МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Гродненский государственный университет имени Янки Купалы»

Факультет математики и информатики

**Кафедра современных технологий программирования**

Филипьев Валерий Дмитриевич

**Программирование действий над матрицами на языке С++**

Курсовая работа

студента 1 курса специальности

«Программное обеспечение информационных технологий»

дневной формы получения образования

Научный руководитель

Домбровский Павел Эдуардович,

преподаватель кафедры

современных технологий

программирования

Гродно, 2017

**РЕЗЮМЕ**

Тема курсовой работы

**Матрицы на языке С++**

Работа содержит: 24 страницы, 5 использованных источника литературы.

**Ключевые слова**: матрицы, С++.

**Цель курсовой работы**: разработка калькулятора матриц

**Объект исследования**: матрицы

**Предмет исследования**: реализация компьютерного решения «Матриц» на ЭВМ с помощью языка программирования С++.

В работе были использованы следующие методы: сравнительный, анализ нормативно- правовой базы, анализ монографий и статей, аналитический.

Авторская характеристика работы.

«Матрицы» написаны на языке С++ и представляет собой простую реализацию, разработанную программой Microsoft Visual Studio.

**SUMMARY**

Topic of the course work

**Development of a computer matrix**

The work contains: 23 pages, 5 sources of literature used.

Keywords: matrices, C ++.

The purpose of the course work: the development of a matrix calculator

Object of study: matrices

The subject of the research: implementation of the computer solution of the Matrix on a computer using the C ++ programming language.

The following methods were used in the work: comparative, analysis of normative and legal base, analysis of monographs and articles, analytical.

Author's characteristic of the work.

"Matrices" are written in C ++ and is a simple implementation developed by Microsoft Visual Studio.

СОДЕРЖАНИЕ

[**ВВЕДЕНИЕ 10**](#_Toc482740448)

[**1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ИЗУЧЕНИЯ ПРОГРАММЫ Матрицы 13**](#_Toc482740449)

[1.1 Основные действия матриц 13](#_Toc482740450)

[1.2 Понятие основных элементов матриц 16](#_Toc482740451)

[1.3 Выводы по главе I 18](#_Toc482740452)

[**2. РАЗРАБОТКА ДЕЙСТВИЙ НАД «Матрицы» 19**](#_Toc482740453)

[2.1 Постановка задач для реализации «Матрицы» 19](#_Toc482740454)

[2.3 Вывод по главе II 1](#_Toc482740456)6

3. [АЛГОРИТМ НАПИСАНИЯ ПРОГРАММЫ «Матрицы». 21](#_Toc482740455)

[3.1 Вывод по главе III](#_Toc482740456) 25

[**ЗАКЛЮЧЕНИЕ 2**](#_Toc482740457)**6**

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 2**](#_Toc482740458)**8**

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

(СОКРАЩЕНИЙ, ТЕРМИНОВ)

ЭВМ – Электронно- вычислительная машина.

С++ – Язык программирования.

Матрицы ­– Упорядоченная запись коэффициентов уравнения при неизвестных.

# ВВЕДЕНИЕ

В наше время, когда развиваются числовые технологические процессы, персональные компьютеры всё более включатся в жизнедеятельность человека. Прежде ЭВМ применялись исключительно с целью трудных точных вычислений. А на сегодняшний день область их употребления значительно расширилась. Благодаря различным математическим алгоритмам мы можем решить какую-либо задачу в нужной нам интерпретации.

**Матрицы** широко применяются в математике для компактной записи систем линейных алгебраических или дифференциальных уравнений. В этом случае количество строк матрицы соответствует числу уравнений, а количество столбцов — количеству неизвестных. В результате решение систем линейных уравнений сводится к операциям над матрицами.

**Актуальность курсовой работы** заключается в создании не сложной программы-калькулятора, составленной на языке С++ , которую сможет использовать каждый пользователь, не смотря на маломощное оборудование своего компьютера.

