Практическая работа №1 Разработка приложению по созданию и управлению базами 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.

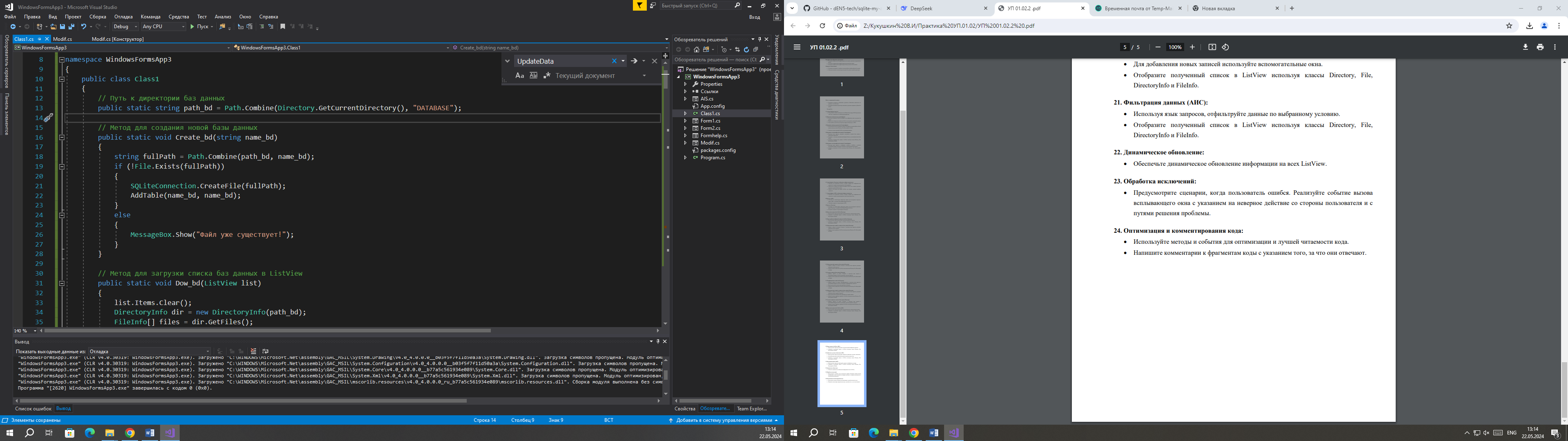


Рисунок 1. Метод для создания новой базы данных.

На Рисунке №1 представлен код на C#, который использует библиотеку SQLite для создания базы данных.

1. `path\_bd` - это статическая переменная, которая хранит путь к директории баз данных. Она объединяет текущую рабочую директорию и строку "DATABASE".

2. `Create\_bd` - это метод, который создает новую базу данных. Он принимает на вход имя базы данных (`name\_bd`).

3. `fullPath` - это переменная, которая хранит полный путь к файлу базы данных. Она объединяет `path\_bd` и `name\_bd`.

4. `SQLiteConnection.CreateFile(fullPath)` - эта строка кода создает новый файл базы данных SQLite с именем, указанным в `fullPath`.

5. `AddTable(name\_bd, name\_bd)` - эта строка кода вызывает метод `AddTable`, передавая ему два раза `name\_bd`. Этот метод, очевидно, добавляет новую таблицу в базу данных.

6. `if (!File.Exists(fullPath))` - это условие, которое проверяет, существует ли файл базы данных. Если файл не существует, то создается новый файл базы данных и вызывается метод `AddTable`. Если файл уже существует, то выводится сообщение об ошибке.

