**Обзор различных способов и форм взаимодействия пользователя с системой.**

Взаимодействие пользователя с системой может происходить различными способами. Сейчас будут рассмотрены самые распространённые их них.

Текстовый интерфейс пользователя (CLI) — это тип пользовательского интерфейса, в котором команды и данные вводятся через командную строку. CLI часто используется в операционных системах и некоторых приложениях для технических задач.

Графический интерфейс пользователя (GUI) — это наиболее распространенный тип интерфейса в IT-системах. В интерфейсе применяются графические элементы (кнопки, поля ввода, поля выбора и др.), что облегчает взаимодействие пользователя с системой.

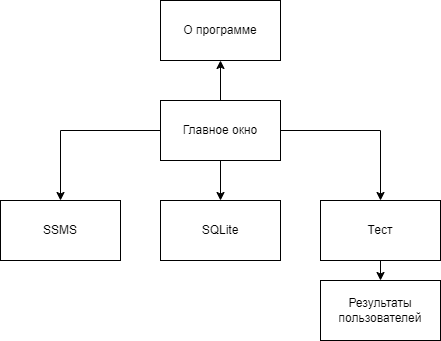
Жестовый интерфейс (GBI, Gesture-Based Interface или GDI — Gesture Driven). Это технология взаимодействия, с помощью которой можно управлять устройствами, которые «понимают» жесты (движения тела). Используется в ноутбуках, смартфонах и планшетах. Бывают двух типов: контактный (тактильный) и бесконтактный.

Голосовой интерфейс (VUI, Voice user interface). Пользователь вводит голосовую команду на своем родном языке, смартфон распознает ее и выполняет. Практическая реализация голосовых интерфейсов – сложная задача, программисты должны учитывать множество параметров. Пример: программы Алиса у Яндекс, OK Google.

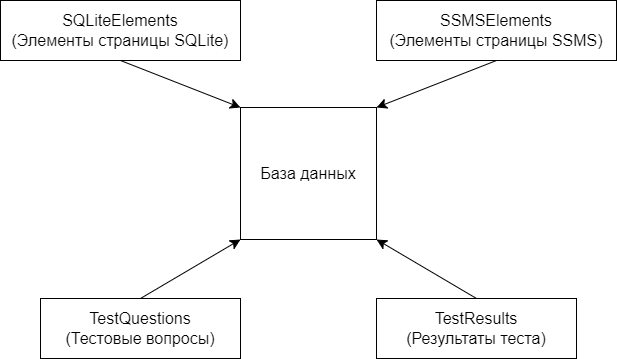
Для общения с пользователем был выбран способ взаимодействия пользователя с системой с помощью графического интерфейса. Этот способ был выбран потому, что подавать и объяснять информацию пользователю так будет более наглядно, доступно и удобно. Это будет достигаться с помощью визуальных примеров, текстовым содержанием.

Определив способ взаимодействия пользователя, пере

**Схема программного приложения**



**Информационная модель базы данных**

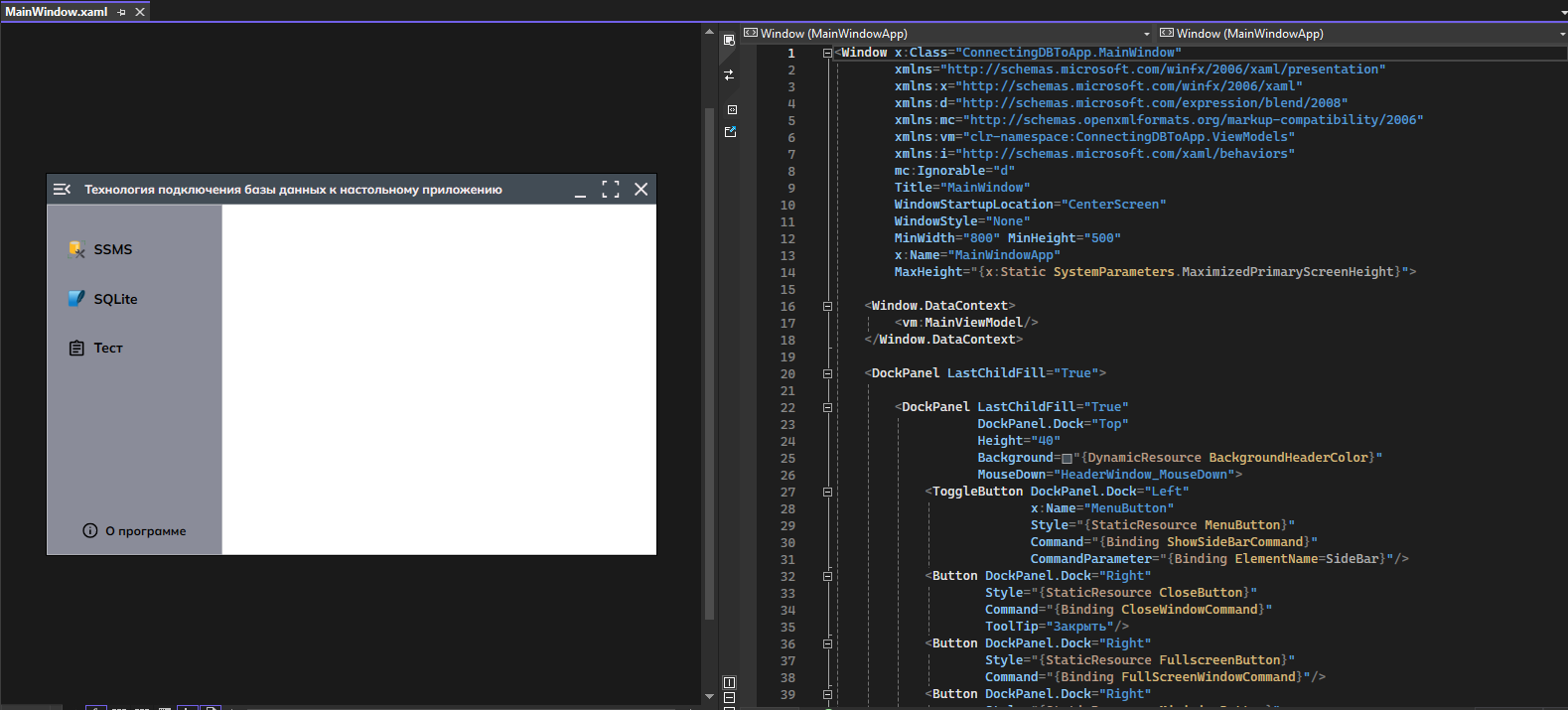


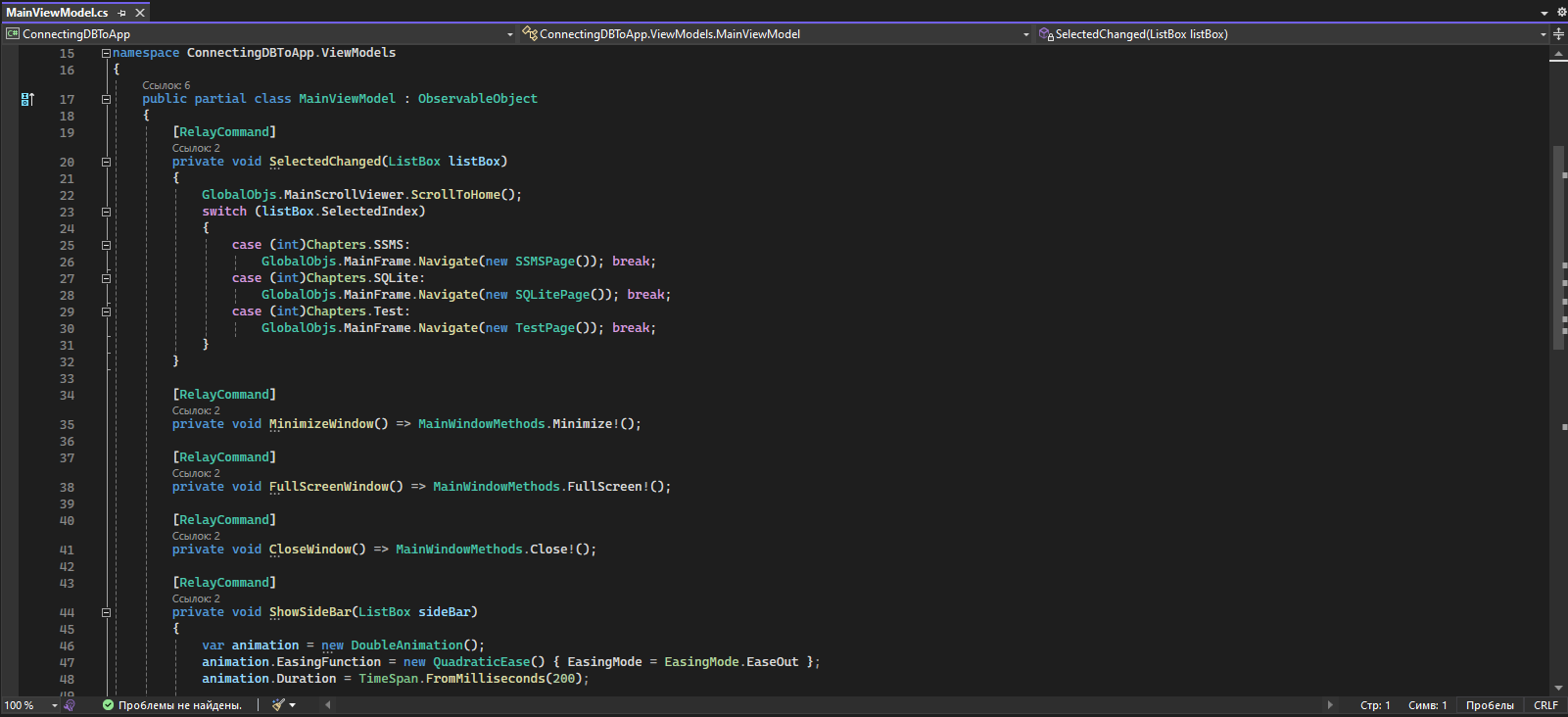
**Разработка интерфейса программного приложения и проектирование программного кода приложения**

*Главное окно приложения* осуществляет навигацию по всему приложению. В левой части окна находится боковая панель приложения, в которой находятся все разделы приложения. В правой части окна непосредственно отображаются страницы этих разделов.

