ГУАП

КАФЕДРА № 42

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| канд. техн. наук, доцент |  |  |  | Н. В. Богословская |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 4 |
| ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСОВ ПОЛЬЗОВАТЕЛЯ. СИНХРОНИЗАЦИЯ ЭЛЕМЕНТОВ УПРАВЛЕНИЯ |
| по курсу: |
| ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ гр. № | 4326 |  |  |  | Г. С. Томчук |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2024

СОДЕРЖАНИЕ

[1 Цель работы 3](#_Toc180732731)

[2 Задача 3](#_Toc180732732)

[3 Ключевые позиции 4](#_Toc180732733)

[3.1 Исследование синхронной работы элементов управления 4](#_Toc180732734)

[3.1 Отрисовка изображения 6](#_Toc180732735)

[4 Тестирование программы 7](#_Toc180732736)

[ВЫВОДЫ 10](#_Toc180732737)

1. Цель работы

Выполнение работы имело следующие цели:

* Научиться проектировать пользовательский интерфейс в Windows Forms с использованием синхронизированных элементов управления.
* Освоить методы привязки данных в C# для связи элементов управления с объектами класса.
* Практически исследовать работу с коллекциями данных и их привязкой к элементам управления, таким как ListBox.
* Научиться использовать свойства класса для отображения и редактирования информации на форме.
* Изучить и применить возможности DataBindings для синхронизации данных между объектами и элементами интерфейса.
* Овладеть навыками работы с диалоговыми окнами для выбора файлов (например, загрузки пользовательской иконки).

1. Задача

11. В приложении пользователь может создать объект класса Текстовый файл, используя классы Файл, Директория. Методы для работы с файлами должны полностью обеспечить пользователя возможностями создания, удаления, изменения, переименования файлов. Подзадачи лабораторной работы № 4 включают в себя:

* Разработать интерфейс в Windows Forms для исследования синхронизации данных, связав ListBox с коллекцией объектов класса FileClass.
* Настроить свойства ListBox (DataSource и DisplayMember) для автоматического отображения данных из коллекции файлов в выбранной директории.
* Реализовать привязку данных (свойств) объекта класса к элементам управления (TextBox, DateTimePicker, CheckBox) с использованием DataBindings.
* Добавить в класс FileClass поле icon\_photo для хранения пути к иконке файла и создать перегруженные методы ShowPhoto() для отображения иконки в PictureBox.
* Создать механизм выбора пользовательской иконки файла с использованием диалогового окна OpenFileDialog.
* Настроить отображение различных типов данных (строковые, числовые, логические, дата-время) через соответствующие элементы управления на форме.
* Обеспечить синхронизацию элементов управления так, чтобы при выборе файла в списке изменялись данные в связанных полях интерфейса.

1. Ключевые позиции
   1. Исследование синхронной работы элементов управления

На рис. 1 изображен код инициализации коллекции fileList типа BindingList<FileClass>, в которой хранятся все объекты файлов, находящихся в рабочей папке программы. Также в конструкторе форм вызываются функции наполнения все того же fileList и установки связей с данными.

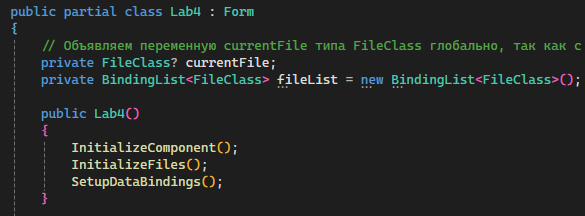
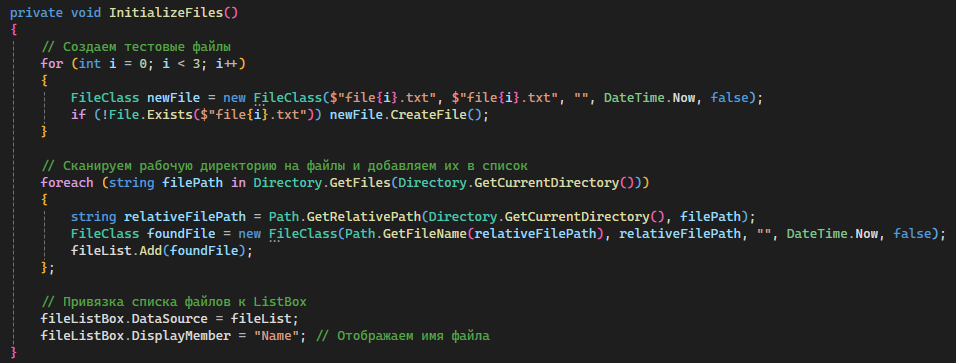


Рисунок 1

На рис. 2 изображена функция InitializeFiles, в которой происходит наполнение списка fileList, а также привязка данного списка к элементу интерфейса ListBox.



Рисунок

На рис. 3 изображена функция установки связей между данными из списка fileList и элементами управления формы.

