/\*Описать заданные по варианту классы (содержащие private поля и методы). Для

каждого класса описать конструктор по умолчанию и конструктор с параметрами, а

также деструктор (по необходимости). Создать функцию, дружественную обоим

классам, и в ней обратиться к их закрытым полям и методам.

Вариант 4

Создать два класса: Число (int) и Матрица (int \*\*). Описать дружественную

функцию, проверяющую на равенство число и сумму элементов на главной диагонали

матрицы.

\*/

#include <iostream>

#include<string>

using namespace std;

class matriza;

class number{

int numb;

void show(){

cout<<"Вы ввели число: \n"<<numb<<endl;

};

public:

number(){

cout<<"Без параметров\n";

};

number(int a){

numb=a;

show();

};

friend bool A(number ob1,matriza ob2);

~number(){

cout<<"Destructor Number\n"<<endl;

}

};

class matriza{

int matr,size,mass[10][10];

void show(){

cout<< "Сумма элементов на главной диагонали матрицы: "<<matr<<endl;

};

public:

matriza(){

cout<<"Без параметров\n";

}

matriza(int b){

int i,j;

matr=0;

size=b;

mass[b][b];

cout<<"Введите "<<size\*size<<" чисел\n";

for(i=0; i<b;i++){

for( j=0;j<b;j++){

cin>> mass[i][j];

if(i==j)

matr+=mass[i][j];

}

}

cout<< "Матрица:\n";

for( i=0;i<size;i++){

for( j=0;j<size;j++)

cout<<mass[i][j]<<" ";

cout<<endl;

}

show();

}

friend bool A(number ob1,matriza ob2);

~matriza(){

cout<<"Destructor Matriza"<<endl;

}

};

bool A(number ob1,matriza ob2){

if(ob1.numb == ob2.matr)return true;

else return false;

}

int main()

{

setlocale(LC\_ALL, "Russian");

number ob1(12);

matriza ob2(3);

if (A(ob1,ob2)) cout<<"Равно\n";

else cout <<"Не Равно\n";

return 0;

}