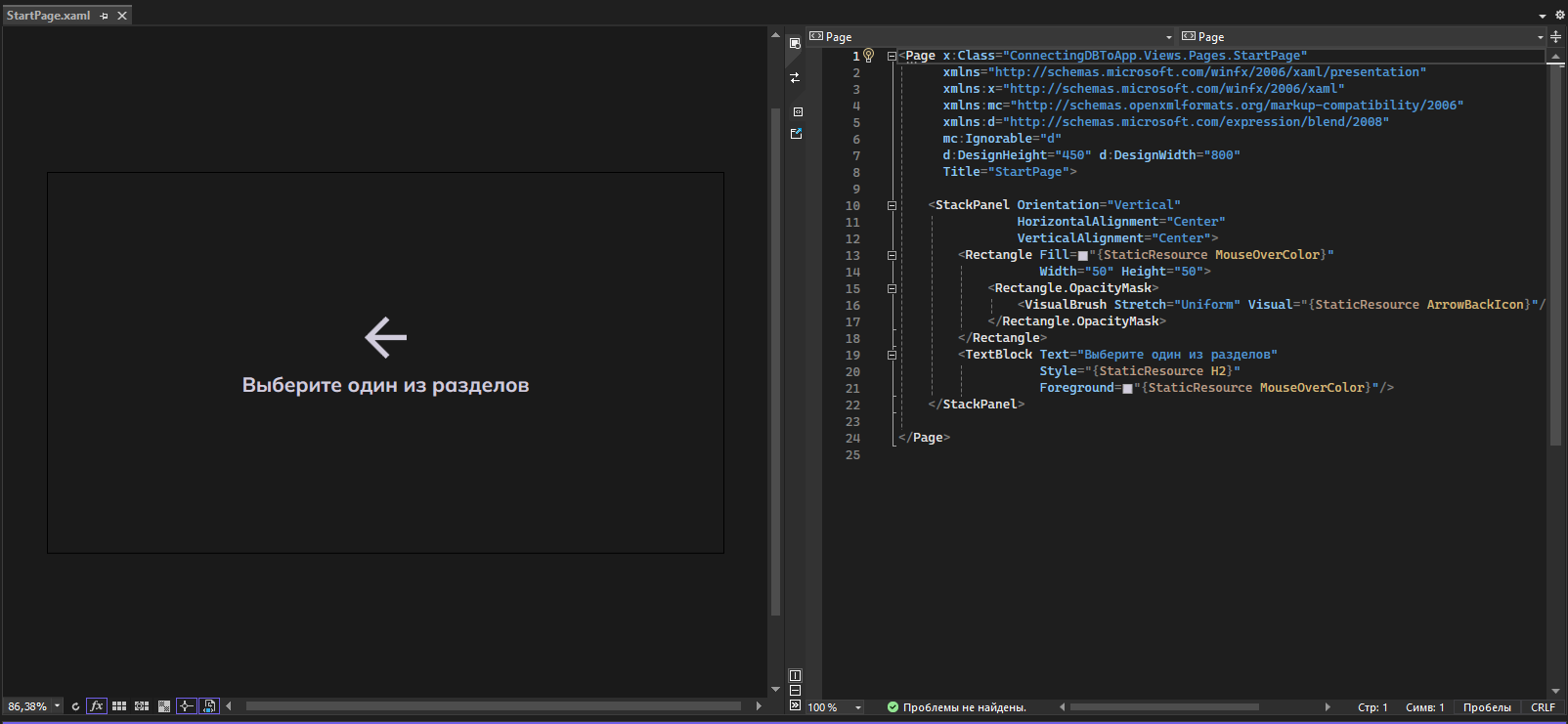
Класс MainViewModel реализуют логику для главного окна. В нём реализованы:

* переходы по разделам
* открытие окна «О программе»
* показ/скрытие меню навигации
* реализация функционала для кастомизированных кнопок «Свернуть», «Полный экран», «Закрыть»

Рисунок 1. Интерфейс и разметка главного окна приложения

 Рисунок 2. Класс с логикой главного окна приложения

*Стартовая страница приложения* – это страница, котораяотображается при запуске программного приложения.

 Рисунок 3. Интерфейс и разметка стартовой страницы

*Страница с теоретическими сведениями о SQL Server Managament Studio* отображается теоретические сведения о работе с такой СУБД как SQL Server Managament Studio. Эта страница выполнена в виде вкладок по которым можно перемещаться. В каждой вкладке отображается часть определённой обучающей информации.

Для этой страницы класс SSMSViewModel реализует получение данных из базы данных, функцию для копирования текста в буфер обмена и функции, реализующие перемещение по разделам с помощью гиперссылок в конце каждого раздела страницы.

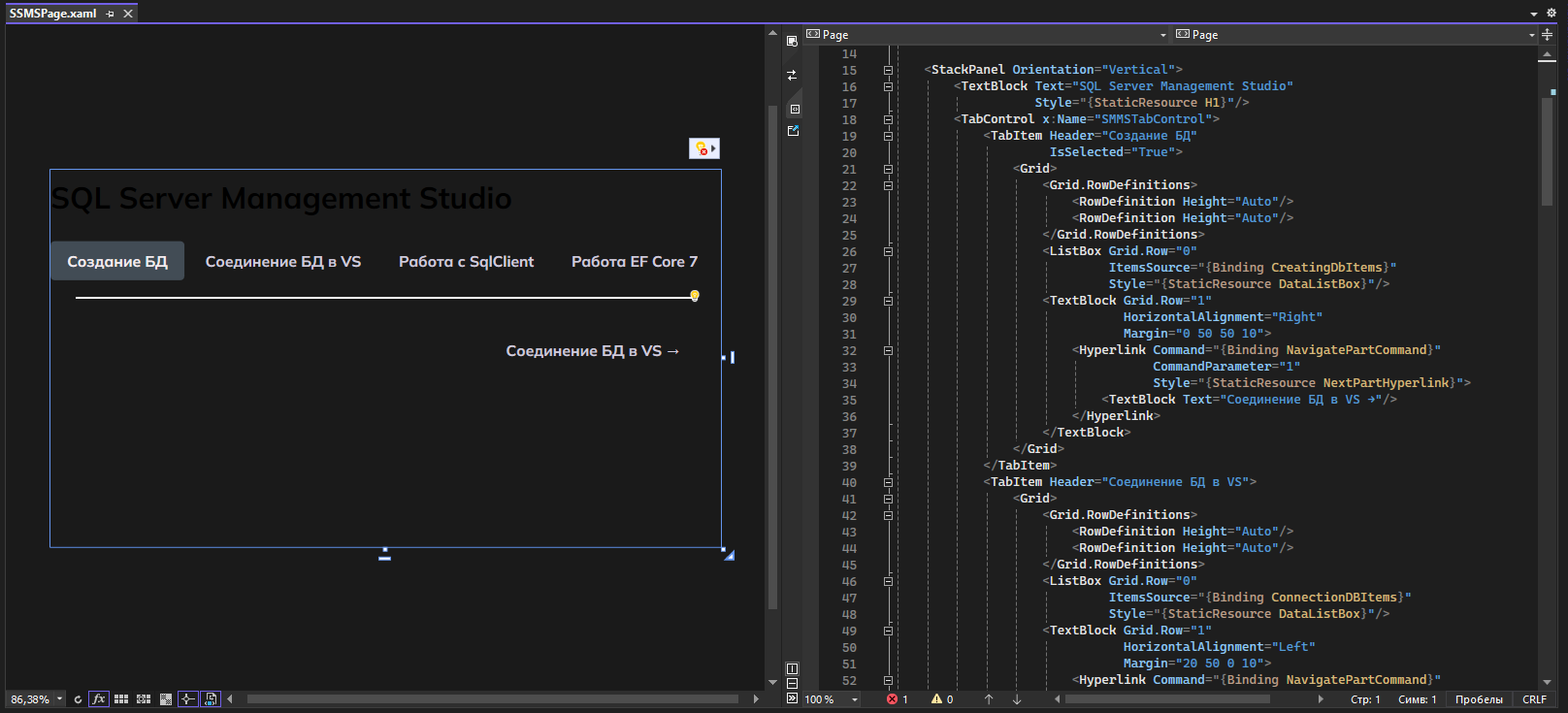
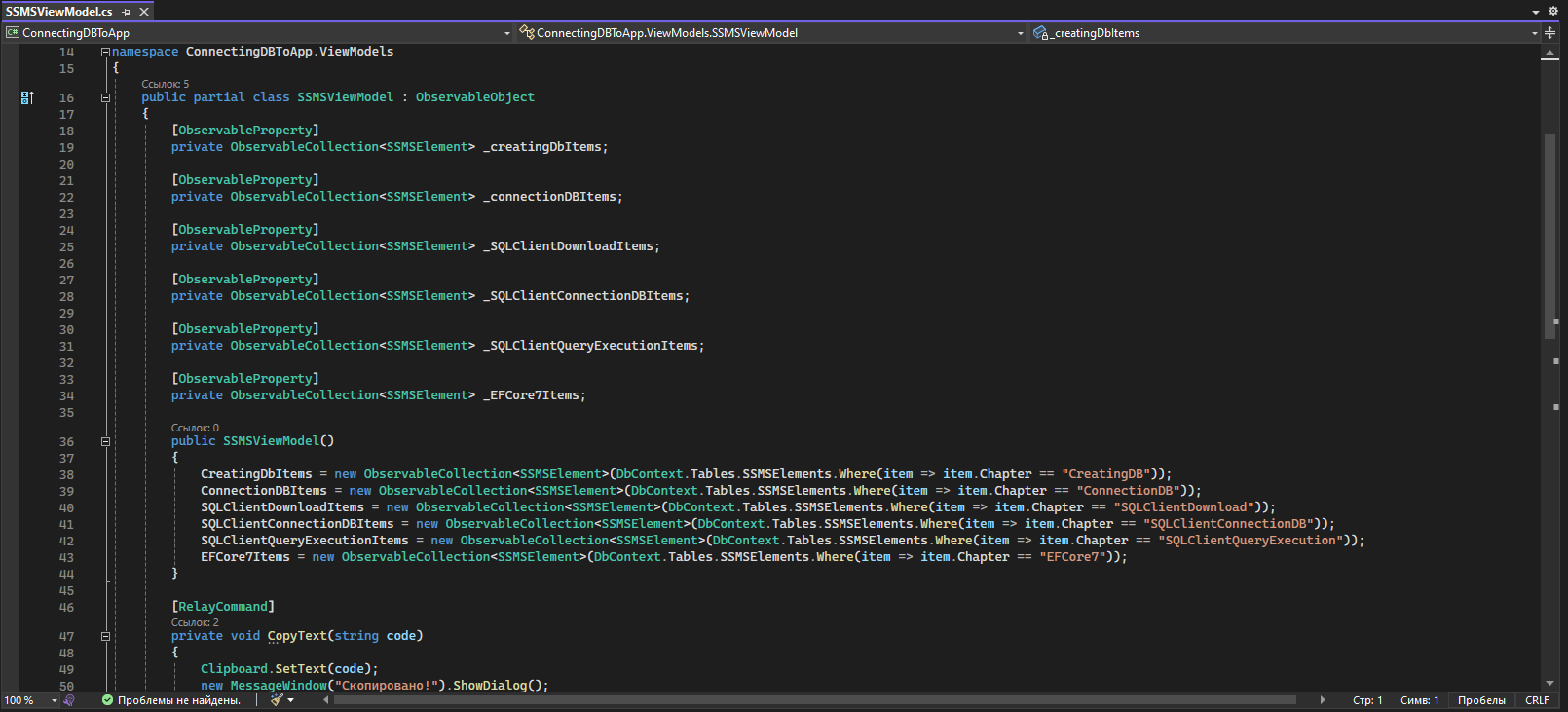
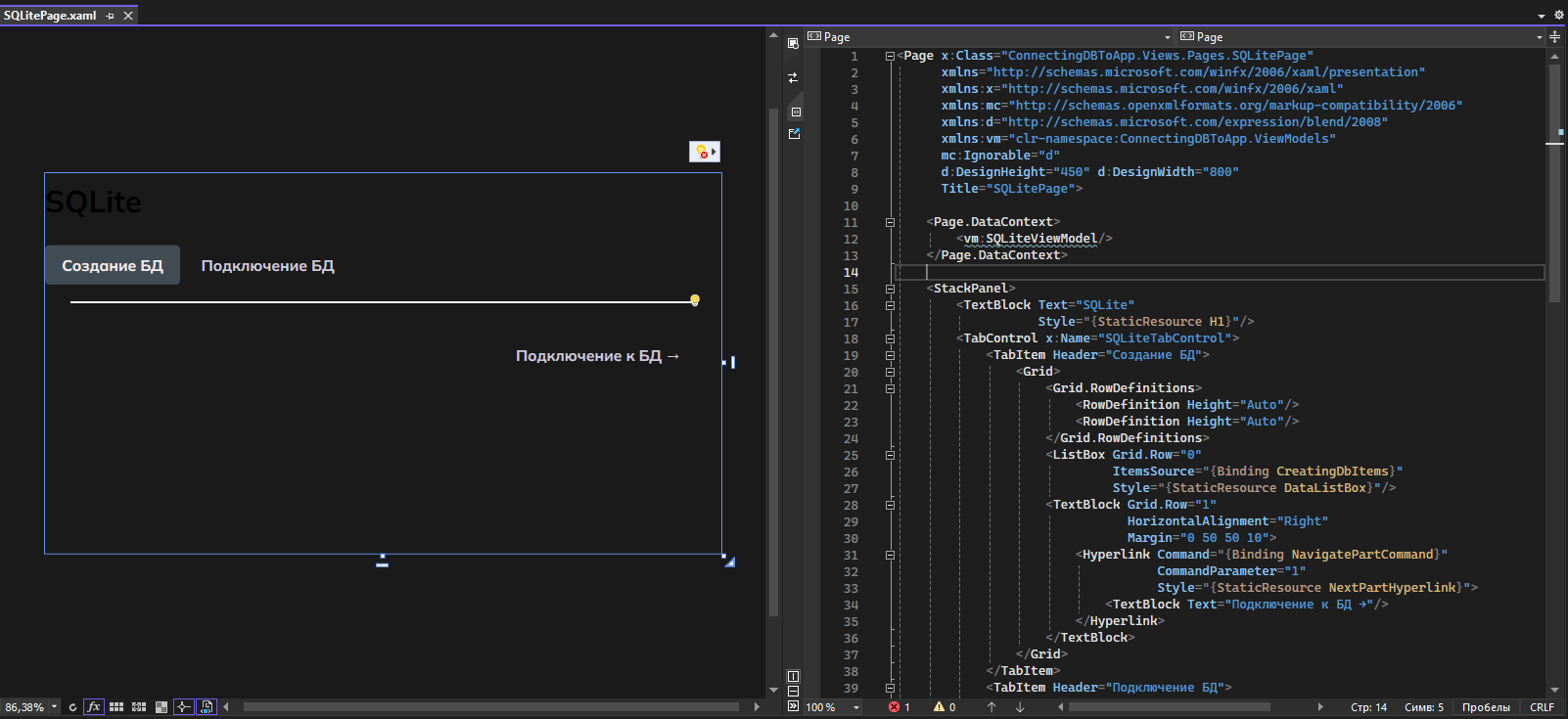


