

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АДЫГЕЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Инженерно-физический факультет
Кафедра автоматизированных систем обработки информации и
управления

ОТЧЕТ ПО ПРАКТИКЕ

Программная реализация численного метода
Найти определитель матрицы

1 курс, группа ИИВТ АСОИУ

Выполнил:

_____ А. Ю. Федосеев
«___» _____ 2025 г.

Руководитель:

_____ С. В. Теплоухов
«___» _____ 2025 г.

Майкоп, 2025 г.

1. Введение

1.1. Текстовая формулировка задачи (Вариант 7)

Найти определитель матрицы.

1.2. Теория метода

Определителем квадратной матрицы

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

второго порядка называется число

$$|A| = a_{11}a_{22} - a_{12}a_{21}.$$

Определителем

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

квадратной матрицы порядка n , $n \geq 3$, называется число

$$|A| = \sum_{k=1}^n (-1)^{k+1} a_{1k} M_k,$$

где M_k — определитель матрицы порядка $n - 1$, полученной из матрицы A вычеркиванием первой строки и столбца с номером k .

2. Ход работы

2.1. Выбор средств для разработки

Для разработки программы, предназначенной для вычисления определителя матрицы, нами был выбран стек технологий, включающий язык программирования C, платформу .NET Core и фреймворк WPF (Windows Presentation Foundation) для построения пользовательского интерфейса. Этот выбор позволяет создать современное, производительное и кроссплатформенное настольное приложение с гибким и интуитивно понятным интерфейсом. Использование паттерна MVVM обеспечивает четкое разделение представления, бизнес-логики и модели данных, что значительно упрощает разработку, тестирование и сопровождение программы. Язык C предоставляет возможности строгой типизации и объектно-ориентированного подхода, а WPF — удобные средства визуализации, включая систему привязок данных, шаблоны элементов управления и поддержку стилей, что делает интерфейс приложения более удобным и наглядным для пользователя.

2.2. Код приложения

Для упрощения разработки было создано несколько вспомогательных классов. Листинг кода для этих классов приведён ниже:

Листинг 1. Класс MatrixCalculate для вычисления определителя

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace StudiesPractice.Core
8  {
9      public class MatrixCalculate
10     {
11         public decimal CalculateDeterminant(decimal[,] matrix)
12         {
13             int n = matrix.GetLength(0);
14
15             if (n == 1)
16                 return matrix[0, 0];
17
18             if (n == 2)
19                 return matrix[0, 0] * matrix[1, 1] - matrix[0, 1]
20                     * matrix[1, 0];
21
22             decimal det = 0;
23
24             for (int p = 0; p < n; p++)
25             {
26                 decimal[,] subMatrix = new decimal[n - 1, n - 1];
27
28                 for (int i = 1; i < n; i++)
29                 {
30                     int colIndex = 0;
31                     for (int j = 0; j < n; j++)
32                     {
33                         if (j == p) continue;
34                         subMatrix[i - 1, colIndex++] = matrix[i, j];
35                     }
36                 }
37
38                 det += matrix[0, p] * CalculateDeterminant(
39                     subMatrix) * (p % 2 == 0 ? 1 : -1);
40             }
41         }
42     }
43 }
```

```

39
40         return det;
41     }
42 }
43 }

```

Листинг 2. Модель одной ячейки матрицы

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Linq;
5  using System.Text;
6  using System.Threading.Tasks;
7
8  namespace StudiesPractice.Model
9  {
10     public class MatrixCell : INotifyPropertyChanged
11     {
12         private string _value;
13
14         public string Value
15         {
16             get => _value;
17             set
18             {
19                 _value = value;
20                 PropertyChanged?.Invoke(this, new
21                     PropertyChangedEventArgs(nameof(Value)));
22             }
23         }
24
25         public event PropertyChangedEventHandler PropertyChanged;
26     }
27 }

```

Листинг 3. Модель всей матрицы

```

1  using System;
2  using System.Collections.Generic;
3  using System.Collections.ObjectModel;
4  using System.Linq;
5  using System.Text;
6  using System.Threading.Tasks;
7
8  namespace StudiesPractice.Model
9  {

```

```

10 public class MatrixModel
11 {
12     public string Size { get; set; }
13     public ObservableCollection<MatrixCell> NumberList { get;
        set; }
14     public decimal Result { get; set; }
15 }
16 }

```

3. Скриншоты программы

Пример внешнего вида программы представлен на рис. 1, рис. 2 и рис. 3.

