**Тесты**

1)Тесты для класса DbService, который предоставляет доступ к базе данных. Используется библиотека NUnit для написания тестов и библиотека Moq для создания мок-объектов.

Тесты проверяют различные методы класса DbService, такие как ExistsUser, IsBannedUser, GetHashPassword и GetDataUserByLogin. Каждый тест имеет три части: Arrange (настройка), Act (действие) и Assert (утверждение).

В разделе Arrange создаются мок-объекты и настраивается их поведение с помощью методов Setup. Это позволяет имитировать работу реального объекта и определить, что он должен возвращать в ответ на определенный вызов метода.

В разделе Act вызывается тестируемый метод и сохраняется его результат.

В разделе Assert проверяется, что результат тестируемого метода соответствует ожидаемому результату.

Эти тесты позволяют убедиться в правильности работы класса DbService и его методов. Они также обеспечивают защиту от ошибок при изменении кода класса в будущем.

2)Тесты для класса DbSettingsService предназначен для работы с файлом настроек базы данных. Он содержит методы для проверки наличия файла настроек, получения настроек из файла, сохранения настроек в файл и удаления файла настроек.

Метод DbSettingsFileExists() проверяет, существует ли файл настроек базы данных, и возвращает true, если файл существует, и false, если файл не существует.

Метод GetDbSettings() получает настройки базы данных из файла и возвращает объект типа DbSettings.

Метод SaveDbSettings(dbSettings) сохраняет настройки базы данных в файл.

Метод RemoveDbSettings() удаляет файл настроек базы данных.

Для тестирования класса DbSettingsService определены два теста:

1. DbSettingsFileExistsWhenFileDoesNotExistReturnsFalse() - проверяет метод DbSettingsFileExists на корректность возвращаемого значения при отсутствии файла настроек базы данных.

2. RemoveDbSettingsWhenFileDoesNotExistDoesNothing() - проверяет метод RemoveDbSettings на корректность работы при отсутствии файла настроек базы данных.

3)Класс UserTests содержит тесты для проверки корректности работы конструктора класса User. Тест ConstructorWithValidArgumentsSetsProperties() проверяет, что при передаче в конструктор корректных аргументов объект User создается с корректными свойствами. В качестве аргументов конструктору передаются значения для свойств объекта User, такие как роль пользователя, логин, пароль, имя, фамилия, дата рождения и статус блокировки. После создания объекта User проверяется, что все его свойства соответствуют переданным значениям.

В этом тесте создаются переменные для каждого из свойств User, затем объект User создается с использованием этих переменных и наконец проверяется , что каждое свойство объекта User установлено на правильные значения с помощью метода Assert.AreEqual().Если все свойства установлены правильно, тест пройдет успешно, иначе завершится неудачей.