Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Кафедра «ВПМ»

Пояснительная записка к курсовой работе по дисциплине

«Алгоритмические языки и программирование»

на тему:  
«Работа с матрицами»

Выполнил: Мелёхин Д.В. студент группы 145 Проверил: Москвитина О.А.

Рязань 2022

Оглавление

[Задание на курсовую работу 3](#_Toc104145761)

[Введение 4](#_Toc104145762)

[Анализ задания и математическая постановка задачи 6](#_Toc104145763)

[Разработка схемы алгоритма и её описание 9](#_Toc104145764)

[Разработка программы и её описание 13](#_Toc104145765)

[Инструкция по использованию разработанной программы 14](#_Toc104145766)

[Результаты отладки 19](#_Toc104145767)

[Решение контрольных примеров и проверка правильности функционирования программы 20](#_Toc104145768)

[Текст программы и её описание 22](#_Toc104145769)

[Календарь выполнения работ 23](#_Toc104145770)

[Заключение 24](#_Toc104145771)

[Библиографический список 25](#_Toc104145772)

[Приложение 26](#_Toc104145773)

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

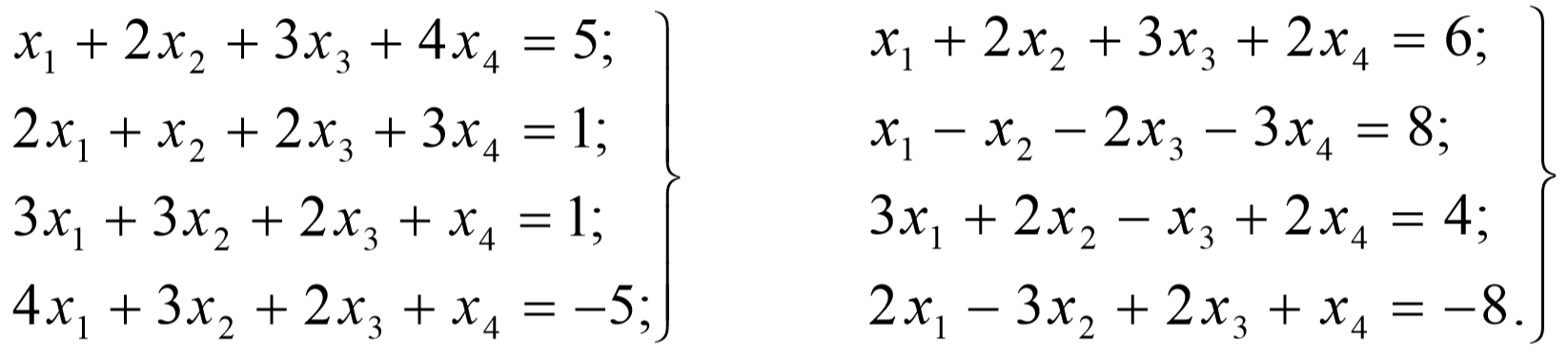
Кафедра «ВПМ»

Задание на курсовую работу по дисциплине

«Алгоритмические языки и программирование»

студенту Мелёхину Д.В группы 145

Составить процедуру вычисления обратной матрицы методом Жордана и с ее помощью решить системы линейных уравнений:



Также предусмотреть возможность ввода коэффициентов и свободных членов уравнений с помощью клавиатуры.

Срок представления работы к защите: 23 мая 2022 года

Дата выдачи задания: 05.02.22

Преподаватель: О.А. Москвитина

## Введение

Численные методы ‒ это, бесспорно, одно из мощнейших средств решения задач в математике. Даже в повседневной жизни человечество постоянно сталкивается хотя бы с простейшими вычислениями, например, выполняя арифметические действия в голове или на бумаге. С течением времени математика, развиваясь, становилась базой для многих других наук, давала толчок к их развитию; один из примеров: вся современная физика, астрономия и инженерия базируются на знаменитом открытии дифференциального и интегрального исчисления Ньютона и Лейбница конца XVII начала XVIII веков.

В математике существует замечательный инструмент ‒ уравнения. Они требуются для решения любой серьёзной научной задачи ‒ физической, инженерной ли экономической; например, для расчёта направления полёта космического корабля, обработки звука или графики, проектирования мостов и других силовых конструкций. Это лишь самая малая часть, для чего требуется данный инструмент. В общем, уравнение ‒ ключевая фигура в решении самых разнообразных вычислительных задач.

Часто при решении задачи условия приводят не к одному уравнению, а к целой их системе с несколькими неизвестными. Если с двумя взаимосвязанными уравнениями вручную справиться несложно, то пятью ‒ задача очень непростая. Естественно, это возможно, но данный процесс займёт немало времени. На помощь приходит программа данного проекта, разработанная в среде PascalABC.NET. Её главная задача ‒ решение систем алгебраических линейных уравнений (матриц или СЛАУ). Программа не содержит никакой лишней информации, работает быстро и безотказно. Простое взаимодействие, интуитивно понятный интерфейс, широкий функционал ‒ всё это её преимущества. От пользователя требуется только запустить исполняемый файл, выбрать способ ввода уравнений, внести нужные равенства и получить результат.