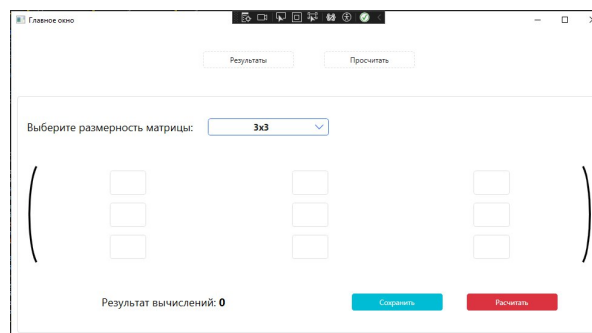


Рис. 1. Страница для заполнения и расчета элементов матрицы

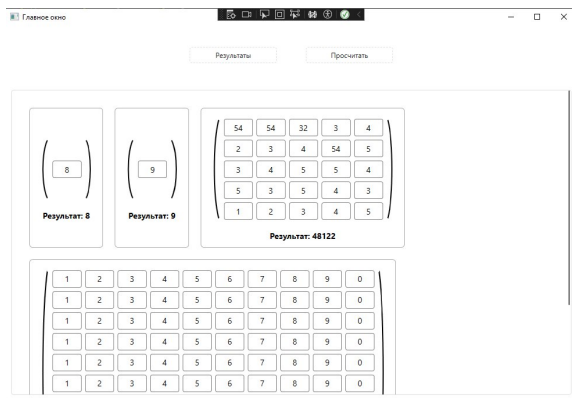


Рис. 2. Страница для просмотра результатов расчётов (1)

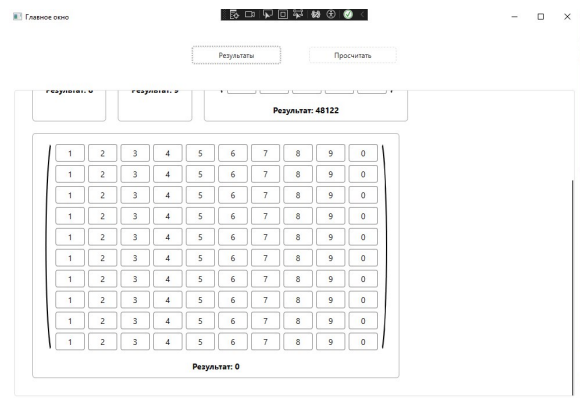


Рис. 3. Страница для просмотра результатов расчётов (2)

4. Источники

Список литературы

- [1] Кнут Д.Э. Всё про Т_EX. — Москва: Изд. Вильямс, 2003. — 550 с.
- [2] Львовский С.М. Набор и верстка в системе Л^AT_EX. — 3-е издание, исправленное и дополненное, 2003.
- [3] Воронцов К.В. Л^AT_EX в примерах. — 2005.
- [4] Документация по платформе .NET. — <https://learn.microsoft.com/dotnet/>, 2024.
- [5] Документация по языку программирования C#. — <https://learn.microsoft.com/dotnet/csharp/>, 2024.
- [6] Официальная документация WPF. — <https://learn.microsoft.com/dotnet/desktop/wpf/>, 2024.
- [7] MVVM Pattern. Архитектура приложений WPF. — <https://learn.microsoft.com/dotnet/architecture/mvvm/>, 2024.
- [8] Бронштейн И.Н., Семендяев К.А. Справочник по математике для инженеров и студентов вузов. — М.: Наука, 1986.