Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по лабораторной работе №1

Тема работы: Действия над матрицами

Выполнил

студент: гр. 151004 Иванов И.И.

Проверил: Фадеева Е.П.

Минск 2022

содержание

[1 Постановка задачи 3](#_Toc97797455)

[2 Описание алгоритмов решения задачи 4](#_Toc97797456)

[3 Структура данных 5](#_Toc97797457)

[3.1 Структура данных программы 5](#_Toc97797458)

[3.2 Структура данных алгоритма MulM 5](#_Toc97797459)

[3.3 Структура данных алгоритма Disp 5](#_Toc97797460)

[3.4 Структура данных алгоритма SumM 6](#_Toc97797461)

[3.5 Структура данных алгоритма MulN 6](#_Toc97797462)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 8](#_Toc97797463)

[4.1 Схема алгоритма решения задачи 8](#_Toc97797464)

[4.2 Схема алгоритма Disp 9](#_Toc97797465)

[4.3 Схема алгоритма SumM 10](#_Toc97797466)

[4.4 Схема алгоритма MulN 11](#_Toc97797467)

[4.5 Схема алгоритма MulM 12](#_Toc97797468)

[5 Результаты расчетов 13](#_Toc97797469)

[Приложение А 14](#_Toc97797470)

[Приложение Б 17](#_Toc97797471)

[Приложение В 20](#_Toc97797472)

# Постановка задачи

**1 постановка**

Даны матрицы A и B размерности 3×3. Вычислить выражение:

2\*A + 3\*B\*(A\*B-2\*A), где

А= В=

**2 постановка**

Осуществить ввод собственных матриц A и B размерностей 3х3 и вычислить выражение:

B\*(A\*B-4\*A)\*6 + 4\*A.

# Описание алгоритмов решения задачи

Таблица 1 – Описание алгоритмов решения задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п.п | Наименование алгоритма | Назначение алгоритма | Формальные параметры | Предполагаемый тип реализации |
| 1. | Основной алгоритм | Вызывает следующие подпрограммы:  Disp, MulN,  MulM, SumM |  |  |
| 3. | MulM(M1,  M2, Res) | Умножает матрицы M1 и M2 с формированием результата  Res | M1, M2 – получает защищённое значение от фактического параметра;  Res – получает адрес от фактического параметра адрес | Процедура |
| 4. | Disp(M) | Выводит на экран матрицу M | M – получает защищённое значение от фактического параметра | Процедура |
| 5. | SumM(M1,  M2, IsPlus, Res) | Вычисляет сумму (разность, если IsPlus = false) двух матриц M1 и M2 с формировнием результата в  Res | M1, M2, IsPlus – получает защищённое значение от фактического параметра;  Res - получает адрес от фактического параметра | Процедура |
| 6. | MulN(M, N, Res) | Умножает матрицу Matr на число Num с формированием результата в  Res | M, N - получает защищённое значение от фактического параметра;  Res - получает адрес от фактического параметра адрес | Процедура |

# Структура данных

## Структура данных программы

Таблица 2 – Структура данных

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| А | Array[1..3, 1..3] Of Real | Основная вводимая матрица |
| B | Array[1..3, 1..3] Of Real | Основная вводимая матрица |
| C1, C2, C3, C4, C5, C6, C7 | Array[1..3, 1..3] Of Real | Матрицы для промежуточных значений |

## Структура данных алгоритма MulM

Таблица 3 – Структура данных алгоритма MulM(M1, M2, Res)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Res | Array[1..3, 1..3] Of Real | Полученная матрица | Формальный |
| M1 | Array[1..3, 1..3] Of Real | Матрица | Формальный |
| M2 | Array[1..3, 1..3] Of Real | Матрица | Формальный |
| I | Integer | Параметр цикла. Значение строки матрицы | Локальный |
| J | Integer | Параметр цикла. Значение столбца матрицы | Локальный |

## Структура данных алгоритма Disp

Таблица 4 – Структура данных алгоритма Disp(M)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| M | Array[1..3, 1..3] Of Real | Матрица | Формальный |

Продолжение Таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| I | Integer | Параметр цикла. Значение строки матрицы | Локальный |
| J | Integer | Параметр цикла. Значение столбца матрицы | Локальный |

## Структура данных алгоритма SumM

Таблица 5 – Структура данных алгоритма SumM(M1, M2, Res, IsPlus)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Res | Array[1..3, 1..3] Of Real | Полученная матрица | Формальный |
| M1 | Array[1..3, 1..3] Of Real | Матрица | Формальный |
| M2 | Array[1..3, 1..3] Of Real | Матрица | Формальный |
| IsPlus | Boolean | Выбор, что делать: вычитание или сложение | Формальный |
| I | Integer | Параметр цикла. Значение строки матрицы | Локальный |
| J | Integer | Параметр цикла. Значение столбца матрицы | Локальный |