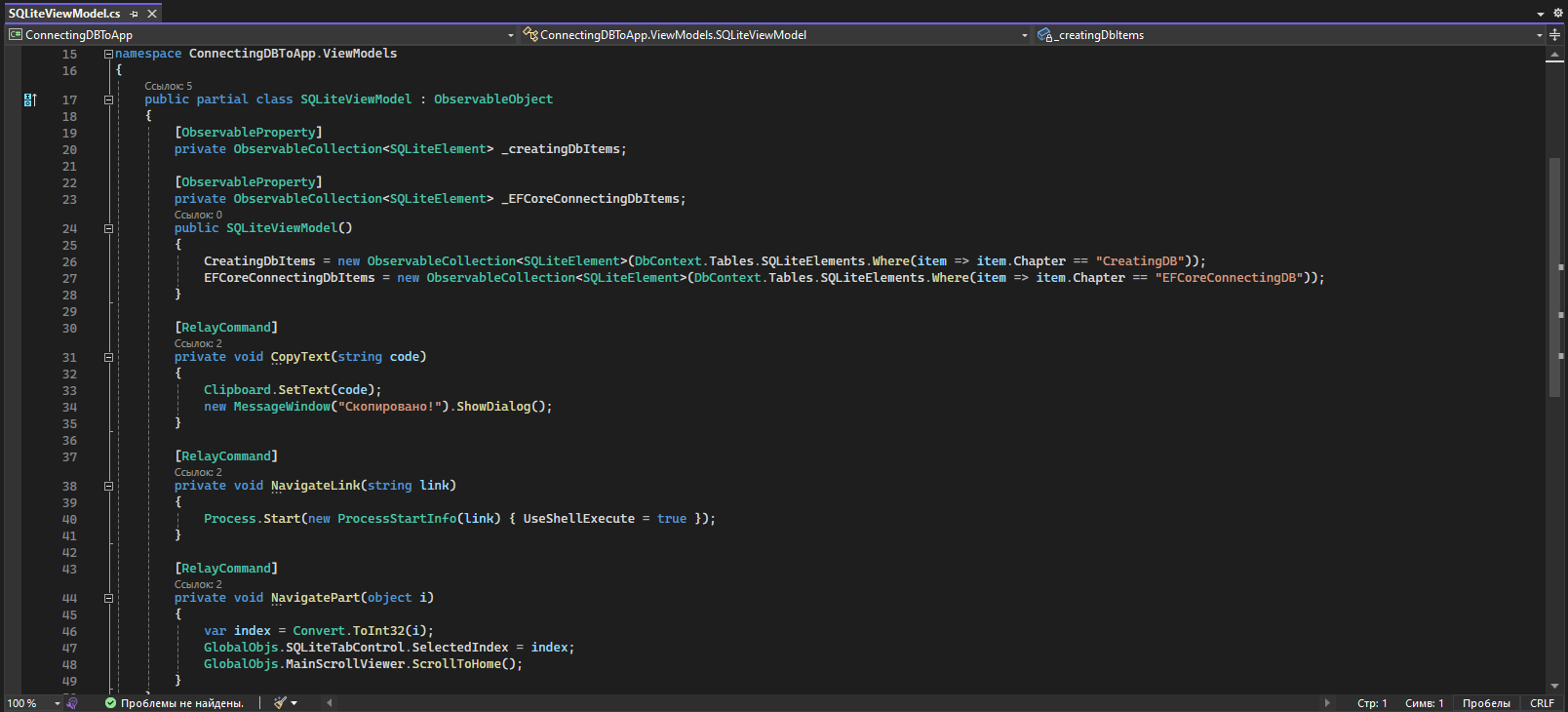
Рисунок 4. Интерфейс и разметка страницы SSMS

Рисунок 5. Класс с логикой страницы SSMS

*Страница с теоретическими сведениями о SQLite* отображается теоретические сведения о работе с такой СУБД как SQLite. Пользовательский интерфейс идентичен странице с SSMS.

