|  |  |
| --- | --- |
| **Изображение выглядит как ночное небо  Автоматически созданное описание** | **автономная некоммерческая образовательная организация**  **высшего образования Центросоюза Российской Федерации**  **«Сибирский университет потребительской кооперации»** |

Кафедра информатики

**ОТЧЕТ ОБ ОЗНАКОМИТЕЛЬНОЙ** **ПРАКТИКЕ**

Место прохождения практики

АНОО ВО Центросоюза РФ СибУПК

Обучающегося 2 курса

*Перетягин Александр Андреевич*

группы ПИБ-11

Руководитель практики

доцент кафедры информатики,

канд. пед. наук

***Колдунова Ирина Дмитриевна***

Оценка после защиты \_\_\_\_\_\_\_\_\_\_\_

Дата защиты 30.12.2021

Новосибирск

2023

**Содержание**

[ВВЕДЕНИЕ 2](#_Toc167177885)

[1 Теоретические основы разработки оконных приложений 3](#_Toc167177886)

[1.1 Основы создания пользовательских интерфейсов 3](#_Toc167177887)

[1.2 Элементы управления 5](#_Toc167177888)

[2. Создание приложения 9](#_Toc167177889)

[2.1 Работа с конструктором Windows Forms 9](#_Toc167177890)

[2.2 Программирование элементов 12](#_Toc167177891)

[2.3 Тест приложения 13](#_Toc167177892)

[Заключение 16](#_Toc167177893)

[Используемая литература 17](#_Toc167177894)

[Приложение 19](#_Toc167177895)

ВВЕДЕНИЕ

Целью учебной практики является формирование у обучающихся профессиональных умений, приобретение первоначального практического опыта, направленное на освоение вида деятельности по специальности 09.03.03 Прикладная информатика и способствующее формированию общих и профессиональных компетенций по специальности базовой подготовки.

Основными задачами учебной практики являются:

- получение практического опыта в профессиональной области: проводить обследование организаций, выявлять информационные потребности пользователей; формировать требования к информационной системе; разрабатывать, внедрять и адаптировать прикладное программное обеспечение; документировать процессы создания информационных систем на стадиях жизненного цикла; программировать приложения и создавать программные прототипы решения прикладных задач;

- формирование практических умений: проектировать ИС в соответствии с профилем подготовки по видам обеспечения; выполнять технико­-экономическое обоснование проектных решений; собирать детальную информацию для формализации требований пользователей заказчика; составлять техническую документацию проектов автоматизации и информатизации прикладных процессов; применять системный подход и математические методы в формализации решения прикладных задач; готовить обзоры научной литературы и электронных информационно-образовательных ресурсов для профессиональной деятельности.

# Теоретические основы разработки оконных приложений

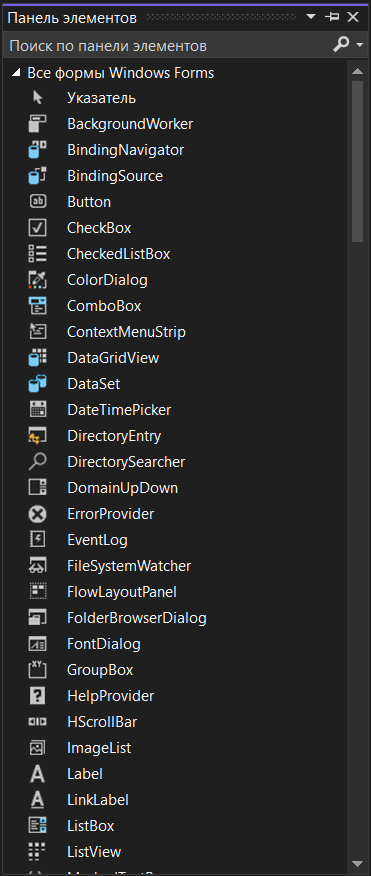
## Основы создания пользовательских интерфейсов

Windows Forms — это технология пользовательского интерфейса для .NET, представляющая собой набор управляемых библиотек, которые упрощают выполнение стандартных задач, таких как чтение из файловой системы и запись в нее. С помощью среды разработки, такой как Visual Studio, можно создавать интеллектуальные клиентские приложения Windows Forms, которые отображают информацию, запрашивают ввод пользователя и взаимодействуют с удаленными компьютерами по сети [4].

