**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 «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Лабораторная работа №2

по дисциплине «Базовые компоненты интернет-технологий»

Выполнил:

студент группы ИУ5-33Б

Рассказов Н.Д.

Проверил:

Гапанюк Ю. Е.

2022 г.

1. **Описание задания.**
2. Необходимо создать виртуальное окружение и установить в него хотя бы один внешний пакет с использованием pip.
3. Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Python 3.
4. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab\_python\_oop.
5. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab\_python\_oop.
6. 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)
7. 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)
8. 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.
9. 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)
10. 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.
11. 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.
12. Текст программы.

*import* random

*import* math

*from* turtle *import* write\_docstringdict

*import* pygame

*from* lab\_python\_oop.rectangle *import* Rectangle

*from* lab\_python\_oop.circle *import* Circle

*from* lab\_python\_oop.square *import* Square

def py\_game():

    pygame.init()

*#s = pygame.mixer.Sound("bird.mp3")*

    width = 400

    heigth = 500

    apple\_width = 10

    apple\_heigth = 10

    square\_width = 60

    square\_heigth = 60

    screen = pygame.display.set\_mode((width, heigth))

    done = False

    color\_red = (255, 0, 0)

    color\_white = (255, 255, 255)

    color\_black = (0, 0, 0)

    x = 0

    x\_last = 0

    y = 0

    y\_last = y

    apple\_x = x + square\_width + (width - x - square\_width)/2

    apple\_last\_x = apple\_x

    apple\_y = y + square\_heigth/2

    apple\_last\_y = apple\_y

    k = 0

*while* not done:

*for* event *in* pygame.event.get():

*if* event.type == pygame.QUIT:

                print(f"Итоговый счёт: {k}")

                done = True

        pressed = pygame.key.get\_pressed()

*if* pressed[pygame.K\_UP]:

            y -= 3

*if* pressed[pygame.K\_DOWN]:

            y += 3

*if* pressed[pygame.K\_LEFT]:

            x -= 3

*if* pressed[pygame.K\_RIGHT]:

            x += 3

*# Скорость по диагонали*

*if* not (pressed[pygame.K\_UP] and pressed[pygame.K\_DOWN]) and \

             not (pressed[pygame.K\_LEFT] and pressed[pygame.K\_RIGHT]):

*if* pressed[pygame.K\_UP] and pressed[pygame.K\_LEFT]:

                x += 3-math.sqrt(3)

                y += 3-math.sqrt(3)

*if* pressed[pygame.K\_UP] and pressed[pygame.K\_RIGHT]:

                x -= 3-math.sqrt(3)

                y += 3-math.sqrt(3)

*if* pressed[pygame.K\_DOWN] and pressed[pygame.K\_LEFT]:

                x += 3-math.sqrt(3)

                y -= 3-math.sqrt(3)

*if* pressed[pygame.K\_DOWN] and pressed[pygame.K\_RIGHT]:

                x -= 3-math.sqrt(3)

                y -= 3-math.sqrt(3)

*if* x + square\_width > width:

            x = width - square\_width

*if* x < 0:

            x = 0

*if* y + square\_heigth > heigth:

            y = heigth - square\_heigth

*if* y < 0:

            y = 0

*if* ((x+square\_width >= apple\_x+apple\_width/2) and (x <= apple\_x+apple\_width/2)

                and (y+square\_heigth >= apple\_y+apple\_heigth/2) and (y <= apple\_y+apple\_heigth/2)) or (k == 0):

*# Generate apple coords*

*while* (x+square\_width >= apple\_x+apple\_width/2) and (x <= apple\_x+apple\_width/2) \

                    and (y+square\_heigth >= apple\_y+apple\_heigth/2) and (y <= apple\_y+apple\_heigth/2):

                apple\_x = random.randrange(width-apple\_width)

                apple\_y = random.randrange(heigth-apple\_heigth)

            k += 1

*# s.play()*

*if* k == 50:

                dog\_surf = pygame.image.load('dog.png')

                dog\_rect = dog\_surf.get\_rect(*width*=width)

                screen.blit(dog\_surf, dog\_rect)

                pygame.display.update()

            print(f"Вы съели яблоко. Итого: {k}")

        pygame.draw.rect(screen, color\_black, pygame.Rect(

            apple\_last\_x, apple\_last\_y, apple\_width, apple\_heigth))

        pygame.draw.rect(screen, color\_red, pygame.Rect(

            apple\_x, apple\_y, apple\_width, apple\_heigth))

        apple\_last\_x = apple\_x

        apple\_last\_y = apple\_y

        pygame.draw.rect(screen, color\_black, pygame.Rect(

            x\_last, y\_last, square\_width, square\_heigth))

        pygame.draw.rect(screen, color\_white, pygame.Rect(

            x, y, square\_width, square\_heigth))

        x\_last = x

        y\_last = y

        pygame.time.wait(7)

        pygame.display.flip()

def main():

    r = Rectangle("синего", 15, 15)

    c = Circle("зеленого", 15)

    s = Square("красного", 15)

    print(r)

    print(c)

    print(s)

*if* input("Запустить игру? (1/0)") == "1":

        py\_game()

*if* \_\_name\_\_ == "\_\_main\_\_":

    main()

1. Экранные формы с примерами выполнения программы.