Для тех, кто умеет решать системы линейных алгебраических уравнений, программа сэкономит огромное количество времени, которое можно потрать на покорение новых вершин в математике. Остальным данный инструмент не только выполнит поставленную задачу, но и научит самостоятельно решать системы такого вида, что полезно для общего саморазвития. Это будет полезно в тех случаях, когда нет возможности воспользоваться программой.

Настоящий инструмент будет всегда актуален, вне зависимости от времени, ведь его можно использовать в инженерных расчётах, составлении баз данных, анализе электрических цепей, решении заданий в университете или для проверки вручную полученных ответов.

Программа реализует метод Жордана (французского математика) ‒ поиск корней уравнения с помощью элементарных преобразований, приводящих матрицу к нижнему треугольному виду, при котором переменные, расположенные ниже главной диагонали, обращаются в ноль. Корни уравнения - это числа, при подстановке которых уравнения принимают вид числового равенства (тождества).

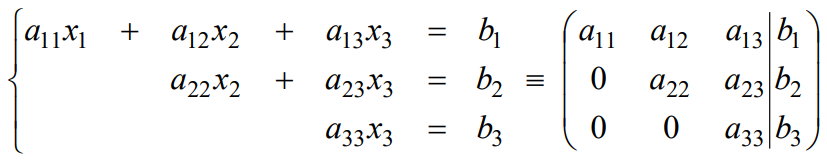
## Анализ задания и математическая постановка задачи

При решении поставленной задачи (см. стр. 3) необходимо выполнить следующие действия:

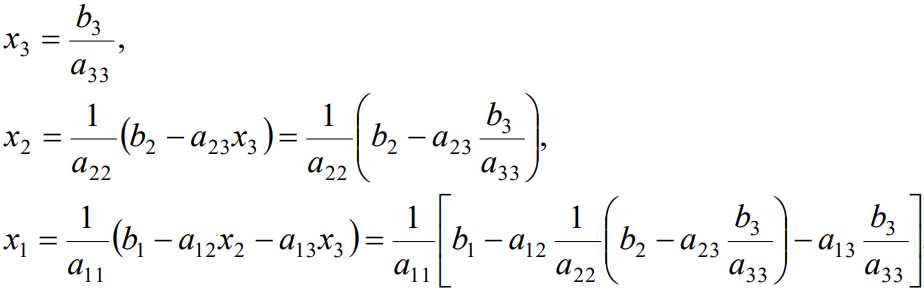
1. Ввести количество строк и столбцов .
2. Ввести элементов основной матрицы .
3. Ввести элементов столбца свободных членов .
4. Получить расширенную матрицу путём подстановки столбца справа от матрицы .
5. Преобразовать элементы, расположенные ниже главной диагонали, получив 0. После каждого обнуления уравнять элементы рассматриваемой строки.
6. Записать получившуюся треугольную матрицу.
7. Получить столбец неизвестных .

Для наглядности процесса решения после выполнения пунктов 2 ‒ 6 следует производить печать полученных матриц. Действия 4 ‒ 7 целесообразно оформить в виде подпрограмм. После выполнения пункта 7 производится печать столбца неизвестных.

Нижняя треугольная матрица имеет вид:



Корни данной системы находятся по формулам:



Укрупнённая блок-схема алгоритма показана на рисунке 1.

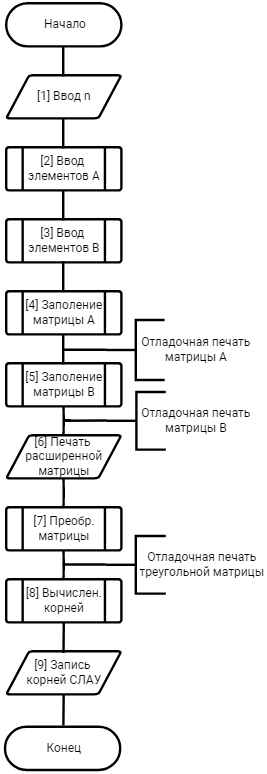


Рисунок 1 ‒ Укрупнённая блок-схема алгоритма

## Разработка схемы алгоритма и её описание

Алгоритм программы состоит из следующих пунктов:

1. Ввод элементов матрицы (блок 2) оформлен в виде подпрограммы с вложенным циклом. В данном блоке определён ввод значений элементов квадратной матрицы.
2. Ввод элементов столбца свободных членов (блок 3) оформлен в виде подпрограммы, содержащей цикл с параметром. В данном блоке определён ввод значений элементов столбца .
3. Заполнение матрицы (блок 4) реализуется в виде подпрограммы с вложенным циклом (рисунок 2). Внешний цикл имеет параметры , ; внутренний ‒ , . Каждый элемент матрицы имеет индексы .
4. Заполнение матрицы (блок 5) осуществляет подпрограмма, содержащая цикл с параметром (рисунок 3). Элемент матрицы имеет индекс .
5. Печать исходных матриц и производится в процедурах совместно с заполнением.
6. Печать расширенной матрицы (блок 6) осуществляется с помощью вложенного цикла.
7. Преобразование матрицы (блок 7) производится с помощью процедуры, показанной на рисунке 4.
8. Печать треугольной матрицы производится в процедуре преобразования матрицы.
9. Вычисление корней (блок 8) выполняет вложенный цикл (рисунок 5), расположенный в процедуре преобразования матриц.
10. Запись корней СЛАУ (блок 9) выполняет цикл с параметром.

