МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

Отчёт по лабораторной работе № 1

«Форматирование документа с использованием стилей»

Выполнил:

Студент гр. АСОИ-181

Остапенко Александр Константинович

Самусев Данила Андреевич

Проверил:

Шебан Татьяна Леонидовна

Содержание

[Введение 3](#_Toc50543778)

[1. Постановка задачи 4](#_Toc50543779)

[1.1. Логическая модель задачи 4](#_Toc50543780)

[1.2. Входные данные 5](#_Toc50543781)

[1.3. Выходные данные 5](#_Toc50543782)

[1.4. Обработка ошибок 5](#_Toc50543783)

[2. Проектирование программного модуля 7](#_Toc50543784)

[2.1. Структурная диаграмма программного модуля 7](#_Toc50543785)

[2.2. Разработка схемы алгоритма программного модуля и ее описание 8](#_Toc50543786)

[2.3. Разработка пользовательского интерфейса 11](#_Toc50543787)

[3. Реализация программного модуля 13](#_Toc50543788)

[3.1. Код программы 13](#_Toc50543789)

[3.2. Описание используемых операторов и функций 14](#_Toc50543790)

[4. Тестирование программного модуля 15](#_Toc50543791)

[Заключение 17](#_Toc50543792)

[Список использованных источников 18](#_Toc50543793)

## Введение

Целью курсовой работы является разработка программного модуля, позволяющего создавать динамические массивы, и выполнять с ними следующие преобразования: найти количество отрицательных элементов массива; рассчитать сумму модулей элементов массива, расположенных после минимального по модулю элемента; преобразовать массив, заменив все отрицательные элементы их квадратами и упорядочить массив по возрастанию.

Для разработки данной курсовой работы был использован язык программирования VBA (Visual Basic for Applications), т.к. он является наиболее простым, достаточно гибким и вполне применимым данного типа расчётов (расчётов с массивами случайных чисел).

Пояснительная записка включает следующие разделы:

1. Постановка задачи – в этой части описываются логическая модель задачи, входные и выходные данные, средства обработки ошибок.
2. Проектирование программного модуля – приводится структурная диаграмма программного модуля, схема алгоритма с описанием всех блоков.
3. Реализация программного модуля – приводится код программы с комментариями, а также использованные операторы языка.
4. Тестирование – производится опытное тестирование разработанной и реализованной программы

В заключении подводятся общие выводы.

## 1. Постановка задачи

## 1.1. Логическая модель задачи

В данной работе необходимо вычислить:

* + 1. Найти количество отрицательных элементов массива.

Для решения этой задачи последовательно перебираются и анализируются значения каждого элемента массива. Если значение элемента меньше нуля, счетчик количества таких элементов увеличивается на 1 (изначально этот счетчик равен нулю).

* + 1. Сумма модулей элементов массива, расположенных после минимального по модулю элемента.

Для решения этой задачи вначале производится поиск минимального по модулю элемента и его индекса. Для этого за минимальный элемент принимается модуль первого элемента массива. Затем минимальный элемент последовательно сравнивается со всеми остальными элементами массива. Если находится элемент, меньше, чем минимальный, то минимальному элементу присваивается значение этого меньшего элемента. Кроме того, запоминается индекс (номер) минимального элемента.

Затем исходный двумерный динамический массив преобразовывается в промежуточный динамический одномерный массив. Затем в цикле, организованном от следующего за сохранённым значением индекса минимального элемента до конца массива, модуль каждого элемента прибавляется к сумме, изначально равной нулю.

1.1.3. Заменить все отрицательные элементы их квадратами и упорядочить массив по возрастанию.

Для решения этой задачи используются промежуточный динамический одномерный массив. Вначале исходный двумерный динамический массив преобразовывается в одномерный динамический массив. Затем в цикле производится поиск отрицательных элементов и замена их значениями в квадрате. Сформированный таким образом одномерный динамический массив преобразовывается в двумерный динамический массив, который затем и выводится на пользовательскую форму и лист Excel.

## 1.2. Входные данные

Входными данными в данной курсовой работе является размерность массива:

- количество строк (m),

- количество столбцов (n),

Входные данные размерности должны быть числовыми, больше 1, целыми числами. Входные данные вводятся в текстовые поля пользовательской формы.