**Главная задача** курсовой работы заключается в правильности составленных алгоритмов и функций компонентов. Потребуется создать полноценную правильно - функционирующую программу в среде C++ и самостоятельно изучить некоторые процедуры и функции.

**Объект исследования:** компьютерная модель «Матрицы».

**Предмет исследования:** реализация модели «Матрицы» на ЭВМ с помощью языка программирования С++

**Цель исследования:** разработка учебной модели «Матрицы».

В соответствии с объектом, предметом и целью исследования были поставлены следующие задачи:

* Применение теоретических знаний по дисциплине «Языки программирования» и по темам, вынесенным на самостоятельное изучение, на практике для создания программного проекта, реализующего модель учебного и логического характера.
* Получение необходимых практических навыков прикладного программирования.
* Приобретение навыков алгоритмического мышления и работы с оформлением документации.

**Методы исследования:** сравнительный, анализ нормативно-правовой базы, анализ монографий и статей, аналитический.

Курсовая работа состоит из двух частей: основной и специальной. В основной части рассказана про матрицы, компоненты, используемые, при написании программы, также, описаны их свойства и возможности. Специальная часть состоит из описания программы. Завершает курсовую работу заключение и список используемой литературы, которая была использована при составлении курсовой работы и программы.

**Общий объем курсовой работы** составляет 21 страница. Список использованной литературы состоит из 4 источников.

## 1. **ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ИЗУЧЕНИЯ ПРОГРАММЫ**

### **1.1 Матрицы**

Ма́трица — математический объект, записываемый в виде прямоугольной таблицы чисел (или элементов кольца) и допускающий алгебраические операции (сложение, вычитание, умножение) между ним и другими подобными объектами. Обычно матрицы представляются двумерными (прямоугольными) таблицами. Иногда рассматривают многомерные матрицы или матрицы непрямоугольной формы.

## *Операции над матрицами*

Пусть aij — элементы матрицы A, а bij — элементы матрицы B.

Линейные операции:

Умножение матрицы A на число λ (обозначение: λA) заключается в построении матрицы B, элементы которой получены путём умножения каждого элемента матрицы A на это число, то есть каждый элемент матрицы B равен

bij = λaij

Сложение матриц A + B есть операция нахождения матрицы C, все элементы которой равны попарной сумме всех соответствующих элементов матриц A и B, то есть каждый элемент матрицы C равен

cij = aij + bij

A+B=\begin{pmatrix} 2 & 0 & -1\\ 1 & 3 & 0\end{pmatrix}+\begin{pmatrix} 3 & 1 & 0\\ 8 & 2 & 3\end{pmatrix}=\begin{pmatrix}2+3 & 0+1 & -1+0\\1+8 & 3+2 & 0+3\end{pmatrix}=\begin{pmatrix}5 & 1 & -1\\9 & 5 & 3\end{pmatrix}

Вычитание матриц A − B определяется аналогично сложению, это операция нахождения матрицы C, элементы которой

cij = aij - bij

A-B=\begin{pmatrix} 2 & 0 & -1\\ 1 & 3 & 0\end{pmatrix}-\begin{pmatrix} 3 & 1 & 0\\ 8 & 2 & 3\end{pmatrix}=\begin{pmatrix}2-3 & 0-1 & -1-0\\1-8 & 3-2 & 0-3\end{pmatrix}=\begin{pmatrix}-1 & -1 & -1\\-7 & 1 & -3\end{pmatrix}

Сложение и вычитание допускается только для матриц одинакового размера.

Существует нулевая матрица Θ такая, что её прибавление к другой матрице A не изменяет A, то есть

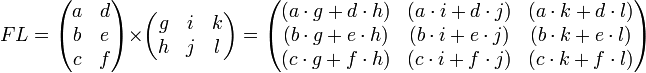
A + Θ = A

Все элементы нулевой матрицы равны нулю.

Умножение матриц (обозначение: AB, реже со знаком умножения A\times B) — есть операция вычисления матрицы C, элементы которой равны сумме произведений элементов в соответствующей строке первого множителя и столбце второго.