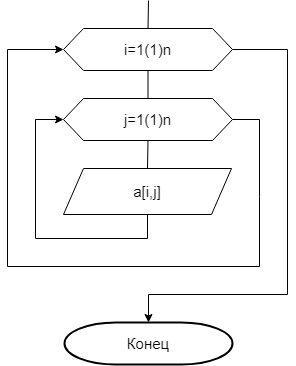


Рисунок 2 ‒ Заполнение матрицы

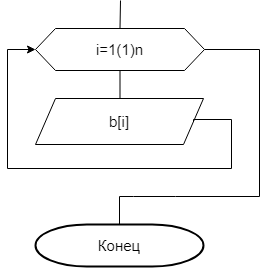


Рисунок 3 ‒ Заполнение столбца

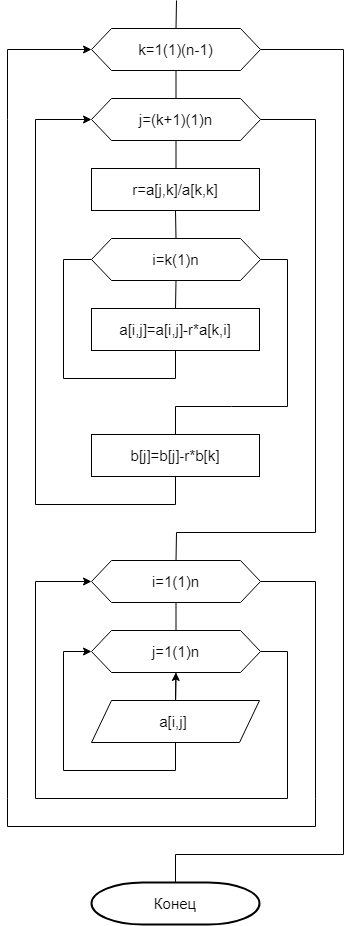


Рисунок 4 ‒ Преобразование матрицы

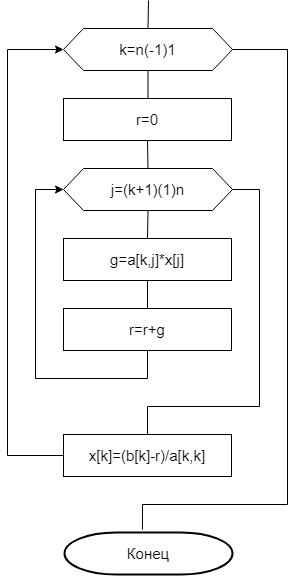


Рисунок 5 ‒ Вычисление корней

## Разработка программы и её описание

Данная программа состоит из 3 модулей. Использование модульного программирования позволило упростить тестирование программы и обнаружение ошибок. Смежные элементы, образующие группы подзадач, строго отделены от остальной составляющей проекта, что улучшило мобильность созданной программы.

Первый модуль является основой программой. Это «скелет» проекта, в котором организован вывод информации на экран и вызов процедур из двух других блоков. Данных модуль содержит оператор выбора case, позволяющий осуществить требуемый способ ввода СЛАУ. Первый и второй способ задают фиксированные системы уравнений, которые необходимо решить для выполнения задания. Третий способ ‒ ручной ввод СЛАУ. Он позволяет выбрать требуемые элементы и размерность матриц. При выборе четвёртого способа программа создаёт матрицы введённой размерности, содержащие случайные элементы. Максимальная введённая размерность не должна превышать 10.

Второй модуль ‒ оформление проекта. В нём описаны процедуры вывода титульного листа, формулировки задания и описания метода решения программы. Данные подпрограммы осуществляют работу с текстовыми файлами, считывая исходные строки. Здесь также содержатся процедуры для вывода разделительной линии и отступа между элементами.

Третий модуль отвечает за вычислительную составляющую программы. Здесь описаны самые главные процедуры программы ‒ приведение расширенной матрицы к треугольному виду и вычисление корней уравнений. Для наглядности печать полученных матриц выполняется после каждого шага преобразования. Также модуль содержит ввод и вывод элементов для ручного составления СЛАУ.

## Инструкция по использованию разработанной программы

Для полноценной работы программы требуется система, соответствующая следующим техническим требованиям:

1. ОС: Windows 10 / 8 / 7 / XP / Vista
2. ЦПУ: Intel или AMD (от 1 ГГц)
3. ОЗУ: 256 Мб
4. HDD: 150 Мб

Обязательным периферийным устройством является клавиатура. Для удобства работы с программой следует использовать компьютерную мышь или трекбол.

Программа поставляется в папке Program, содержащей 4 файла: Jordan.exe, title.txt, task.txt, method.txt

Файл title.txt содержит текст титульного листа; в task.txt размещён текст задания, в method.txt ‒ метод, по которому программа осуществляет вычисления.

Для запуска программы необходимо открыть исполняемый файл Jordan.exe. Это можно сделать несколькими способами. С помощью мыши:

* Дважды кликнуть основной кнопкой мыши на исполняемый файл.
* Один раз кликнуть вспомогательной кнопкой мыши; появится выпадающий список, в котором основной кнопкой необходимо выбрать пункт «открыть».

