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: Операции над матрицами

Выполнил

студент: гр. 151003 Барановский Р.А.

Проверил: Фадеева Е.П.

Минск 2021

содержание

[1 Постановка задачи 3](#_Toc98335784)

[2 Описание алгоритмов решения задачи 4](#_Toc98335785)

[3 Структура данных 7](#_Toc98335786)

[3.1 Структура данных основной программы 7](#_Toc98335787)

[3.2 Структура данных алгоритма AddMatrix 7](#_Toc98335788)

[3.3 Структура данных алгоритма SubMatrix 7](#_Toc98335789)

[3.4 Структура данных алгоритма MultNum 8](#_Toc98335790)

[3.5 Структура данных алгоритма MultMatrix 8](#_Toc98335791)

[3.6 Структура данных алгоритма OutputMatrix 8](#_Toc98335792)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 9](#_Toc98335793)

[4.1 Схема алгоритма основной программы 9](#_Toc98335794)

[4.2 Схема алгоритма AddMatrix 11](#_Toc98335795)

[4.3 Схема алгоритма SubMatrix 13](#_Toc98335796)

[4.4 Схема алгоритма MultNum 15](#_Toc98335797)

[4.5 Схема алгоритма MultMatrix 17](#_Toc98335798)

[4.6 Схема алгоритма OutputMatrix 19](#_Toc98335799)

[5 Результаты расчетов 20](#_Toc98335800)

[Приложение А 21](#_Toc98335801)

[Приложение Б 25](#_Toc98335802)

# Постановка задачи

 Разработать программу для расчета матричного выражения:

2\*(A - B)\*(A2 + B), где

A = B =

Результаты расчетов сравнить в MathCad 14.

Отчет 2: Использование только функций

# 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 подпрограммы  AddMatrix  (X,Y, res),  SubMatrix  (X,Y, res),  MultNum  (num, X, res),  MultMatrix  (X,Y, res),  OutputMatrix  (X) |  |  |
| 2 | AddMatrix  (X,Y, res) | Складывает матрицы X и Y, результат формируется в переменной res | X – получает от фактического параметра адрес с защитой 1 массива со значениями слагаемого  Y – получает от фактического параметра адрес с защитой 2  массива со значениями слагаемого  res – получает от фактического параметра адрес, возвращаемый параметр | Функция  res – возвращаемый функцией параметр |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | SubMatrix  (X,Y, res) | Вычитает  матрицы X и Y, результат формируется в переменной res | X – получает от фактического параметра адрес с защитой массива со значениями уменьшаемого  Y – получает от фактического параметра адрес с защитой массива со значениями вычитаемого  res – получает от фактического параметра адрес, возвращаемый параметр | Функция  res – возвращаемый функцией параметр |
| 4 | MultNum  (num,X, res) | Умножает число num на матрицу X, результат формируется в переменной res | num – получает от фактического параметра адрес с защитой числа  X – получает от фактического параметра адрес с защитой массива  res - получает от фактического параметра адрес, возвращаемый параметр | Функция  res - возвращаемый функцией параметр |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | MultMatrix  (X,Y, res) | Умножает матрицы X и Y, результат формируется в переменной res | X – получает от фактического параметра адрес с защитой 1 массива со значениями множителя  Y – получает от фактического параметра адрес с защитой 2 массива со значениями множителя  res - получает от фактического параметра адрес, возвращаемый параметр | Функция  res – возвращаемый функцией параметр |
| 6 | OutputMatrix  (X) | Выводит на экран матрицу X | X – получает от фактического параметра адрес с защитой массива для вывода | Процедура |

# Структура данных

## Структура данных основной программы

Таблица 2 – Структура данных основной программы

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| res1, res2, res3, res4 | Array[1..3,1..3] of integer | Промежуточные вычисления |
| finalRes | Array[1..3,1..3] of integer | Результат вычислений |

## Структура данных алгоритма AddMatrix

Таблица 3 – Структура данных алгоритма AddMatrix(X, Y, res)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| X | Array[1..3, 1..3] of integer | Матрица 1 | Формальный |
| Y | Array[1..3, 1..3] of integer | Матрица 2 | Формальный |
| res | Array[1..3, 1..3] of integer | Возвращаемый функцией параметр | Формальный |
| temp1 | Array[1..3, 1..3] of integer | Результат сложения матриц | Локальный |
| i | integer | Счетчик 1 | Локальный |
| j | integer | Счетчик 2 | Локальный |

