

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра КСУП

Первый графический проект. Матричные операции

Отчет по лабораторной работе № 1
по дисциплине «Компьютерная графика»

Студенты гр. 582-1

_____ К. Н. Полушвайко

_____ А. Д. Рязанов

_____ А. А. Юрьев

«__» _____ 2024 г.

Проверил

канд. техн. наук,

доцент каф. КСУП

_____ Н. Ю. Хабибулина

«__» _____ 2024 г.

2024 г.

1 Цель работы и постановка задачи

Цель работы – закрепление знаний создания Windows-приложения, в частности создание приложения типа WindowsForms Application в среде Visual Studio (далее по тексту - VS) и работа в командах, реализация индивидуальных задач.

Варианты задач:

1. Умножение матрицы на константу (Вариант 1) – А. Юрьев;
2. Скалярное произведение векторов (Вариант 3) – А. Юрьев;
3. Векторное произведение векторов (Вариант 4) – К. Полушвайко;
4. Умножение матрицы на вектор (Вариант 6) – А. Рязанов.

Роли:

- Руководитель – А. Рязанов;
- Технический писатель – К. Полушвайко;
- Инженер-программист – А. Юрьев.

2 Анализ задачи

Рассмотрим подробнее каждую из задач.

Умножение матрицы на константу является одной из основных операций в линейной алгебре. Эта операция позволяет умножить каждый элемент матрицы на заданное число. Результатом умножения матрицы на константу является новая матрица, в которой каждый элемент получен путем умножения соответствующего элемента исходной матрицы на эту константу. Пример представлен на рисунках 2.1 и 2.2.

$$4 \cdot \begin{pmatrix} 6 & 5 & -7 \\ -9 & 8 & 3 \\ -3 & -2 & 7 \\ 8 & -8 & 1 \end{pmatrix} = \begin{pmatrix} 6 \cdot 4 & 5 \cdot 4 & -7 \cdot 4 \\ -9 \cdot 4 & 8 \cdot 4 & 3 \cdot 4 \\ -3 \cdot 4 & -2 \cdot 4 & 7 \cdot 4 \\ 8 \cdot 4 & -8 \cdot 4 & 1 \cdot 4 \end{pmatrix}$$

Рисунок 2.1 – Пример умножения матрицы на константу.

$$\begin{pmatrix} 6 \cdot 4 & 5 \cdot 4 & -7 \cdot 4 \\ -9 \cdot 4 & 8 \cdot 4 & 3 \cdot 4 \\ -3 \cdot 4 & -2 \cdot 4 & 7 \cdot 4 \\ 8 \cdot 4 & -8 \cdot 4 & 1 \cdot 4 \end{pmatrix} = \begin{pmatrix} 24 & 20 & -28 \\ -36 & 32 & 12 \\ -12 & -8 & 28 \\ 32 & -32 & 4 \end{pmatrix}$$

Рисунок 2.2 – Пример умножения матрицы на константу.

Скалярное произведение векторов позволяет вычислить числовое значение, между двумя векторами. Результатом скалярного произведения является число, которое равно сумме произведений соответствующих элементов этих векторов. Таким образом, если у нас есть два вектора А и В размерности n, то скалярное произведение между ними обозначается как $A \cdot B$ и вычисляется следующим образом: $A \cdot B = A_1 \cdot B_1 + A_2 \cdot B_2 + \dots + A_n \cdot B_n$. Также его можно рассчитать, как произведение модулей векторов на косинус между ними:

$$\vec{a} \cdot \vec{b} = |\vec{a}| \cdot |\vec{b}| \cdot \cos(\angle \vec{a} \vec{b}) \quad (2.1)$$

Векторное произведение векторов - это еще одна важная операция в линейной алгебре. В отличие от скалярного произведения, результатом векторного произведения является вектор, а не число. Векторное произведение двух векторов А и В обозначается как $A \times B$ и определяется следующим образом:

$$a \times b = \begin{vmatrix} i & j & k \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} \quad (2.2)$$

$$a \times b = (a_y b_z - a_z b_y, a_z b_x - a_x b_z, a_x b_y - a_y b_x) \quad (2.3)$$

Умножение матрицы на вектор. Результатом умножения является новый вектор, полученный путем комбинации строк или столбцов матрицы с

компонентами вектора. Умножение матрицы на вектор выполняется путем умножения каждого элемента строки матрицы на соответствующий элемент вектора и последующего суммирования полученных произведений. Например, если у нас есть матрица A размером $m \times n$ и вектор B размером $n \times 1$, то результатом умножения будет новый вектор C размером $m \times 1$, где каждый элемент C_i вычисляется как сумма произведений элементов строки i матрицы A на соответствующие элементы вектора B .