С помощью клавиатуры:

* Направляющими стрелками выбрать требуемый файл и нажать enter.

После открытия программа выведет на экран текстовые файлы и способы ввода системы уравнения. Для выбора требуемого способа с помощью клавиатуры необходимо ввести соответствующую цифру (рисунок 6):

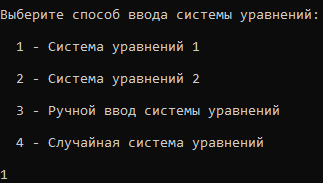


Рисунок 6 ‒ Выбор способа ввода СЛАУ

При выборе пунктов 1 и 2 программа выведет решение предустановленных систем уравнений.

На примере ручного ввода СЛАУ рассмотрим процесс решения. Первый шаг ‒ ввод размерности матрицы (рисунок 7):



Рисунок 7 ‒ Ввод размерности матрицы

Далее указываем значения СЛАУ. Каждый элемент матрицы имеет 2 индекса, первый из которых является номером строки, а второй ‒ номером столбца. После ввода значений программа выведет на экран полученную матрицу (рисунок 8):

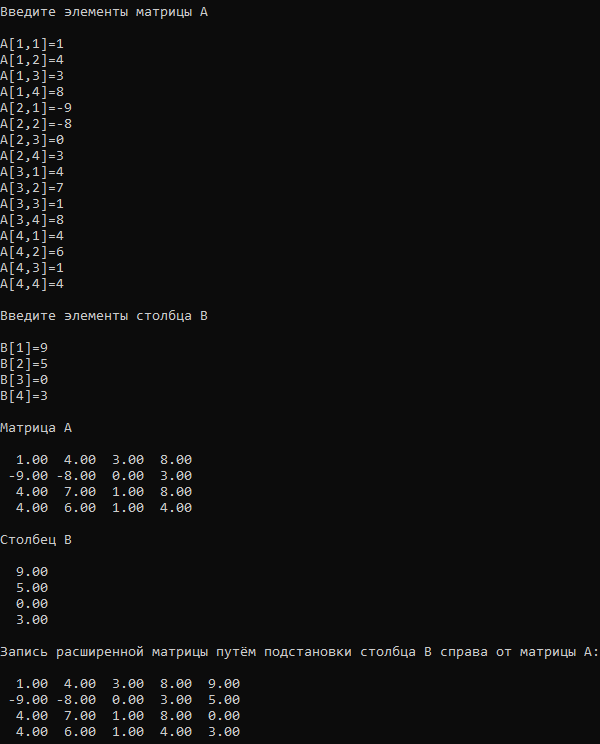


Рисунок 8 ‒ Полученная расширенная матрица

Следующий этап ‒ приведение расширенной матрицы к треугольному виду. Программа выводит на экран каждый шаг преобразования (рисунок 9):

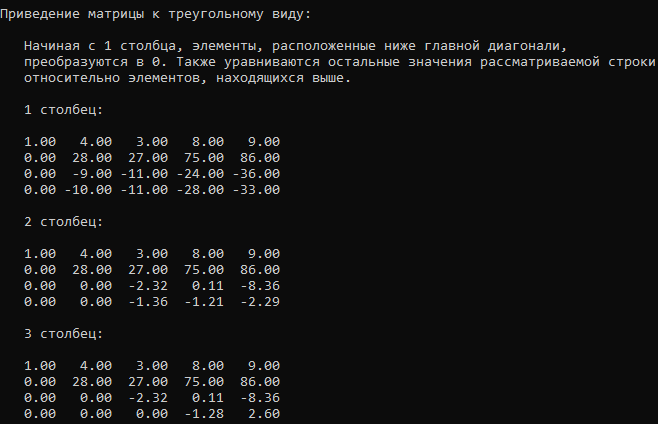


Рисунок 9 ‒ Приведение матрицы к треугольному виду

В зависимости от результата программа определяет совместность матрицы. Если система совместна, то происходит вычисление корней и вывод на экран (рисунок 10):

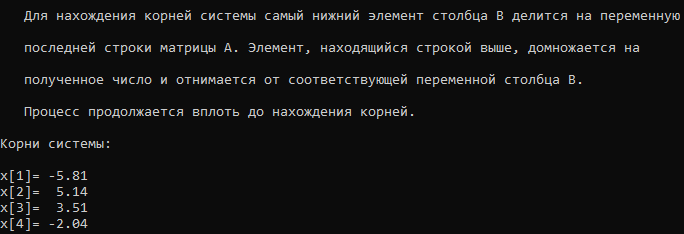


Рисунок 10 ‒ Запись корней уравнения

Если же система несовместна, то программа предупредит об этом (рисунок 11):

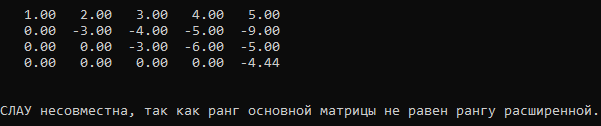


Рисунок 11 ‒ Несовместная СЛАУ

После решения системы есть возможность сразу перейти к следующей, проделав вышеописанные шаги снова. При этом предыдущие решения сохраняются на экране.

