Министерство науки и высшего образования РФ
федеральное государственное автономное образовательное учреждение
высшего образования
«Омский государственный технический университет»

Лабораторная работа №1

По дисциплине:

Документирование программного обеспечения

Студент: Горшенин Л.И.
Группа ПИН-202
Руководитель:
ст. преподаватель Фатеева А.С.

Омск 2023

ВВЕДЕНИЕ

Задание:

1. Взять программный продукт, доступ к исходным кодам которого есть у исполнителя (курсовые работы, pet-project, open source).
2. Создать технологическую документацию к данному продукту, используя специальный формат комментариев к программному коду и задействовав средства автоматизированного формирования программной документации (javadoc, doxygen, и т.п.).
3. Обязательному комментированию подлежат компоненты программы (классы, методы, свойства и т.п.), которые будут доступны внешним программам (модификатор доступа public).

Ход работы

Для лабораторной работы был выбран pet-проект Vegetarian Community.
Было разработано классическое CRUD приложение «Вегетарианское сообщество» с использованием Windows Forms. Приложение представляет собой окно с обсуждениями на разные темы, в котором пользователи могут оставлять комментарии.

В качестве СУБД использовалась MS SQL Server. Вся информация хранится на локальном сервере в базе данных.

Для работы с базой данных использовался пакет Microsoft.Data.SqlClient.

Проект написанн с использований технологий :

- C#
- MS SQL Server
- Windows Forms

Функционал реализованный в проекте:

1. Был реализован класс, отвечающий за работу с постами. В классе реализованы методы для создания и переключения постов и последующее сохранение изменений в локальную базу данных.
2. Был реализован класс, отвечающий за работу с комментариями. В классе реализованы методы для создания, удаления и редактирования комментариев и последующее сохранение изменений в локальную базу данных.
3. Был реализован класс, отвечающий за работу с пользователями. В классе реализованы методы создания пользователей к определенному посту и сохранение изменений в локальную базу данных.
4. Был реализован класс, представляющий отображение приложения с возможность взаимодействия.