В Windows Forms форма — это визуальная поверхность, на которой выводится информация для пользователя. Обычно приложение строится путем добавления элементов управления в формы и создания кода для реагирования на действия пользователя, такие как щелчки мыши или нажатия клавиш [1]. Элемент управления — это отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных.

При выполнении пользователем какого-либо действия с формой или одним из ее элементов управления создается событие. Приложение реагирует на эти события, как задано в коде, и обрабатывает события при их возникновении [7].

В Windows Forms предусмотрено множество элементов управления, которые можно добавлять в формы. Например, элементы управления могут отображать текстовые поля, кнопки, раскрывающиеся списки, переключатели и даже веб-страницы. Если предусмотренные элементы управления не подходят для ваших целей, в Windows Forms можно создавать собственные пользовательские элементы управления с помощью класса UserControl [2].



*Рис. 1. Панель элементов*

Рассмотрим назначение некоторых элементов управления поподробнее:

* Button – кнопка, которая служит для выполнения действия с помощью мыши. При нажатии на неё вызывается обработчик события Click;
* ComboBox используется для выбора одного значения из нескольких возможных;
* ListBox — список, в который можно добавлять данные, удалять их программно, сортировать, связывать элементы управления с базой данных, заполнять элементы данными из разных источников данных программно, подогнать ширину элемента под текст;
* TextBox — Этот элемент управления является основным, предназначенным для ввода пользователем текстовых данных. Использовать TextBox можно в однострочном или многострочном режиме. Однако данный элемент управления имеет ограничение — до 64 килобайт текста.

В Windows Forms имеются многофункциональные элементы управления пользовательского интерфейса, позволяющие эмулировать функции таких сложных приложений, как Microsoft Office. С помощью элементов управления ToolStrip и MenuStrip вы можете создавать панели инструментов и меню, которые содержат текст и изображения, отображают подменю и размещают другие элементы управления, такие как текстовые поля и поля со списками [3].

Используя функцию перетаскивания конструктора Windows Forms в Visual Studio, можно легко создавать приложения Windows Forms. Просто выделите элемент управления с помощью курсора и поместите его на нужное место в форме. Для преодоления трудностей, связанных с выравниванием элементов управления, конструктор предоставляет такие средства, как линии сетки и линии привязки. С помощью элементов управления FlowLayoutPanel, TableLayoutPanel и SplitContainer можно гораздо быстрее создавать сложные макеты форм [9].

Наконец, если нужно создать свои собственные элементы пользовательского интерфейса, пространство имен System.Drawing содержит широкий набор классов, необходимых для отрисовки линий, кругов и других фигур непосредственно на форме.

## Элементы управления

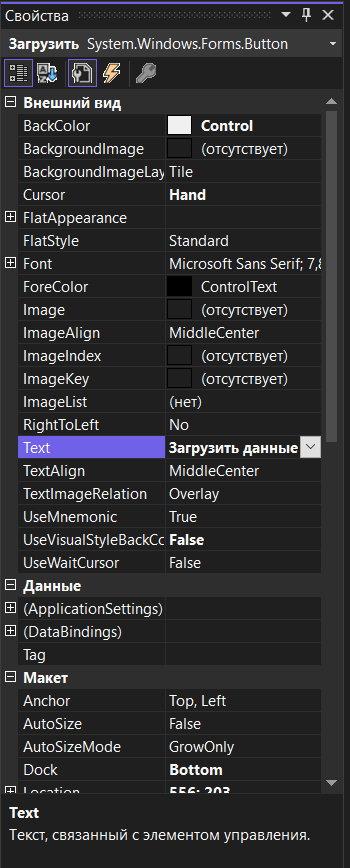
Для того чтобы запустить приложение в режиме отладки, нажмем на клавишу F5 или на зеленую стрелочку на панели Visual Studio.



*Рис. 2. Зелёная стрелочка запуска*

После запуска приложения студия компилирует его в файл с расширением exe. Найти данный файл можно, зайдя в папку проекта и далее в каталог \bin\Debug\net7.0-windows.

Чтобы перейти к свойствам элемента управления, необходимо щелкнуть его правой кнопкой мыши и выбрать пункт меню «Свойства».