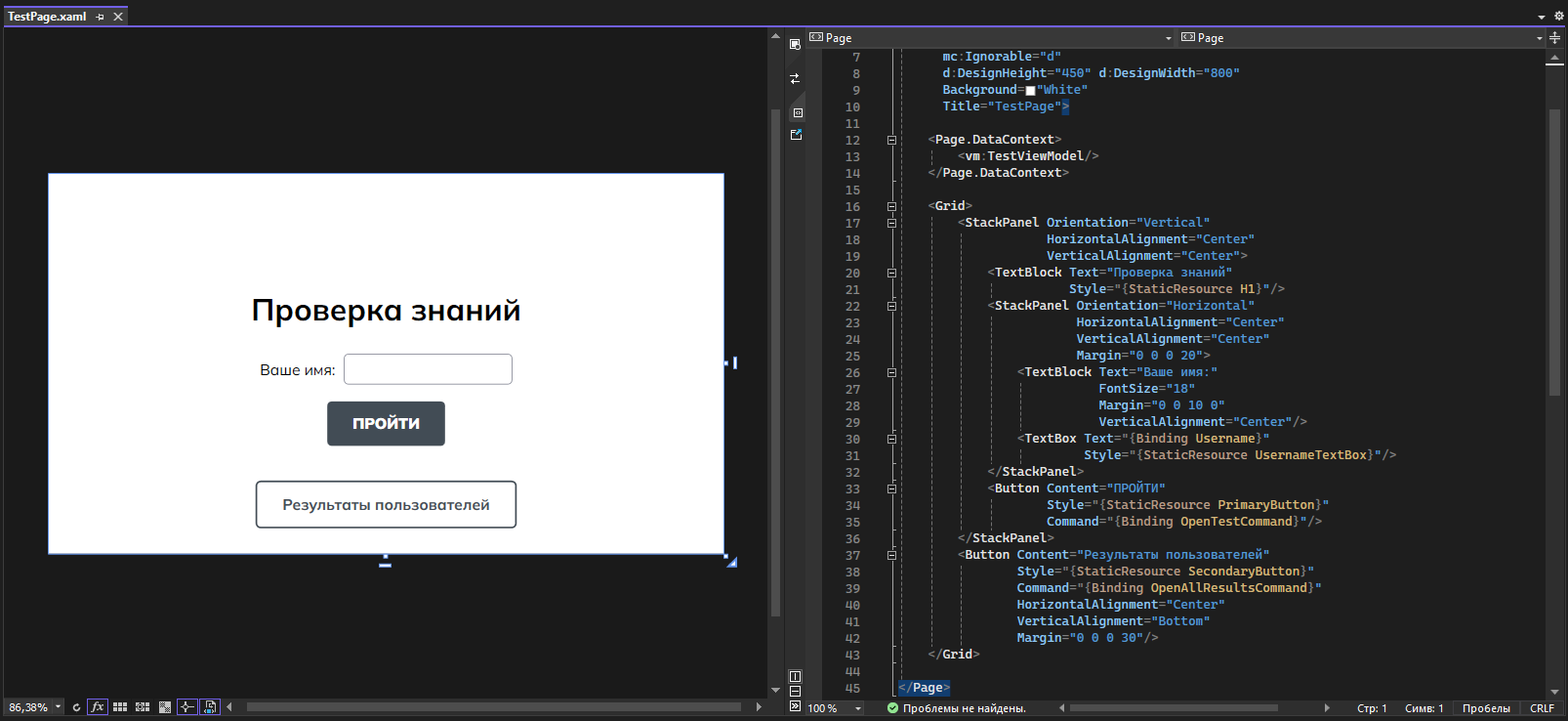
Класс SQLiteViewModel реализует тот же самый функционал, как и класс SSMSViewModel, только для страницы с SQLite.

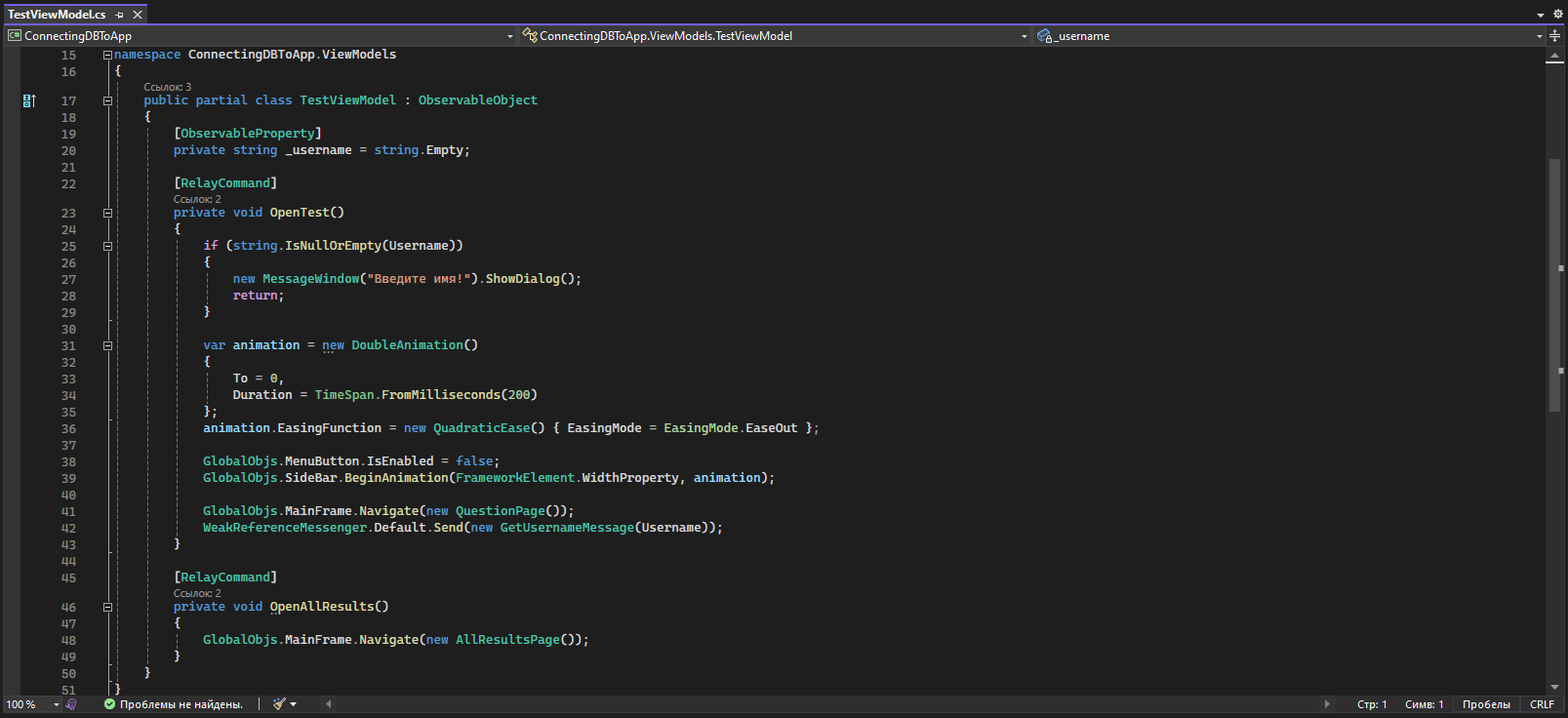
 Рисунок 6. Интерфейс и разметка страницы SQLite

 Рисунок 7. Класс с логикой страницы SQLite

*Начальная страница теста* отображается при переходе в раздел с тестом. С помощью неё можно ввести имя и начать прохождение теста, либо просмотреть все результаты пользователей.

Класс TestViewModel реализует такой функционал как переход на страницу со всеми результатами и функцию начала теста с проверкой введённого имени в поле ввода.

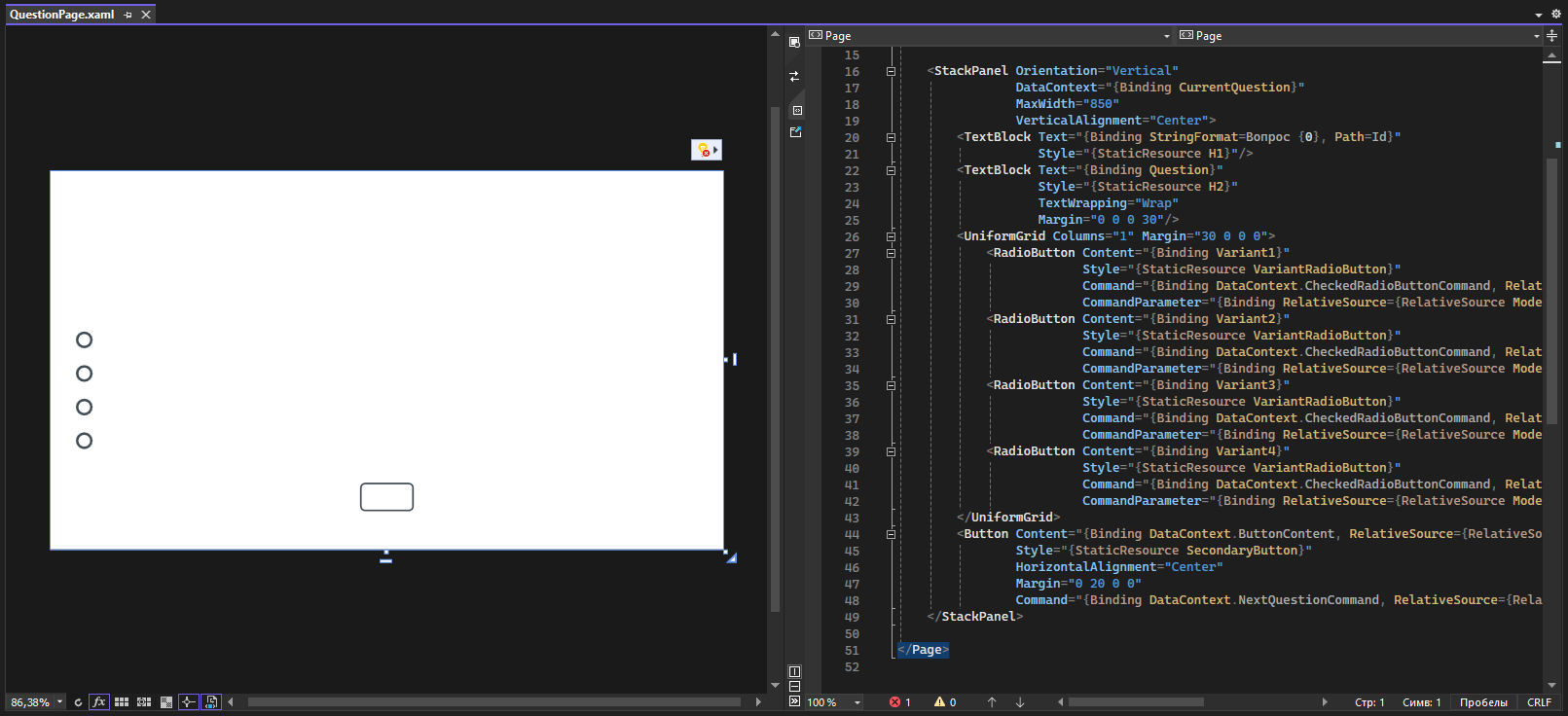
 Рисунок 8. Интерфейс и разметка начальной страницы теста

