Муниципальное бюджетное

Общеобразовательное учреждение

города Новосибирска

«Аэрокосмический лицей имени Ю. В. Кондратюка»

Научно-практическая конференция

«У истоков освоения космоса»

Исследовательская работа

“Решение систем уравнений по методу Гаусса”

Выполнили:

Машкин Николай, Жданов Роман

ученики 9Б класса

Руководитель:

Клековкина Т. В.

Педагог ВКК

**Оглавление**

[Введение 2](#_Toc98008926)

[Цели и задачи 2](#_Toc98008927)

[Матрицы 2](#_Toc98008928)

[Определение 2](#_Toc98008929)

[Общий вид матрицы 3](#_Toc98008930)

[Операции над матрицами 3](#_Toc98008931)

[Операции внутри матриц 5](#_Toc98008932)

[Решение систем уравнений методом Гаусса 6](#_Toc98008933)

[Определение 6](#_Toc98008934)

[Алгоритм решения 6](#_Toc98008935)

[Пример 7](#_Toc98008936)

[Реализация метода Гаусса на Python 8](#_Toc98008937)

[Создание приложения 10](#_Toc98008938)

[Вывод 14](#_Toc98008939)

[Список литературы 15](#_Toc98008940)

# Введение

Часто нам приходится решать различные системы уравнений. Иногда это легкие системы с одной или двумя переменными, но бывают и системы посложнее. Их удобно решать с помощью матриц. Они позволяют избавиться от переменных и оперировать только коэффициентами.

Но всегда ли нам хочется делать работу вручную? Конечно же проще и быстрее будет написать систему в поле для ввода текста, а программа сделает все сама, да еще и распишет подробное решение.

# Цели и задачи

Цель: написать программу, которая сможет решать системы уравнений.

Задачи:

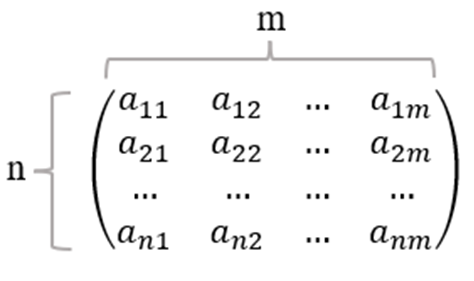
1. Узнать, что такое матрицы и какие действия можно над ними проводить
2. Научиться решать системы уравнений методом Гаусса
3. Написать алгоритм решения на языке Python
4. Создать интерфейс приложения на C#

# Матрицы

## Определение

Матрица — математический объект, записываемый в виде прямоугольной таблицы элементов, который представляет собой совокупность строк и столбцов, на пересечении которых находятся его элементы. Количество строк и столбцов задает размер матрицы.