## 1.3. Выходные данные

Выходными данными являются:

Массив, заполненный случайными числами в диапазоне [-10;10];

Количество отрицательных элементов массива;

Сумма модулей элементов массива, расположенных после минимального по модулю элемента;

Массив, преобразованный таким образом, что все отрицательные элементы заменены их квадратами и упорядочены по возрастанию

Выходные данные выводятся на лист Excel и на пользовательскую форму.

## 1.4. Обработка ошибок

В программе предусмотрена проверка на допустимость входных данных. При нажатии кнопки «Заполнить» происходит проверка введённых данных. При неправильно введённых данных или при их отсутствии выдаётся сообщение об ошибке. После нажатия кнопки «ОК» окно сообщения закрывается, и курсор устанавливается в поле ввода. Так, при вводе размерности, не соответствующей требованиям, указанным в п.1.2, появляется одно из окон сообщения, представленных на рисунках 1-3. Окно сообщения об ошибке, если не выбран режим расчета, представлено на рисунке 4.

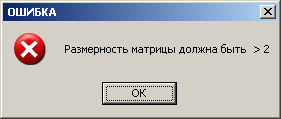


Рисунок 1 – Окно сообщения об ошибке ввода размерности меньше двух

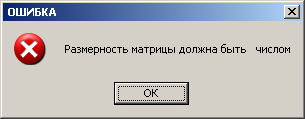


Рисунок 2 – Окно сообщения об ошибке при вводе нечисловой размерности

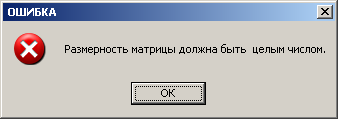


Рисунок 3 – Окно сообщения об ошибке при вводе не целого значения размерности

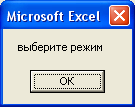


Рисунок 4 – Окно сообщения об ошибке, если не выбран режим расчета

## 2. Проектирование программного модуля

## 2.1. Структурная диаграмма программного модуля

Структурная диаграмма включает четыре уровня. Первый уровень – Workbook\_open()– открытие рабочей книги Excel. Второй уровень – пользовательская форма Userform1. Третий уровень состоит из процедур, которые вызываются событиями, связанными с элементами пользовательской формы. Четвертый уровень – процедура вывода отсортированной матрицы. Структурная диаграмма программного модуля представлена на рисунке 5.

Userform\_initialize() – процедура начальной инициализации пользовательской формы.

Userform1\_Initialize()

CommandButton2\_Click()

CommandButton4\_Click()

CommandButton3\_Click()

CommandButton5\_Click()

Userform1

Userform1

Workbook\_open()

Рисунок 5 – Структурная диаграмма программного модуля

CommandButton1\_Click()

out\_matr(A(), n , m , k)

OptionButton5\_Click()

Commandbutton1\_Click() – процедура, срабатывающая при нажатии кнопки “Заполнить ” и выполняющая заполнение массива случайными числами от -10 до 10.

Commandbutton2\_Click() – процедура, срабатывающая при нажатии кнопки “Выполнить”, используемая для нахождения заданных параметров массива и его преобразования.

Commandbutton3\_Click() – процедура, срабатывающая при нажатии кнопки “Перейти на лист”, для перехода на лист Excel.

Commandbutton4\_Click() – процедура, срабатывающая при нажатии кнопки «Выход», используется для выхода из приложения.

Commandbutton5\_Click() – процедура, срабатывающая при нажатии кнопки «Очистить» для очистки полей ввода и списков формы.

OptionButton5\_Click() - процедура, срабатывающая при выборе переключателя ввода данных с листа Excel.

## 2.2. Разработка схемы алгоритма программного модуля и ее описание

Схема алгоритма программного модуля представлена на рисунке 6.