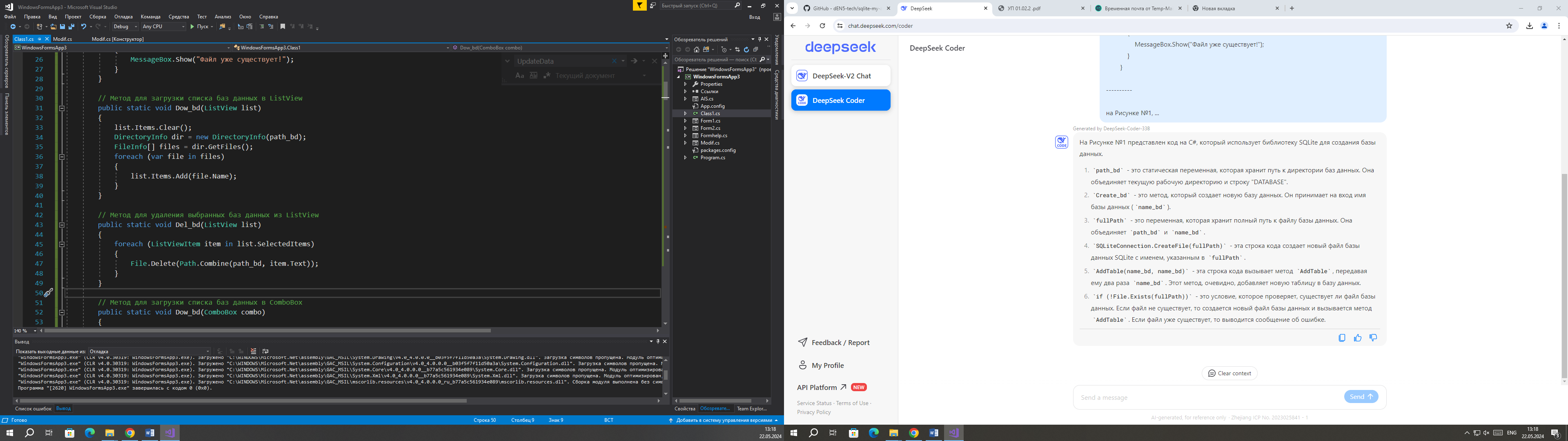


Рисунок 2. Метод для загрузки списка баз данных в ListView. Метод для удаления выбранных баз данных из ListView.

На Рисунке №2 представлены методы для работы с базами данных в приложении, использующем SQLite.

Dow\_bd - это метод, который загружает список баз данных из директории, указанной в path\_bd, в ListView.

list.Items.Clear() - эта строка кода очищает текущий список элементов в ListView.

DirectoryInfo dir = new DirectoryInfo(path\_bd) - создается новый экземпляр DirectoryInfo для директории, указанной в path\_bd.

FileInfo[] files = dir.GetFiles() - метод GetFiles() вызывается для получения массива всех файлов в директории.

foreach (var file in files) - это цикл, который проходит по каждому файлу в массиве files.

list.Items.Add(file.Name) - для каждого файла в массиве files, его имя добавляется в ListView.

Del\_bd - это метод, который удаляет выбранные базы данных из ListView.

foreach (ListViewItem item in list.SelectedItems) - это цикл, который проходит по каждому выбранному элементу в ListView.

File.Delete(Path.Combine(path\_bd, item.Text)) - для каждого выбранного элемента в ListView, файл с именем, соответствующим тексту элемента, удаляется из директории, указанной в path\_bd.

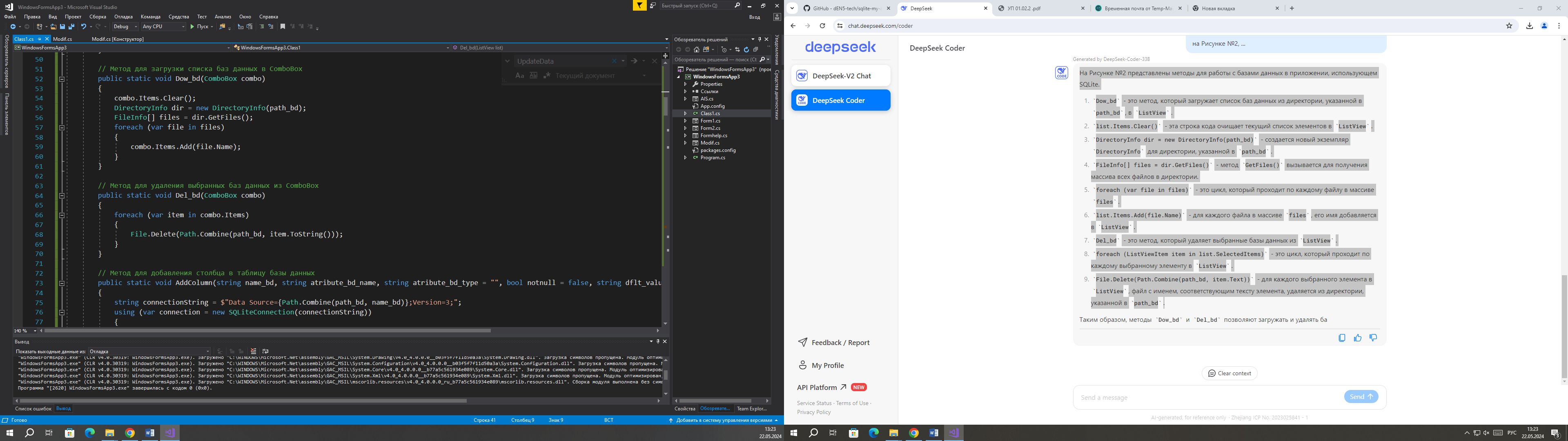


Рисунок 3. Метод для загрузки списка баз данных в ComboBox. Метод для удаления выбранных баз данных из ComboBox.