## Общий вид матрицы



n – количество строк

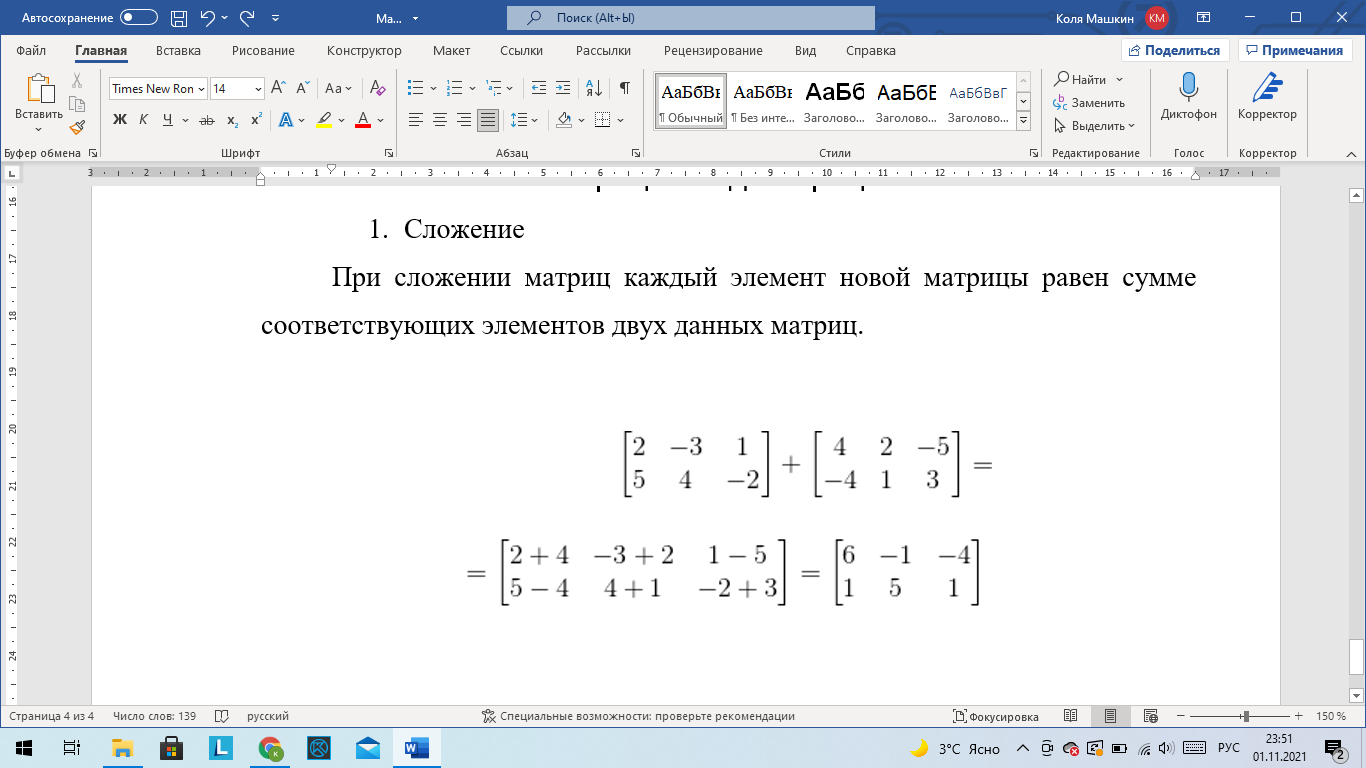
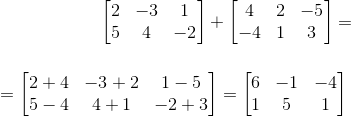
m – количество столбцов

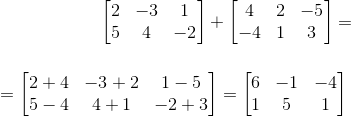
a – элемент матрицы

## Операции над матрицами

1. **Сложение**

При сложении матриц каждый элемент новой матрицы равен сумме соответствующих элементов двух данных матриц.





*Нельзя складывать матрицы разных размеров!!!*

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

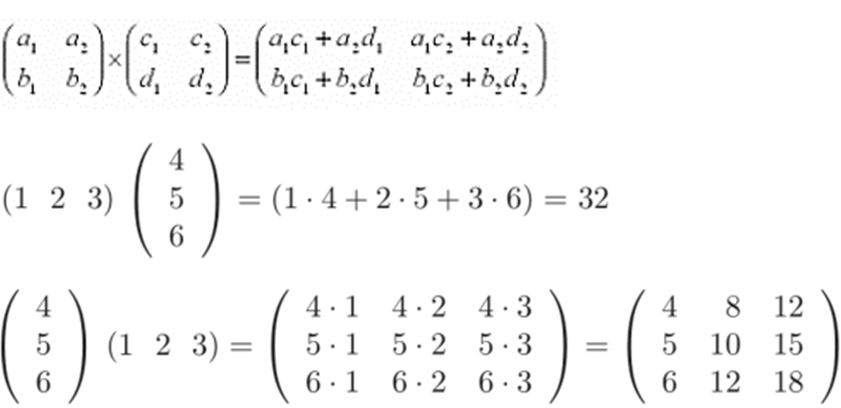
Где Θ – нулевая матрица (все элементы равны 0)

1. **Умножение**

Чтобы матрицу А можно было умножить на матрицу В нужно, чтобы число столбцов матрицы А равнялось числу строк матрицы В.

