МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Лабораторная работа №4**

по дисциплине: Объектно-ориентированное программирование

тема: «Классы»

Выполнил: ст. группы ПВ-233

Ситников Алексей Павлович

Проверил:

Белгород 2025 г.

**Цель работы:** приобретение практических навыков создания класса на языке С++.

Вариант 3 (13)

**Задание 1.**

Выполнить построение объектной модели следующей предметной области: “Простой тетрис”. (простым тетрисом называется такой тетрис, в котором нет ломаных фигур, т. е. все фигуры квадраты, прямоугольники)

Изображение выглядит как текст, диаграмма, линия, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

**Задание 2.**

Создать класс Rectangle, разработав следующие элементы класса:

а. Поля:

int a, b;

б. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.

в. Методы, позволяющие:

вывести длины сторон прямоугольника на экран;

рассчитать периметр прямоугольника;

рассчитать площадь прямоугольника.

г. Перегрузить (переопределить):

сравнение(меньше);

сложение(бинарное).

Заголовочный файл:

#ifndef **UNTITLED10\_RECTANGLE\_H**#define **UNTITLED10\_RECTANGLE\_H**class Rectangle {  
 int a;  
 int b;  
 public:  
 Rectangle(int a, int b);  
 void PrintSize();  
 int Perimeter();  
 int Square();  
 friend bool operator**<**(Rectangle& rect1, Rectangle& rect2);  
 friend Rectangle operator**+**(const Rectangle& rect1, const Rectangle& rect2);  
};  
#endif //UNTITLED10\_RECTANGLE\_H

Реализация методов класса:

#include <iostream>  
#include "Rectangle.h"  
  
Rectangle::Rectangle(int a, int b){  
 this->a = a;  
 this->b = b;  
}  
  
void Rectangle::PrintSize(){  
 std::cout **<<** a **<<** "x" **<<** b **<<** std::endl;  
}  
  
int Rectangle::Perimeter(){  
 return 2 \* (a + b);  
}  
int Rectangle::Square(){  
 return a\*b;  
}  
  
bool operator**<**(Rectangle& rect1, Rectangle& rect2) {  
 return rect1.Square() < rect2.Square();  
}  
  
Rectangle operator**+**(const Rectangle& rect1, const Rectangle& rect2) {  
 return Rectangle(rect1.a + rect2.a, rect1.b + rect2.b);  
}

main.cpp

#include <iostream>  
#include "cmake-build-debug/Rectangle.h"  
  
int main() {  
 Rectangle A(10, 15);  
 Rectangle B(11,8);  
  
 A.PrintSize();  
 B.PrintSize();  
  
 int a = A.Square();  
 std::cout **<<** a **<<** std::endl;  
 a = A.Perimeter();  
 std::cout **<<** a **<<** std::endl;  
  
 a = B.Square();  
 std::cout **<<** a **<<** std::endl;  
 a = B.Perimeter();  
 std::cout **<<** a **<<** std::endl;  
  
 if(A **<** B){  
 std::cout **<<** true;  
 }  
 else{  
 std::cout **<<** false;  
 }  
  
 Rectangle N = A **+** B;  
 std::cout **<<** std::endl;  
 N.PrintSize();  
 a = N.Square();  
 std::cout **<<** a **<<** std::endl;  
 a = N.Perimeter();  
 std::cout **<<** a **<<** std::endl;  
 return 0;  
}

Вывод программы:  
Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

**Вывод:** в ходе проделанной работы я научился создавать классы, создавать методы класса и перегружать операторы.