## Структура данных алгоритма SubMatrix

Таблица 4 – Структура данных алгоритма SubMatrix(X, Y, res)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| X | Array[1..3, 1..3] of integer | Матрица 1 | Формальный |
| Y | Array[1..3, 1..3] of integer | Матрица 2 | Формальный |
| res | Array[1..3, 1..3] of integer | Возвращаемый функцией параметр | Формальный |
| temp2 | Array[1..3, 1..3] of integer | Результат вычитания матриц | Локальный |
| i | integer | Счетчик 1 | Локальный |
| j | integer | Счетчик 2 | Локальный |

## Структура данных алгоритма MultNum

Таблица 5 – Структура данных алгоритма MultNum(X, num, res)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| num | integer | Число | Формальный |
| X | Array[1..3, 1..3] of integer | Матрица | Формальный |
| res | Array[1..3, 1..3] of integer | Возвращаемый функцией параметр | Формальный |
| temp3 | Array[1..3, 1..3] of integer | Результат умножения матрицы и числа | Локальный |
| i | integer | Счетчик 1 | Локальный |
| j | integer | Счетчик 2 | Локальный |

## Структура данных алгоритма MultMatrix

Таблица 6 – Структура данных алгоритма MultMatrix(X, Y, res)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| X | Array[1..3, 1..3] of integer | Матрица 1 | Формальный |
| Y | Array[1..3, 1..3] of integer | Матрица 2 | Формальный |
| res | Array[1..3, 1..3] of integer | Возвращаемый функцией параметр | Формальный |
| temp4 | Array[1..3, 1..3] of integer | Результат умножения матриц | Локальный |
| i | integer | Счетчик 1 | Локальный |
| j | integer | Счетчик 2 | Локальный |
| k | integer | Счетчик 3 | Локальный |

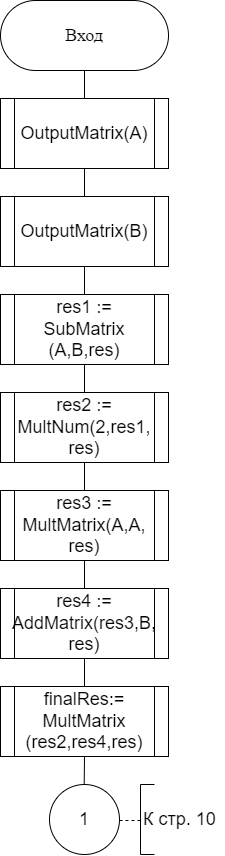
## Структура данных алгоритма OutputMatrix

Таблица 7 – Структура данных алгоритма OutputMatrix(X)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| X | mas | Выводимая матрица | Формальный |
| i | integer | Счетчик 1 | Локальный |
| J | integer | Счетчик 2 | Локальный |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

## Схема алгоритма основной программы



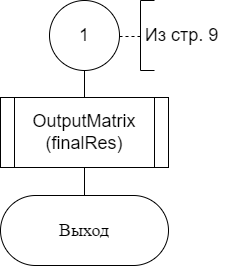
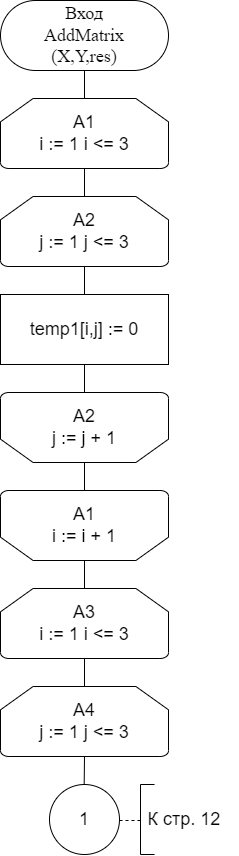


Рисунок 1 – Схема алгоритма основной программы

## Схема алгоритма AddMatrix



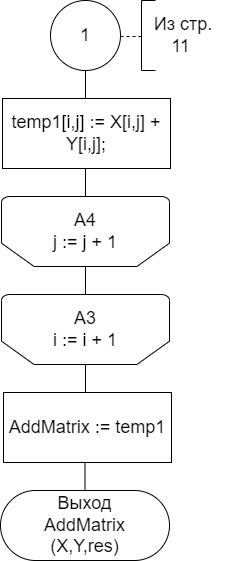
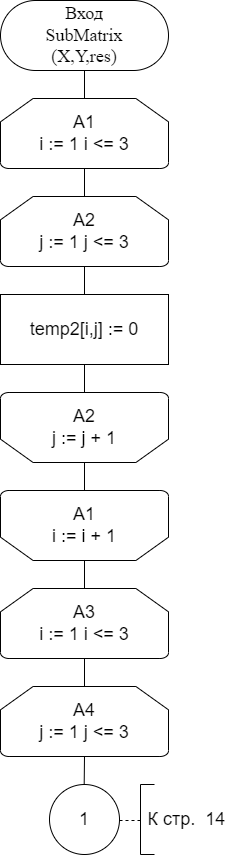