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} \times \begin{vmatrix} x \\ y \\ z \end{vmatrix} = \begin{vmatrix} ax + by + cz \\ dx + ey + fz \\ gx + hy + iz \end{vmatrix} \quad (2.4)$$

2 Описание структуры программы

Для удобства реализации и компоновки проекта был создан репозиторий на github с рабочим процессом Git-flow. Каждая лабораторная реализована в отдельной форме. А каждая задача была реализована в отдельном пользовательском интерфейсе для удобства соединения всех задач в одно целое. Также такой подход уменьшал количество конфликтов, при слиянии веток в репозитории.

3 Описание основных функций

В программе используются расчетные методы, такие как:

- `VectorProductOfVectors::UpdateCVector()` – для расчета векторного произведения векторов;
- `SM::UpdateCVector()` – для расчета скалярного произведения векторов;
- `MultiplyingMatricesByVector::resultButton_Click()` – для расчета умножения матрицы на вектор;
- `MatrixByAConstant::ResultButton_Click()` – для расчета умножения матрицы на константу.

4 Руководство пользователя

При запуске программы вас встретит главное меню с кнопками (рисунок 4.1). Единственная активная кнопка – это «лабораторная работа 1», нажав на нее откроется новое окно с вкладками, где находятся задачи на первую лабораторную работу (рисунки 4.2-4.5).

Каждая вкладка имеет поля ввода с валидацией, если ввести некорректное значение, поле подсветится красным.