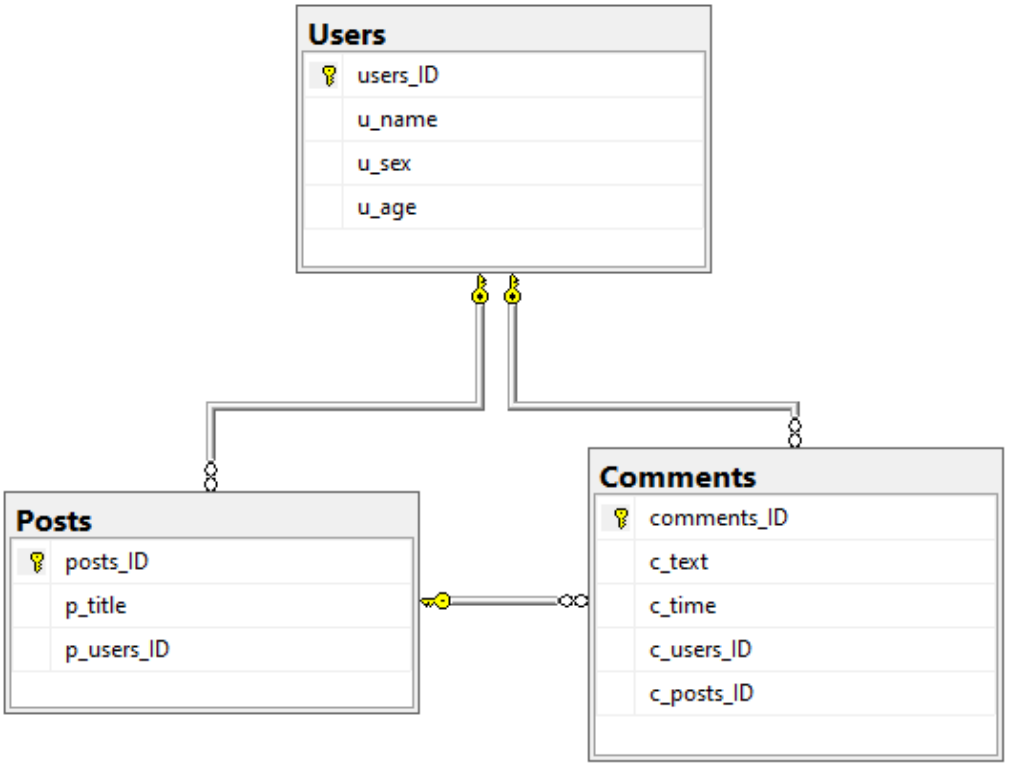

Рисунок 1 – Er-модель базы данных

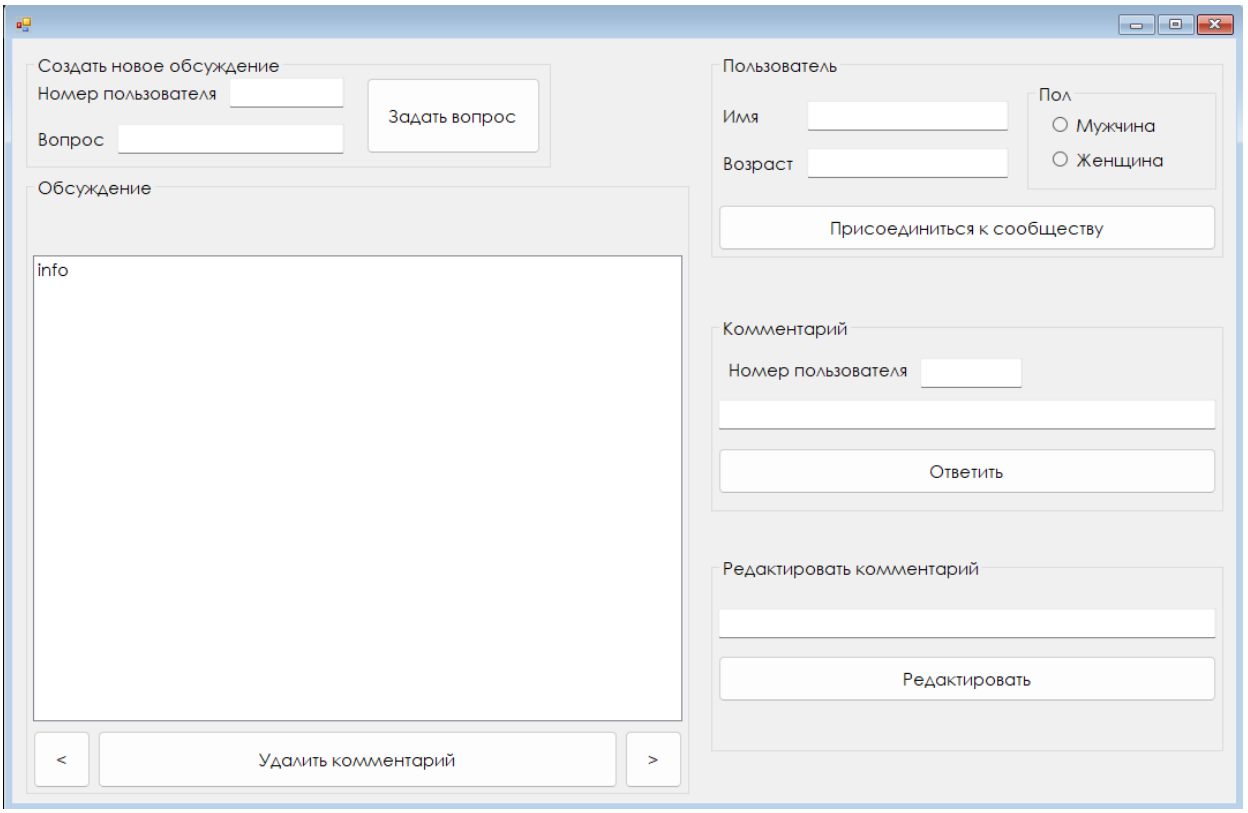

Рисунок 2 – Отображение приложения

Для написание документации на код использовался инструмент Summary C#.

```
Ссылок: 6
public sealed class Post
{
    /// <summary>
    /// Класс поста
    /// </summary>
    /// <param name="currentId">Уникальный идентификатор поста</param>
    /// <param name="text">Заголовок поста</param>
    /// <param name="userId">Уникальный идентификатор пользователя, создавшего текущий пост</param>
    Ссылок: 1
    public Post(int currentId, string text, int userId)
    {
        Id = currentId;
        Text = text;
        UserId = userId;
    }

    /// <summary>
    /// Уникальный идентификатор поста
    /// </summary>
    Ссылок: 2
    public int Id { get; }

    /// <summary>
    /// Заголовок поста
    /// </summary>
    Ссылок: 2
    public string Text { get; }

    /// <summary>
    /// Уникальный идентификатор пользователя, создавшего текущий пост
    /// </summary>
    Ссылок: 2
    public int UserId { get; }
}
```