\! c_{ij} = \sum^n_{k=1} a_{ik} b_{kj}

В первом множителе должно быть столько же столбцов, сколько строк во втором. Если матрица A имеет размерность m \times n, B — n \times k, то размерность их произведения AB = C есть m \times k.



A B=\begin{pmatrix} 2 & 3\\ 5 & 7\end{pmatrix}\times\begin{pmatrix} -1 & 2\\ -2 & 3\end{pmatrix}=\begin{pmatrix}2 \cdot (-1) + 3 \cdot (-2) & 2 \cdot 2 + 3 \cdot 3\\5 \cdot (-1) + 7 \cdot (-2) & 5 \cdot 2 + 7 \cdot 3\end{pmatrix}=\begin{pmatrix}-8 & 13\\-19 & 31\end{pmatrix}

B A=\begin{pmatrix} -1 & 2\\ -2 & 3\end{pmatrix}\times\begin{pmatrix} 2 & 3\\ 5 & 7\end{pmatrix}=\begin{pmatrix}-1 \cdot 2 + 2 \cdot 5 & -1 \cdot 3 + 2 \cdot 7\\-2 \cdot 2 + 3 \cdot 5 & -2 \cdot 3 + 3 \cdot 7\end{pmatrix}=\begin{pmatrix}8 & 11\\11 & 15\end{pmatrix}

## *Определитель матрицы*

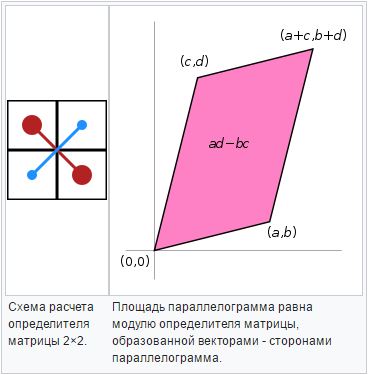
## Для матрицы 2\*2{\displaystyle 2\times 2} определитель вычисляется как:



Эта матрица **A** может быть рассмотрена как матрица линейного отображения, преобразующего единичный квадрат в параллелограмм с вершинами (0, 0), (*a*, *b*), (*a* + *c*, *b* + *d*), и (*c*, *d*).

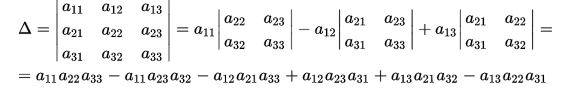
Абсолютное значение определителя  равно площади этого параллелограмма, и, таким образом, отражает коэффициент, на который масштабируются площади при преобразовании **А.**

Значение определителя со знаком (*ориентированная площадь* параллелограмма) помимо коэффициента масштабирования также показывает, выполняет ли преобразование **A** отражение. (приведено ниже)



### Матрицы 3 x 3

Определитель матрицы 3\*3{\displaystyle 3\times 3} можно вычислить по формуле:



Для более удобного вычисления определителя третьего порядка можно воспользоваться правилом Саррюса или правилом треугольника.

Определитель матрицы, составленной из **векторов a b c** {\displaystyle \mathbf {a} ,\mathbf {b} ,\mathbf {c} }\mathbf {a} ,\mathbf {b} ,\mathbf {c}  равен их смешанному произведению в правой декартовой системе координат. Аналогично двумерному случаю, определитель такой матрицы равен ориентированному объёму параллелепипеда, натянутого на {\displaystyle \mathbf {a} ,\mathbf {b} ,\mathbf {c} } **a b c** .

В общем случае, для матриц более высоких порядков (выше 2-го порядка) {\displaystyle n\times n}определитель можно вычислить, применив другие рекурсивные формулы.

1.2 Понятие основных элементов матриц

Для матрицы определены следующие алгебраические операции:

* сложение матриц, имеющих один и тот же размер;
* умножение матриц подходящего размера (матрицу, имеющую {\displaystyle n} столбцов, можно умножить справа на матрицу, имеющую {\displaystyle n}N\*N строк);
* в том числе умножение на матрицу вектора (по обычному правилу матричного умножения; вектор является в этом смысле частным случаем матрицы);
* умножение матрицы на элемент основного кольца или поля (то есть скаляр).

