

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”



Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по лабораторной работе №4
“Объектно-ориентированное программирование на Kotlin”

Выполнил:
Студент группы ИУ5-34Б
Жеребенков А.Ю.
Преподаватель:
Нардид А.Н.

Москва 2025

Задание

1. Необходимо разработать программу, реализующую работу с классами. Программа должна быть разработана в виде консольного приложения на языке Kotlin.
2. Все файлы проекта (кроме основного файла main.py) должны располагаться в пакете lab_Kotlin_oop.
3. Каждый из нижеперечисленных классов должен располагаться в отдельном файле пакета lab_Kotlin_oop.
4. Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры. Подробнее про абстрактные классы и методы Вы можете прочитать [здесь](#).
5. Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры. Подробнее про описание свойств Вы можете прочитать [здесь](#).
6. 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.
7. Класс «Круг» создается аналогично классу «Прямоугольник», задается параметр «радиус». Для вычисления площади используется константа math.pi из модуля [math](#).
8. 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.

9. В корневом каталоге проекта создайте файл main.ru для тестирования Ваших классов (используйте следующую конструкцию - https://docs.Kotlin.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.

Код(Main.kt)

```
import lab_kotlin_oop.Circle
import lab_kotlin_oop.Rectangle
import lab_kotlin_oop.Square

fun main() {
    val rectangle = Rectangle(3.0, 4.0, "синий")
    val circle = Circle(5.0, "красный")
    val square = Square(2.0, "зеленый")

    println(rectangle)
    println(circle)
    println(square)
}
```

Код Circle.kt

```
package lab_kotlin_oop

import kotlin.math.PI
import kotlin.math.pow

class Circle(private val radius: Double, colorName: String) : Figure() {
    override val name = "Круг"
    private val color = Color(colorName)

    override fun area(): Double {
        return PI * radius.pow(2)
    }

    override fun toString(): String {
```

```
        return "Фигура: $name, радиус: ${radius}, цвет: ${color.color}, площадь: ${area()}"  
    }  
}
```

Код Rectangle.kt

```
package lab_kotlin_oop  
  
open class Rectangle(protected val width: Double, private val height: Double, colorName: String) : Figure() {  
    override val name = "Прямоугольник"  
    val color = Color(colorName)  
  
    override fun area(): Double {  
        return width * height  
    }  
  
    override fun toString(): String {  
        return "Фигура: $name, ширина: $width, высота: $height, цвет: ${color.color}, площадь: ${area()}"  
    }  
}
```

Код Figure.kt

```
package lab_kotlin_oop  
  
abstract class Figure {  
    abstract val name: String  
    abstract fun area(): Double  
    abstract override fun toString(): String  
}
```

Код Color.kt

```
package lab_kotlin_oop  
  
data class Color(val color: String)
```

Код square.py

```
package lab_kotlin_oop  
  
class Square(side: Double, colorName: String) : Rectangle(side, side, colorName) {  
    override val name = "Квадрат"
```

```
override fun toString(): String {
    val side = width
    return "Фигура: $name, ширина: $side, цвет: ${color.color}, площадь: ${area()}"
}
}
```

Работа приложения

```
C:\Users\ADM\jdk8\openjdk-25\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.2\lib\idea_rt.jar=56339" -Dfile.encoding=UTF-8
Фигура: Прямоугольник, ширина: 3.0, высота: 4.0, цвет: синий, площадь: 12.0
Фигура: Круг, радиус: 5.0, цвет: красный, площадь: 78.53981633974483
Фигура: Квадрат, ширина: 2.0, цвет: зеленый, площадь: 4.0

Process finished with exit code 0
```