**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №2

# «Объектно-ориентированные возможности языка Go»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-35Б |  | Гапанюк Ю. Е. |
| Бердников Н.О. |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 г

**Описание программы**Программа предназначена для работы с классами и их наследованием. Она выполняет задачу, связанную с объектно-ориентированным программированием (ООП) и состоит из классов, реализующих геометрические фигуры (прямоугольник, квадрат, круг). Каждый класс имеет методы для вычисления площади фигуры и представления её свойств, включая цвет.

Структура программы:

Абстрактный класс «Shape»:

В этом классе объявлен абстрактный метод для вычисления площади. Конкретные фигуры (прямоугольник, круг, квадрат) наследуют этот класс и реализуют его метод.

Класс «Color»:

Данный класс содержит свойство для описания цвета каждой геометрической фигуры. Цвет передаётся как параметр при создании объектов фигур и сохраняется внутри объекта.

Класс «Rectangle»:

Этот класс наследует «Shape» и представляет собой прямоугольник с определёнными шириной, высотой и цветом. Для вычисления площади он переопределяет абстрактный метод из базового класса.

Класс «Circle»:

Аналогично классу «Rectangle», этот класс описывает фигуру круга. Радиус круга передаётся как параметр.

Класс «Square»:

Класс «Square» наследуется от «Rectangle». Он использует длину стороны в качестве единственного параметра для вычисления площади.

**Текст программы:**

**Main.go:**  
package main

import (

"fmt"

"lab2/classes"

)

func printShapeInfo(shape classes.Shape) {

fmt.Println(shape)

}

func main() {

n := 2.0

rect := classes.NewRectangle(n, n, classes.NewColor("blue"))

circ := classes.NewCircle(n, classes.NewColor("green"))

sq := classes.NewSquare(n, classes.NewColor("red"))

printShapeInfo(rect)

printShapeInfo(circ)

printShapeInfo(sq)

}

**Shape.go:**  
package classes

type Shape interface {

Area() float64

String() string

}

**Color.go:**  
package classes

type Color struct {

Name string

}

func NewColor(name string) \*Color {

return &Color{Name: name}

}

func (c \*Color) String() string {

return c.Name

}

**Rectange.go:**  
package classes

import "fmt"

type Rectangle struct {

Width float64

Height float64

Color \*Color

}

func NewRectangle(width, height float64, color \*Color) \*Rectangle {

return &Rectangle{Width: width, Height: height, Color: color}

}

func (r \*Rectangle) Area() float64 {

return r.Width \* r.Height

}

func (r \*Rectangle) String() string {

return fmt.Sprintf("Rectangle(width=%.2f, height=%.2f, color=%s, area=%.2f)", r.Width, r.Height, r.Color, r.Area())

}

**Circle.go:**  
package classes

import (

"fmt"

"math"

)

type Circle struct {

Radius float64

Color \*Color

}

func NewCircle(radius float64, color \*Color) \*Circle {

return &Circle{Radius: radius, Color: color}

}

func (c \*Circle) Area() float64 {

return math.Pi \* c.Radius \* c.Radius

}

func (c \*Circle) String() string {

return fmt.Sprintf("Circle(radius=%.2f, color=%s, area=%.2f)", c.Radius, c.Color, c.Area())

}

**Square.go:**package classes

import "fmt"

type Square struct {

Side float64

Color \*Color

}

func NewSquare(side float64, color \*Color) \*Square {

return &Square{Side: side, Color: color}

}

func (s \*Square) Area() float64 {

return s.Side \* s.Side

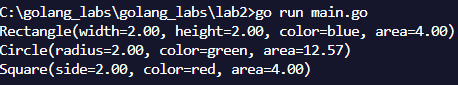
}

func (s \*Square) String() string {

return fmt.Sprintf("Square(side=%.2f, color=%s, area=%.2f)", s.Side, s.Color, s.Area())

}

**Результат вывода:**

****

**Использованные источники:**

1. [Электронный ресурс] Курс ПиКЯП (Парадигмы и конструкции языков программирования). URL:<https://github.com/ugapanyuk/courses_content/wiki/COURSE_PCPL_MAIN>