Для умножения двух матриц мы по очереди каждый столбец первой матрицы скалярно умножаем на каждую строку второй матрицы.

При скалярном умножении двух векторов мы находим сумму произведения каждых соответствующих элементов обоих векторов:

****

Свойства умножения матриц:

* Ассоциативность: (AB)C = A(BC)
* Некоммутативность (в общем случае): AB ≠ BA

Исключение: произведение коммутативно в случае умножения с единичной матрицей: AI = IA

* Дистрибутивность: (A+B)C = AC + BC ⇔ A(B+C) = AB + AC
* Ассоциативность и коммутативность относительно умножения на число n: (nA)\*B = n\*(AB) = A\*(nB)

# Операции внутри матриц

1. **Сложение строк**

При сложении двух строк данной матрицы получается новая строка, которая заменяет одну из исходных строк или добавляется к данной матрице. Каждый элемент новой строки равен сумме соответствующих элементов исходных строк.

Также можно прибавлять к одной строке другую, умноженную на число.



Изображение выглядит как текст

Автоматически созданное описание



1. **Умножение строки на коэффициент**

При умножении строки матрицы на коэффициент каждый элемент строки умножается на этот коэффициент. Получившаяся строка либо заменяет исходную, либо добавляется к матрице.





Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Строки матрицы можно переставлять местами.

Например, в этой матрице можно безболезненно переставить первую и вторую строки:



# Решение систем уравнений методом Гаусса

## Определение

Метод Гаусса - метод последовательного исключения переменных, когда с помощью преобразований система уравнений приводится к равносильной системе треугольного вида, из которой последовательно, начиная с последних, находятся все переменные системы.

Матрица треугольного вида – матрица, у которой все элементы под главной диагональю равны нулю.

Изображение выглядит как текст, часы, коллекция картинок, датчик

Автоматически созданное описание

## Алгоритм решения

1. Ищем максимальный по модулю элемент в n-столбце
2. Переставляем строку с найденным элементом наверх и нормализуем (делим всю строку на этот элемент)
3. Обрабатываем нижележащие строки (добиваемся того, чтобы в остальных строках под элементом стояли нули)
4. Повторяем действия с каждым столбцом
5. Переходим обратно к системе уравнений
6. Подставляем переменные снизу вверх

Представленный выше алгоритм удобен для программирования, но если в матрице есть строка с первым элементом равным 1, то можно переместить ее наверх и обойтись без деления, сразу перейдя к нормализации нижележащих строк.

## Пример

Решим данную систему уравнений:



На первом этапе нужно записать расширенную матрицу системы:  


*Матрица системы* – это матрица, составленная только из коэффициентов при неизвестных.

*Расширенная матрица системы* – обычная матрица системы плюс столбец свободных членов.

Далее умножаем первую строку на –2:

Прибавляем первую строку ко второй строке: 

Теперь первую строку можно обратно разделить на –2: 

Делим вторую строку на 3:

Переходим обратно к системе:



Подставляем *у* в первое уравнение и находим *х*:



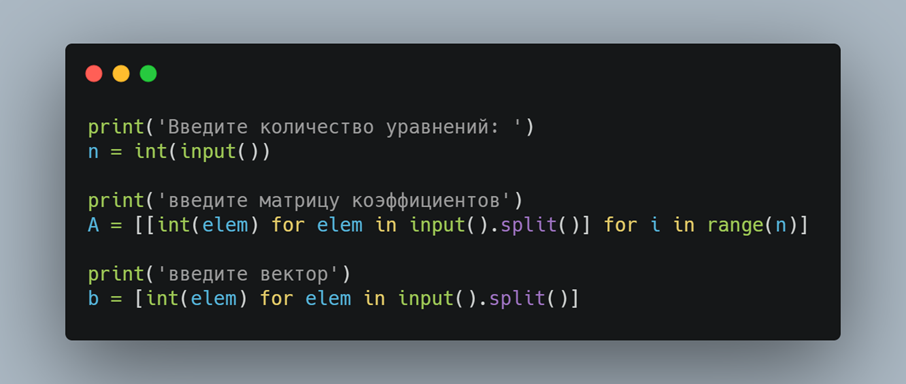


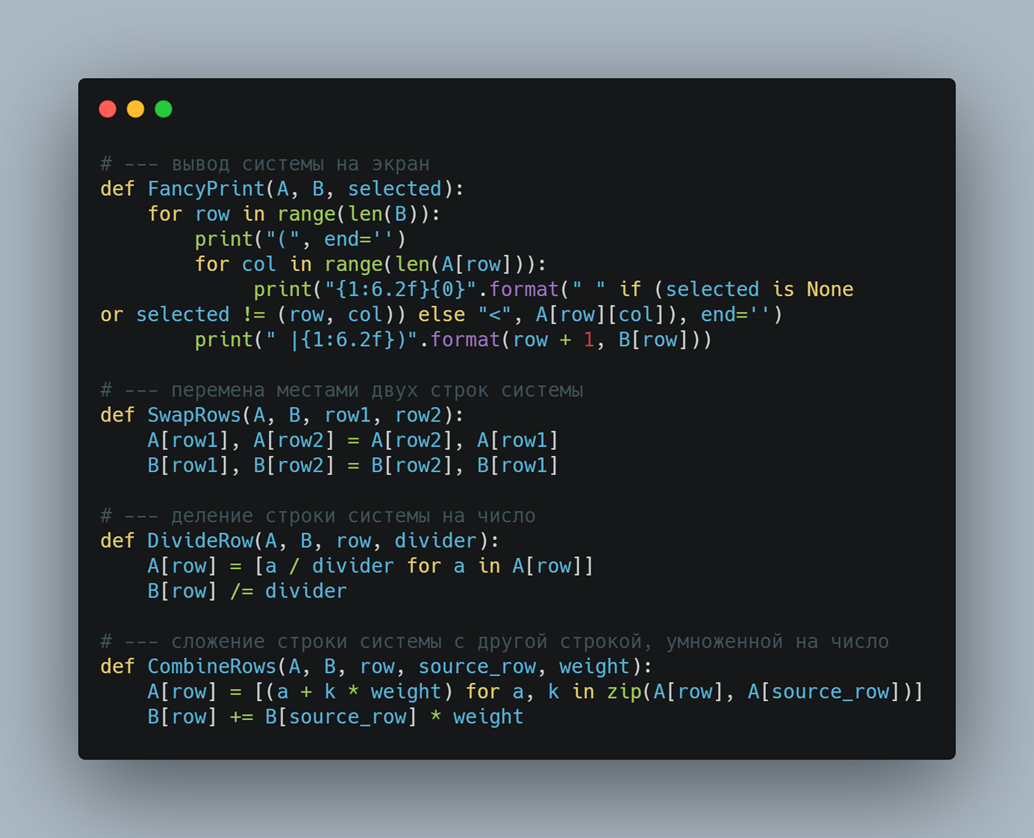
Ответ:

# Реализация метода Гаусса на Python

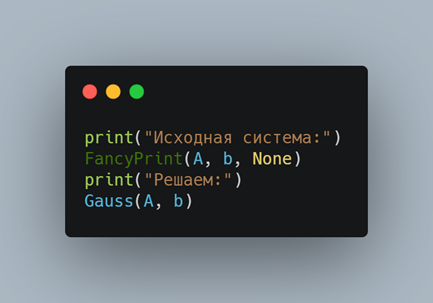
Для работы программы нам нужно ввести размерность матрицы, матрицу коэффициентов при переменных и вектор свободных коэффициентов.

