# Текст программы

## main.py

from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.color import Color  
from lab\_python\_oop.rectangle import Rectangle  
from lab\_python\_oop.square import Square  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 print(Rectangle(5, 5, Color("Blue")))  
 print(Circle(5, Color("Green")))  
 print(Square(5, Color("Red")))  
  
 rectangle = Rectangle(15, 15, "Pink")  
 print(rectangle.get\_area())  
from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.color import Color  
from lab\_python\_oop.rectangle import Rectangle  
from lab\_python\_oop.square import Square  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 print(Rectangle(5, 5, Color("Blue")))  
 print(Circle(5, Color("Green")))  
 print(Square(5, Color("Red")))  
  
 rectangle = Rectangle(15, 15, "Pink")  
 print(rectangle.get\_area())

## shape.py

from abc import ABC, abstractmethod  
  
  
class Shape(ABC):  
 @abstractmethod  
 def get\_area(self) -> float:  
 pass  
from abc import ABC, abstractmethod  
  
  
class Shape(ABC):  
 @abstractmethod  
 def get\_area(self) -> float:  
 pass

## color.py

class Color:  
 def \_\_init\_\_(self, value: str):  
 self.\_value = value  
  
 @property  
 def value(self) -> str:  
 return self.\_value  
  
 def \_\_repr\_\_(self):  
 return f"Color({self.value})"  
  
 def \_\_str\_\_(self):  
 return self.value  
class Color:  
 def \_\_init\_\_(self, value: str):  
 self.\_value = value  
  
 @property  
 def value(self) -> str:  
 return self.\_value  
  
 def \_\_repr\_\_(self):  
 return f"Color({self.value})"  
  
 def \_\_str\_\_(self):  
 return self.value

## circle.py

import math  
  
from .color import Color  
from .shape import Shape  
  
  
class Circle(Shape):  
 def \_\_init\_\_(self, radius: float, color: Color):  
 self.\_radius = radius  
 self.\_color = color  
  
 def get\_area(self) -> float:  
 return self.\_radius \* math.pi  
  
 def \_\_repr\_\_(self):  
 return f"Circle(radius: {self.\_radius}, color: '{self.\_color}')"  
import math  
  
from .color import Color  
from .shape import Shape  
  
  
class Circle(Shape):  
 def \_\_init\_\_(self, radius: float, color: Color):  
 self.\_radius = radius  
 self.\_color = color  
  
 def get\_area(self) -> float:  
 return self.\_radius \* math.pi  
  
 def \_\_repr\_\_(self):  
 return f"Circle(radius: {self.\_radius}, color: '{self.\_color}')"

## rectangle.py

from lab\_python\_oop.color import Color  
from .shape import Shape  
  
  
class Rectangle(Shape):  
 def \_\_init\_\_(self, width: float, height: float, color: Color):  
 self.\_width = width  
 self.\_height = height  
 self.\_color = color  
  
 def get\_area(self) -> float:  
 return self.\_height \* self.\_width  
  
 def \_\_repr\_\_(self):  
 return f"Rectangle(width: {self.\_width}, height: {self.\_height}, color: '{self.\_color}')"  
from lab\_python\_oop.color import Color  
from .shape import Shape  
  
  
class Rectangle(Shape):  
 def \_\_init\_\_(self, width: float, height: float, color: Color):  
 self.\_width = width  
 self.\_height = height  
 self.\_color = color  
  
 def get\_area(self) -> float:  
 return self.\_height \* self.\_width  
  
 def \_\_repr\_\_(self):  
 return f"Rectangle(width: {self.\_width}, height: {self.\_height}, color: '{self.\_color}')"

## square.py

from .color import Color  
from .rectangle import Rectangle  
  
  
class Square(Rectangle):  
 def \_\_init\_\_(self, side\_length: float, color: Color):  
 super().\_\_init\_\_(side\_length, side\_length, color)  
  
 def \_\_repr\_\_(self):  
 return f"Square(side\_length: {self.\_width}, color: '{self.\_color}')"  
from .color import Color  
from .rectangle import Rectangle  
  
  
class Square(Rectangle):  
 def \_\_init\_\_(self, side\_length: float, color: Color):  
 super().\_\_init\_\_(side\_length, side\_length, color)  
  
 def \_\_repr\_\_(self):  
 return f"Square(side\_length: {self.\_width}, color: '{self.\_color}')"

## Результат выполнения программы

Rectangle(width: 5, height: 5, color: 'Blue')  
Circle(radius: 5, color: 'Green')  
Square(side\_length: 5, color: 'Red')  
225