МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ПИ

Факультет КНТ

Отчет по лабораторной работе №4

Тема: «Создание самодокументирующегося кода».

Курс: «Профессиональная практика программной инженерии»

Выполнил

ст. гр. ПИ-18в

Настеко И.Р.

Донецк – 2022

Примеры докблоков

/// <summary>

/// Модель администратора

/// </summary>

public class Admin : IUser

{

/// <summary>

/// Идентификационный номер

/// </summary>

public Guid Id { get; }

/// <summary>

/// Имя

/// </summary>

public string Name { get; set; }

/// <summary>

/// Логин

/// </summary>

public string Login { get; set; }

/// <summary>

/// Пароль

/// </summary>

public string Password { get; set; }

/// <summary>

/// Электронная почта

/// </summary>

public string Email { get; set; }

/// <summary>

/// Права

/// </summary>

public List<UserRightsEnum> Rights { get; set; }

}

/// <summary>

/// Интерфейс сервиса для работы с сообщениями

/// </summary>

public interface IMessageService

{

/// <summary>

/// Функция отправки сообщения в ветку

/// </summary>

void SendMessage(Guid branchId, Message message);

/// <summary>

/// Функция удаления сообщения по Id

/// </summary>

void DeleteMessage(Guid messageId);

/// <summary>

/// Функция редактирования сообщения

/// </summary>

void EditMessage(Guid messageId, string messageText);

}

/// <summary>

/// Интерфейс сервиса для работы с ветками обсуждений

/// </summary>

public interface IBranchService

{

/// <summary>

/// Создать ветку

/// </summary>

void CreateBranch(string branchName);

/// <summary>

/// Удалить ветку

/// </summary>

void DeleteBranch(Guid branchId);

/// <summary>

/// Переименовать ветку

/// </summary>

void RenameBranch(Guid branchId, string newName);

}

/// <summary>

/// Интерфейс сервиса для работы с панелью админа

/// </summary>

internal interface IAdminService

{

/// <summary>

/// Удаление пользователя по Id

/// </summary>

bool DeleteUser(Guid userId);

/// <summary>

/// Назачение пользователю прав модератора

/// </summary>

bool AppointModerator(Guid userId);

/// <summary>

/// Назначение пользователю определённых прав

/// </summary>

void SetRights(Guid userId, UserRightsEnum[] rights);

}

Содержимое файла DocBook

<?xml version="1.0" encoding="UTF-8"?>

<article xmlns="http://docbook.org/ns/docbook" version="5.0"

xmlns:xlink="http://www.w3.org/1999/xlink">

<title>Docbook Настеко И. ПИ-18В</title>

<section>

<title>Краткое описание разработанного продукта</title>

<para>Данная документация написана для проекта "Форум с ветками-обсуждениями, с обязательной реализациейгрупповых ролей: гость, посетитель, модератор, администратор."</para>

<para>Разработанный продукт может быть использован в различных целях. Например как площадка для обсуждения программирования, решения проблем и помощи друг другу. Или как торговая площадка, где люди могут продавать, покупать, обмениватся различными вещами</para>

<para>Реализована минимальная часть для функционирования форума, а именно: регистрация и авторизация пользователя, панель администратора, создания веток обсуждения и отправка сообщений</para>

<para>Написаны Unit тесты для правильной работы программы, а также выявлены все возможные ошибки</para>

<para>Проект написан на языке C# с использованием фреймворка ASP.NET Core MVC</para>

</section>

<section>

<title>Сценарии использования приложения</title>

<para><emphasis role="bold">Сценарий 1. Работа с ветками</emphasis></para>

<para>Пользователь создаёт ветку функцией CreateBranch <programlisting>

public void CreateBranch(string branchName)

{

//INSERT BranchTable branchName

}</programlisting></para>

<para>Затем может переименовать её функцией RenameBranch <programlisting>

public void RenameBranch(Guid branchId, string newName)

{

//UPDATE BranchTable SET Name=newName WHERE Id=branchId

}</programlisting></para>

<para>И удалить DeleteBranch <programlisting>

public void DeleteBranch(Guid branchId)

{

//DELETE BranchTable WHERE Id=branchId

}</programlisting></para>

<para><emphasis role="bold">Сценарий 2. Работа с сообщениями</emphasis></para>

<para>Пользователь создаёт ветку функцией CreateBranch <programlisting>

public void CreateBranch(string branchName)

{

//INSERT BranchTable branchName

}</programlisting></para>

<para>Затем может отправить сообщение в ветку функцией SendMessage<programlisting>

public void SendMessage(Guid branchId, Message message)

{

//INSERT MessageTable message.MessageText,message.UserId,message.CreatedDate,branchId

}</programlisting></para>

<para>После отправки можно редактировать сообщение, если было отправлено с ошибкой<programlisting>

public void EditMessage(Guid messageId, string messageText)

{

//UPDATE MessageTeble SET MessageText=messageText WHERE Id=messageId

}</programlisting></para>

<para>Или удалить при необходимости<programlisting>

public void DeleteMessage(Guid messageId)

{

//DELETE MessageTable WHERE Id=messageId

}</programlisting></para>

<para><emphasis role="bold">Сценарий 3. Работа с пользователями</emphasis></para>

<para>Пользователь зайдя под гостем, может зарегистрироваться на форуме, что бы стать посетителем<programlisting>

public bool ResgisterNewUser(IUser user)

{

if(true)

return true;