Возможные реакции программы:

1. Неправильно введённый способ решения СЛАУ. Если выбранный вариант находится вне существующих [1..4], то программа будет предлагать ввести значение заново до тех пор, пока оно не будет находиться в допустимом интервале.
2. Неправильно введённая размерность матрицы. Если указанное число находится вне диапазона [1..10], то программа будет предлагать ввести значение заново до тех пор, пока оно не будет допустимым.

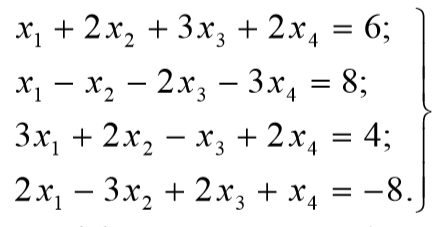
Для закрытия программы необходимо нажать на крестик в правом верхнем углу экрана.

## Результаты отладки

1. computing.pas(1) : Имя модуля должно совпадать с именем файла.
2. Jordan.pas(23) : Неизвестное имя 'title'. Данная ошибка была получена при вызове процедуры чтения текстового документа, описанной в модуле. Причиной этому было отсутствие имени процедуры в разделе интерфейсов.
3. Jordan.pas(166) : Данный объект не может быть передан как var-параметр. Ошибка была получена после применения в заголовке процедуры параметра var к символьному типу.

## Решение контрольных примеров и проверка правильности функционирования программы

Для контрольного подсчета возьмём систему уравнений 2:



При решении данной СЛАУ программа определила, что система совместна и имеет решения, так как число ненулевых строк основной матрицы равно числу ненулевых строк расширенной матрицы. Корни, полученные программой (рисунок 12):

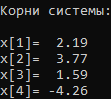


Рисунок 12 ‒ Корни системы уравнений 2

Для проверки правильности работы программы обратимся к проверенному калькулятору СЛАУ[[1]](#footnote-2), результат решения которого показан на рисунке 13:

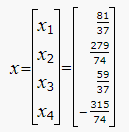


Рисунок 13 ‒ Проверка решений

Сравнив результаты, можно сделать вывод, что они идентичны. Программа работает правильно.

## Текст программы и её описание

Программа для решения систем линейных алгебраических уравнений методом Жордана разработана в среде PascalABC.NET. Для начала работы программы необходимо запустить исполняемый файл Jordan.exe, расположенный в папке Program. Исходные данные, имеющие смысл значений элементов массива и размерности, вводятся в диалоговом режиме. Значения элементов массива, состоящего из корней СЛАУ, вычисляются по заданной формуле (см. стр. 7). В процессе получения треугольной матрицы реализованы следующие действия с массивами:

* Умножение и деление на число
* Сложение и вычитание с другими элементами

Для наглядности решения каждая операция с матрицами сопровождается печатью соответствующего массива (данные действия описаны как в подпрограмме модуля, так и в теле основной программы).

## Календарь выполнения работ

|  |  |  |  |
| --- | --- | --- | --- |
| *№*  *n/n* | Этапы решения задачи | Ориентировочная трудоёмкость, ч | Реально затраченное время, ч |
| 1 | Анализ предметной области и формализация поставленной задачи | 8 | 3 |
| 2 | Разработка структуры организации данных | 8 | 4 |
| 3 | Составление алгоритма | 12 | 12 |
| 4 | Написание программы | 8 | 10 |
| 5 | Устранение синтаксических ошибок | 4 | 3 |
| 6 | Подготовка тестов | 8 | 2 |
| 7 | Тестирование и отладка | 24 | 8 |
| 8 | Подготовка документации | 8 | 12 |
| Итого: | | 80 | 54 |

## Заключение

По итогу мы получили завершённый проект, позволяющий решать системы линейных алгебраических уравнений. Данная программа является удобным и быстрым инструментом. Она позволяет работать с системами, имеющими до 10 неизвестных переменных включительно. Пользователь имеет возможность вводить уравнения как вручную, так и случайным способом, при этом программа автоматически генерирует значения элементов. Самым сложным этапом была разработка алгоритма, по которому преобразуются матрицы и вычисляются неизвестные. Программа имеет минималистичный дизайн, что даёт заказчику возможность оформить её под свои цели. Данный проект является базой для создания полноценного калькулятора СЛАУ. В программу можно внедрить альтернативные способы вычисления матриц: метод Крамера, Гаусса, Зейделя, Якоби; метод прогонки, релаксации, простых итераций; матричный и многосеточный методы.