19

Min = Abs(B(1)) minind = 1

16

18

17

да

Min = Abs(B(i)) minind = i

Abs(B(i)) < Min

i = 1 to m \* n Step -1

нет

12

SUMN = 0 k = 1

11

нет

14

15

B(k) = A(i, j)

k = k + 1

j = 1 To m

i = 1 To n

Да

OptionButton2 = True

10

нет

TextBox6.Text = kolot

Worksheets(1).Cells(2, n + 2) = kolot

7

9

8

Да

kolot =kolot + 1

A(i, j) < 0

4

6

5

3

1

А

j = 1 To m

i = 1 To n

Да

i = 1 j = 1

Max = Abs(A(i, j))

OptionButton1 = True

нет

Да

Выберите режим

OptionButton1 = False And OptionButton2 = False And OptionButton3 = False

SUMN = 0 Max = 0 kolot = 0

Начало

2

нет

13

36

32

B(p - 1) = B(p)

B(p) = tmp

Нет

B(p - 1) > B(p)

tmp = B(p - 1)

p = n \* m To k Step2

k = 2 To n \* m

30

Нет

Нет

Да

B(k) < 0

22

TextBox5.Text = SUMN Cells(4, n + 2) = SUMN

39

Конец

out\_matr A, n, m, 1

37

38

A(i, j) = C(l)

l = l + 1

j = 1 To m

i = 1 To n

29

B(k) = B(k) ^ 2

j = 1 To m\*т

24

l = 1

k = 1

26

B(k) = A(i, j)

k = k + 1

j = 1 To m

i = 1 To n

OptionButton1 = True

25

20

SUMN = SUMN + Abs(B(i))

j = minind + 1 To m\*n

Да

21

23

27

28

31



34

33

35

Да

Блок 1 – обнуление сумм.

Блок 2 – проверка выбора режима вычисления, если режим не выбран, то выполняется блок 3 – предложение выбрать режим и выход из процедуры.

Блок 4 – если выбран переключатель OptionButton1 (определение количества отрицательных элементов), то выполняются блоки 5-9.

Блоки 5-9 – определение количества отрицательных элементов.

Блок 10 – вывод количества отрицательных элементов.

Блок 11 – проверка. если выбран переключатель OptionButton2 (определение суммы модулей элементов массива, расположенных после минимального по модулю элемента), то выполняются блоки 12-22.

Блок 13-15 – преобразование двумерного массива в одномерный.

Блок 16-19 - определение модуля минимального элемента и его номера.

Блоки 21 -22 - расчет суммы модулей после модуля минимального элемента.

Блок 23 – вывод суммы модулей после модуля минимального элемента

Блок 24 – проверка. если выбран переключатель OptionButton3 (преобразовать массив), то выполняются блоки 25-39.

Блок 25-27 – преобразование двумерного массива в одномерный.

Блок 28-30 – замена отрицательных элементов их квадратами.

Блок 31-35 – сортировка преобразованного массива.

Блок 36-38 – преобразование одномерного массива в двумерный.

Блок 39 – вызов процедуры вывода матрицы.

## 2.3. Разработка пользовательского интерфейса

Пользовательский интерфейс предназначен для обеспечения простоты и удобства пользования программой.

Программа состоит из одной пользовательской формы, которая представлена на рисунке 7.

На форме расположены следующие элементы управления.

Текстовые поля для ввода размерности матрицы (TextВox1 и TextВox2),

списки (ListBox1, ListBox2 , ListBox3) для отображения введенного и преобразованного массива, а также массива, содержащего максимальные элементы строк,

кнопка «Заполнить» для заполнения массива и отображения результатов на форме (CommandButton1),

кнопка «Выполнить» для выполнения операций с массивом (CommandButton2),

кнопка «Перейти на лист» для перехода к листу Excel (CommandButton3),

кнопка «Очистить» для очистки полей и списков формы (CommandButton5),

кнопка «Выход» для выхода из приложения (CommandButton4),

переключатели выбора режимов действий, выполняемых над массивом (OptionButton1 - OptionButton3),

переключатели выбора режимов заполнения массива числами от -10 до 10 (OptionButton4 - OptionButton5).