*Рис. 3. Свойства*

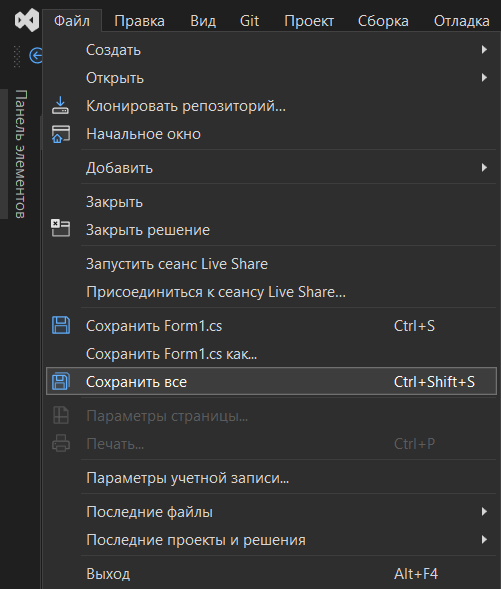
Рассмотрим назначение некоторых элементов свойств поподробнее:

* Name: устанавливает имя формы - точнее имя класса, который наследуется от класса Form;
* BackColor: указывает на фоновый цвет формы. Щелкнув на это свойство, мы сможем выбрать тот цвет, который нам подходит из списка предложенных цветов или цветовой палитры;
* Cursor: определяет тип курсора, который используется на форме;
* Enabled: если данное свойство имеет значение false, то она не сможет получать ввод от пользователя, то есть мы не сможем нажать на кнопки, ввести текст в текстовые поля и т. д.;
* Text: определяет заголовок формы.

С помощью значений свойств в окне “Свойства” мы можем изменить по своему усмотрению внешний вид формы, но все то же самое мы можем сделать динамически в коде. Дважды щелкните по списку в конструкторе форм. Делая это, Visual Studio автоматически откроет файл кода для формы. И он автоматически добавит метод события в код. Этот метод события будет запускаться всякий раз, когда мы будем кликать мышкой по нему в приложении [6].

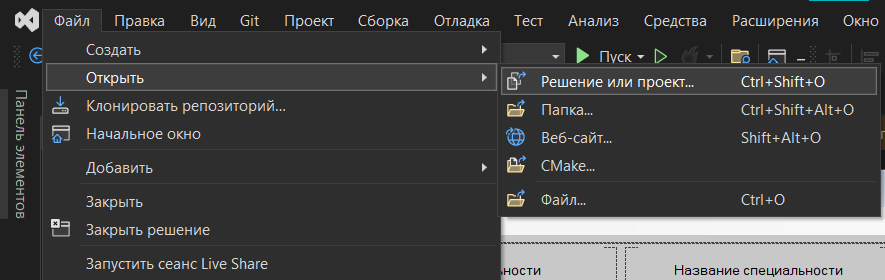
Чтобы сохранить все составляющие проекта приложения Windows выполните следующие действия [10]:

1. Выберите команду “Сохранить все” в меню “Файл”. В результате откроется окно “Сохранить проект”. Либо выполнить сохранение можно добиться, при сочетании клавиш “Ctrl” + “Shift” + “S”, также откроется диалоговое окно, если у проекта ещё не определено место сохранения.
2. В поле Расположение выберите из списка папку, в которой необходимо сохранить проект. При необходимости можно нажать кнопку Обзор для поиска требуемой папки. Если папка отсутствует, то следует включить команду Создать каталог для решения.
3. В дальнейшем проект будет сохраняться в выбранную папку.



*Рис. 4. Сохранение проекта*

Для того, чтобы открыть заранее сохранённый проект, необходимо нажать на кнопку “Файл”, расположенную вверху Windows Forms, выбрать пункт “Открыть” далее нажать “Решение или проект” (сочетание клавиш “Ctrl” + “Shift” + “O”). Откроется диалоговое окно, в котором выбрать файл для открытия и нажать кнопку “Открыть” [8].



*Рис. 5. Открытие проекта*

1. Создание приложения

## 2.1 Работа с конструктором Windows Forms

Для создания необходимых элементов управления необходимо обратиться к панели управления (*рис. 1*). В данной программе необходимо реализовать заполнение поля “Факультет” из внешнего текстового файла, вручную выбрать из выпадающего списка форму обучения и наличие оригинала документов, написать в полях ввода номер специальности, название специальности, ФИО, баллы ЕГЭ. Вывести строку в listView с данными об абитуриенте и далее сохранить в текстовом файле.