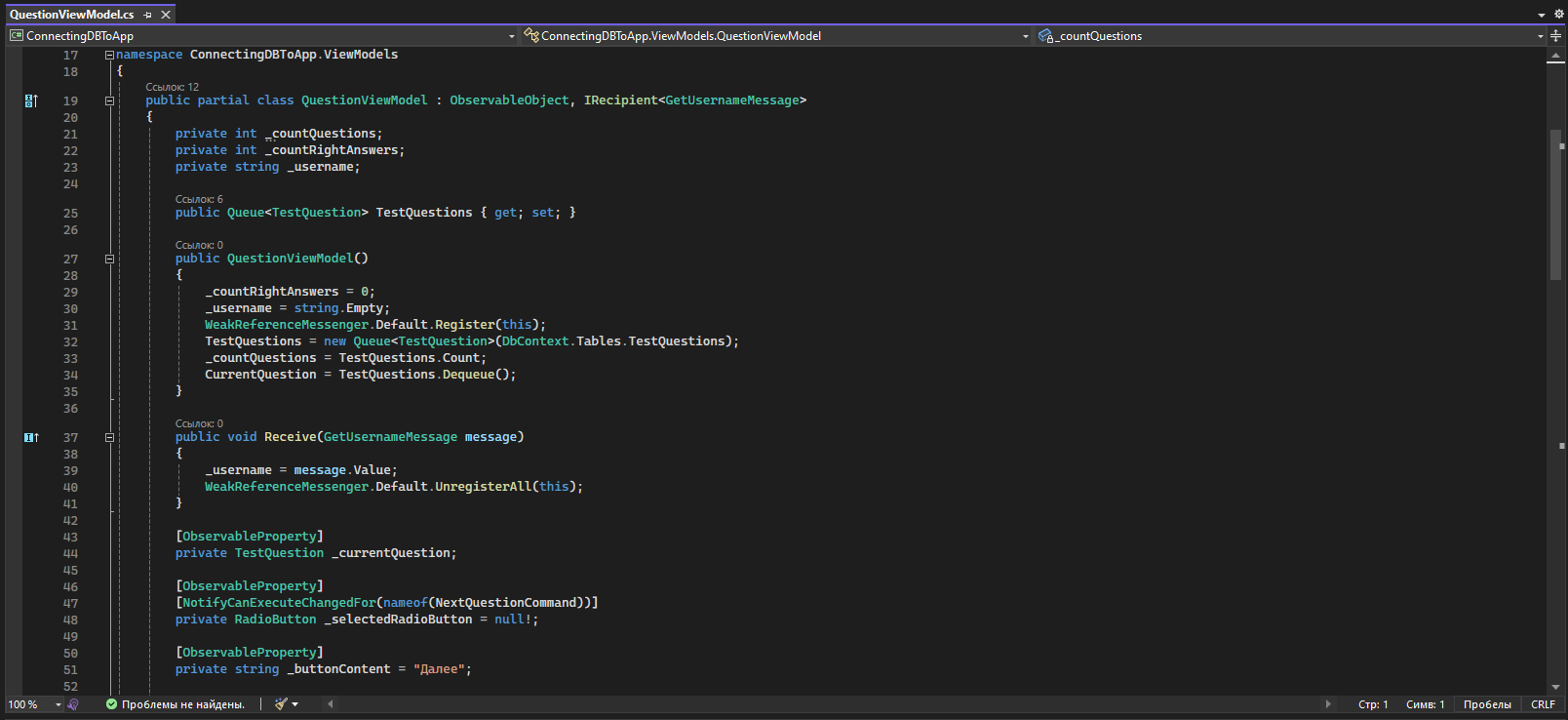
 Рисунок 9. Класс с логикой начальной страницы теста

*Страница с вопросом* отображает текущий вопрос теста. Интерфейс состоит из вопроса, четырёх вариантов ответа и кнопкой перехода на следующий вопрос.

Класс QuestionViewModel реализует:

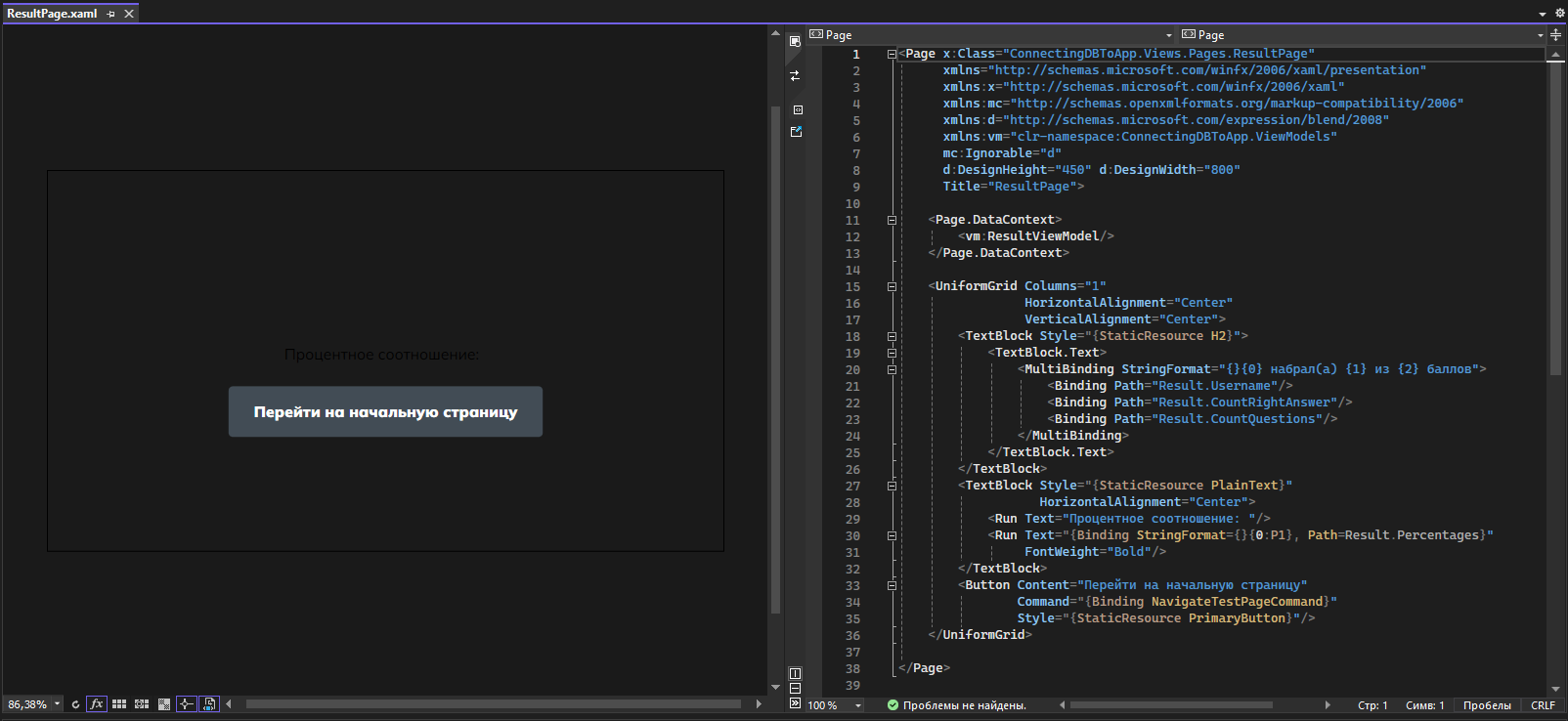
* переход между вопросами теста
* высчитывание результата
* отправка результата на страницу с итоговым результатом

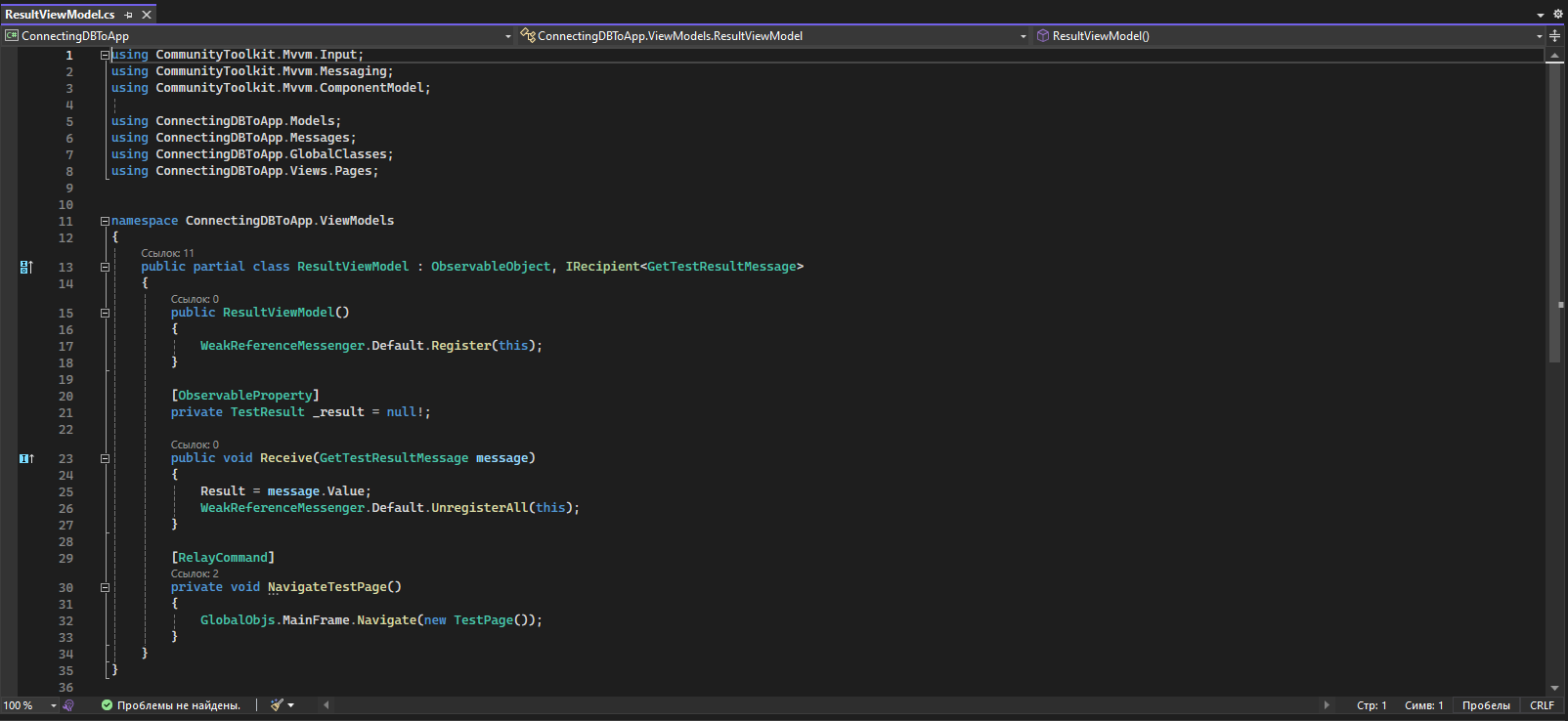
Рисунок 10. Интерфейс и разметка страницы с вопросом

Рисунок 11. Класс с логикой страницы с вопросом

*Страница с результатом теста* отображает результат теста. На ней находится текстовые блоки с результатом и кнопкой возвращения на начальную страницу теста.

