

Практическая работа №1 Разработка приложения по созданию и управлению базами данных.

Введение

В рамках учебной практики по программированию мы предлагаем вам захватывающий проект, направленный на разработку приложения по управлению базами данных. Этот проект призван предоставить вам возможность углубленного изучения и практической отработки различных аспектов работы с языком баз данных SQLite и C# Form, что является важным навыком для каждого разработчика.

Целью данной практической работы является создание полноценного редактора баз данных, способного выполнять разнообразные операции над ними. В процессе выполнения задания вы будете осваивать следующие ключевые аспекты:

1. **Создание и Удаление баз данных:** Приобретение навыков программного взаимодействия с системой баз данных для создания, удаления и переименования баз данных.
2. **Добавление, изменения сущностей и их атрибутов:** Разработка механизмов создания таблиц внутри баз данных с добавлением к ним сущностей.
3. **Работы с данными:** Добавление, изменение и удаление данных из базы данных.
4. **Реализация автоматизированной информационной системы:** Реализация функционала автоматизированной информационной системы с использованием созданных баз данных.
5. **Реализация использования директорий:** Разработка алгоритма для взаимодействия с разными директориями для получения файлов баз данных.

Цель проекта:

Разработать функционального много форменного приложения для взаимодействия с базами данных на языке программирования C# Form, способный выполнять основные операции работы с базами данных, обеспечивая удобства по созданию и удалению, и просмотра содержимого базы данных.

Основные требования:

Интерфейс много форменного приложения:

- Реализовать графический интерфейс формы для создания, удаления и открытия базы данных.
- Реализовать графический интерфейс формы редактора базы данных.
- Реализовать графический интерфейс формы вывода данных из базы данных.

Операции с базами данных:

- Реализовать возможность создания и удаления файлов и выбора файлов баз данных.
- Обеспечить механизм для просмотра доступных баз данных в нужной директории.

Работа с содержимым баз данных:

- Реализовать механизм по добавлению, удалению и обновлению сущностей и их атрибутов с их свойствами.
- Реализация возможности просмотра, изменения и удаления содержимого файлов баз данных.

Ход работы

1. Создание WinForms проекта:

- В Visual Studio создайте новый проект типа Windows Forms App (.NET Framework или .NET Core, в зависимости от вашей среды разработки).

2. Добавление компонентов для главной формы:

- Разместите компонент ListView на форме, для отображения списка доступных баз данных.
- Переместите на форму две кнопки под ListView для создания и удаления баз данных.
- Также добавьте две кнопки для перехода к формам «Редактор» и «АИС».

3. Функционал вывода доступных баз (Главная форма):

- Для компонента ListView напишите алгоритм для вывода доступных файлов с базами данных используя классы Directory, File, DirectoryInfo и FileInfo.
- Заполните соответствующий ListView полученными данными.

4. Функционал создания файлов баз данных (Главная форма):

- Используя язык запросов, реализуйте возможность создания базы данных с добавлением инкремента.
- Реализуйте с помощью дополнительной формы область для написания имени базы данных.
- Отобразите созданную базу данных в ListView используя классы Directory, File, DirectoryInfo и FileInfo.

5. Функционал удаления файлов баз данных (Главная форма):

- Используя язык запросов и выбранные элементы в ListView с помощью SelectedIndexChanged, реализуйте возможность удаления файла базы данных.
- Отобразите изменения в ListView используя классы Directory, File, DirectoryInfo и FileInfo.

6. Создание формы «Редактор» и добавление на форму компонентов:

- Разместите два компонента ListView на форме. Первый для отображения всех сущностей, а второй для редактирования списка атрибутов.
- Переместите на форму три кнопки над первым ListView для создания, удаления и изменения сущностей базы данных.
- Добавьте две кнопки над вторым ListView для создания, удаления атрибутов сущностей.

7. Создание формы «АИС» и добавление на форму компонентов:

- Разместите два компонента ListView на форме. Первый для отображения всех сущностей, а второй для редактирования и просмотра данных базы.

8. Переход в «АИС»:

- Реализуйте на главной форме выбор базы данных для последующего перехода в форму «АИС» с помощью SelectedIndexChanged в ListView.
- Реализуйте открытие и загрузки формы «АИС».