## Библиографический список

* Новиков А. И. Начала линейной алгебры и аналитическая геометрия. — М.: ФИЗМАТЛИТ, 2015. — 376 с. —ISBN 978-5-9221-1618-3.
* Линейная алгебра: учеб. пособие / Н.В. Гредасова, М.А. Корешникова, Н.И. Желонкина [и др.]; Мин-во науки и высш. образования РФ.— Екатеринбург: Изд-во Урал. ун-та, 2019.— 88 с.
* Белоусов И. В. МАТРИЦЫ И ОПРЕДЕЛИТЕЛИ: учебное пособие по линейной алгебре. / Кишинев: 2006/.
* Гарслян А.Е. МЕТОД ЖОРДАНА - ГАУССА. Методические указания к практическим занятиям. – М.: МИИТ, 2010.–28 с.
* Долинер, Л. И. Основы программирования в среде PascalABC.NET: учебное пособие / Л. И. Долинер. – Екатеринбург: Изд-во Урал. ун-та, 2014. – 128 с. ISBN 978-5-7996-1260-3.
* Кадырова, Г. Р. Основы алгоритмизации и программирования : учебное пособие / Г. Р. Кадырова. – Ульяновск : УлГТУ, 2014. – 95 с.

## Приложение

Код основной программы:

**Program** Jordan;

**USES**

titlepage, computing; {подключаемые модули}

**var** {описание переменных}

f1:text;

s:string;

a,a1: Aarray;

b: Barray;

x: Barray;

n, i, j, v: integer;

**begin** {Jordan}

title(f1,s);

otstup;

sharp;

otstup;

task(f1,s);

otstup;

sharp;

otstup;

method(f1,s);

otstup;

sharp;

otstup;

writeln;

**repeat** {выбор способа ввода уравнений}

writeln('Выберите способ ввода системы уравнений:');

writeln;

writeln(' 1 - Система уравнений 1');

writeln;

writeln(' 2 - Система уравнений 2');

writeln;

writeln(' 3 - Ручной ввод системы уравнений');

writeln;

writeln(' 4 - Случайная система уравнений');

writeln;

read(v);

Writeln;

**case** v **of**

1: **begin** {система уравнений 1}

n:=4;

A[1, 1] := 1; A[1, 2] := 2; A[1, 3] := 3; A[1, 4] := 4; {значения элементов матрицы A и столбца B}

A[2, 1] := 2; A[2, 2] := 1; A[2, 3] := 2; A[2, 4] := 3;

A[3, 1] := 3; A[3, 2] := 3; A[3, 3] := 2; A[3, 4] := 1;

A[4, 1] := 4; A[4, 2] := 3; A[4, 3] := 2; A[4, 4] := 1;

b[1] := 5; b[2] := 1; b[3] := 1; b[4] := -5;

writeln('Матрица A: '); {печать матрицы A}

writeln;

**for** i := 1 **to** n **do**

**begin**

**for** j := 1 **to** n **do**

**begin**

write(a[i,j]:7:2);

a1[i,j]:=a[i,j];

**end**;

writeln;

**end**;

writeln;

writeln('Столбец B: '); {печать столбца B}

writeln;

**for** i:=1 **to** n **do**

writeln(b[i]:7:2);

writeln;

writeln('Запись расширенной матрицы путём подстановки столбца B справа от матрицы A:');

writeln;

**for** i := 1 **to** n **do** {запись расширенной матрицы}

**begin**

**for** j := 1 **to** n **do**

**begin**

write(a[i,j]:7:2);

**end**;

writeln(b[i]:7:2);

**end**;

writeln;

matrix(n,a,b,x); {вызов процедуры приведения матрицы к треугольному виду}

writeln;

**if** a[n,n]=0 **then** {условие несовместной матрицы}

writeln('СЛАУ несовместна, так как ранг основной матрицы не равен рангу расширенной.');

writeln;

**if** a[n,n]<>0 **then** {условие совместной матрицы}

**begin**

writeln(' Для нахождения корней системы самый нижний элемент столбца B делится на переменную');

writeln;

writeln(' последней строки матрицы A. Элемент, находящийся строкой выше, домножается на');

writeln;

writeln(' полученное число и отнимается от соответствующей переменной столбца В.');

writeln;

writeln(' Процесс продолжается вплоть до нахождения корней.');

writeln;

writeln('Корни системы:');

**for** i := 1 **to** n **do** {печать корней системы}

writeln('x[', i, ']=', x[i]:6:2, ' ');

**end**;

**end**;

2: **begin** {система уравнений 2}

n:=4;

A[1, 1] := 1; A[1, 2] := 2; A[1, 3] := 3; A[1, 4] := 2; {значения элементов матрицы A и столбца B}

A[2, 1] := 1; A[2, 2] := -1; A[2, 3] := -2; A[2, 4] := -3;

A[3, 1] := 3; A[3, 2] := 2; A[3, 3] := -1; A[3, 4] := 2;

A[4, 1] := 2; A[4, 2] := -3; A[4, 3] := 2; A[4, 4] := 1;

b[1] := 6; b[2] := 8; b[3] := 4; b[4] := -8;

writeln('Матрица A: '); {печать матрицы A}

writeln;

**for** i := 1 **to** n **do**

**begin**

**for** j := 1 **to** n **do**

**begin**

write(a[i,j]:6:2);

a1[i,j]:=a[i,j];

**end**;

writeln;

**end**;

writeln;

writeln('Столбец B: '); {печать столбца B}

writeln;

**for** i:=1 **to** n **do**

writeln(b[i]:7:2);

writeln;

writeln('Запись расширенной матрицы путём подстановки столбца B справа от матрицы A:');

writeln;

**for** i := 1 **to** n **do** {запись расширенной матрицы}

**begin**

**for** j := 1 **to** n **do**

**begin**

write(a[i,j]:6:2);