Класс ResultViewModel принимает вычисленный результат от класса QuestionViewModel и передаёт его странице с результатом теста для его отображения. Ещё этот класс реализует ту самую функцию возвращения на начальную страницу раздела с тестом.

Рисунок 12. Интерфейс и разметка страницы с результатом теста

Рисунок 13. Класс с логикой страницы с результатом теста

*Страница со списком всех результатов* отображает все результаты пользователей, проходивших тест. Состоит из текстового блока, списка элементов и кнопкой удаления результатов.

Класс AllResultsViewModel передаёт данные о всех результатах пользователей данной странице, реализует функции возвращения на начальную страницу раздела с тестом и удаления всех результатов.

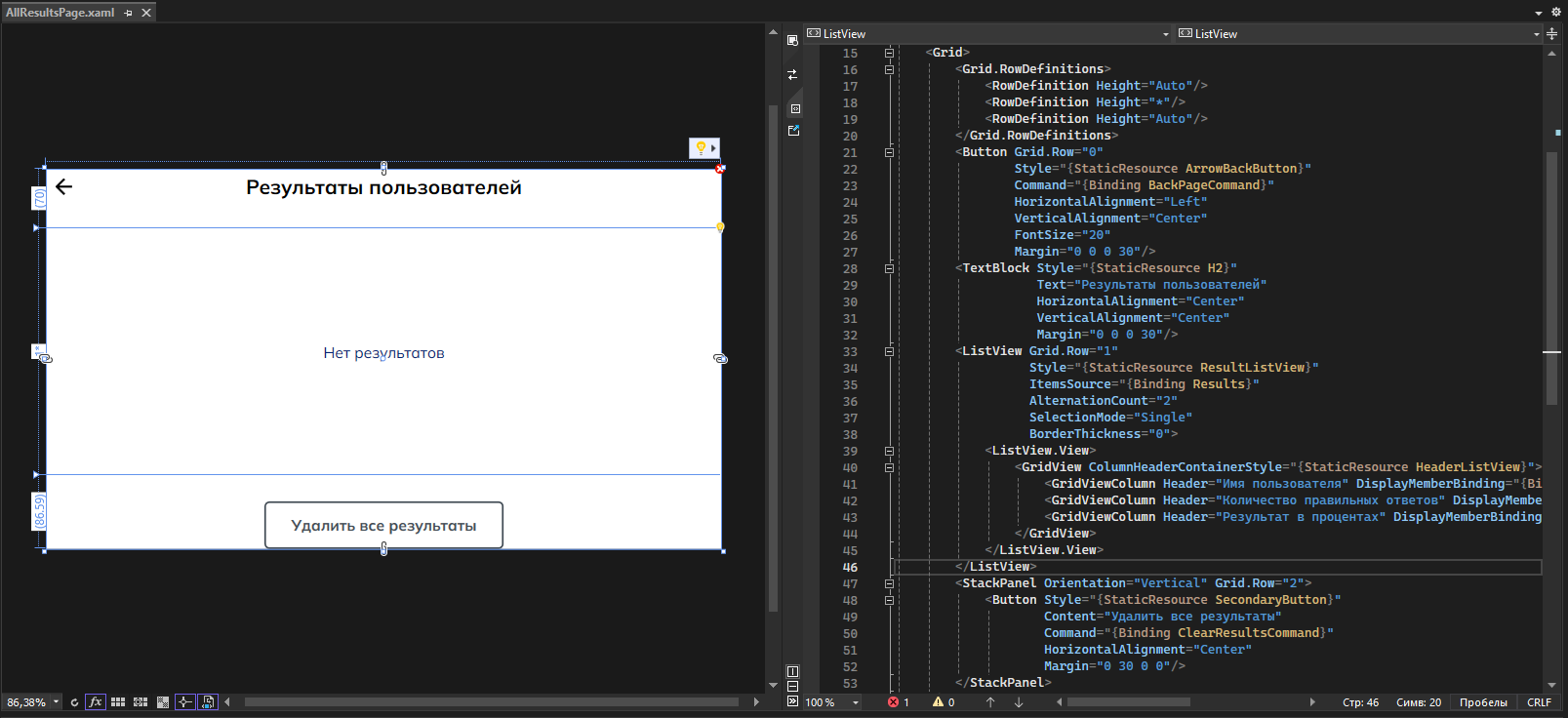


Рисунок 14. Интерфейс и разметка страницы со списком всех результатов

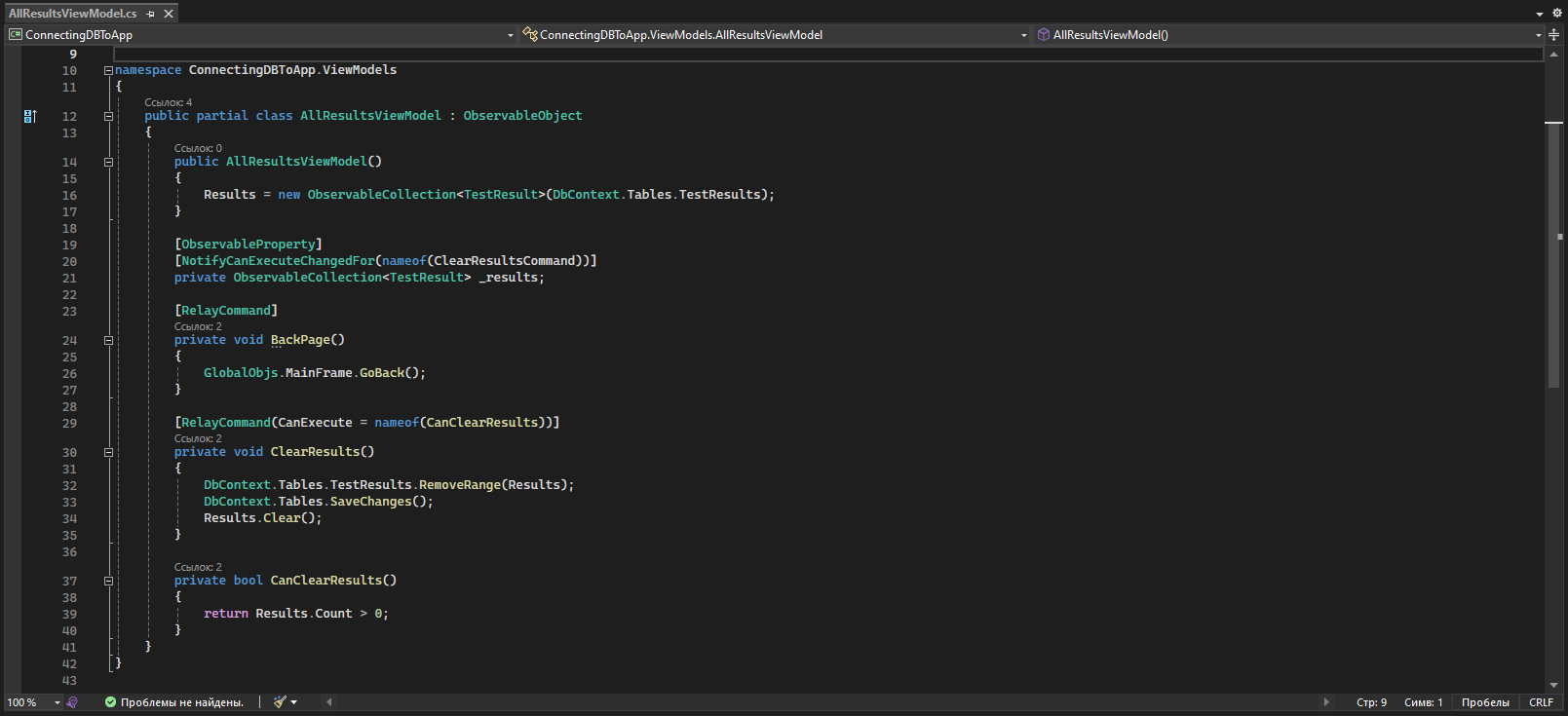
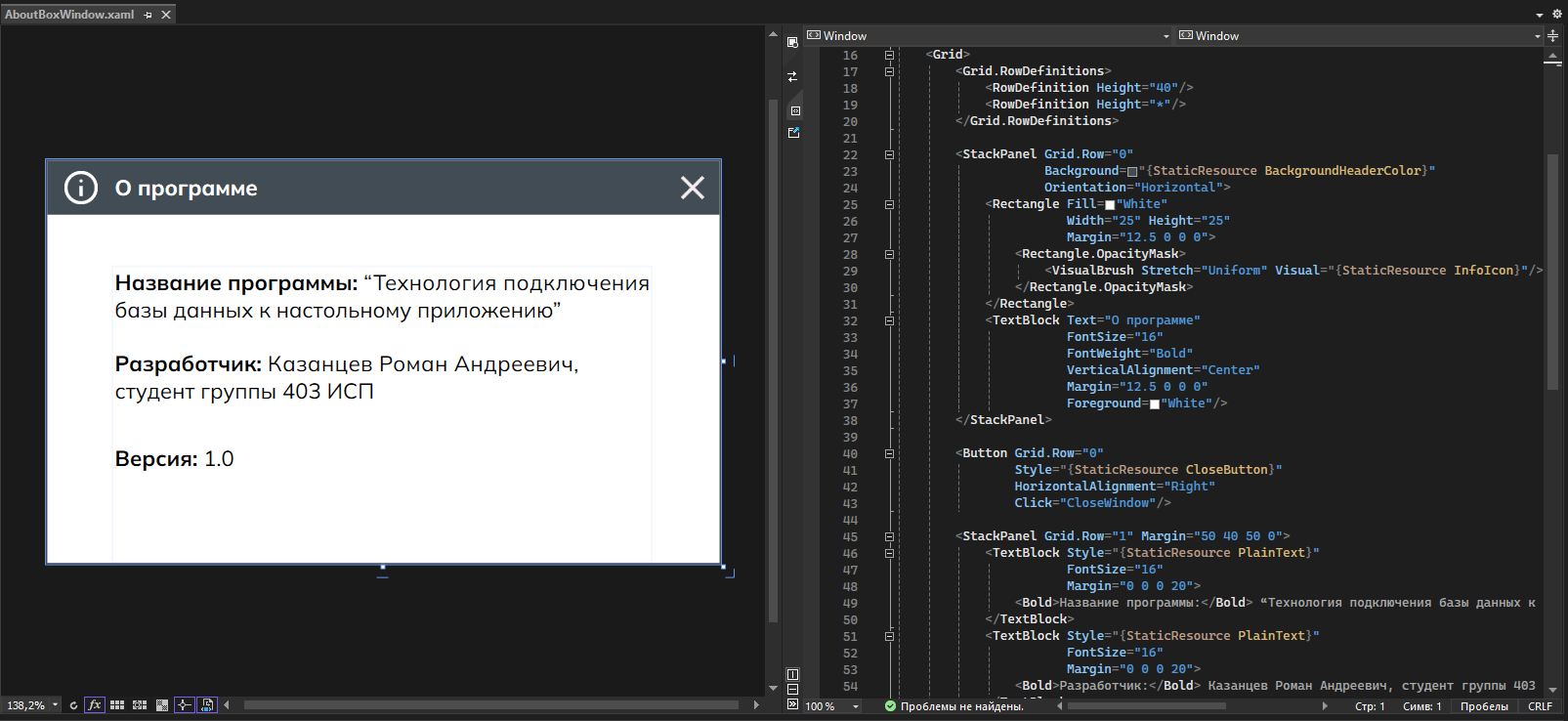