На Рисунке №3 представлены методы для работы с базами данных в приложении, использующем SQLite и управляемые через ComboBox.

Dow\_bd - это метод, который загружает список баз данных из директории, указанной в path\_bd, в ComboBox.

combo.Items.Clear() - эта строка кода очищает текущий список элементов в ComboBox.

DirectoryInfo dir = new DirectoryInfo(path\_bd) - создается новый экземпляр DirectoryInfo для директории, указанной в path\_bd.

FileInfo[] files = dir.GetFiles() - метод GetFiles() вызывается для получения массива всех файлов в директории.

foreach (var file in files) - это цикл, который проходит по каждому файлу в массиве files.

combo.Items.Add(file.Name) - для каждого файла в массиве files, его имя добавляется в ComboBox.

Del\_bd - это метод, который удаляет выбранные базы данных из ComboBox.

foreach (var item in combo.Items) - это цикл, который проходит по каждому элементу в ComboBox.

File.Delete(Path.Combine(path\_bd, item.ToString())) - для каждого элемента в ComboBox, файл с именем, соответствующим тексту элемента, удаляется из директории, указанной в path\_bd.

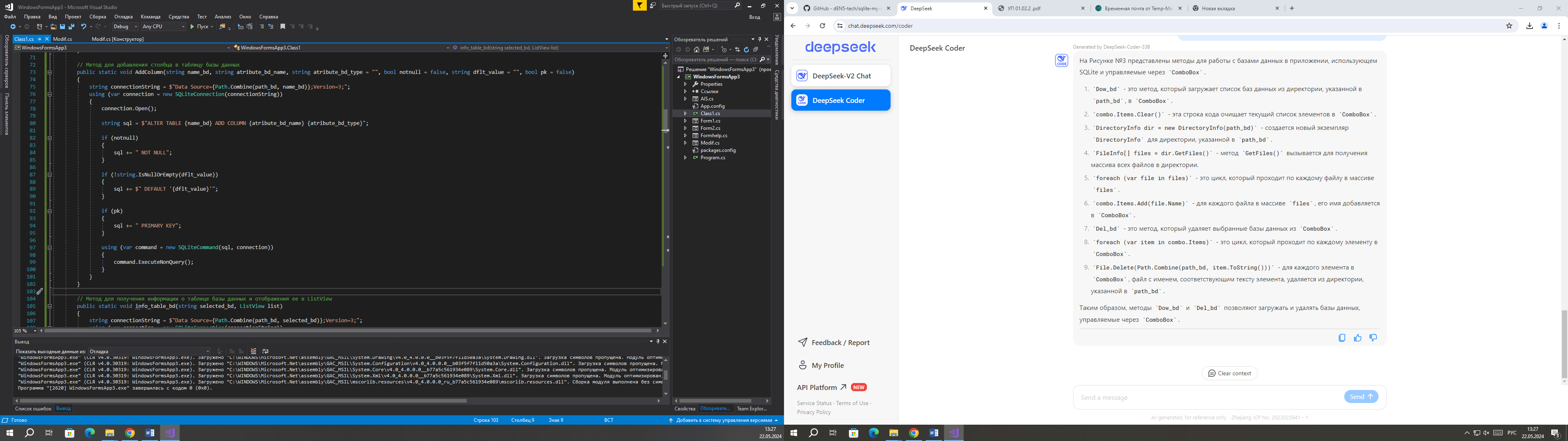


Рисунок 4. Метод для добавления столбца в таблицу базы данных

На Рисунке №4 представлен метод AddColumn, который используется для добавления столбца в таблицу базы данных SQLite.

AddColumn - это метод, который принимает следующие параметры:

name\_bd - имя базы данных, в которой нужно добавить столбец.

atribute\_bd\_name - имя нового столбца.

atribute\_bd\_type - тип данных нового столбца (например, TEXT, INTEGER, REAL, BLOB).

notnull - флаг, указывающий, может ли столбец содержать NULL значения.

dflt\_value - значение по умолчанию для столбца.

pk - флаг, указывающий, является ли столбец первичным ключом.

string connectionString = $"Data Source={Path.Combine(path\_bd, name\_bd)};Version=3;"; - строка подключения к базе данных, где path\_bd - путь к директории баз данных, а name\_bd - имя базы данных.

using (var connection = new SQLiteConnection(connectionString)) - создается новое соединение с базой данных и оно автоматически закрывается после выполнения блока кода.

connection.Open(); - открывает соединение с базой данных.

string sql = $"ALTER TABLE {name\_bd} ADD COLUMN {atribute\_bd\_name} {atribute\_bd\_type}"; - формируется SQL-запрос для добавления столбца в таблицу.