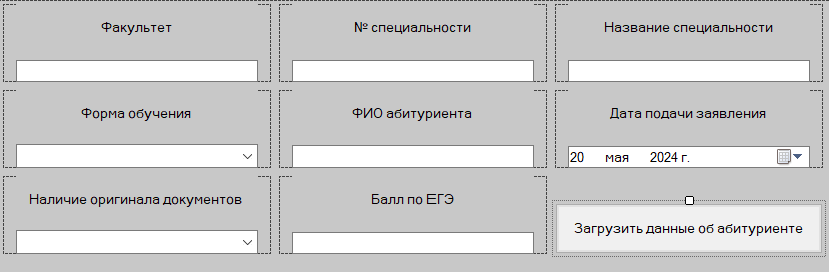
Для решения поставленной программы, необходимо:

1. Добавить элементы управления в конструктор Windows Forms;
2. Программирование элементов;
3. Тест программы.

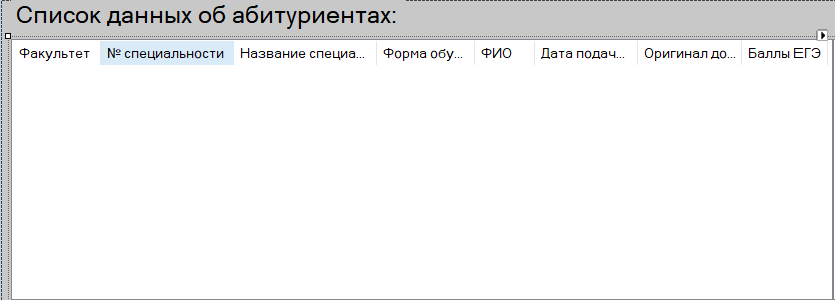
Элементы управления, которые использовались:

* textBox;
* button;
* comboBox;
* listBox;
* listView;
* menuStrip;
* datetimePicker;
* panel;
* openFileDialog;
* toolTip;
* saveFileDialog;
* colorDialog.

Для удобства, было решено разместить поля ввода (Факультет, № специальности, Название специальности, Форма обучения, ФИО абитуриента, Дата подачи заявления, Наличие оригинала документов, баллы по ЕГЭ) через textBox сверху *(рис. 7)*, вывод чeрез listView1 снизу. *(рис. 7)*

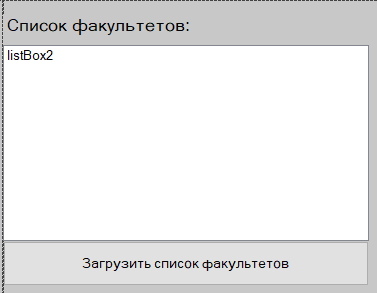


*Рис. 6. Поля ввода*

**

*Рис. 7. Поле вывода*

Чтобы реализовать выбора формы обучения из текстового файла, было решено добавить в правую верхнюю часть приложения кнопку, при нажатии на которую будут передаваться данные из внешнего файла и записываться в listBox2 *(рис.9),* которые будут присваиваться полю “Факультет”.



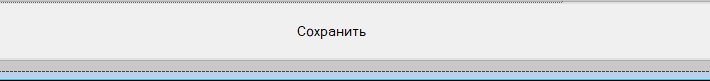
*Рис. 8. Загрузка из внешнего файла*

При помощи дополнительной кнопки “Загрузить” *(рис. 10)*, было сделано сохранение введённых параметров в виде строки в listView1.



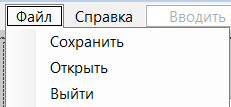
*Рис. 9. Загрузка из полей ввода в поле вывода*

Для удобства в дальнейшем добавлении и удалении информации об абитуриенте, имеется кнопка сохранения во внешний файл на компьютере *(рис. 11),* которая расположена ниже поля вывода listView1 *(рис. 8)*.