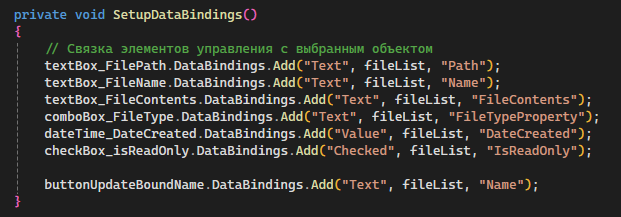


Рисунок 3

На рис. 4 показан обработчик события смены выбранного файла в списке fileList.

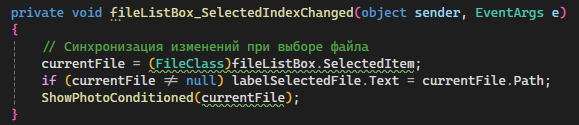


Рисунок 4

На рис. 5 изображен обработчик клика по кнопке несвязанного обновления имени файла. Это нужно, чтобы продемонстрировать разницу между несинхронизированными и синхронизированными элементами управления.

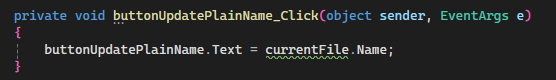


Рисунок 5

* 1. Отрисовка изображения

На рис. 6 показано объявление полей iconPhotoPath и iconGraphics в классе FileClass.

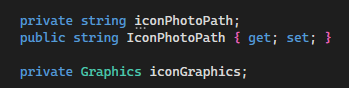


Рисунок 6

На рис. 7 показаны методы класса FileClass для отображения иконок файлов. Два перегруженных метода отрисовки и два метода удаления изображения – для PictureBox и для Form.



Рисунок 7

На рис. 8 изображена функция в классе формы, необходимая для корректного отображения иконки в зависимости от способа отображения.

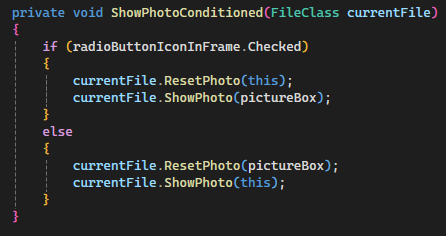


Рисунок 8

На рис. 9 изображено использование OpenFileDialog для получения файла картинки.

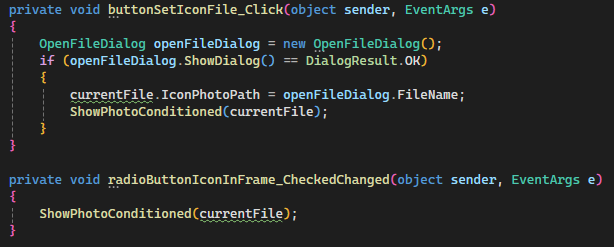


Рисунок 9

1. Тестирование программы

На рис. 10-13 представлены результаты тестирования программы.