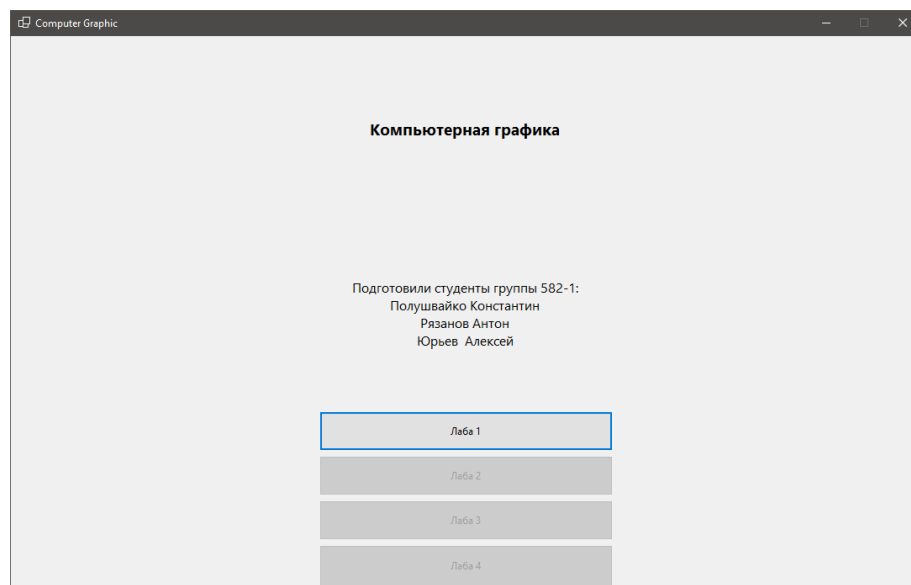


Рисунок 4.1 – Главное меню

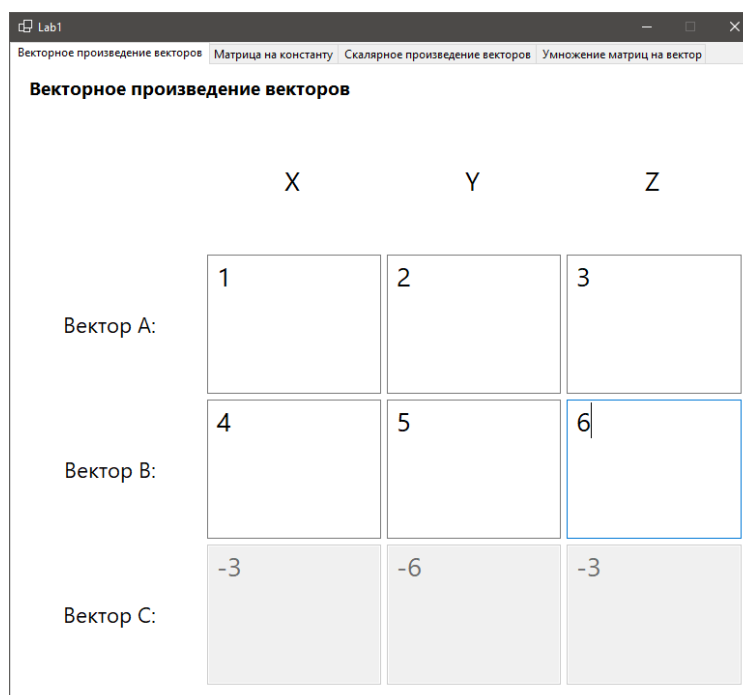


Рисунок 4.2 – Векторное произведение векторов

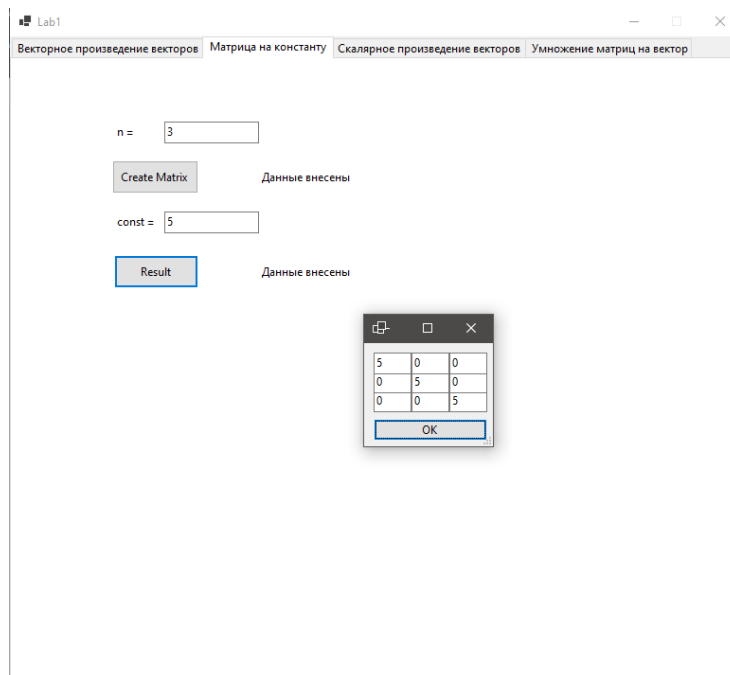


Рисунок 4.3 – Умножение матрицы на константу

Lab1

Векторное произведение векторов Матрица на константу **Скалярное произведение векторов** Умножение матриц на вектор

X
Y
Z

Вектор А

1	2	3
---	---	---

Вектор Б

3	2	1
---	---	---

Результат

10

Рисунок 4.4 – Скалярное произведение векторов

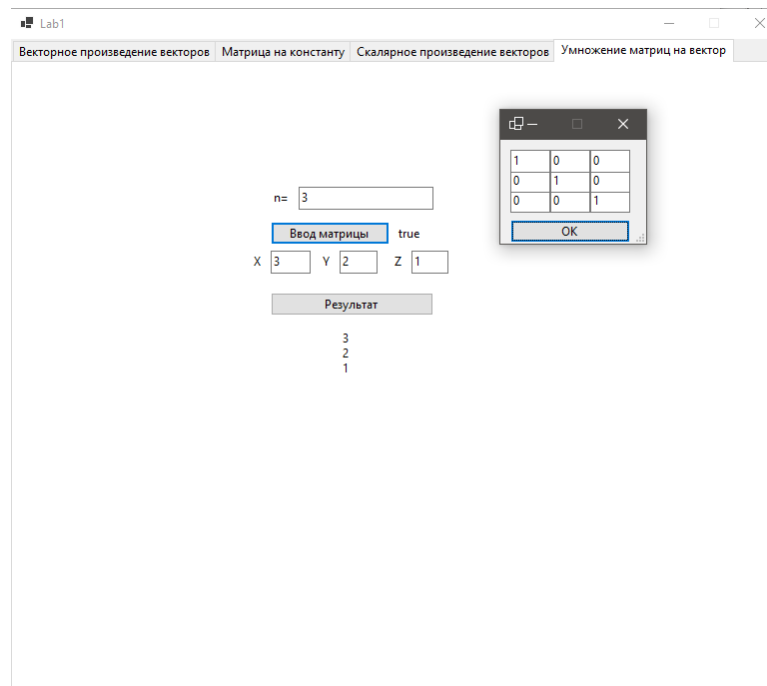


Рисунок 4.5 – Умножение матриц на вектор

5 Контрольные вопросы по лабораторной работе № 1

1. VS (Visual Studio) - это интегрированная среда разработки (IDE), разработанная компанией Microsoft, которая предоставляет разработчикам инструменты для создания различных программных продуктов, включая приложения, веб-сайты, мобильные приложения и т.д.

