

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

Факультет Радиотехнический  
Кафедра “Системы обработки информации и управления”

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №2

Выполнил:  
студент группы РТ5-31Б:  
Стукановский Максим  
Владимирович  
Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Юрий  
Евгеньевич  
Подпись и дата:

Москва, 2025 г.

### 1. Постановка задачи:

Написать программу, которая будет реализовывать работу с классами по следующим пунктам:

- Создать абстрактный класс «Геометрическая фигура» для вычисления площади фигуры, класс «Прямоугольник», который наследуется от класса «Геометрическая фигура», класс «Квадрат», наследуемый от класса «Прямоугольник», класс «Круг», который наследуется от класса «Геометрическая фигура»;
- Для классов «Прямоугольник», «Квадрат», «Круг» переопределить виртуальный метод `Object.ToString()`;
- Разработать интерфейс `IPrint`. Для классов «Прямоугольник», «Квадрат», «Круг» реализовать наследование от интерфейса `IPrint`.

Лабораторная работа была выполнена на языке программирования «Rust», поэтому некоторые требования были изменены в силу невозможности их реализации в данном языке программирования

## 2. Код программы:

```
#![allow(unused)]

trait GeometricalShape {
    fn find_area(&self) -> f64;
}

// rectangle --point
struct Rectangle {
    name: String,
    width: f64,
    height: f64,
}
impl GeometricalShape for Rectangle {
    fn find_area(&self) -> f64 {
        self.width * self.height
    }
}
impl Rectangle {
    fn print_info(&self) {
        println!("Фигура: {}, Ширина: {}, Длина: {}, Площадь: {}",
            self.name,
            self.height,
            self.width,
            self.find_area()
        );
    }
}
// rectangle --endpoint

// Square --point
struct Square {
    name: String,
    side: f64,
}
impl GeometricalShape for Square {
    fn find_area(&self) -> f64 {
        self.side*self.side
    }
}
impl Square {
    fn print_info(&self) {
        println!("Фигура: {}, Сторона: {}, Площадь {}",
            self.name,
            self.side,
            self.find_area()
        );
    }
}
// Square --endpoint

// circle --point
static PI: f64 = 3.141592;
struct Circle {
    name: String,
    radius: f64,
}
impl GeometricalShape for Circle {
    fn find_area(&self) -> f64 {
        PI*self.radius*self.radius
    }
}
impl Circle {
    fn print_info(&self) {
        println!("Фигура: {}, Радиус: {}, Площадь: {}",
```

```

        self.name,
        self.radius,
        self.find_area()
    );
}
}
// circle --endpoint

enum Shape {
    Rectangle(Rectangle),
    Square(Square),
    Circle(Circle),
}
impl Shape {
    fn print_info(&self) {
        match self {
            Shape::Rectangle(r) => r.print_info(),
            Shape::Square(s) => s.print_info(),
            Shape::Circle(c) => c.print_info(),
        }
    }
}

fn main() {
    let shapes: Vec<Shape> = vec! [
        Shape::Rectangle(Rectangle { name: "Прямоугольник".to_string(), width:
12.0, height: 3.9 }),
        Shape::Square(Square {name: "Квадрат".to_string(), side: 11.8}),
        Shape::Circle(Circle { name: "Круг".to_string(), radius: 4.3 })
    ];

    for shape in shapes {
        shape.print_info();
        println!("-----");
    }
}

```

3. Анализ результатов:

Фигура: Прямоугольник, Ширина: 3.9, Длина: 12, Площадь: 46.8

-----

Фигура: Квадрат, Сторона: 11.8, Площадь 139.24

-----

Фигура: Круг, Радиус: 4.3, Площадь: 58.08803608

-----