Рисунок 3 – Документирование класса модели на примере класса Post

```
/// <summary>
/// Коллекция постов
/// </summary>
Ссылок: 1
public PostsCollection() { }

#region Properties

/// <summary>
/// Уникальный идентификатор текущего отображаемого поста
/// </summary>
Ссылок: 9
public int CurrentPost { get; private set; } = -1;

Ссылок: 2
private int MaxValue...

Ссылок: 2
private string Title...

#endregion

#region Public methods

/// <summary>
/// Создание экземпляра класса <see cref="Post"/> и добавление его в базу данных
/// </summary>
/// <param name="title">Заголовок поста</param>
/// <param name="userId">Уникальный идентификатор пользователя, создавшего текущий пост</param>
Ссылок: 1
public void CreatePost(string title, int userId)
{
    var post = new Post(MaxValue, title, userId);
    InsertPost(post);
}

/// <summary>
/// Переключение текущего поста на следующий/предыдущий
/// </summary>
/// <param name="isNextDirection">Задает направление какой пост будет выбран. По умолчанию предыдущий</param>
Ссылок: 2
public void Move(bool isNextDirection = false)
{
    if(isNextDirection && CurrentPost < MaxValue - 1)
    {
        CurrentPost++;
        OnShowPost?.Invoke(Title);
    }
    else if(isNextDirection == false && CurrentPost > 0)
```

Рисунок 4 – Документирование класса и его методов на примере класса PostCollection

```
/// <summary>
/// Основное приложение, в котором происходит взаимодействие компонентов
/// </summary>
/// <param name="postsCollection">Коллекция написанных постов</param>
/// <param name="commentsCollection">Коллекция написанных комментариев</param>
/// <param name="title">Label заголовка определенного поста</param>
/// <param name="box">ListBox истории сообщений определенного поста</param>
Ссылок: 1
public App(PostsCollection postsCollection, CommentsCollection commentsCollection, Label title, ListBox box)
{
    _postCollection = postsCollection;
    _commentsCollection = commentsCollection;
    _title = title;
    _box = box;


    _postCollection.OnShowPost += ShowPost;
    _commentsCollection.OnShowComments += ShowComments;
}
```

Рисунок 5 – Документирование класса

```
/// <summary>
/// Создание экземпляра класса <see cref="User"/> и добавление его в базу данных
/// </summary>
/// <param name="name">Имя пользователя</param>
/// <param name="currentSex">Пол пользователя</param>
/// <param name="age">Возраст пользователя</param>
Ссылок: 1
public void CreateUser(string name, string currentSex, int age)
{
    var user = new User(Count, name, currentSex, age);
    InsertUser(user);
}
```

Рисунок 6 – Документирование метода

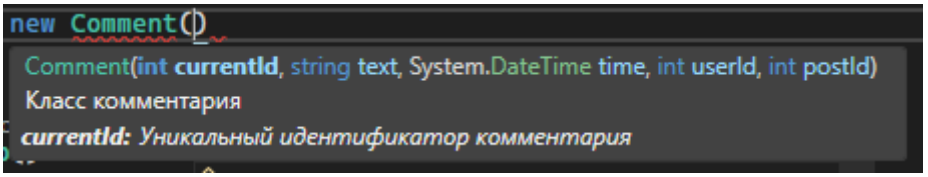

Рисунок 7 – Подсказки редактора после написания документации

ЗАКЛЮЧЕНИЕ

В результате выполнения лабораторной работы была написана документация на проект, в следствие чего улучшилась читаемость кода, также был изучен инструмент Summary C# и получен практический навык работы с ним.

СПИСОК ИСТОЧНИКОВ

1. Документация Swagger – Режим доступа: https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/documentation-comments
2. Задокументированный код Режим доступа: https://github.com/len6q/database-with-winforms

ПРИЛОЖЕНИЕ

Код с документацией : https://github.com/len6q/database-with-winforms