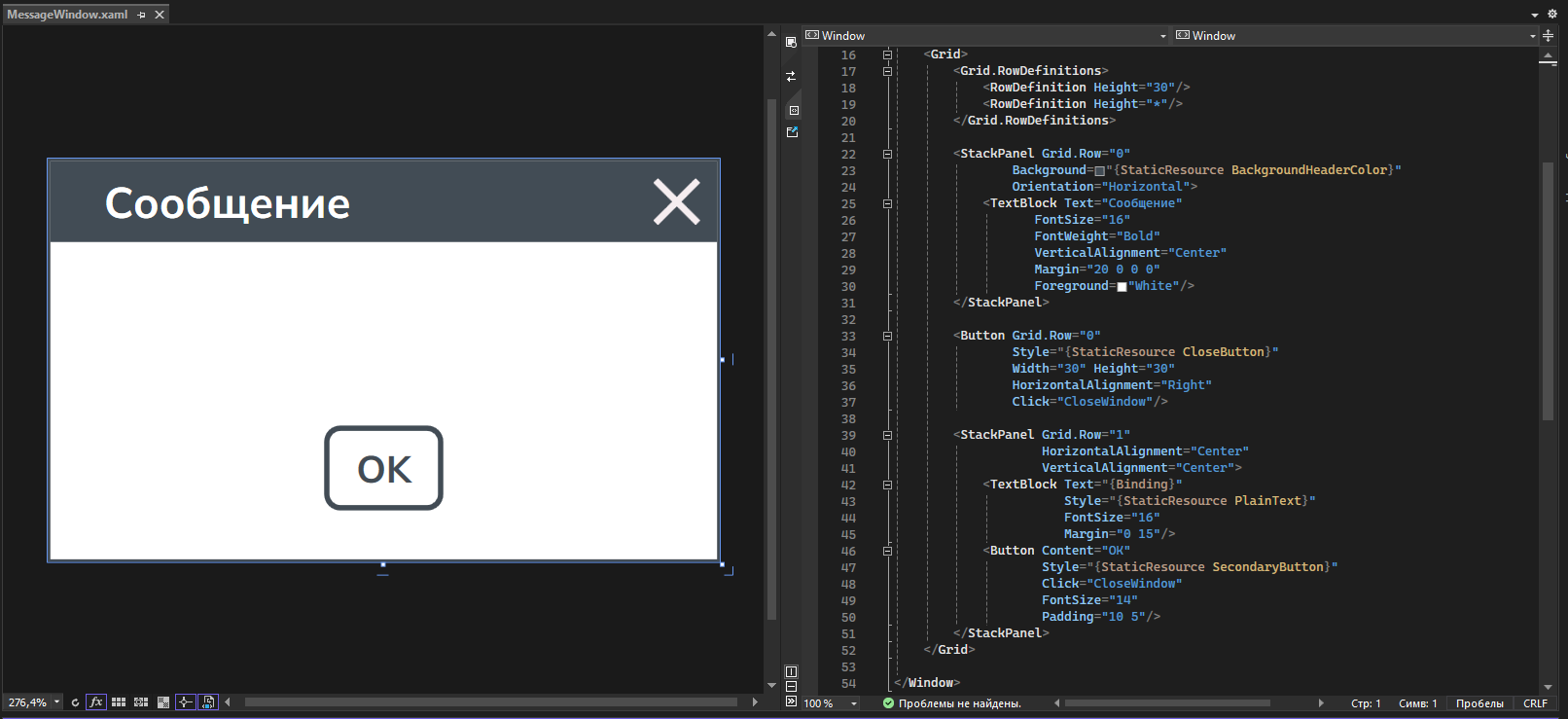
Рисунок 15. Класс с логикой страницы со списком всех результатов

Данная страница

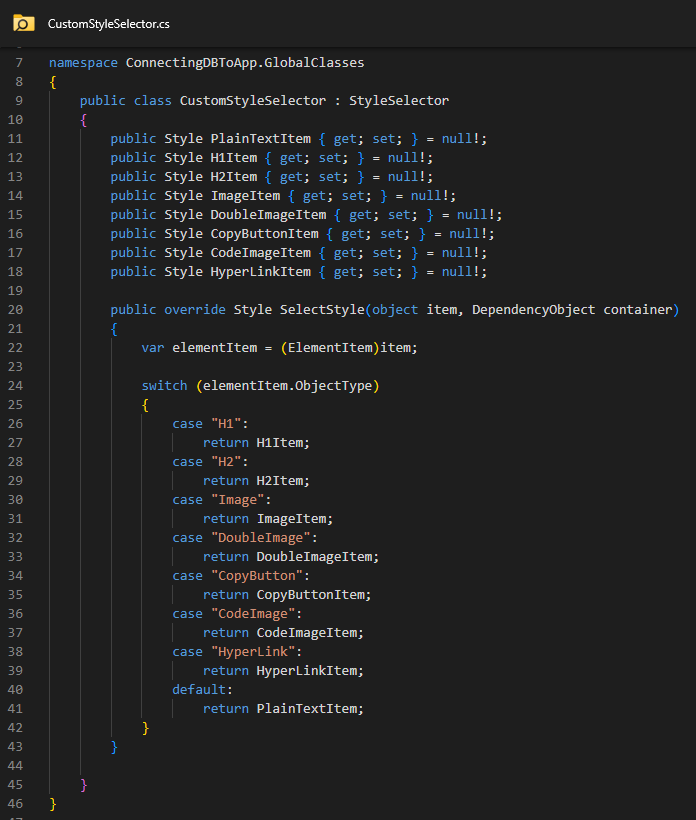
*Окно «О программе»*, состоящее только из текстовых блоков, отображает общие сведения о программе.

 Рисунок 15. Интерфейс и разметка окна «О программе»

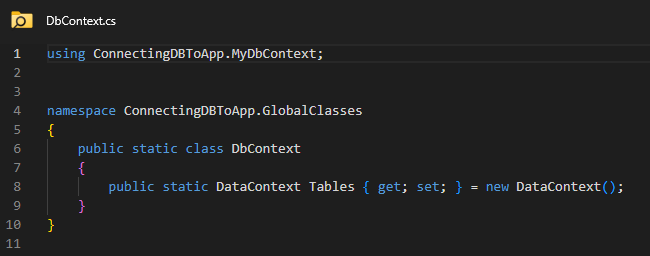
*Всплывающее окно с сообщением* предназначено для передачи какого-либо сообщения пользователю. Окно содержит текстовый блок с сообщением и кнопкой «ОК», закрывающей окно.

 Рисунок 16. Интерфейс и разметка всплывающего окна

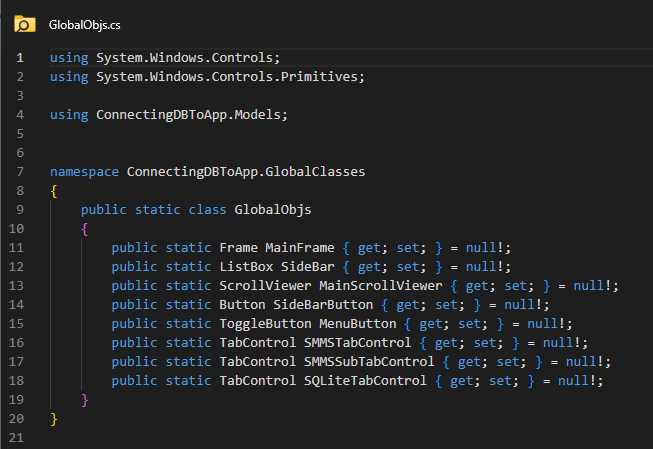
Класс CustomStyleSelector предназначен для выбора стиля, который нужно применить элементам на таких страницах как SSMS и SQLite.

