МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Белорусский государственный университет транспорта»

Кафедра «Информационно-управляющие системы и технологии»

Отчет  
по лабораторным работам

по дисциплине «Объектно-ориентированное программирование»

Выполнил Проверил

студент группы ГИ-21 ст. преп. каф. «ИУСиТ»

Зайцев И. С. Голдобина Т. А.

Гомель, 2020

Оглавление

[1 Лабораторная работа № 1 Тема: «Организация ввода/вывода, динамическое выделение памяти» 3](#_Toc51230999)

[1.1 Задания 3](#_Toc51231000)

[1.2 Контрольные вопросы 5](#_Toc51231001)

# Лабораторная работа № 1 Тема: «Организация ввода/вывода, динамическое выделение памяти»

### **Цель**

Изучить организацию ввода/вывода и работу с динамической памятью при программировании алгоритмов в C++.

## Задания

### Задание 1.1

### Условие

Реализовать функции для работы с матрицами:

а) функция производит перемножение матриц;

б) функция производит сложение двух матриц.

Память под матрицы выделять динамически. Необходимо освобождать память, выделенную под матрицы.

Программный код

#include <iostream>

#include <cstdlib>

using namespace std;

int addition()

{

int row1, row2, col1, col2, i, j;

cout << "\nВведите количество строк первой матрицы(сложение): ";

cin >> row1;

cout << "Введите количество столбцов первой матрицы(сложение): ";

cin >> col1;

cout << "Введите количество строк второй матрицы(сложение): ";

cin >> row2;

cout << "Введите количество столбцов второй матрицы(сложение): ";

cin >> col2;

int Mass[row1][col1], Mass2[row2][col2], MassRes[row1][col2];

if (row1 != row2 || col1 != col2)

{

cout << "\nСложение невозможно\n";

return 0;

}

for (i=0; i<row1; i++)

{

for (j=0; j<col1; j++)

{

Mass[i][j]=rand() % 25;

cout << Mass[i][j] << "\t";

}

cout << endl;

}

cout << "\n";

for (i=0; i<row2; i++)

{

for (j=0; j<col2; j++)

{

Mass2[i][j]=rand() % 25;

cout << Mass2[i][j] << "\t";

}

cout << endl;

}

cout << "\nРезультат сложения:\n";

for (i=0; i<row1; i++)

{

for(j=0; j<col2; j++)

{

MassRes[i][j]=Mass[i][j]+Mass2[i][j];

cout << MassRes[i][j] << "\t";

}

cout << endl;

}

}

int multiplication()

{

int row1, row2, col1, col2, i, j;

cout << "\nВведите количество строк первой матрицы(умножение): ";

cin >> row1;

cout << "Введите количество столбцов первой матрицы(умножение): ";

cin >> col1;

cout << "Введите количество строк второй матрицы(умножение): ";

cin >> row2;

cout << "Введите количество столбцов второй матрицы(умножение): ";

cin >> col2;

if (col1 != row2)

{

cout << "Умножение невозможно!";

return 0;

}

int Mass[row1][col1], Mass2[row2][col2], MassRes[row1][col2];

for (int i = 0; i < row1; i++)

{

for (int j = 0; j < col1; j++)

{

Mass[i][j]=rand() %50;

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

}

cout << endl;

}

cout << "\n";

for (int i = 0; i < row2; i++)

{

for (int j = 0; j < col2; j++)

{

Mass2[i][j]=rand() %50;

cout << Mass2[i][j] << " ";

}

cout << endl;

}

for (int i = 0; i < row1; i++)

{

for (int j = 0; j < col2; j++)

{

MassRes[i][j] = 0;

for (int k = 0; k < col1; k++)

MassRes[i][j] += Mass[i][k] \* Mass2[k][j];

}

}

cout << "Матрица произведения:" << endl;

for (int i = 0; i < row1; i++)

{

for (int j = 0; j < col2; j++)

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

cout << endl;

}

}

int main()

{ int key;

setlocale(LC\_ALL,"");

cout << "Сложение матриц(1);\nУмножение матриц(2);\nВыход(0);";

while(key!=0)

{

cout << "\nВведите номер: ";

cin >> key;

if(key==1)

{

addition();

}

else if(key==2)

{

multiplication();

}

else{return 0;}

}

return 0;

}

### Результат

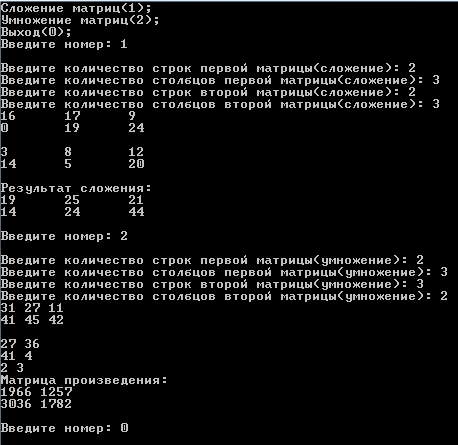


Рисунок – Результат задания 1

## Контрольные вопросы

1. Как осуществляется ввод/вывод в C++?

В C++ ввод и вывод данных производится потоками байт. Поток (последовательность байт) – это логическое устройство, которое выдает и принимает информацию от пользователя и связано с физическими устройствами ввода/вывода. При операциях ввода байты направляются от устройства в основную память. В операциях вывода – наоборот.

Для ввода информации с клавиатуры используется поток cin, для вывода – cout.

1. Что такое манипулятор ввода/вывода.

Манипулятор – объект особого типа, который управляет потоками ввода/вывода, для форматирования передаваемой в потоки информации. Манипуляторы помещаются в операторы ввода/вывода непосредственно перед форматируемым значением.

1. Для чего необходимы операторы new и delete? В чем их отличие от функций malloc() и free()?

Операторы new и delete используют для выделения и освобождения памяти соответственно.

С указателями на объект нельзя использовать функции выделения и освобождения динамической памяти malloc() и free() , так как выделение памяти под объект не вызывает его конструктор, в результате возникает неинициализированный объект, который ведет себя непредсказуемо, поэтому в C++ используются операторы new и delete.

1. Как создать и удалить массив объектов?

Для создания массива объектов используется оператор new, объявляется следующим образом:

new имя\_типа [выражение];

Для удаления используется оператор delete, объявляется следующим образом:

delete [ ] переменная\_указатель;

1. Как можно выделить память под двумерный массив?

Используя «указатель на указатель», в языке C++ допустимо объявлять переменные, имеющие тип «указатель на указатель». Объявляется он следующим образом:

int \*\*mass;

### Вывод по практической работе

Изучил организацию ввода/вывода и работал с динамической памятью при программировании алгоритмов в C++.