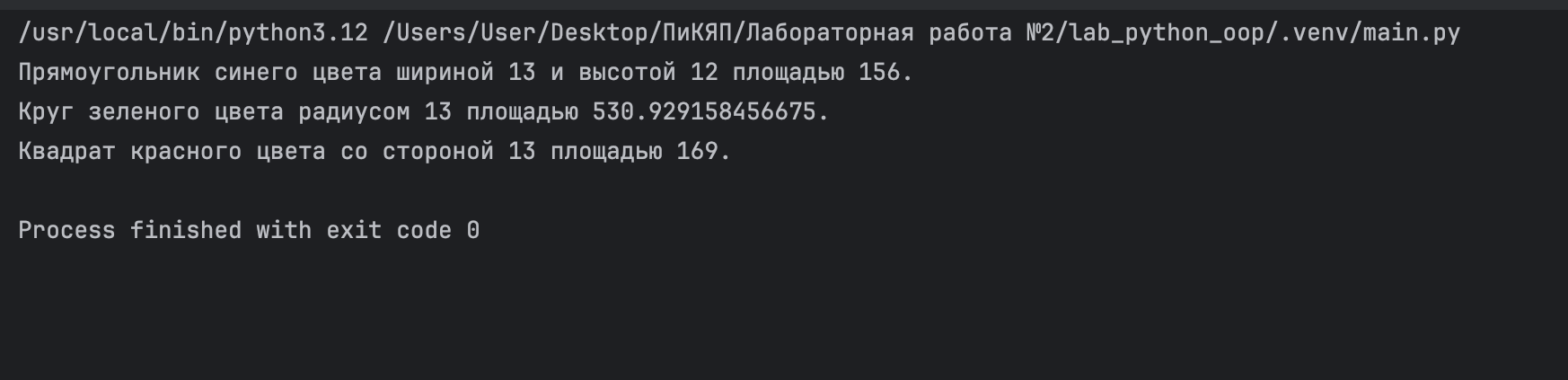
|  |
| --- |
| from rectangle import Rectangle   class Square(Rectangle):   FIGURE\_TYPE = "Квадрат"   @classmethod  def get\_figure\_type(cls):  return cls.FIGURE\_TYPE   def \_\_init\_\_(self, color\_param, side\_param):  *"""  Класс должен содержать конструктор по параметрам «сторона» и «цвет».  """* 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()  ) |

Файл main.py для тестирования классов, в котором созданы объекты классов и выводится информация о них:

* Прямоугольник синего цвета шириной 13 и высотой 12.
* Круг зеленого цвета радиусом 13.
* Квадрат красного цвета со стороной 13.

|  |
| --- |
| from rectangle import Rectangle from circle import Circle from square import Square   def main():  r = Rectangle("синего", 13, 12)  c = Circle("зеленого", 13)  s = Square("красного", 13)  print(r)  print(c)  print(s)  if \_\_name\_\_ == "\_\_main\_\_":  main() |

**Результат работы:**

****