## Структура данных алгоритма MulN

Таблица 6 – Структура данных алгоритма MulN (M, N, Res)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Res | Array[1..3, 1..3] Of Real | Полученная матрица | Формальный |
| N | Real | Число, на которое умножается матрица | Формальный |
| M | Array[1..3, 1..3] Of Real | Матрица | Формальный |
| I | Integer | Параметр цикла. Значение строки матрицы | Локальный |
| J | Integer | Параметр цикла. Значение столбца матрицы | Локальный |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

## Схема алгоритма решения задачи

|  |
| --- |
|  |
| Рисунок 1 – Схема алгоритма решения задачи |

## Схема алгоритма Disp

|  |
| --- |
|  |
| Рисунок 2 – Схема алгоритма Disp |

## Схема алгоритма SumM

|  |
| --- |
|  |
| Рисунок 3 – Схема алгоритма SumM |

## Схема алгоритма MulN

|  |
| --- |
|  |
| Рисунок 4 – Схема алгоритма MulN |

## Схема алгоритма MulM

|  |
| --- |
|  |
| Рисунок 5 – Схема алгоритма MulM |

# Результаты расчетов

Вследствие результатов программы на экран выводятся следующие результаты расчетов:

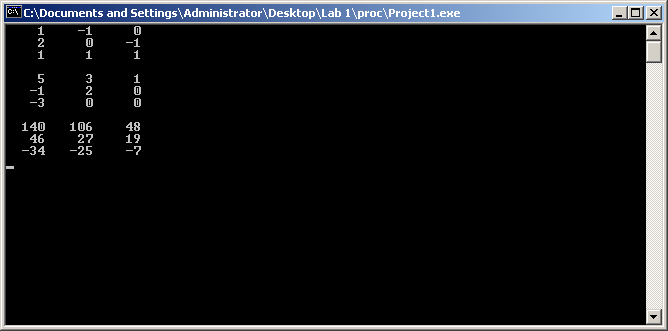


Рисунок 6 – Результаты расчетов

Приложение А

(обязательное)

Исходный код программы (постановка 1)

Program Lab1P;

{Operations with matrix exps}

//Use app

{$APPTYPE CONSOLE}

//Declare modules

Uses

SysUtils;

//Declare types

Type

TMatrix = Array[1..3, 1..3] Of Real;

//TMatrix - matrix for using in subprograms

//Declare vars

Var

A:TMatrix = ((1, -1, 0), (2, 0, -1), (1, 1, 1));

B:TMatrix = ((5, 3, 1), (-1, 2, 0), (-3, 0, 0));

C1, C2, C3, C4, C5, C6, C7:TMatrix;

//I, J - loop params

//A, B - start matrixs

//C1, C2, C3, C4, C5, C6, C7 - Res matrixs

//This func calculates Matrix M \* Number N, saves in Res

Procedure MulN(const M:TMatrix; const N:Real; var Res:TMatrix);

Var

I, J:Integer;

//I, J - loop params

Begin

//Mulle every cell of matrix

For I:= 1 To 3 Do

For J:= 1 To 3 Do

Res[I, J]:= M[I, J] \* N;

End;

//This func calculates Matrix M1 + Matrix M2, saves in Res

Procedure SumM(const M1:TMatrix; const M2:TMatrix; var Res:TMatrix; IsPlus: Boolean);

Var

I, J:Integer;

//I, J - loop params

Begin

//Sum every cell with every cell of matrix

For I:= 1 To 3 Do

For J:= 1 To 3 Do

Begin

if IsPlus then

Res[I, J]:= M1[I, J] + M2[I, J]

else

Res[I, J]:= M1[I, J] - M2[I, J];

End;

End;

//This func calculates Matrix M1 \* Matrix M2, saves in Res

Procedure MulM(const M1:TMatrix; const M2:TMatrix; var Res:TMatrix);

Var

I, J:Integer;

//I, J - loop params

Begin

//Sum every line with every column of matrix

For I:= 1 To 3 Do

For J:= 1 To 3 Do

Res[I, J]:= M1[I, 1] \* M2[1, J] + M1[I, 2] \* M2[2, J] + M1[I, 3] \* M2[3, J];

End;

//This func shows formatted matrix M

Procedure Disp(const M:TMatrix);

Var

I, J:Integer;

//I, J - loop params

Begin

//Display lines with spacing

For I:= 1 To 3 Do

Begin

For J:= 1 To 3 Do

Write(FloatToStr(M[I, J]):5, ' ');

Writeln;

End;

End;

Begin

Disp(A);

Writeln;

Disp(B);

Writeln;

MulN(A, 2, C1);

MulN(B, 3, C2);

MulM(A, B, C3);

MulN(A, 2, C4);

SumM(C3, C4, C5, False);

MulM(C2, C5, C6);

SumM(C1, C6, C7, True);

Disp(C7);

Readln;

End.