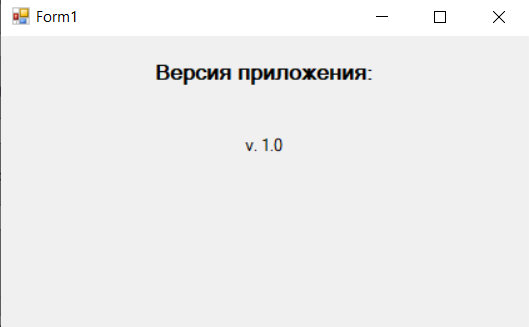


*Рис. 10. Сохранение во внешний файл*

В самом верху конструктора приложения в Windows Forms, было добавлено menuStrip *(рис. 12)* с кнопками: файл и справка. При нажатии на “Файл” откроется список с полями: сохранить, открыть и выйти *(рис. 12)*. При выборе кнопки выйти, приложение закроется. При нажатии на кнопку “Справка” откроется новое диалоговое окно с информацией версии данного приложения, аналогично можно начать на кнопку “F1” *(рис. 13).*



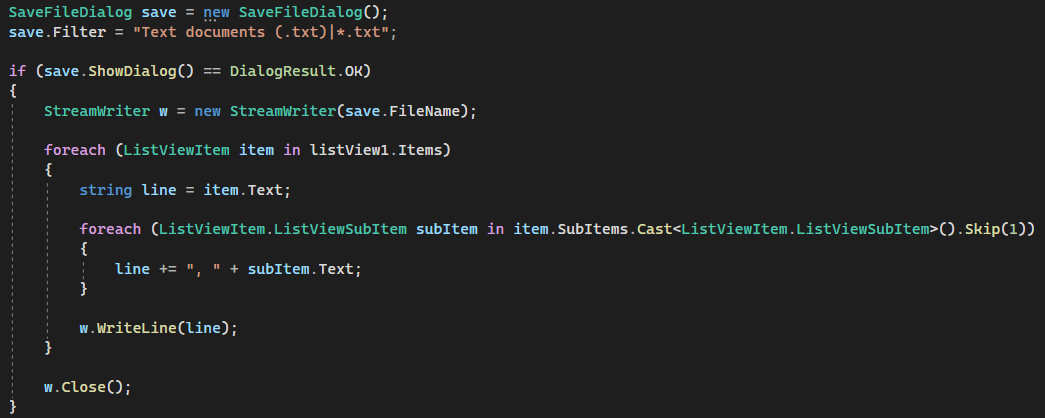
*Рис. 11. menuStrip*



*Рис. 12. Окно справки*

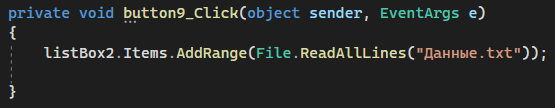
## 2.2 Программирование элементов

Для кнопок сохранения файла в текстовый файл формата .txt (button5 и сохранитьToolStripMenuItem), был написан код, который вызывает диалоговое окно, в котором необходимо создать или выбрать файл *(рис. 14).* После сохранения файла, если всё прошло успешно, окно само закроется, иначе выскочит ошибка сохранения.



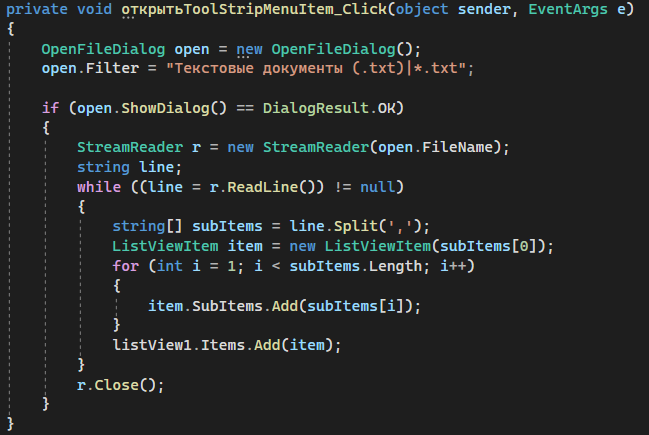
*Рис. 13. Сохранение*

Для кнопки (button9) открытия из внешнего файла “Данные”, которая выводит список факультетов, был написан код *(рис. 15)*.



*Рис. 14. Открыть список факультетов*

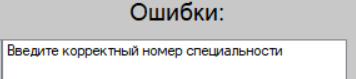
Кнопка “открытьToolStripMenuItem”, предназначена для выбора и открытия текстового файла с данными об абитуриенте, имеет следующий код (*рис. 15*).