}</programlisting></para>

<para>Затем ему нужно авторизоваться, для чего сначала вызывается функция проверки на существование пользователя в базе данных<programlisting>

public bool CheckIsUserExist(Guid userID)

{

return true;

}</programlisting></para>

<para>После чего происходит авторизация<programlisting>

public bool AuthorizeUser(string userName, string password)

{

return true;

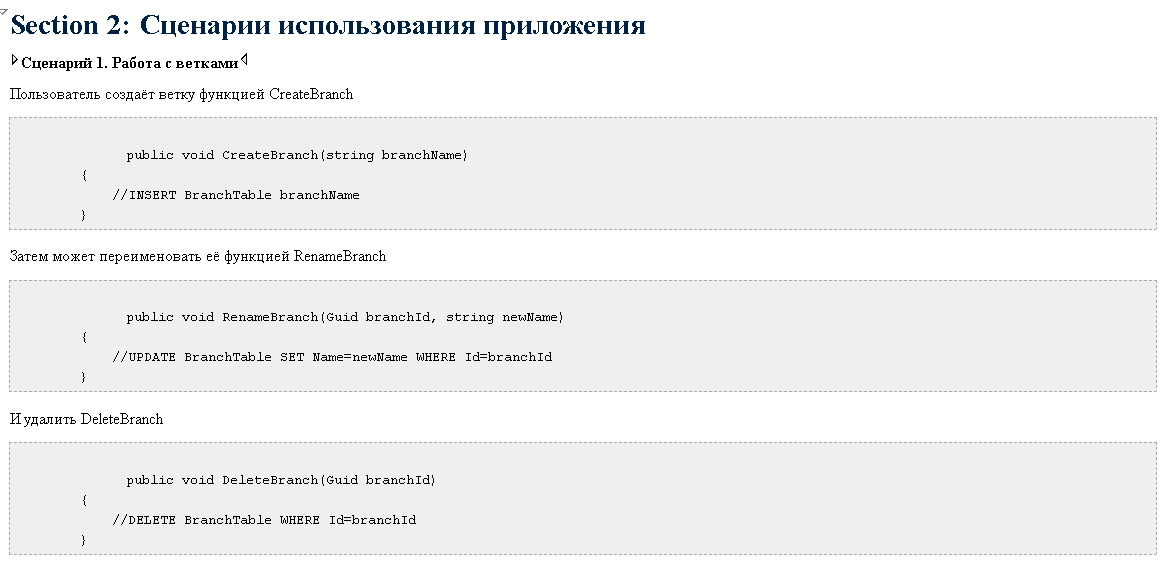
}</programlisting></para>

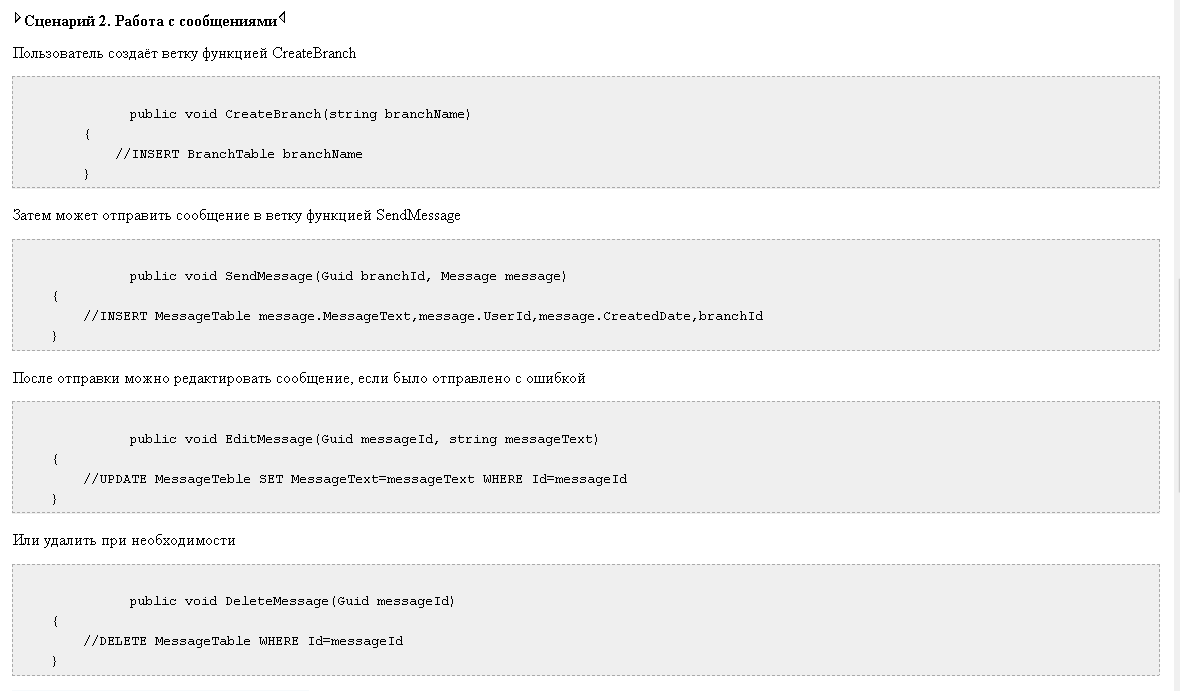
</section>

</article>

Просмотр ДокБука в oXygen XML Editor









Валидация



