**МИНОБРНАУКИ РОССИИ**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

Дисциплина: Обьектно-ориентированное программирование

по теме «Классы»

Выполнил: ст. группы ВТ-22  
Ковалев Павел ВТ-22

Проверил: Буханов Д.Г.

Белгород 2019

Вариант 3

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

**Задание**: Выполнить построение диаграммы объектов в соответствии с заданием варианта (табл. 1). По заданному описанию класса в соответствии с вариантом создать класс на языке С++ и переопределить указанные операции.

|  |  |
| --- | --- |
| 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) | 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). |

<main.cpp>

#include <QCoreApplication>

#include <iostream>

using namespace std;

const int CAPACITY = 32; // разрядность

// Получить бит по номеру позиции

bool **getBit**(int value, int position) {

return (value >> position) & 1;

}

// Установить значение бита по номеру позиции

int **setBit**(int value, int position, bool bit) {

int mask = 1 << position;

if(bit) return value | mask; // ставим 1

return value & ~mask; // ставим 0

}

// Сложение чисел побитово

double **addNumbers**(int num1, int num2, int position) {

if(position > CAPACITY) return num1;

// Получаем биты

bool bit1 = getBit(num1, position);

bool bit2 = getBit(num2, position);

if(bit1 && bit2) {

// Если 1 и 1, то заполняем нулями, пока есть единицы

int pos = position;

do {

num1 = setBit(num1, pos++, 0);

} while (getBit(num1, pos));

num1 = setBit(num1, pos, 1);

return addNumbers(num1, num2, position+1);

} else {

num1 = setBit(num1, position, bit1 | bit2);

return addNumbers(num1, num2, position+1);

}

}

class **Rectangle**

{

double a;

double b;

public:

**Rectangle**() {

cout << "Read side A:";

cin >> a;

cout << "Read side B:";

cin >> b;

}

void **setASide**(double t){

a = t;

}

void **setBSide**(double t){

b = t;

}

void **setSideRectangle**(double t1,double t2){

setASide(t1);

setBSide(t1);

}

void **viewRectangleSide**(){

cout << "Side A:" << a << " Side B:" << b << endl;

}

double **perimeterCalculation**(){

double P;

P = (a+b)\*2;

cout << "Perimeter:" << P << endl;

return P;

}

double **areaCalculation**(){

double S;

S = a\*b;

cout << "Area:" << S << endl;

return S;

}

void **newRectangle**(){

cout << "Read side A:";

cin >> a;

cout << "Read side B:";

cin >> b;

}

friend Rectangle operator <(Rectangle& A,Rectangle& B){

Rectangle temp;

double PA = A.perimeterCalculation();

double PB = B.perimeterCalculation();

double SA = A.areaCalculation();

double SB = B.areaCalculation();

if((PA<PB)&&(SA<SB)){

temp.a = A.a;

temp.b = A.b;

} else {

temp.a = B.a;

temp.b = B.b;

}

}

friend Rectangle operator &(Rectangle& A,Rectangle& B){

Rectangle temp;

temp.a = addNumbers(A.a,B.a,1);

temp.b = addNumbers(A.b,B.b,1);

return temp;

}

};

int **main**(int argc, char \*argv[])

{

QCoreApplication app(*argc*, argv);

Rectangle A;

Rectangle B;

A = A&B;

A.viewRectangleSide();

return app.exec();

}