Следующие условия добавляют к SQL-запросу дополнительные параметры, такие как NOT NULL, значение по умолчанию и первичный ключ, если они указаны.

using (var command = new SQLiteCommand(sql, connection)) - создается новый экземпляр SQLiteCommand с SQL-запросом и соединением к базе данных.

command.ExecuteNonQuery(); - выполняет SQL-запрос, который добавляет столбец в таблицу.

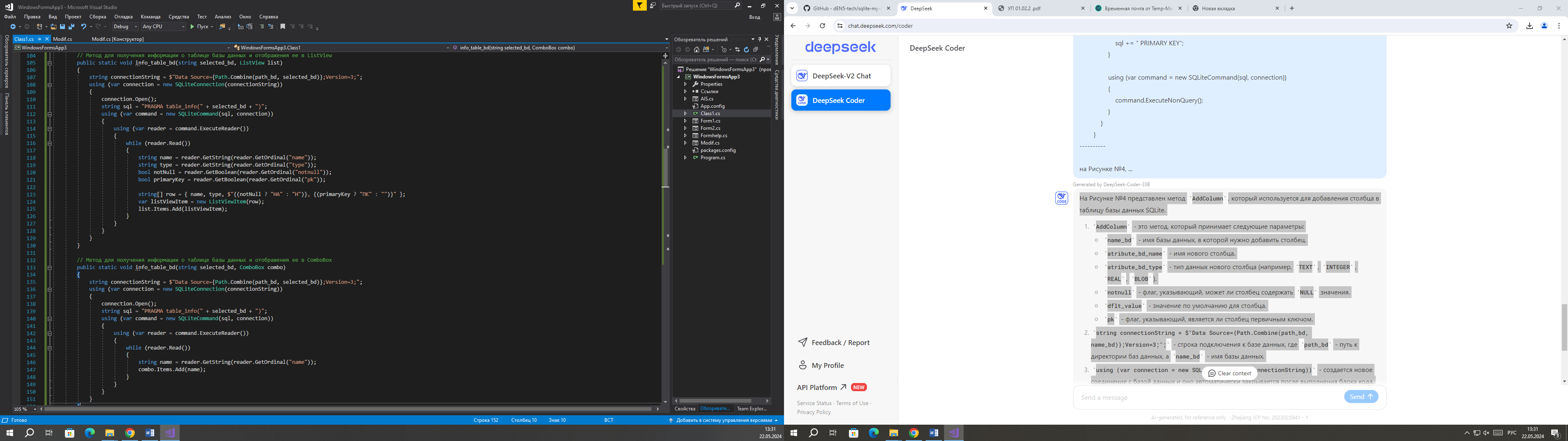


Рисунок 5. Метод для получения информации о таблице базы данных и отображения ее в ListView. Метод для получения информации о таблице базы данных и отображения ее в ComboBox.

На Рисунке №5 представлены методы для получения информации о таблице базы данных и ее отображения в ListView и ComboBox.

info\_table\_bd - это метод, который принимает на вход имя выбранной базы данных (selected\_bd) и ListView (list), а также ComboBox (combo).

string connectionString = $"Data Source={Path.Combine(path\_bd, selected\_bd)};Version=3;"; - строка подключения к базе данных, где path\_bd - путь к директории баз данных, а selected\_bd - имя выбранной базы данных.

using (var connection = new SQLiteConnection(connectionString)) - создается новое соединение с базой данных и оно автоматически закрывается после выполнения блока кода.

connection.Open(); - открывает соединение с базой данных.

string sql = "PRAGMA table\_info(" + selected\_bd + ")"; - SQL-запрос, который используется для получения информации о таблице базы данных.

using (var command = new SQLiteCommand(sql, connection)) - создается новый экземпляр SQLiteCommand с SQL-запросом и соединением к базе данных.

using (var reader = command.ExecuteReader()) - выполняет SQL-запрос и возвращает SQLiteDataReader для чтения результатов запроса.

while (reader.Read()) - цикл, который проходит по каждой строке результата запроса.

Внутри цикла считываются данные из каждой строки результата запроса и добавляются в ListView или ComboBox.

Для ListView создается новый экземпляр ListViewItem с данными столбца и добавляется в ListView.

Для ComboBox просто добавляется имя столбца в ComboBox.