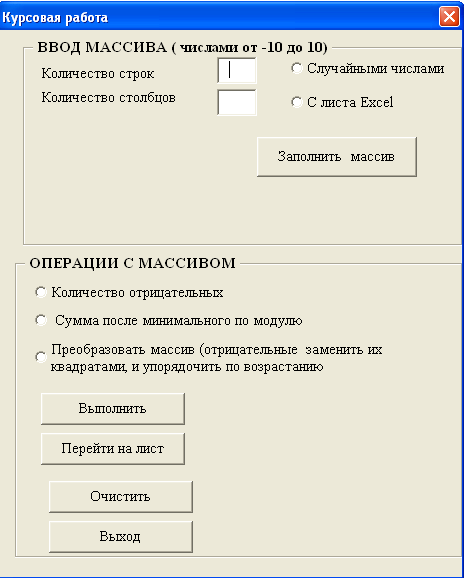


Рисунок 7 – Пользовательская форма

## 3. Реализация программного модуля

## 3.1. Код программы

Программный модуль разработан в приложении Microsoft Excel на языке программирования Visual Basic For Application.

'Объявление глобальных переменных

Dim n As Integer

Dim m As Integer

Dim sum As Long

Dim A() As Double

Dim B() As Double

Dim C() As Double

Dim U() As Double

Public Sub out\_matr(A() As Double, n As Integer, m As Integer, k As Integer) ' процедура вывода массива на лист

For i = 1 To n

For j = 1 To m

Worksheets(1).Cells(i + (n + 2) \* k, j) = A(i, j)

Next j

Next i

'End If

End Sub

Private Sub CommandButton3\_Click()

Application.Visible = True

UserForm1.Hide

End Sub

Public Sub CommandButton4\_Click()

Application.Quit

End Sub

## 3.2. Описание используемых операторов и функций

"**=**" – оператор присваивания;

" **<=**, " – оператор сравнения;

ИнструкцияDim Предназначена для описания типа данных переменной на уровне модуля или процедуры, синтаксис:

Dim Имя переменной [Аs Тип переменной ]

Типы переменных: Double (вещественный двойной точности), Single (вещественный обычной точности), Integer (целочисленный).

If Then Else - оператор условного перехода. Если условие принимает значение True, то выполняется инструкция (или инструкции) после Then если False, то выполняется инструкция (или инструкции) после Else. Ветвь Else является необязательной.

Инструкция For Next

Повторяет выполнение инструкций, заданных в теле цикла столько раз, сколько раз интервал шага помещается в интервале между начальным и конечным значение счетчика, плюс единица.

Синтаксис: For <счётчик\_цикла> =<начало> To <конец> [Step <шаг>]

<тело цикла>

[Exit For]

. . .

Next [<счётчик\_цикла>]

## 4. Тестирование программного модуля

При тестировании программы ошибок и сбоев не выявлено.

При вводе неверных исходных данных выводятся окна с сообщениями об ошибке, представленные на рисунках 1-4.

Программа протестирована при размерности массива 3 на 4 и заполнении случайными числами . Пользовательская форма с результатом решения 1-го задания представлена на рисунке 8. Пользовательская форма с результатом решения 2-го задания представлена на рисунке 9. Пользовательская форма с результатом решения 3-го задания представлена на рисунке 10 Фрагмент листа Excel с результатом решения по всем пунктам представлен на рисунке 11.