*Рис. 15. Открыть данные об абитуриенте*

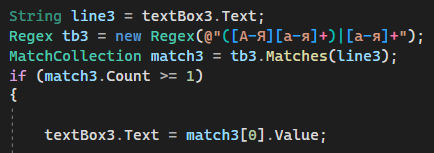
## 2.3 Тест приложения

При пропуске или неправильно введённом формате данных одного из полей ввода, в listBox1, будет выводиться соответствующее сообщение с названием этого поля *(рис. 16)*, при этом, поля сохранят ранее введённую информацию. Если необходимо удалить это сообщение, просто дважды кликните левой кнопкой мыши по этому сообщению.



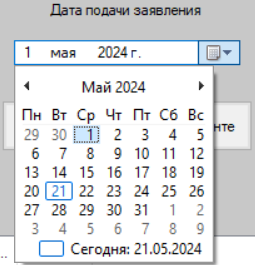
*Рис. 16. Сообщение о пропущенном поле ввода*

Для полей ввода была реализована маска для их заполнения, при помощи команды regex *(рис. 17)*, которая предотвращает ввод рандомной информации.



*Рис. 17. Команды regex для поля “Название специальности”*

Для удобства ввода даты подачи заявления, был добавлен dateTimePicker2 *(рис. 18).*



*Рис. 18. dateTimePicker2*

Тестированию подверглись comboBox1 и comboBox2 *(рис. 19)*, благодаря которым из выпадающего списка можно выбрать наполнение поля ввода.

Изображение выглядит как стол

Автоматически созданное описание

*Рис. 19. comboBox1*

В итоге проведения нескольких тестов программы, не было выявлено ошибок или дефектов, что позволяет сделать вывод о полной готовности и работоспособности приложения.

# Заключение

В итоге проведённой работы, было сделано полноценное приложение, которое полностью удовлетворяет всем требованиям данным для выполнения. Для разработки был выбран Visual Studio 2022 на языке программирования C#, в котором был выбран шаблон Windows Forms (.NET Framework). Работа с WF позволяет создавать приложения с высокой производительностью и надежностью, а также обеспечивает удобство использования и интуитивно понятный интерфейс для пользователей.

В ходе выполнения работы, было получено как много новой информации, так и закрепление ранее изученной как работы с Visual Studio, так и с языком C#.

В заключении отчета по данному заданию можно отметить, что была разработана программа для ввода, отображения и дальнейшее сохранение данных об абитуриентах. В системе реализована загрузка списка факультетов из файла. Для отображения данных использован компонент listView. В конце были проведены тесты для удостоверения в том, что всё корректно работает.

# Используемая литература

1. C # Приложение Windows Forms [Электронный ресурс]: CoderLessons – [сайт]. — URL: <https://coderlessons.com/tutorials/kompiuternoe-programmirovanie/uchebnik-c-kompiuternoe-programmirovanie/20-c-prilozhenie-windows-forms> (дата обращения: 24.12.2023).

2. Windows Forms / [Электронный ресурс] // WikipediA : [сайт]. — URL: <https://en.wikipedia.org/wiki/Windows_Forms> (дата обращения: 24.12.2023).

3. Введение в Windows Forms Первое приложение с .NET CLI / [Электронный ресурс] // METANIT.COM : [сайт]. — URL: <https://metanit.com/sharp/windowsforms/1.3.php> (дата обращения: 18.12.2023).

4. Введение в приложения C # Windows Forms / [Электронный ресурс] // GG: [сайт]. — URL: <https://www.geeksforgeeks.org/introduction-to-c-sharp-windows-forms-applications/> (дата обращения: 24.12.2023).

5. Общие сведения о Windows Forms / [Электронный ресурс] // D: [сайт]. — URL: <https://msdn-whiteknight.github.io/dotnet-docs-ru/html/framework/winforms/windows-forms-overview.html> (дата обращения: 24.12.2023).

6. Разработка оконного приложения на C# в Microsoft Visul Studio / [Электронный ресурс] // dzen: [сайт]. — URL: <https://dzen.ru/a/YE7vTfypmqTEMN8> (дата обращения: 19.12.2023).

7. Руководство по классическим приложениям (Windows Forms .NET) [Электронный ресурс]: Microsoft. Документация. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/?source=recommendations&view=netdesktp-6.0> (дата обращения: 20.12.2023).

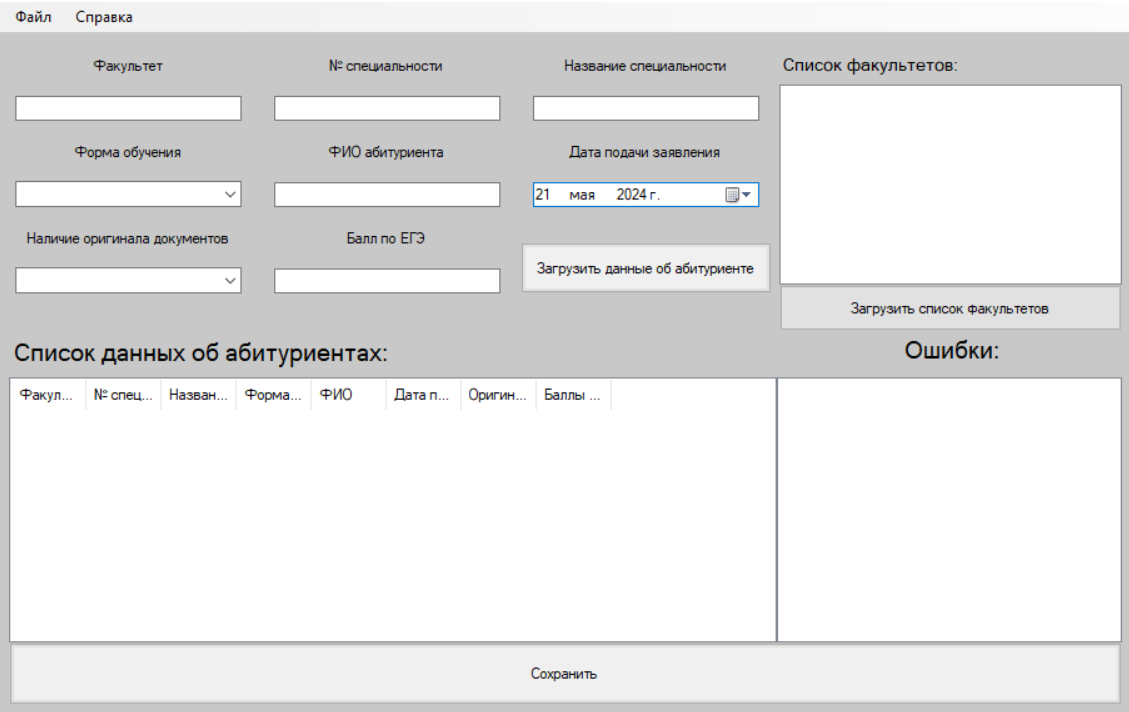
8. Руководство по программированию в Windows Forms / [Электронный ресурс] // METANIT.COM: [сайт]. — URL: <https://metanit.com/sharp/windowsforms/> (дата обращения: 24.12.2023).

9. Создание приложения Windows Forms на C# в Visual Studio / [Электронный ресурс] // Microsoft : [сайт]. — URL: <https://learn.microsoft.com/ru-ru/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022> (дата обращения: 24.12.2023).