Для начала нам надо было обеспечить ввод всех данных:



Функции действий над матрицами:

Вывод исходной матрицы и вызов функции:



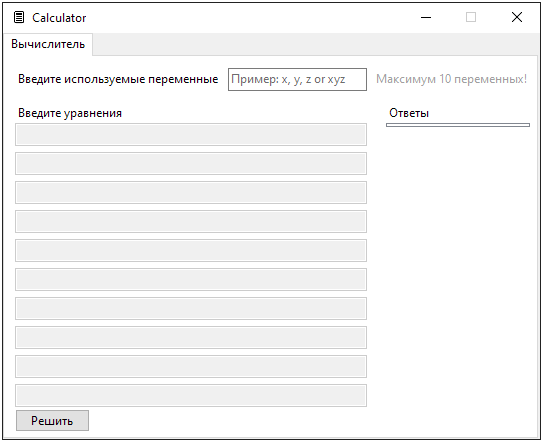
Здесь представлена реализация самой функции решения:

# Создание приложения

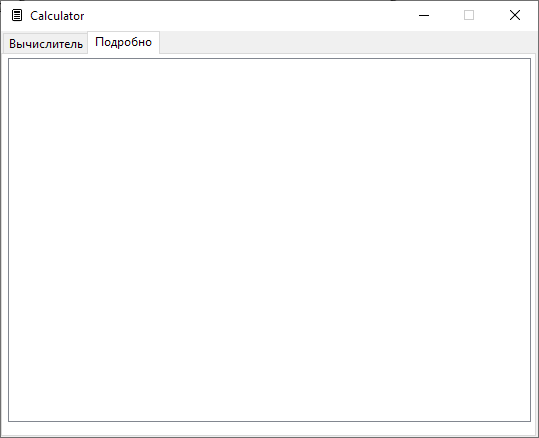
Теперь у нас есть все необходимые знания для написания своего приложения по решению систем уравнений. Для написания программы был выбран язык С# и редактор кода Microsoft Visual Studio 2022.

Для начала мы создали форму нашего приложения и добавили туда:

* Поле для ввода используемых переменных
* 10 полей для ввода уравнений
* Поле для вывода ответов
* Кнопку “Решить”
* Надпись, предупреждающую о превышении допустимого количества переменных (10 штук)

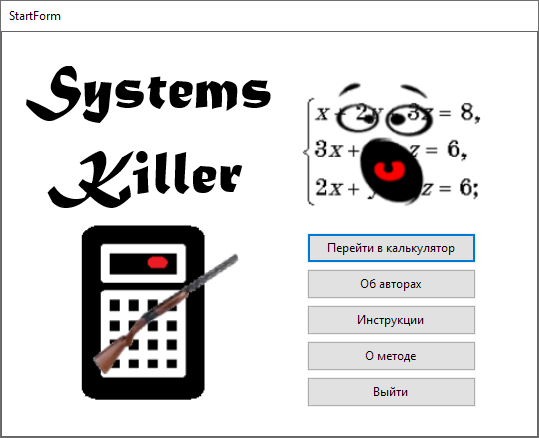


Затем мы добавили вторую вкладку для вывода подробного решения методом Гаусса.



Следующим шагом стал перевод приведенной ранее программы с языка Python на C#.

Далее мы решили сделать начальный экран, с которого можно перейти в калькулятор, а также посмотреть информацию об авторах, прочитать инструкцию по пользованию приложения и узнать информацию о методе Гаусса.



# Вывод

В результате проделанной работы мы узнали, что такое матрицы и как с ними работать. Научились решать системы уравнений с помощью метода Гаусса и написали собственное приложение, решающее системы уравнений.

# Список литературы

1. Метод Гаусса - mathprofi.ru
2. Метод Гаусса - Википедия
3. Матрица - Википедия
4. Матрицы: определение и основные понятия – onlinemschool
5. Помощь при написании кода – stackoverflow.com
6. Помощь при написании кода - cyberforum.ru
7. Официальная документация по С# - docs.microsoft.com