Library

Оглавление

[To do 2](#_Toc97026125)

[Содержание 2](#_Toc97026126)

[Добавить в библиотеку из существующего 2](#_Toc97026127)

[Долгий ящик 2](#_Toc97026128)

[Иметь в виду 2](#_Toc97026129)

[Цели 2](#_Toc97026130)

[Структура 3](#_Toc97026131)

[Типы библиотек 3](#_Toc97026132)

[WPF и Windows Forms 3](#_Toc97026133)

[UWP 3](#_Toc97026134)

[Язык 3](#_Toc97026135)

[Решение 3](#_Toc97026136)

[Использование 3](#_Toc97026137)

# To do

* Сделать структуру решения
* Отработать тестирование
* Отработать подключение к другим решениям
* Отправить проект на GitHub
* В UWP сделать Navigation панель для разных классов

## Содержание

* Свести все классы работы с базами данных в одну библиотеку
  + Продумать её возможности и структуру
  + Получение структуры базы данных. Таблицы, поля, хранимые процедуры.
* Отличия баз данных MS SQL Server и SQLite
  + в SQLite похоже не работает N'' для русского языка. похоже требуется просто ''.
  + формат даты отличается от MS SQL Server
  + в MS SQL Server «SCOPE\_IDENTITY() » в SQLite «last\_insert\_rowid()»
* Сделать метод получения текущего времени
  + Использовать TimeType перечисление
  + С использованием TimeType отработать запись и чтение даты и времени в базах данных. И отразить это в нахваниях TimeType
    - MS SQL Server
    - SQLite
* Менеджер управления базой данных, либо настройка доступа UWP приложения
* Сделать Log систему
  + На основе баз данных
  + Системных сообщений

## Добавить в библиотеку из существующего

* Описание подключения к базе данных SQLite
* Из сборки 2\_assemble классы работы с базой данных
* Парсер html
  + C:\Users\e.yumagulov\Desktop\проекты\программа перевода\отработка\browser
  + Плюс посмотреть, что ещё есть в
    - C:\Users\e.yumagulov\Desktop\проекты\программа перевода\отработка\
* Работа с word
  + Парсер html

## Долгий ящик

* Добавлять данные через параметры. Там преимущество в предотвращении «SQL injection attacks»
  + command.Parameters.AddWithValue("@Entry", inputText);

# Иметь в виду

* Если есть возможность использовать не методы, а свойства. Поскольку при вызове удобнее не писать скобки.
  + Сейчас нет авто подстановки скобок при вызове метода. Их приходится дописывать самому.

# Цели

Облегчение жизни, экономия времени, унификация, порядок, получение удовольствия.

Библиотека на практике делается для трёх типов проектов

* Windows Forms
* WPF
* UWP
* (возможно для сервисов)
* (возможно для Linux)

# Структура

## Типы библиотек

### WPF и Windows Forms

Для библиотеки следует выбирать «netstandard2.1» тогда её можно применять для большего количества проектов. Хотя в моём случае их будет только 2: WPF и Windows Forms.

<TargetFramework>netstandard2.1</TargetFramework>

### UWP

Для UWP следует выбирать отдельный тип проекта библиотеки – именно для UWP.

Минимальная версия проекта предназначена для обновления Fall Creators Update

Существует ряд преимуществ повышения минимальной версии проекта UWP для обновления Fall Creators Update.

В первую очередь можно использовать библиотеки .NET Standard 2.0 вместо стандартных библиотек классов. Это означает, что вы можете передать свой код доступа к данным любому другому приложению на основе .NET. Например, приложениям для WPF, Windows Forms, Android, iOS или ASP.NET.

## Язык

Язык следует выбирать последний или 10-й на данный момент.

<LangVersion>latest</LangVersion>

## Решение

Поэтому в одном решении следует иметь 7 проектов:

* Библиотека для WPF и Windows Forms
* Тест библиотеки для WPF и Windows Forms
* Библиотека для UWP
* Тест библиотеки для UWP
* Проект Windows Forms
* Проект WPF
* Проект UWP

# Использование

Библиотеку иметь только в одном экземпляре.

Когда её требуется подключить к другим решениям – ссылаться на это решение.

Периодически сохранять всё на GitHub.