**Свойства сложения матриц:**

* коммутативность: *A+B = B+A*;
* ассоциативность: *(A+B)+C =A+(B+C)*;
* сложение с нулевой матрицей: *A + Θ = A*;
* существование противоположной матрицы: *A + (-A) = Θ*;

**Свойства умножения матриц на число:**

* умножение на единицу: 1A = A;
* ассоциативность: (λβ)A = λ(βA);
* дистрибутивность: (λ+β)A = λA + βA;
* дистрибутивность: λ(A+B) = λA + λB;

1.3 Выводы по главе I

В 1985 году вышло первое издание «Языка программирования C++», обеспечивающее первое описание этого языка, что было чрезвычайно важно из-за отсутствия официального стандарта. В 1989 году состоялся выход C++ версии 2.0. Его новые возможности включали множественное наследование, абстрактные классы, статические функции-члены, функции-константы и защищённые члены. В 1990 году вышло «Комментированное справочное руководство по C++», положенное впоследствии в основу стандарта. Последние обновления включали шаблоны, исключения, пространства имён, новые способы приведения типов и булевский тип.

Подводя итог по истории разработки на С++ можно еще раз удостовериться что данный язык играет огромную роль в нынешние дни на мировой арене, который помогает разрабатывать программы различного рода для коммерческих приложений.

Программа состоит из функий: ввода размерности, сложения-вычитания, умножения(так же и на число), определителя.

## 2. РАЗРАБОТКА ДЕЙСТВИЙ НАД «Матрицы»

### **2.1 Постановка задач для реализации «Матрицы»**

Нужно разработать калькулятор, который будет включать в себя различные функции для действий над матрицами.

В ходе разработки изучались программы «Матрицы» написанные раннее и были выявлены различные недостатки. Например, в программах, были очень плохо написанные коды, которые не были разбиты по функциям, не имеющие связи между собой, действия и работающих в каких-то определенных условиях.

Алгоритм написания программы:

1)Создание всех нужных функций

2)Разбить грамотно и по порядку

3)Сделать проверки для различных условий

4)Тестирование.

**2.2 Вывод по главе II**

В результате работы над курсовой работой формируется целостное представление о процессе создания реального приложения работающего в среде операционной системы Windows, закрепляются навыки алгоритмического, модульного и структурного анализа решаемой задачи, закрепляются навыки разработки пользовательского интерфейса приложения. На основе визуального программирования, закрепляются навыки написания программного кода приложения, а также его отладки, тестирования и документирования.

В работе описано и представлено полное описание программы и её действий, что она выполняет и каким образом работает. Описано подробное описание входных и выходных данных, переменных и функций программ. Описан алгоритм работы программы и листинг программы.

### 

### **3. Алгоритм написание программы «Солнечная система»**

**Код (с пояснениями и комментариями)**

#include <iostream>

#include <cstdio>

#include <fstream>

#include <cstdlib>

#include <Windows.h>

#include <stdio.h>

using namespace std;

(функция ввода матриц)

void inputMatrix(int \*\*a,int n2,int m2) {

cout << "Input variables of matrix" << endl;

for (int i = 0; i < n2; i++)

for (int j = 0; j < m2; j++)

cin >> a[i][j];

}

(функция вывода)

void outputMatrix(int \*\*a, int n2, int m2) {

cout << "Answer " << endl;

for (int i = 0; i < n2; i++) {

for (int j = 0; j < m2; j++) {

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

}

cout << endl;

}

}

(проверка размерности перед сложением-вычитанием)

bool checkingDimensionAddSub(int n1, int m1, int n2, int m2) {

if (n1 == n2 && m1 == m2)return true;

else false;