Рисунок 2 – Схема алгоритма addMatrix

## Схема алгоритма SubMatrix



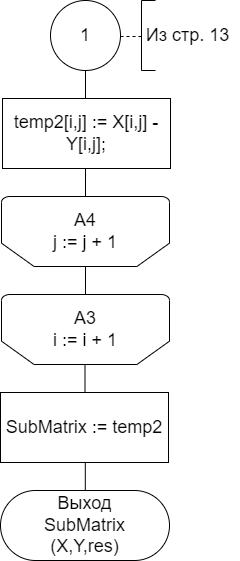
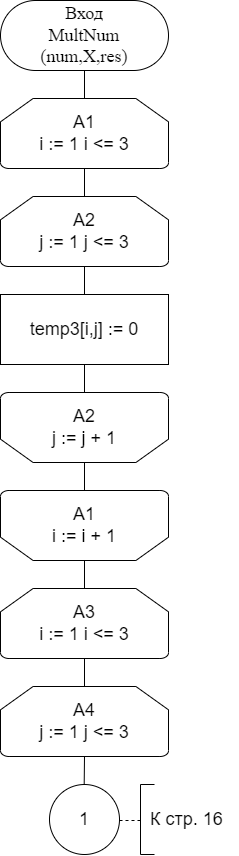


Рисунок 3 – схема алгоритма subMatrix

## Схема алгоритма MultNum



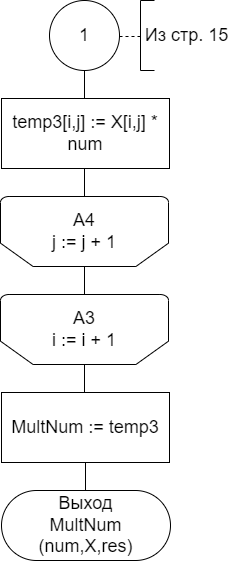
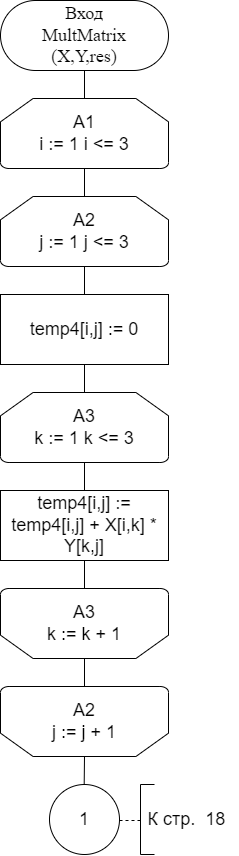


Рисунок 4 – Схема алгоритма multNum

## Схема алгоритма MultMatrix



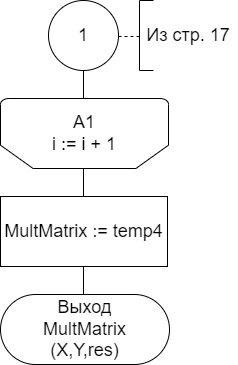


Рисунок 5 – Схема алгоритма multMatrix

## Схема алгоритма OutputMatrix

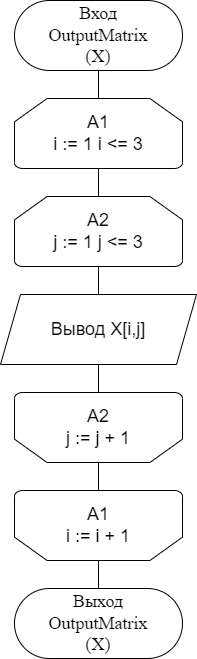


Рисунок 6 – Схема алгоритма outputMatrix

# Результаты расчетов

Вследствие результатов программы на экран выводятся следующие результаты расчетов:

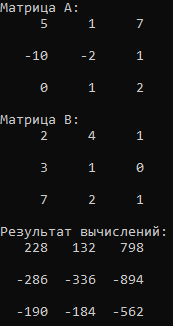


Рисунок 7 – результаты расчетов

Приложение А

(обязательное)

Исходный код программы

program Var2;

