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

по теме: «Классы»

Выполнил: ст. группы ПВ202

Буйвало Анастасия Андреевна

Проверил: Картамышев С.В

Белгород 2022 г.

# Лабораторная работа №4

**«Классы»**

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

**Задание:** выполнить построение диаграммы объектов (не менее 7-ми 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.

1. Графический редактор простых фигур

Изображение выглядит как текст

Автоматически созданное описание

1. Создать класс для работы с двумерным массивом вещественных чисел. Разработать следующие элементы класса:

а. Поля:

double \*\*ptr;

int n, m.

б. Конструктор, позволяющий создать массив размерности n×m.

в. Методы, позволяющие:

ввести элементы массива с клавиатуры;

вывести элементы массива на экран;

отсортировать элементы каждой строки массива в порядке убывания.

г. Перегрузить (переопределить):

умножение

обращение к элементу массива

interface\_class.h

//  
// Created by настя буйвало on 23/02/2022.  
//  
  
#ifndef OOP4\_INTERFACE\_CLASS\_H  
#define OOP4\_INTERFACE\_CLASS\_H  
#include <iostream>  
  
class Arr{  
public:  
 static void sort(double\*, int);  
};  
  
class Matr{  
 double \*\*ptr;  
 int n, m;  
public:  
 Matr(int, int);  
 void set\_Matr();  
 void get\_Matr();  
  
 void sort\_Matr();  
  
 Matr& operator\* (Matr& right);//умножение  
 double operator()(int, int); //обращение  
};  
  
  
#endif //OOP4\_INTERFACE\_CLASS\_H

realize\_class.cpp

//  
// Created by настя буйвало on 23/02/2022.  
//  
#include "interface\_class.h"  
  
//обмен  
void swap(double\* a, double\* b)  
{  
 double t = \*a;  
 \*a = \*b;  
 \*b = t;  
}  
  
void Arr::sort(double \*ptr, int n) //в порядке убывания  
{  
 for(int i = 0; i < n - 1; i++){  
 int imin = i;  
 for(int j = i + 1; j < n; j++) {  
 if (ptr[j] > ptr[imin]) {  
 imin = j;  
 }//находим индекс минимального элемента  
 }  
 swap(&ptr[i], &ptr[imin]);// обмениваем минимальный элемент с текущим  
 }  
}  
  
//конструктор  
Matr:: Matr(int count\_str, int count\_col)  
{  
 m = count\_str;  
 n = count\_col;  
 ptr = new double\*[m];  
 for(int i = 0; i < n; i++)  
 ptr[i] = new double[n];  
};  
  
//ввод с клавиатуры  
void Matr::set\_Matr()  
{  
 for(int i = 0; i < m; i++)  
 for(int j = 0; j < n; j++)  
 std:: cin >> ptr[i][j];  
}  
  
//вывод  
void Matr::get\_Matr()  
{  
 for(int i = 0; i < m; i++) {  
 for (int j = 0; j < n; j++)  
 std::cout << ptr[i][j] << " ";  
 std:: cout << "\n";  
 }  
}  
  
void Matr::sort\_Matr()  
{  
 for(int i = 0; i < m; i++)  
 Arr:: sort(ptr[i], n);  
}  
  
//перегрузка умножение матриц  
Matr& Matr::operator\*(Matr &right)  
{  
 Matr res(m, n);  
 for (size\_t i = 0; i < m; i++) {  
 for (size\_t j = 0; j < n; j++) {  
 res.ptr[i][j] = 0;  
 for (size\_t k = 0; k < n; k++)  
 res.ptr[i][j] = res.ptr[i][j] + (ptr[i][k] \* right.ptr[k][j]);  
 }  
 }  
 return res;  
}  
  
double Matr::operator()(int i, int j)  
{  
 return(ptr[i][j]);  
}

main.cpp

#include <iostream>  
#include "interface\_class.h"  
  
int main() {  
 int n, m;  
 std::cout << "Введите кол-во строк и столбцов ";  
 std:: cin >> m >> n;  
  
 std::cout << "Введите матрицу\n";  
 Matr matr1(m, n);  
 matr1.set\_Matr();  
 matr1.get\_Matr();  
  
 std::cout << "Введите матрицу\n";  
 Matr matr2(m, n);  
 matr2.set\_Matr();  
 matr2.get\_Matr();  
 std:: cout << "\n";  
  
 Matr matr3(m, n);  
 matr3 = matr1 \* matr2;  
 std:: cout << "Результат умножения матриц:\n";  
 matr3.get\_Matr();  
  
 std:: cout << "Отсортированная по убыванию матрица:\n";  
 matr3.sort\_Matr();  
 matr3.get\_Matr();  
  
 std:: cout << "\nmatr3[0][0] = " << matr3(0, 0);  
 return 0;