**end**;

writeln(b[i]:6:2);

**end**;

writeln;

matrix(n,a,b,x); {вызов процедуры приведения матрицы к треугольному виду}

**if** a[n,n]=0 **then** {условие несовместной матрицы}

writeln('СЛАУ несовместна, так как ранг основной матрицы не равен рангу расширенной.');

**if** a[n,n]<>0 **then** {условие совместной матрицы}

**begin**

writeln(' Для нахождения корней системы самый нижний элемент столбца B делится на переменную');

writeln;

writeln(' последней строки матрицы A. Элемент, находящийся строкой выше, домножается на');

writeln;

writeln(' полученное число и отнимается от соответствующей переменной столбца В.');

writeln;

writeln(' Процесс продолжается вплоть до нахождения корней.');

writeln;

writeln('Корни системы:');

writeln;

**for** i := 1 **to** n **do** {печать корней системы}

writeln('x[', i, ']=', x[i]:6:2, ' ');

writeln;

**end**;

**end**;

3: **begin** {ручной ввод системы уравнений}

**repeat**

writeln('Введите число строк и столбцов матрицы A <= ',AmatrixSize,' '); {ввод размерности матрицы}

read(n);

writeln;

**until** n **in** [1..AmatrixSize];

inArr1(n,a,'A'); {процедура ввода элементов матрицы А}

inArr2(n,b,'B'); {процедура ввода элементов столбца B}

outArr1(a,n); {процедура заполнения и печати матрицы А}

outArr2(b,n); {процедура заполнения и печати столбца В}

writeln('Запись расширенной матрицы путём подстановки столбца B справа от матрицы A:');

writeln;

**for** i := 1 **to** n **do** {запись расширенной матрицы}

**begin**

**for** j := 1 **to** n **do**

**begin**

write(a[i,j]:6:2);

**end**;

writeln(b[i]:6:2);

**end**;

writeln;

matrix(n,a,b,x); {вызов процедуры приведения матрицы к треугольному виду}

**if** a[n,n]=0 **then** {условие несовместной матрицы}

writeln('СЛАУ несовместна, так как ранг основной матрицы не равен рангу расширенной.');

writeln;

**if** a[n,n]<>0 **then** {условие совместной матрицы}

**begin**

writeln(' Для нахождения корней системы самый нижний элемент столбца B делится на переменную');

writeln;

writeln(' последней строки матрицы A. Элемент, находящийся строкой выше, домножается на');

writeln;

writeln(' полученное число и отнимается от соответствующей переменной столбца В.');

writeln;

writeln(' Процесс продолжается вплоть до нахождения корней.');

writeln;

writeln('Корни системы:');

writeln;

**for** i := 1 **to** n **do** {печать корней системы}

writeln('x[', i, ']=', x[i]:6:2, ' ');

writeln;

**end**;

**end**;

4: **begin** {случайная система уравнений}

randomize;

**repeat**

writeln('Введите число строк и столбцов матрицы A <= ',AmatrixSize,' '); {ввод размерности матрицы}

read(n);

writeln;

**until** n **in** [1..AmatrixSize];

writeln('Матрица A:');

writeln;

**for** i:=1 **to** n **do** {заполнение матрицы А}

**for** j:=1 **to** n **do**

**repeat**

a[i][j]:=-9+random(19);

**until** a[i][j]<>0;

**for** i:=1 **to** n **do** {печать матрицы А}

**begin**

**for** j:=1 **to** n **do**

write(a[i][j]:7:2);

writeln;

**end**;

writeln;

writeln('Столбец B:');

writeln;

**for** i:=1 **to** n **do** {заполнение столбца В}

**repeat**

b[i]:=-9+random(19);

**until** b[i]<>0;

**for** i:= 1 **to** n **do** {печать столбца В}

**begin**

write(b[i]:7:2);

writeln;

**end**;

writeln;

writeln('Запись расширенной матрицы путём подстановки столбца B справа от матрицы A:');

writeln;

**for** i := 1 **to** n **do** {печать расширенной матрицы}

**begin**

**for** j := 1 **to** n **do**

**begin**

write(a[i,j]:6:2);

**end**;

writeln(b[i]:6:2);

**end**;

writeln;

matrix(n,a,b,x); {вызов процедуры приведения матрицы к треугольному виду}

**if** a[n,n]=0 **then** {условие несовместной матрицы}

writeln('СЛАУ несовместна, так как ранг основной матрицы не равен рангу расширенной.');

writeln;

**if** a[n,n]<>0 **then** {условие совместной матрицы}

**begin**

writeln(' Для нахождения корней системы самый нижний элемент столбца B делится на переменную');

writeln;

writeln(' последней строки матрицы A. Элемент, находящийся строкой выше, домножается на');

writeln;

writeln(' полученное число и отнимается от соответствующей переменной столбца В.');

writeln;

writeln(' Процесс продолжается вплоть до нахождения корней.');

writeln;

writeln('Корни системы:');

writeln;

**for** i := 1 **to** n **do** {печать корней системы}

writeln('x[', i, ']=', x[i]:6:2, ' ');

writeln;

**end**;

**end**;

**end**;

**until** v=-1;

**end**. {Jordan}

Модуль titlepage:

**UNIT** titlepage;

**INTERFACE**

**procedure** title(f1:text; s:string);

**procedure** sharp;

**procedure** otstup;

**procedure** task(f1:text; s:string);

**procedure** method(f1:text; s:string);

**IMPLEMENTATION**

**procedure** title(f1:text; s:string); {печать титульного листа}

**begin**

assign(f1,'title.txt');

reset(f1);

**while not** eof(f1) **do**

**begin**

readln(f1,s);

writeln(s);

**end**;

**end**;

**procedure** sharp; {разделительная линия}

**var** i:integer;

**begin**

**for** i:=1 **to** 237 **do**

**begin**

write('#');

write;

**end**;

**end**;

**procedure** otstup; {отступ}

**var** i:integer;

**begin**

**for** i:=1 **to** 4 **do**

**begin**

writeln(' ');

**end**;

**end**;

**procedure** task(f1:text; s:string); {печать формулировки задания}

**begin**

assign(f1,'task.txt');

reset(f1);

**while not** eof(f1) **do**

**begin**

readln(f1,s);

writeln(s);

**end**;

**end**;

**Procedure** method(f1:text; s:string); {печать метода вычисления программы}

**begin**

assign(f1,'method.txt');

reset(f1);

**while not** eof(f1) **do begin**

readln(f1,s);

writeln(s);

**end**;

**end**;

**End**.

Модуль computing:

**UNIT** computing;

**INTERFACE**

**const**

AmatrixSize = 10;

**Type**

Aarray = **array**[1..AmatrixSize, 1..AmatrixSize] **of** real;

Barray = **array**[1..AmatrixSize] **of** real;

**procedure** inArr1(n:integer; **var** a:Aarray; Name:char);

**procedure** inArr2(n:integer; **var** b:Barray; Name:char);

**procedure** outArr1(**var** a:Aarray; n:integer);

**procedure** outArr2(**var** b:Barray; n:integer);

**procedure** matrix(**var** n:integer; **var** a: Aarray; **var** b,x: Barray);

**IMPLEMENTATION**

**procedure** inArr1(n:integer; **var** a:Aarray; Name:char); {ввод элементов матрицы A}

**var**

i, j:integer;

**begin**

writeln('Введите элементы матрицы ',Name);

writeln;

**for** i:= 1 **to** n **do**

**for** j:= 1 **to** n **do**

**begin**

write(Name,'[',i,',',j,']=');

read(a[i,j])

**end**;

writeln;

**end**;

**procedure** inArr2(n:integer; **var** b:Barray; Name:char); {ввод элементов столбца B}

**var**

i:integer;

**begin**

writeln('Введите элементы столбца ',Name);

writeln;

**for** i:= 1 **to** n **do**

**begin**

write(Name,'[',i,']=');

read(b[i])

**end**;

writeln;

**End**;

**procedure** outArr1(**var** a:Aarray; n:integer); {заполнение и печать матрицы A}

**var**

i, j:integer;

**begin**

writeln('Матрица A');

writeln;

**for** i:= 1 **to** n **do**

**begin**

**for** j:= 1 **to** n **do**

write(a[i,j]:6:2);

writeln;

**end**;

writeln;

**end**;

**procedure** outArr2(**var** b:Barray; n:integer); {заполнение и печать стобца B}

**var**

i:integer;

**begin**

writeln('Столбец B');

writeln;

**for** i:= 1 **to** n **do**

**begin**

write(b[i]:6:2);

writeln

**end**;

writeln

**end**;

**procedure** matrix(**var** n:integer; **var** a: Aarray; **var** b,x: Barray); {приведение матрицы к треугольному виду и отладочная печать}

**var** k,i,j:integer; r,g: real;

**begin**

writeln('Приведение матрицы к треугольному виду:');

writeln;

writeln(' Начиная с 1 столбца, элементы, расположенные ниже главной диагонали, ');

writeln;

writeln(' преобразуются в 0. Также уравниваются остальные значения рассматриваемой строки');

writeln;

writeln(' относительно элементов, находящихся выше.');

writeln;

**for** k := 1 **to** (n-1) **do**

**begin**

**for** j := k + 1 **to** n **do**

**begin**

r := a[j, k] / a[k, k];

**for** i := k **to** n **do**

**begin**

a[j, i] := a[j, i] - r \* a[k, i];

**end**;

b[j] := b[j] - r \* b[k];

**end**;

writeln(' ',k,' столбец:');

writeln;

**for** i := 1 **to** n **do**

**begin**

**for** j := 1 **to** n **do**

**begin**

write(a[i,j]:7:2);

**end**;

writeln(b[i]:7:2);

**end**;

writeln;

**end**;

**for** k := n **downto** 1 **do** {нахождение корней уравнений}

**begin**

r := 0;

**for** j := k + 1 **to** n **do**

**begin**

g := a[k, j] \* x[j];

r := r + g;

**end**;

x[k] := (b[k] - r) / a[k, k];

**end**;

**end**;

**end**.

1. Мир математики [Электронный ресурс] //matworld.ru: Мир математики, 2012. URL: <https://matworld.ru/calculator> (дата обращения 16.05.2022). [↑](#footnote-ref-2)