9. Переход в «Редактор»:

- Реализуйте на главной форме выбор базы данных для последующего перехода в форму «Редактор» с помощью SelectedIndexChanged в ListView.
- Реализуйте открытие и загрузки формы «Редактор».

10. Вывод имен сущностей в ListView (Редактор):

- Используя язык запросов, реализуйте получения имен все сущностей базы данных.
- Отобразите полученный список в ListView используя классы Directory, File, DirectoryInfo и FileInfo.

11. Вывод атрибутов выбранной сущности в ListView (Редактор):

- Используя язык запросов, реализуйте получения атрибутов сущности.
- Отобразите полученный список в ListView используя классы Directory, File, DirectoryInfo и FileInfo.

12. Добавления нового атрибута сущности в базу данных (Редактор):

- Добавьте событие на кнопку «Создать» и используя язык запросов, реализуйте добавление нового атрибута.
- Обновите список атрибутов в ListView используя классы Directory, File, DirectoryInfo и FileInfo.

13. Добавления новой сущности в базу данных (Редактор):

- Добавьте событие на кнопку «Создать» и используя язык запросов, реализуйте добавление новой сущности.
- Обновите список сущностей в ListView используя классы Directory, File, DirectoryInfo и FileInfo.

14. Удаление сущности базы данных (Редактор):

- Добавьте событие на кнопку «Удалить» и используя язык запросов и SelectedIndexChanged в ListView, реализуйте удаление сущности.
- Обновите список сущностей в ListView используя классы Directory, File, DirectoryInfo и FileInfo.

15. Модифицирование сущностей (Редактор):

- Добавьте событие на кнопку «Модифицировать» и используя язык запросов и SelectedIndexChanged в ListView, реализуйте модифицирование сущности.
- Используя вспомогательные окна, пропишите изменения.
- Обновите список сущностей в ListView используя классы Directory, File, DirectoryInfo и FileInfo.

16. Добавления нового атрибута сущности в базу данных (Редактор):

- Добавьте событие на кнопку «Создать» и используя язык запросов, реализуйте добавление нового атрибута сущности.
- Используя вспомогательные окна, пропишите все нужные данные.
- Обновите список атрибутов сущности в ListView используя классы Directory, File, DirectoryInfo и FileInfo.

17. Удаление атрибутов сущности базы данных (Редактор):

- Добавьте событие на кнопку «Удалить» и используя язык запросов и SelectedIndexChanged в ListView, реализуйте удаление атрибутов сущности.
- Обновите список атрибутов сущности в ListView используя классы Directory, File, DirectoryInfo и FileInfo.

18. Вывод имен сущностей в ListView (АИС):

- Используя язык запросов, реализуйте получения имен все сущностей базы данных.
- Отобразите полученный список в ListView используя классы Directory, File, DirectoryInfo и FileInfo.

19. Вывод данных в ListView (АИС):

- Используя язык запросов, реализуйте получения данных выбранной сущности.
- Отобразите полученный список в ListView используя классы Directory, File, DirectoryInfo и FileInfo.

20. Добавление, удаление, обновление (АИС):

- Используя язык запросов, реализуйте события на добавление, удаление, обновление.
- Для добавления новых записей используйте вспомогательные окна.
- Отобразите полученный список в ListView используя классы Directory, File, DirectoryInfo и FileInfo.

21. Фильтрация данных (АИС):

- Используя язык запросов, отфильтруйте данные по выбранному условию.
- Отобразите полученный список в ListView используя классы Directory, File, DirectoryInfo и FileInfo.

22. Динамическое обновление:

- Обеспечьте динамическое обновление информации на всех ListView.

23. Обработка исключений:

- Предусмотрите сценарии, когда пользователь ошибся. Реализуйте событие вызова всплывающего окна с указанием на неверное действие со стороны пользователя и с путями решения проблемы.

24. Оптимизация и комментирование кода:

- Используйте методы и события для оптимизации и лучшей читаемости кода.
- Напишите комментарии к фрагментам кода с указанием того, за что они отвечают.