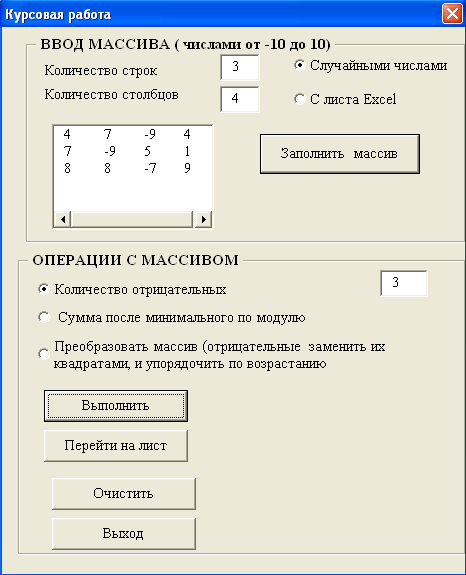


Рисунок 8 – Форма с результатом решения 1-го задания

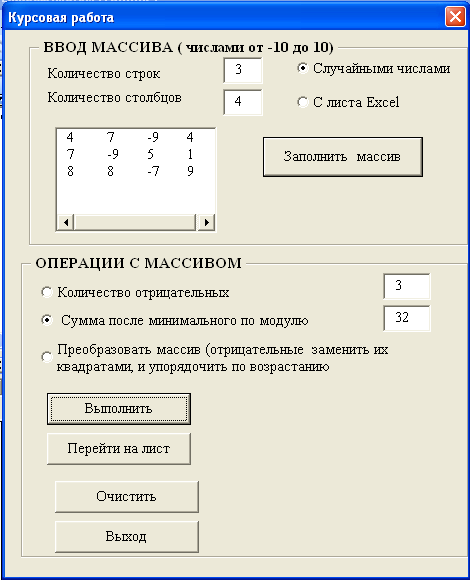


Рисунок 9 – Форма с результатом решения 2-го задания

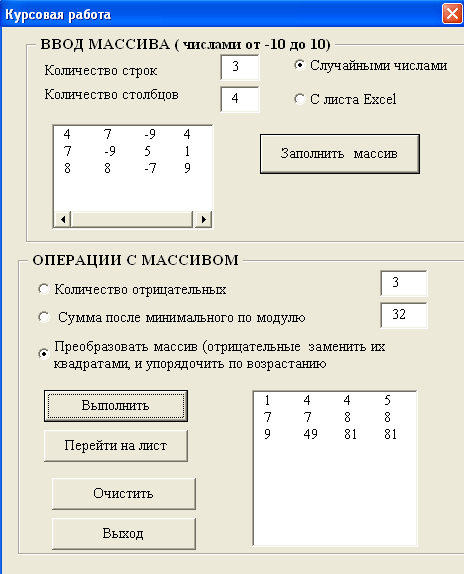


Рисунок 10 – Форма с результатом решения 3-го задания

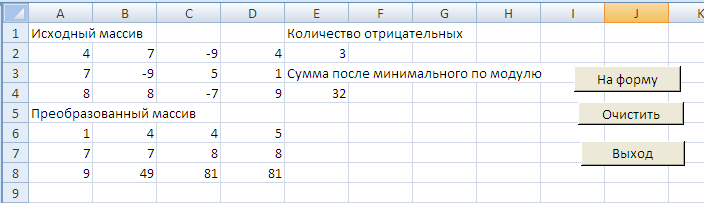


Рисунок 11 – Фрагмент листа Excel с результатом решения

## Заключение

Согласно заданию был разработан программный модуль, осуществляющий решение заданных действий над элементами массива заданной размерности. Программа позволяет ввести исходные: размерность матрицы, автоматически заполнить массив случайными целыми числами, а также вручную с листа Excel. Программа также позволяет с помощью переключателей выбрать действие над элементами массива в соответствии с заданием и выполнить эти действия.

При тестировании программного модуля разработанного в данной курсовой работе никаких отклонений не обнаружено: каждая функция корректно выполняет все операции. Таким образом, можно сделать вывод о том, что курсовая работа выполнена в полном объёме в соответствии с заданием, т.е. поставленная задача выполнена полностью.

## Список использованных источников

1. Ананьев А.И., Федоров А.Ф. Самоучитель Visual Basic 6.0. – СПб.: БХВ-Петербург, 2002. – 624 с. ил.

2. Гарнаев А.Ю. Самоучитель по VBA. – СПб.: БХВ-Петербург, 2000 – 512 с., ил.

3. Герасимович А.И. Рысюк А.И. Математический анализ. – Мн.; 1986г

4. Гурский Е.И. Основы линейной алгебры и аналитической геометрии. – Мн.: Высшая школа, 1982. – 272 с., ил.

5. Король В.И. Visual Basic 6.0 for Application. Язык программирования Справочник с примерами. – М.:“Кудиц-образ”, 2000 г., 442 с.

6. Мышкин А.Д. Лекции по высшей математике – Мн 1991г.

7. Ракитин В.И. “Практическое руководство по методам программного вычисления с приложением для ПК”. — М.: “Высшая школа”, 1998 г., с.38

8. Турчак Л.И. Основы численных методов. – Учеб. Пособие. – Наука, 1987 – 320 с., ил.

Приложение А