Рисунок 17. Класс CustomStyleSelector

Класс DbContext предназначен для хранения контекста данных из базы данных. Этот класс объявлен глобально для того, чтобы работать с данными в любом классе, где это необходимо.

 Рисунок 18. Класс DbContext

Класс GlobalObjs предназначен для хранения пользовательских элементов глобально. Это сделано для того, чтобы редактировать и управлять их состоянием из любого места программы.

 Рисунок 19. Класс GlobalObjs

Класс MainWindowMethods предназначен для делегирования таких функций окна как свёртывание, открытие в оконном/полноэкранном режиме, закрытие.

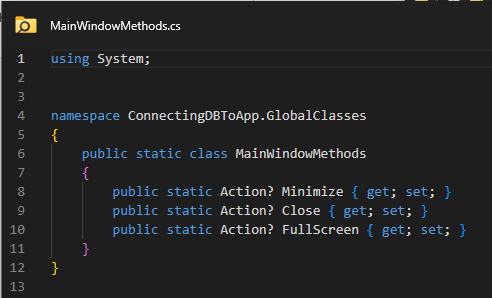


Рисунок 18. Класс MainWindowMethods

Классы GetTestResultMessage и GetUsernameMessage предназначены для создания сообщений, которые передаются между страницами.

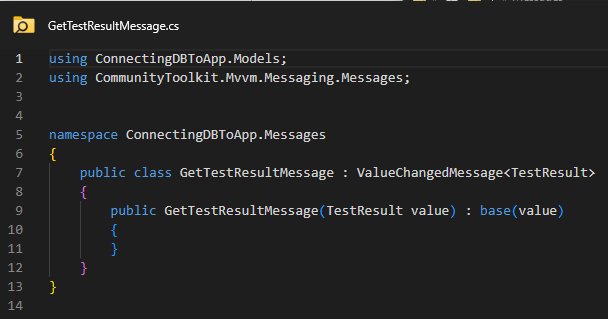


Рисунок 18. Класс GetTestResultMessage

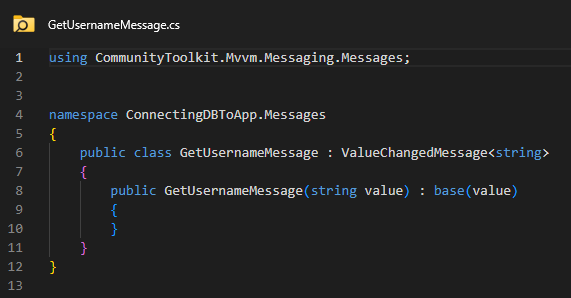


Рисунок 18. Класс GetUsernameMessage

Интерфейс ElementItem описывает данные элементов страниц SSMS и SQLite.

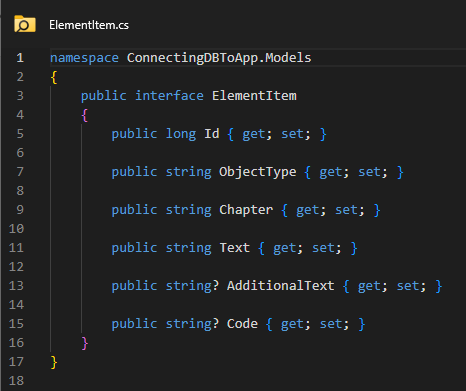
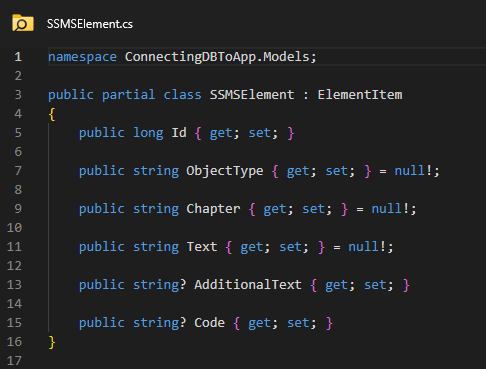
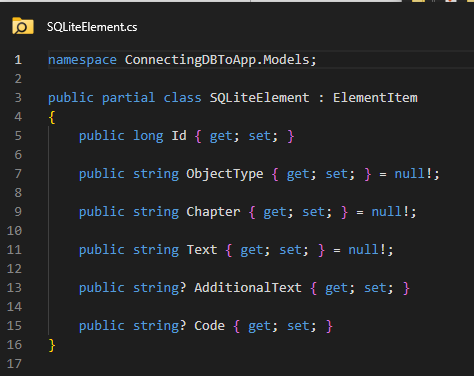


Рисунок 18. Класс ElementItem

Класс SQLiteElement и SSMSElement абсолютно идентичны и реализуют интерфейс ElementItem. Они описывают элементы соответствующих страниц из базы данных.

 Рисунок 18, 19. Классы SQLiteElement и SSMSElement

Класс TestQuestion описывает вопрос теста, который берётся из базы данных.

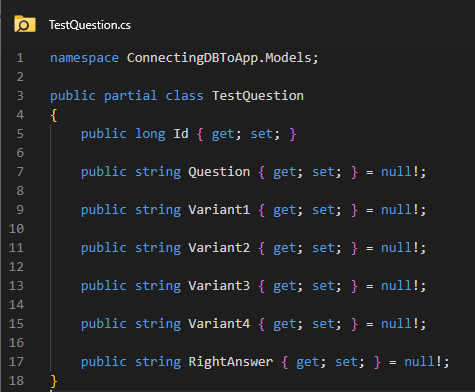


Рисунок 18. Класс TestQuestion

Класс TestResult описывает результат теста.

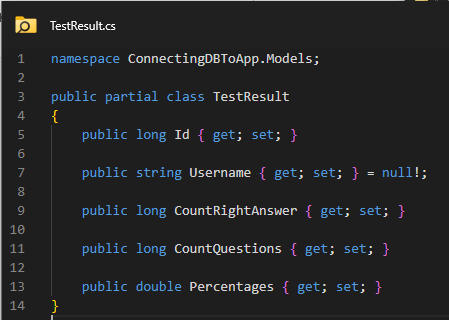
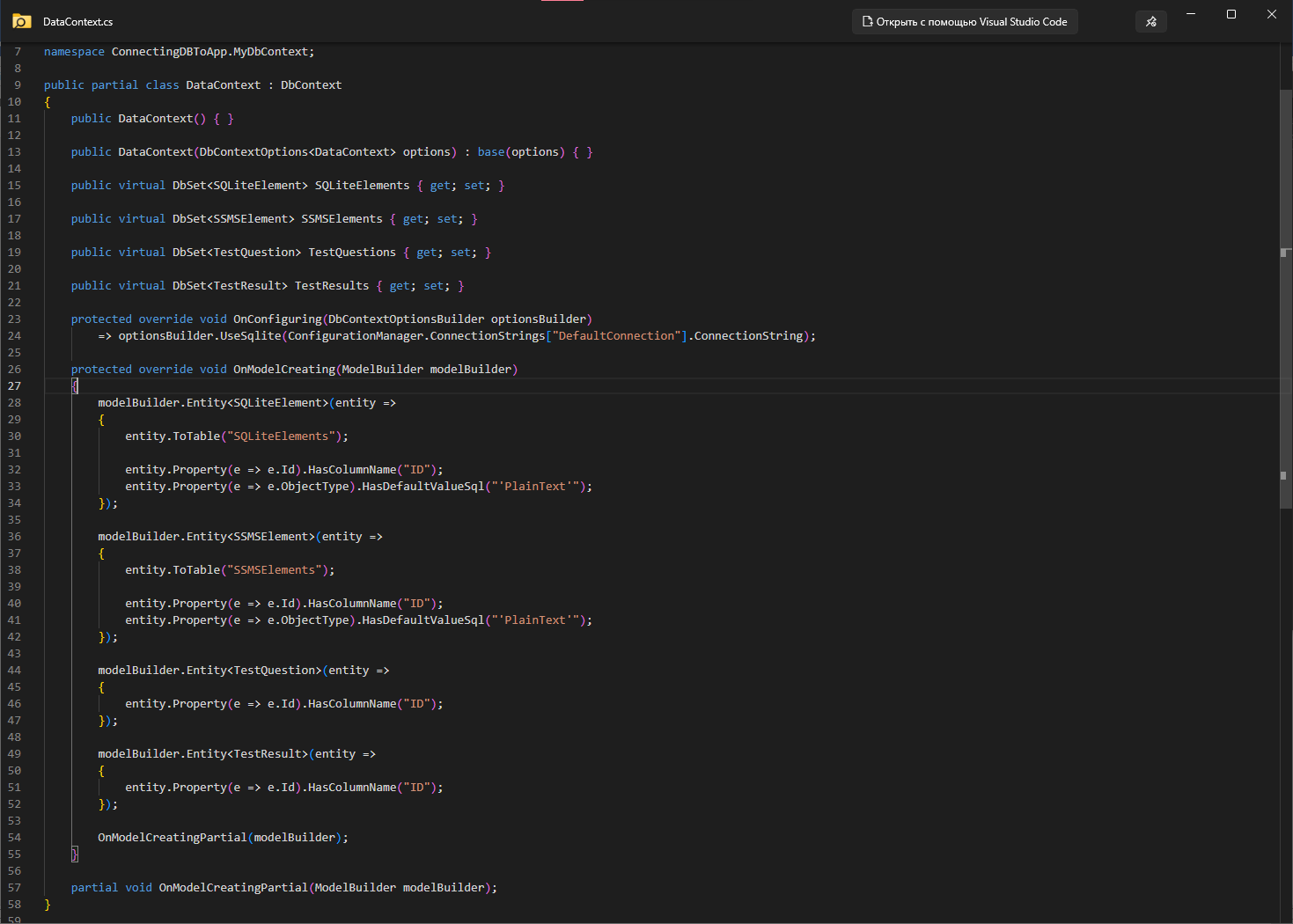


Рисунок 18. Класс TestResult

Класс DataContext предназначен для взаимодействия с базой данных. С помощью него реализуется связь с ней. Также с помощью него осуществляется добавление, редактирование и удаление данных.

 Рисунок 18. Класс DataContext