Приложение Б

(обязательное)

Исходный код программы (постановка 2)

Program Lab1P;

{Operations with matrix exps}

//Use app

{$APPTYPE CONSOLE}

//Declare modules

Uses

SysUtils;

//Declare types

Type

TMatrix = Array[1..3, 1..3] Of Real;

//TMatrix - matrix for using in subprograms

//Declare vars

Var

A, B:TMatrix;

C1, C2, C3, C4, C5, C6, C7:TMatrix;

//I, J - loop params

//A, B - start matrixs

//C1, C2, C3, C4, C5, C6, C7 - Res matrixs

//This func calculates Matrix M \* Number N, saves Res

Procedure MulN(const M:TMatrix; const N:Real; var Res:TMatrix);

Var

I, J:Integer;

//I, J - loop params

Begin

//Multiple every cell of matrix

For I:= 1 To 3 Do

For J:= 1 To 3 Do

Res[I, J]:= M[I, J] \* N;

End;

//This func calculates Matrix M1 + Matrix M2, saves Res

Procedure SumM(const M1:TMatrix; const M2:TMatrix; var Res:TMatrix; IsPlus: Boolean);

Var

I, J:Integer;

//I, J - loop params

Begin

//Sum every cell with every cell of matrix

For I:= 1 To 3 Do

For J:= 1 To 3 Do

Begin

if IsPlus then

Res[I, J]:= M1[I, J] + M2[I, J]

else

Res[I, J]:= M1[I, J] - M2[I, J];

End;

End;

//This func calculates Matrix M1 \* Matrix M2, saves Res

Procedure MulM(const M1:TMatrix; const M2:TMatrix; var Res:TMatrix);

Var

I, J:Integer;

//I, J - loop params

Begin

//Sum every line with every column of matrix

For I:= 1 To 3 Do

For J:= 1 To 3 Do

Res[I, J]:= M1[I, 1] \* M2[1, J] + M1[I, 2] \* M2[2, J] + M1[I, 3] \* M2[3, J];

End;

//This func reads matrix M

Procedure ReadM(const M:TMatrix);

Var

I, J:Integer;

//I, J - loop params

Begin

//Read items of Matrix

For I:= 1 To 3 Do

Begin

For J:= 1 To 3 Do

ReadLn(M[I, J]);

Writeln;

End;

End;

Begin

ReadM(A);

Writeln;

ReadM(B);

Writeln;

MulN(A, 4, C1);

MulN(B, 6, C2);

MulM(A, B, C3);

MulN(A, 4, C4);

SumM(C3, C4, C5, False);

MulM(C2, C5, C6);

SumM(C1, C6, C7, True);

Disp(C7);

Readln;

End.

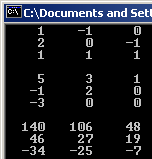
Приложение В

(обязательное)

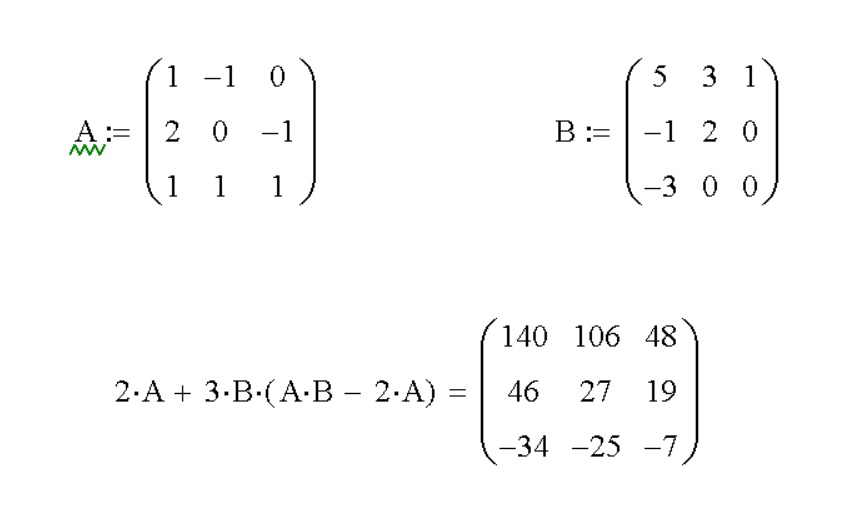
Тестовые наборы

Тестовая ситуация: верно ли программа выполняет операции над двумя матрицами.

Результат программы



Результат в MathCad



Очевидно, что результаты написанной программы совпадают с проверкой в MathCad.