}

(функция умножения матриц)

void multiply(int \*\*a, int \*\*b, int n1, int m1, int n2, int m2) {

cout << "Answer " << endl;

int ans = 0;

for (int i = 0; i < n2; i++) {

for (int j = 0; j < m1; j++) {

ans = 0;

for (int k = 0; k < m1; k++) {

ans += (a[i][k] \* a[k][j]);

}

cout << ans << " ";

}

cout << endl;

}

}

(функция сложения матриц)

void amount(int\*\*a, int\*\*b, int\*\*c, int n, int m, int n2, int m2){

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

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

c[i][j] = a[i][j] + b[i][j];

}

(функция вычитания матриц)

void subtraction(int\*\*a, int\*\*b, int\*\*c, int n1, int m1, int n2, int m2) {

for (int i = 0; i < n1; i++)

for (int j = 0; j < m1; j++)

c[i][j] = a[i][j] - b[i][j];

}

int search(int \*\*a, int m, int n, int what,

bool match, unsigned int &uI, unsigned int &uJ, unsigned int starti, unsigned int startj) {

( Поиск в матрице a[m][n] элемента с указанным значением what

Возвращаеются его номер строки и столбца uI, uJ, если элемент найден.

match - искать равный элемент или отличный от указанного.

Вернёт 0 - не найдено, не 0 – найдено)

if ((!m) || (!n)) return 0;

if ((starti >= n) || (startj >= m)) return 0;

for (unsigned int i = starti; i < n; i++)

for (unsigned int j = startj; j < m; j++) {

if (match == true) {

if (a[i][i] == what) {

uI = i; uJ = j; return 1;

}

}

else if (a[i][j] != what) {

uI = i; uJ = j; return 1;

}

}

return 0;

}

void swaprows(int \*\*a, int n, int m, unsigned int x1, unsigned int x2) {

(Меняет в матрице a[n][m] строки с номерами x1 и x2 местами)

if ((!n) || (!m)) return;

if ((x1 >= n) || (x2 >= n) || (x1 == x2)) return;

double tmp;

for (unsigned int x = 0; x < m; x++) {

tmp = a[x1][x];

a[x1][x] = a[x2][x];

a[x2][x] = tmp;

}

return;

};

void swapcolumns(int \*\*a, int n, int m, unsigned int x1, unsigned int x2) {

(Меняет в матрице a[n][m] столбцы с номерами x1 и x2 местами)

if ((!n) || (!m)) return;

if ((x1 >= m) || (x2 >= m) || (x1 == x2)) return;

double tmp;

for (unsigned int x = 0; x < n; x++) {

tmp = a[x][x1];

a[x][x1] = a[x][x2];

a[x][x2] = tmp;

}

return;

};

double determinant(int \*\*a, int n) {

(Вычисление определителя квадратной матрицы a[m][n])

unsigned int m = n;

if (m == 0) return 0;

if (m == 1) return a[0][0];

if (m == 2) return (a[0][0] \* a[1][1] - a[1][0] \* a[0][1]);

bool sign = false; (смена знака определителя. по умолчанию – не меняет)

double det = 1; (определитель)

double tmp;

unsigned int x, y;

for (unsigned int i = 0; i < n; i++) { (цикл по всей главной диагонали)

if (a[i][i] == 0) { (если элемент на диагонали равен 0, то ищем ненулевой элемент в матрице)

if (!search(a, m, n, 0, false, y, x, i, i)) return 0; (если все элементы нулевые, то отпр. = 0)

if (i != y) { (меняем i-ую строку с y-ой)

swaprows(a, m, n, i, y);

sign = !sign;

}

if (i != x) { (меняем i-ый столбец с x-ым)

swapcolumns(a, m, n, i, x);

sign = !sign;

}

(таким образом, в a[i][i], теперь ненулевой элемент)

}

(выносим элемент a[i][i] за определитель)

det \*= a[i][i];

tmp = a[i][i];

for (x = i; x < m; x++) {

a[i][x] = a[i][x] / tmp;

}

(таким образом a[i][i] теперь равен 1)

