

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Парадигмы и конструкции языков
программирования»

Отчет по лабораторной работе №1

Выполнил: Сорокин Михаил
Группа ИУ5-33Б
Вариант 15 В

Проверил: Гапанюк Юрий
Евгеньевич
преподаватель каф. ИУ5

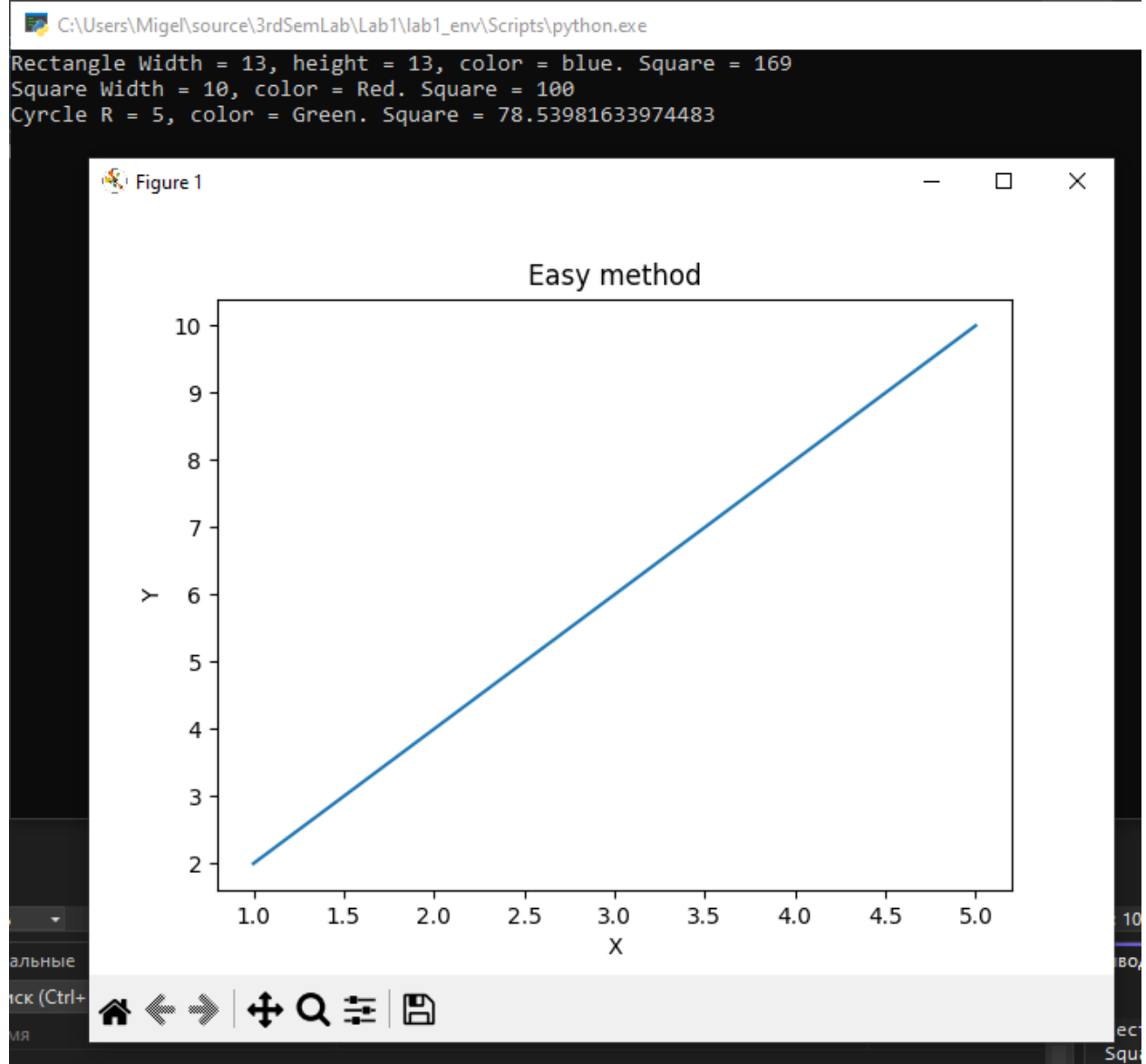
Подпись и дата: 23.10.2024



Подпись и дата:

Москва, 2024 г.

Принт-скрин выполнения программы



Листинг

Пакет lab_python_oop модуль box.py

from lab_python_oop import Rectangle

```
class Box(Rectangle.Rectangle):  
    """description of class"""  
  
    type = "Square"  
  
    def __init__(self, width, col = "white"):  
        self.width = width  
        self.col = col  
  
    def square(self): return self.width ** 2  
  
    def __repr__(self):  
        return ('{} Width = {}, color = {}. Square = {}'.format(  
            self.print_type(),  
            self.width,  
            self.col,  
            self.square())
```

```

    )

модуль Cyrcle.py
import math
from lab_python_oop import FigureColor
from lab_python_oop import Shape

class Cyrcle(Shape.Shape):
    """description of class"""

    type = "Cyrcle"

    def __init__(self, r, col = "white"):
        self.r = r
        self.col = col

    def square(self): return (self.r ** 2) * math.pi

    def __repr__(self):
        return ('{ } R = { }, color = { }. Square = { }'.format(
            self.print_type(),
            self.r,
            self.col,
            self.square()
        ))

```

```

модуль FigureColor.py
class FigureColor:
    """description of figure's color class"""

```

```

    def __init__(self, name):
        self.__color = name

    @property
    def color(self):
        return self.__color

    @color.setter
    def color(self, new_name):
        self.__color = new_name

```

```

модуль Rectangle.py
from lab_python_oop import Shape
from lab_python_oop import FigureColor

```

```

class Rectangle(Shape.Shape):
    """description of main class"""

    type = "Rectangle"

    def __init__(self, width, height, col = "white"):
        self.width = width
        self.height = height
        self.col = FigureColor.FigureColor(col)

    def square (self): return self.width * self.height

    def __repr__(self):
        return ('{ } Width = { }, height = { }, color = { }. Square = { }'.format(
            self.print_type(),
            self.width,
            self.height,
            self.col,
            self.square()
        ))

```

модуль Shape.py

```
import abc
class Shape(abc.ABC):
    """description of abstract class"""

    @classmethod
    def print_type(arg):
        return(arg.type)

    @abc.abstractmethod
    def square(self): pass
```

Исполняемый файл main.py

```
from lab_python_oop import Rectangle
from lab_python_oop import Box
from lab_python_oop import Circle
import matplotlib.pyplot as plt

def main():
    rect = Rectangle.Rectangle(13, 13, "blue")
    print(rect)

    b = Box.Box(10, "Red")
    print(b)

    c = Circle.Circle(5, "Green")
    print(c)

    # Enviroment
    x = [1, 2, 3, 4, 5]
    y = [2, 4, 6, 8, 10]
    plt.plot(x, y)
    plt.xlabel('X')
    plt.ylabel('Y')
    plt.title('Easy method')
    plt.show()
if __name__ == "__main__":
    main()
```

requirements.txt

```
contourpy==1.1.1
cycler==0.12.1
fonttools==4.54.1
importlib_resources==6.4.5
kiwisolver==1.4.7
matplotlib==3.7.5
numpy==1.24.4
packaging==24.1
pillow==10.4.0
pip==24.2
pyparsing==3.1.4
python-dateutil==2.9.0.post0
setuptools==75.2.0
six==1.16.0
zipp==3.20.2
```