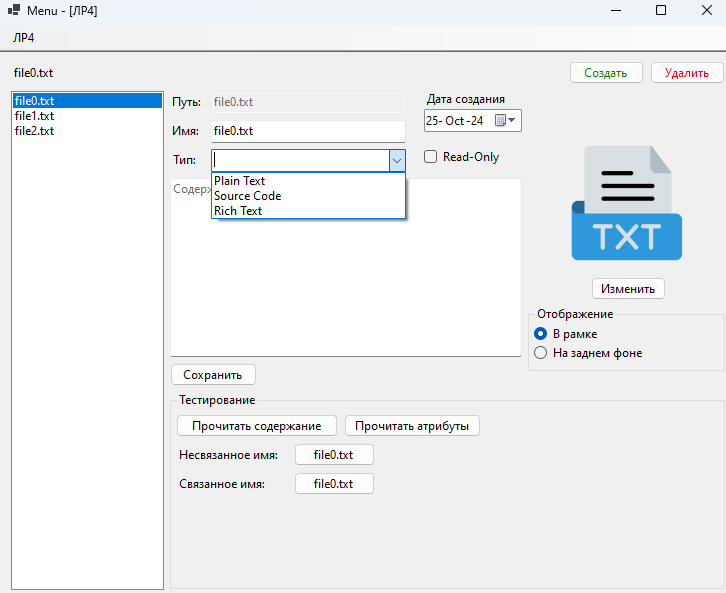


Рисунок 10

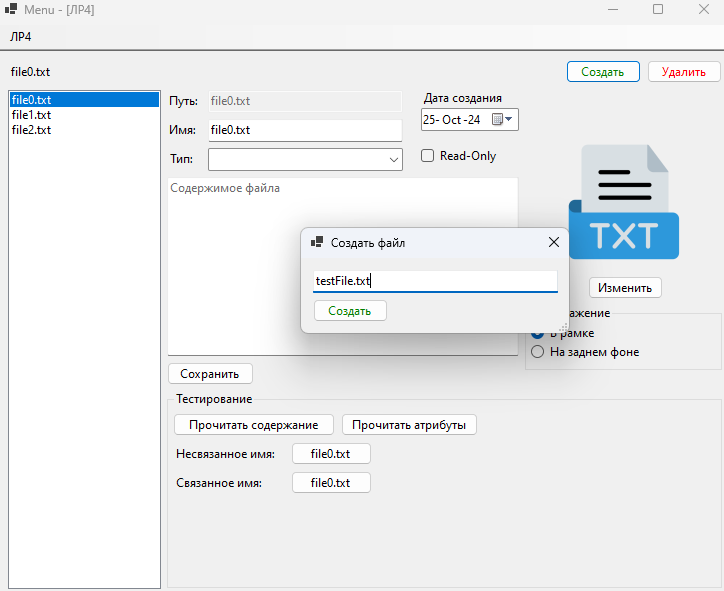


Рисунок 11

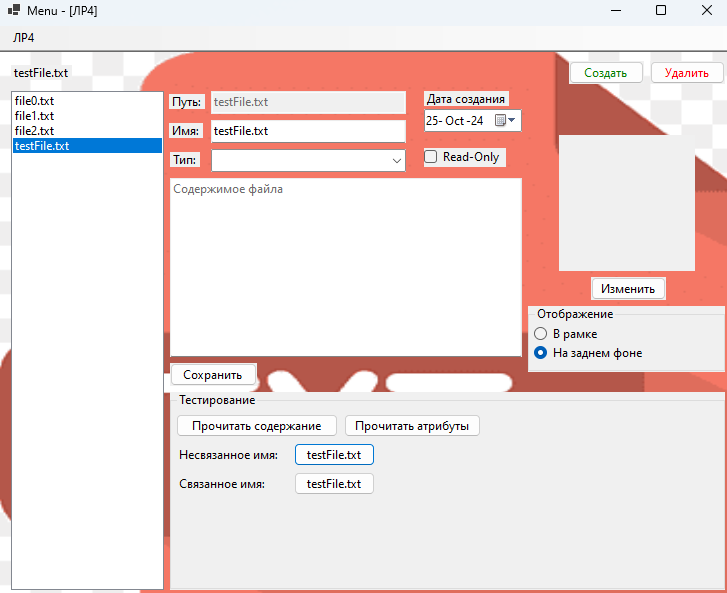


Рисунок 12

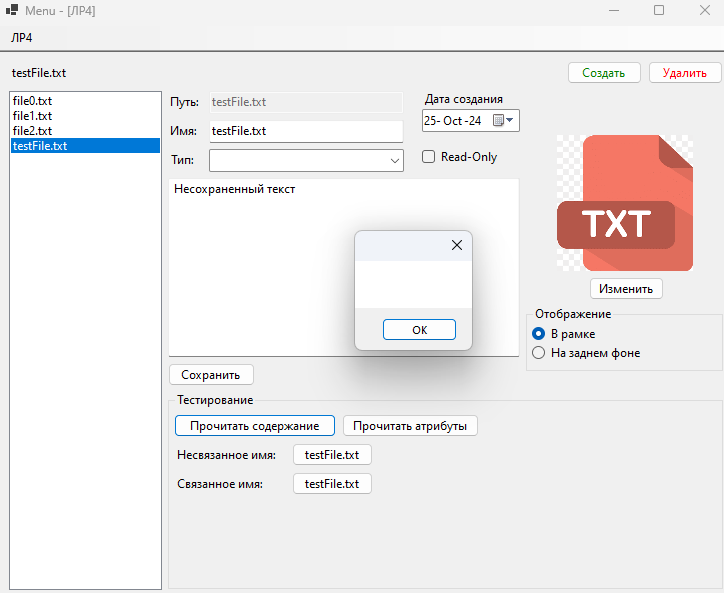


Рисунок 13

ВЫВОДЫ

В ходе выполнения работы мной были освоены основные принципы синхронизации элементов управления в Windows Forms с объектами класса и их свойствами. Я научился привязывать коллекции данных к элементам интерфейса, что позволяет упростить отображение и обновление информации для пользователя.

* Использование DataSource и DataBindings позволяет связать элементы управления с коллекциями объектов и отдельными свойствами класса, обеспечивая динамическое обновление данных при взаимодействии с интерфейсом. Это значительно улучшает удобство и скорость работы с интерфейсом.
* Привязка свойств таких элементов, как TextBox, DateTimePicker, и CheckBox, к свойствам класса помогает легко отображать и редактировать данные объекта, реагируя на изменения выбора в ListBox.
* Реализация методов для загрузки и отображения пользовательских иконок с использованием PictureBox, Graphics и OpenFileDialog позволила изучить методы работы с графическими элементами в Windows Forms и дала возможность добавить наглядные визуальные элементы в интерфейс.

Таким образом, я научился эффективно связывать данные и элементы управления, создавать удобный интерфейс для пользователя и реализовывать возможности настройки и изменения данных с минимальным кодом.