(зануляем все элементы стоящие под (i, i)-ым)

(при помощи вычитания с опр. Коеффициентом)

for (y = i + 1; y < n; y++) {

tmp = a[y][i];

for (x = i; x < m; x++)

a[y][x] -= (a[i][x] \* tmp);

}

}

if (sign) return det\*(-1);

return det;

};

(умножение матрицы на число)

void multipleOnNumber(int \*\*a, int m, int n) {

int number;

cout << "Enter the number for the first matrix, which you want to multiply" << endl;

cin >> number;

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

cout << number\*a[i][j] << " ";

}

cout << endl;

}

}

int main()

{

(команда для смены цвета в консоли)

system("color 8E");

int n1, m1;

cout << "Input dimension of first matrix A " << endl;

cin >> n1 >> m1;

int \*\*a = new int\*[n1];

for (int i = 0; i < n1; i++) {

a[i] = new int[m1];

}

inputMatrix(a, n1, m1);

int n2, m2;

cout << "Input dimension of second matrix B "<<endl;

cin >> n2 >> m2;

int \*\*b = new int\*[n2];

int \*\*c = new int\*[n2];

for (int i = 0; i < n2; i++) {

b[i] = new int[m2];

c[i] = new int[m2];

}

(различные проверки для ввода и вывода матриц)

inputMatrix(b, n2, m2);

if (checkingDimensionAddSub(n1, m1, n2, m2) == true) {

amount(a, b, c, n1, m1, n2, m2);

outputMatrix(c, n1, m1);

}

else cout << "Matrices have different dimension" << endl;

if (checkingDimensionAddSub(n1, m1, n2, m2) == true) {

subtraction(a, b, c, n1, m1, n2, m2);

outputMatrix(c, n1, m1);

}

else cout << "Matrices have different dimension" << endl;

if (m1 != n2)

cout << "Matrices cannot multiply"<< endl; else {

multiply(a, b, n1, m1, n2, m2);

}

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

multipleOnNumber(a, n1, m1);

cout << determinant(a, n1)<<" = Answer determinant our the first matrix"<< endl;

system("pause");

return 0;

}

### 

### **3.1 Вывод по главе III**

Таким образом, поставленные задачи к действиям над матрицами были осуществленны при помощи знаний языка высокого уровня С++ и при помощи знаний высшей математики.

В ходе выполнения работы удалось добиться хорошего калькулятора над матрицами, который многофункционален.

# ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы были изучены новые возможности языка высшего уровня C++,а также были изучены его свойства. В том числе была изучена подсистема Visual Studio. Был разработан программный код для реализации модели на платформе Visual Studio, создан простой интерфейс приложения. Также было проведено исследование компонентов программной среды С++, которые использовались при создании программы.

Написание программы содействовало закреплению абстрактного использованного материала на практике. Данная курсовая работа, несомненно, помогла прояснить некоторые определенные теоретические вопросы, а также практические проблемы, которые были сопряжены с визуальной средой программирования С++. При использовании дополнительной учебной литературе, написание данной программы было несложным. С++ и был создан для того, чтобы облегчить работу программистам. И чтобы написание различных программ и игр казалось не нудным, а интересным занятием.

Благодаря методу оценки эффективности хода был реализован калькулятор над матрицами. Итак, несмотря на внешнюю простоту, программа представляет собой сложную систему взаимодействующих друг с другом функций. Модель была протестирована на наличие ошибок, которых в результате проверки в работе программы не обнаружено. В процессе создания приложения был реализован алгоритм, который в дальнейшем можно усовершенствовать. Все поставленные задачи были выполнены и разработка калькулятора матриц прошла успешно.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Свободная энциклопедия http://ru.wikipedia.org/ - тема «Матрицы и их свойства»
2. Рябушко А.П. Задачник (в 3 томах)
3. Видеохостинговая компания: https://www.youtube.com – тема «Матрицы»
4. Поисковая система https://www.google.com (для различных нюансов при написании программы)