//Program calculates mathematical expression

//Console app

{$APPTYPE CONSOLE}

//Modules declaration

uses

System.SysUtils;

//Types declaration

type

mas = array[1..3,1..3] of integer;

//Constants declaration

const

A:mas = ((5,1,7),

(-10,-2,1),

(0,1,2));

B:mas = ((2,4,1),

(3,1,0),

(7,2,1));

//Function calculates sum of matrices

function AddMatrix(const X, Y: mas): mas;

var

i, j: integer;

temp1: mas;

{ X - matrix 1

Y - matrix 2

temp1 - result

i - counter 1

j - counter 2 }

begin

//Filling temp1 with 0

for i := 1 to 3 do

for j := 1 to 3 do

temp1[i,j] := 0;

//Calculating sum of matrices

for i := 1 to 3 do

for j := 1 to 3 do

temp1[i,j] := X[i,j] + Y[i,j];

//Return sum of matrices

AddMatrix := temp1;

end;

//Procedure calculates difference of matrices

function SubMatrix(const X, Y: mas): mas;

var

i, j: integer;

temp2: mas;

{ X - matrix 1

Y - matrix 2

temp2 - result

i - counter 1

j - counter 2 }

begin

//Filling temp2 with 0

for i := 1 to 3 do

for j := 1 to 3 do

temp2[i,j] := 0;

//Calculating difference of matrices

for i := 1 to 3 do

for j := 1 to 3 do

temp2[i,j] := X[i,j] - Y[i,j];

//Return difference of matrices

SubMatrix := temp2;

end;

//Procedure calculates result of multiplication

//of matrix and number

function MultNum(const num: integer;const X: mas): mas;

var

i, j: integer;

temp3: mas;

{ X - matrix 1

Y - matrix 2

temp3 - result

i - counter 1

j - counter 2 }

begin

//Filling temp1 with 0

for i := 1 to 3 do

for j := 1 to 3 do

temp3[i,j] := 0;

//Calculate result of multiplication

for i := 1 to 3 do

for j := 1 to 3 do

temp3[i,j] := X[i,j] \* num;

//Return result of multiplication of num and matrix

MultNum := temp3;

end;

//Procedure calculates result of multiplication of

//matrices

function MultMatrix(const X, Y: mas): mas;

var

i, j, k: integer;

temp4: mas;

{ X - matrix 1

Y - matrix 2

temp4 - result

i - counter 1

j - counter 2

k - counter 3 }

begin

//Calculate result of multiplication

for i := 1 to 3 do

for j := 1 to 3 do

begin

temp4[i,j] := 0;

for k := 1 to 3 do

temp4[i,j] := temp4[i,j] + X[i,k] \* Y[k,j];

end;

//Return result of multiplication of matrices

MultMatrix := temp4;

end;

//Output matrix

procedure outputMatrix(const X: mas);

var

i, j: integer;

{ X - matrix

i - counter

j - counter }

begin

//Output matrix

for i := 1 to 3 do

begin

for j := 1 to 3 do

write(X[i,j]:6);

writeln;

writeln;

end;

end;

//Variables declaration

var

res1, res2, res3, res4, finalRes: mas;

{ res1 - 1-st part of calculations

res2 - 2-nd part of calculations

res3 - 3-rd part of calculations

res4 - 4-th part of calculations

finalRes - final result of calculations }

//Start the program

begin

//Output matrix A

writeln('Матрица A: ');

OutputMatrix(A);

//Output matrix B

writeln('Матрица B: ');

OutputMatrix(B);

//Calculate result of operations

res1 := SubMatrix(A, B); { A - B }

res2 := MultNum(2, res1); { 2\*(A - B)}

res3 := MultMatrix(A, A); { A\*A }

res4 := AddMatrix(res3, B); { A\*A + B }

finalRes := MultMatrix(res2, res4);

{ 2\*(A - B)\*(A\*A + B) }

//Output result of calculations

writeln('Результат вычислений: ');

OutputMatrix(finalRes);

//Final operations

readln;

end.

Приложение Б

(обязательное)

Тестовые наборы(Расчеты)

Тест 1

Тестовая ситуация: Стандартный режим работы программы

Исходные данные: A = 5 1 7

-10 -2 1

0 1 2

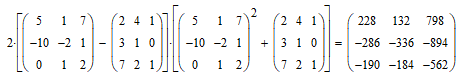
B = 2 4 1

3 1 0

7 2 1

Найти: 2\*(A - B)\*(A2 + B)

Ожидаемый результат:



Полученный результат:

