|  |  |
| --- | --- |
| **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(ейся) 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

2022

**Содержание**

[ВВЕДЕНИЕ 2](#_Toc122909237)

[1. Теоретические основы разработки оконных приложений 3](#_Toc122909238)

[1.1 Основы создания пользовательских интерфейсов 3](#_Toc122909239)

[1.2 Элементы управления 5](#_Toc122909240)

[2. Создание приложения 9](#_Toc122909241)

[2.1 Работа с конструктором Windows Forms 9](#_Toc122909242)

[2.2 Программирование элементов 11](#_Toc122909243)

[2.3 Тест приложения 12](#_Toc122909244)

[Заключение 15](#_Toc122909245)

[Используемая литература 16](#_Toc122909246)

[Приложение 17](#_Toc122909247)

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

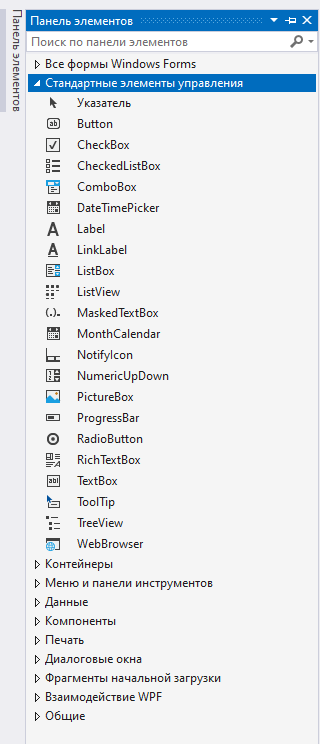
## Основы создания пользовательских интерфейсов

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.

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. 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.

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.



*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.

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.

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

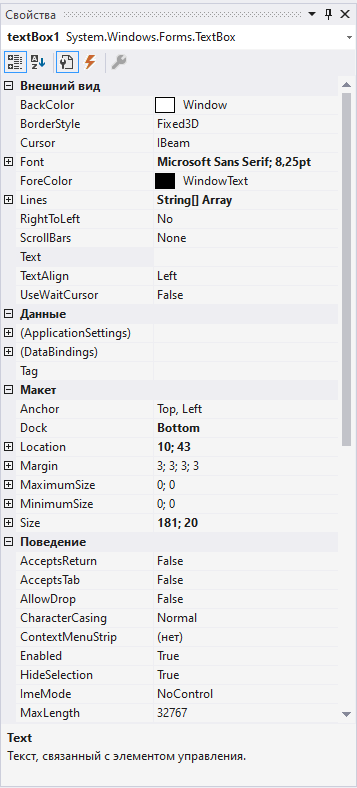
Для того чтобы запустить приложение в режиме отладки, нажмем на клавишу F5 или на зеленую стрелочку на панели Visual Studio.



*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».

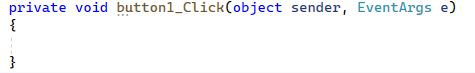


*Рис. 3. Свойства*

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.



*Рис. 4. Метод события кнопки “button\_1”*

Чтобы сохранить все составляющие проекта приложения Windows выполните следующие действия:

1. Выберите команду Сохранить все в меню Файл. В результате откроется окно “Сохранить проект”.
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.

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 5. 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”. Откроется диалоговое окно, в котором выбрать файл для открытия и нажать кнопку “Открыть”.

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 6. 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 listBox с данными об абитуриенте и далее сохранить в текстовом файле.

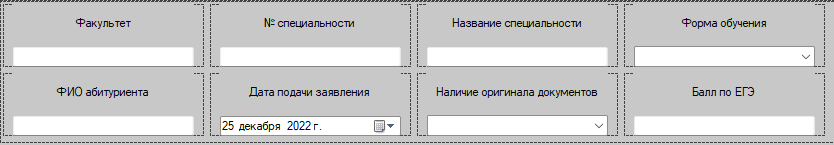
Для решения поставленной программы, необходимо:

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

Элементы управления, которые использовались:

* textBox;
* button;
* comboBox;
* listBox;
* 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 listBox1 снизу. *(рис. 7)*



*Рис. 7. Поля ввода*

**

*Рис. 8. Поля вывода*

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

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 9. Загрузка из внешнего файла*

При помощи дополнительной кнопки “Загрузить” *(рис. 10)*, было сделано сохранение введённых параметров в виде строки в listBox1.



*Рис. 10. 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 listBox1 *(рис. 8)*.



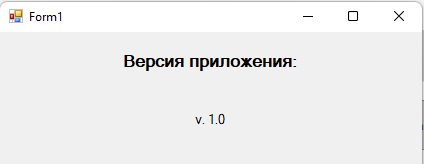
*Рис. 11. Сохранение во внешний файл*

В самом верху конструктора приложения в Windows Forms, было добавлено menuStrip *(рис. 12)* с кнопками: файл, правка 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 *(рис. 13).*

Изображение выглядит как стол

Автоматически созданное описание

*Рис. 12. menuStrip*



*Рис. 13. Окно справки*

## 2.2 Программирование элементов

Для кнопок сохранения файла (button5 и сохранитьToolStripMenuItem), был написан код *(рис. 14).*

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 14. Сохранение*

Для кнопки (button9) открытия из внешнего файла “Данные”, был написан код *(рис. 15)*, аналогично и для кнопки “открытьToolStripMenuItem” текстового файла “Данные абитуриента”.

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 15. Открыть*

## 2.3 Тест приложения

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



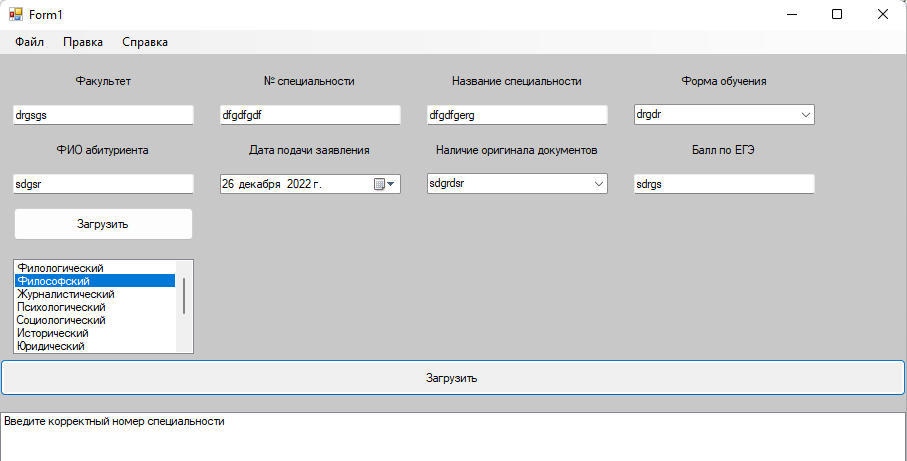
*Рис. 16. Сообщение о пропущенном поле ввода*

Для полей ввода была реализована маска для их заполнения, при помощи команды regex *(рис. 17)*, которая предотвращает ввод рандомной информации *(рис 18)*.

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 17. Команды regex*



*Рис. 18. Ввод рандомной информации*

Для удобства ввода даты подачи заявления, был добавлен dateTimePicker2 *(рис. 19).*

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 19. dateTimePicker2*

Тестированию подверглись comboBox1 и comboBox2 *(рис. 20)*, благодаря которым из выпадающего списка можно выбрать наполнение поля ввода.

Изображение выглядит как стол

Автоматически созданное описание

*Рис. 20. comboBox1*

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

# Заключение

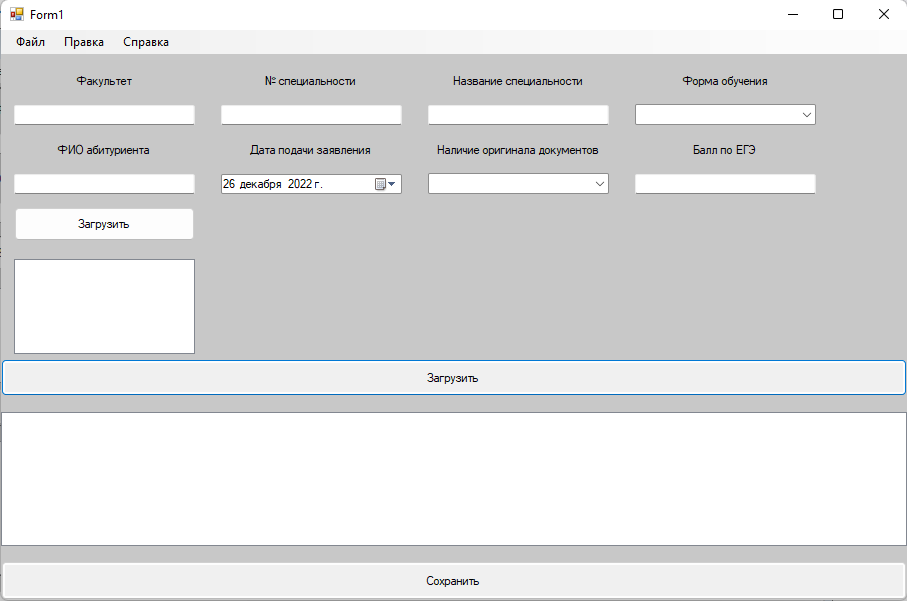
В итоге проведённой работы, было сделано полноценное приложение, которое полностью удовлетворяет всем требованиям данным для выполнения. Для разработки был выбран Visual Studio 2022 на языке программирования C#, в котором был выбран шаблон Windows Forms (.NET Framework).

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#.

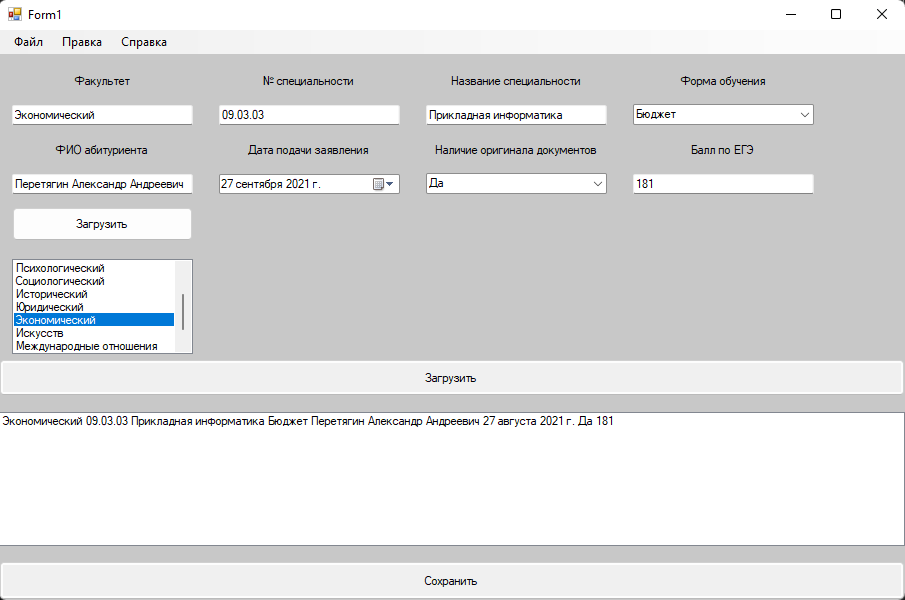
# Используемая литература

1. 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=netdesktop-6.0> (дата обращения: 23.12.2022).
2. ListBox — список [Электронный ресурс]: ListBox. – Режим доступа: <http://fb3809fm.bget.ru/index.php> (дата обращения: 24.12.2022).
3. C # Приложение Windows Forms [Электронный ресурс]: CoderLessons – Режим доступа: <https://coderlessons.com/tutorials/kompiuternoe-programmirovanie/uchebnik-c-kompiuternoe-programmirovanie/20-c-prilozhenie-windows-forms> (дата обращения: 25.12.2022).

# Приложение

****

*Рис. 21. Чистое окно приложения*



*Рис. 22. Заполненный вид полей*

Изображение выглядит как текст, монитор, снимок экрана, внутренний

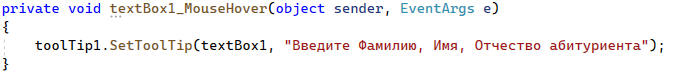
Автоматически созданное описание

*Рис. 23. Окно сохранения*

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 24. Код для выбора строки из listBox1*

**

*Рис. 25. Код для показа всплывающего окна при наведении мыши*

*Изображение выглядит как текст

Автоматически созданное описание*

*Рис. 26. Код для открытия окна справки при помощи клавиши F1*

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 27. Код для очищения полей ввода*

*Изображение выглядит как текст

Автоматически созданное описание*

*Рис. 28. Код кнопки для выхода из приложения*