2. Для запуска VS нужно сначала установить ее на компьютер. После установки можно найти ярлык VS на рабочем столе или в меню "Пуск" и запустить его.

3. «Кнопка быстрого доступа» - это специальная панель инструментов в VS, которая позволяет быстро получить доступ к часто используемым функциям или командам. Кнопки быстрого доступа можно настроить по своему усмотрению.

4. «Обозреватель решений (Solution Explorer)» - это панель в VS, которая позволяет просматривать и управлять файлами и проектами в текущем решении. Она отображает структуру проекта и файлы, включая формы, модули, ресурсы и т.д.

5. «Окно свойств (Properties)» - это панель в VS, которая отображает свойства выбранного элемента в проекте. Например, если выбрана форма, то в окне свойств можно изменить ее свойства, такие как размер, цвет фона, шрифт и т.д.

6. «Панель инструментов (Toolbox)» - это панель в VS, которая содержит набор инструментов и элементов управления, которые можно перетаскивать на форму или другую область разработки. Например, из панели инструментов можно добавить кнопку, текстовое поле или изображение на форму.

7. «Дизайнер форм» - это режим работы в VS, который позволяет разработчику создавать и настраивать визуальные элементы пользовательского интерфейса (UI) для приложения. Дизайнер форм предоставляет удобный графический интерфейс для добавления и настройки элементов управления на форму. Отличие дизайнера форм от самой формы заключается в том, что дизайнер форм позволяет визуально редактировать

форму, а сама форма является объектом, который будет отображаться и использоваться в приложении.

8. «Свойство» - это характеристика или параметр объекта, который определяет его состояние или поведение. В контексте разработки в VS, свойства используются для настройки и управления визуальными элементами пользовательского интерфейса (UI), такими как размер, цвет, шрифт и т.д.

9. «Событие» - это действие или сигнал, который может быть сгенерирован объектом и на который можно отреагировать кодом. События используются для обработки пользовательских действий или изменений состояния объектов. Например, щелчок на кнопке может генерировать событие "Click", на которое можно назначить обработчик кода.

10. Для доступа к свойствам, расположенным на странице «События» (Events), нужно выбрать элемент управления на форме в дизайнера форм, затем в окне свойств перейти на вкладку "События" и выбрать нужное событие. После этого можно дважды щелкнуть на поле рядом с событием, чтобы автоматически создать обработчик кода для этого события.

11. Окно редактора кода в VS предоставляет возможность написания, редактирования и отладки кода программы. В окне редактора кода можно писать код на выбранном языке программирования (например, C#), просматривать и изменять код уже созданных файлов проекта.

12. «Проект» - это структурированная коллекция файлов и настроек, которая используется для разработки и сборки программного продукта. В VS можно создать новый проект с помощью команды "Создать проект" в меню "Файл" или используя сочетание клавиш Ctrl+Shift+N. Есть несколько способов создания нового проекта в зависимости от типа приложения, которое вы хотите разработать (например, Windows Forms, ASP.NET, WPF и т.д.).

13. Чтобы сохранить проект в VS, можно воспользоваться командой "Сохранить все" в меню "Файл" или использовать сочетание клавиш Ctrl+Shift+S. Также можно щелкнуть правой кнопкой мыши на проекте в обозревателе решений и выбрать команду "Сохранить".

14. Чтобы переключиться между формой и модулем, можно использовать вкладки в окне редактора кода. Если открыт файл формы, то можно переключиться на файл модуля, выбрав соответствующую вкладку, и наоборот.

15. Основные файлы, которые входят в проект VS, включают исходные коды (файлы с расширением .cs или .vb), файлы форм (с расширением .designer.cs или .designer.vb), файлы ресурсов (с расширением .resx), файлы конфигурации (например, app.config), файлы проекта (с расширением .csproj или .vbproj) и другие файлы, необходимые для сборки и запуска приложения.

16. В данной лабораторной работе мы познакомились с формами (Form), кнопками (Button), текстовыми полями (TextBox), метками (Label).

6 Заключение

В данной лабораторной работе был изучен способ создания приложения для Windows используя WinForms. Также был получен опыт с работой в команде и использовании системы контроля версий git.