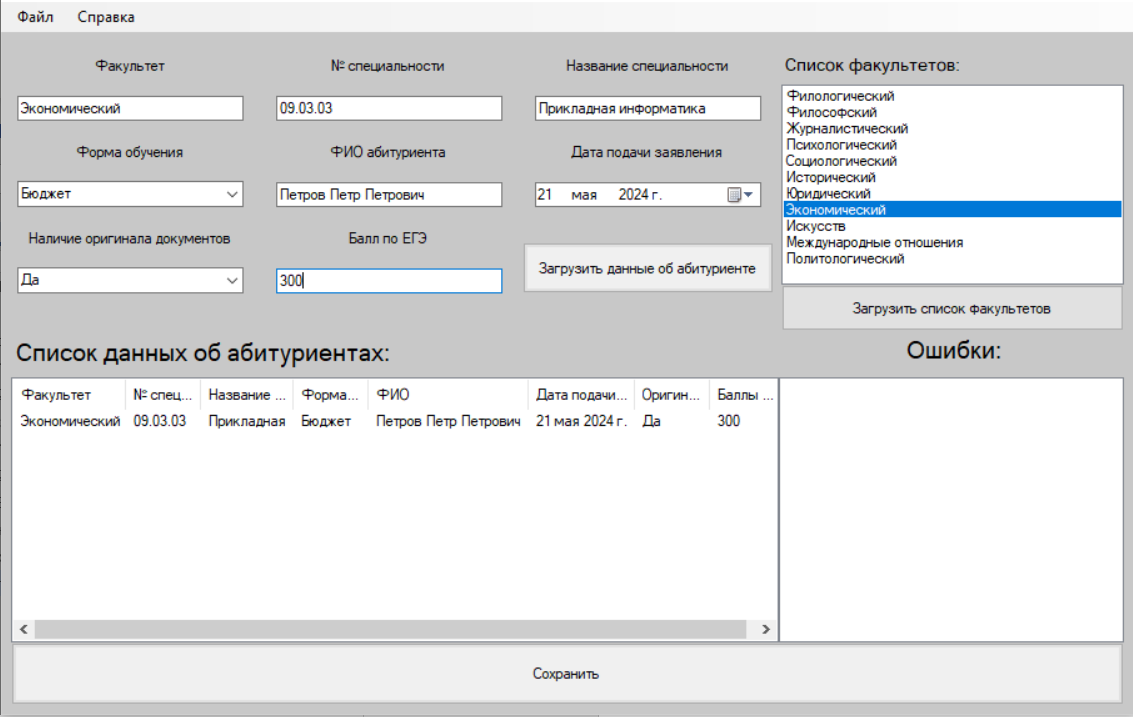
10. Учебное пособие по применению C # Windows Forms / [Электронный

ресурс] // GURU99: [сайт]. — URL: [https://www.guru99.com/c-sharp-windows- forms-application.html](https://www.guru99.com/c-sharp-windows-%20forms-application.html) (дата обращения: 24.12.2023).

# Приложение 1

****

*Рис. 20. Чистое окно приложения*

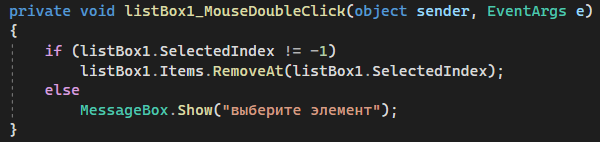


*Рис. 21. Заполненный вид полей*

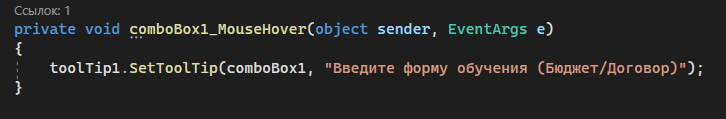
Изображение выглядит как текст, монитор, снимок экрана, внутренний

Автоматически созданное описание

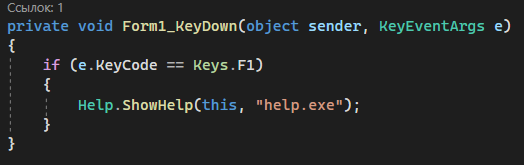
*Рис. 22. Окно сохранения*



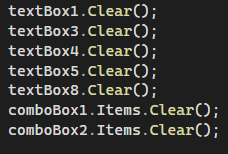
*Рис. 23. Код для выбора строки из listBox1 и её удаление*

**

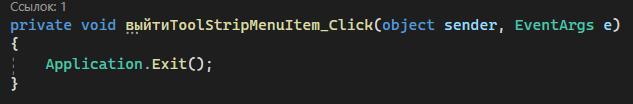
*Рис. 24. Код для показа всплывающего окна при наведении мыши*

**

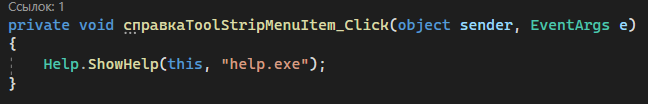
*Рис. 25. Код для открытия окна справки при помощи клавиши F1*



*Рис. 26. Код для очищения полей ввода*

**

*Рис. 27. Код кнопки для выхода из приложения*

**

*Рис. 28. Код кнопки для открытия окна справки*