

Курсовая по ООП - ПО0901

Инструкции для разработчиков - часть II

Составил: Додев П.Н.

Версия: 0.3

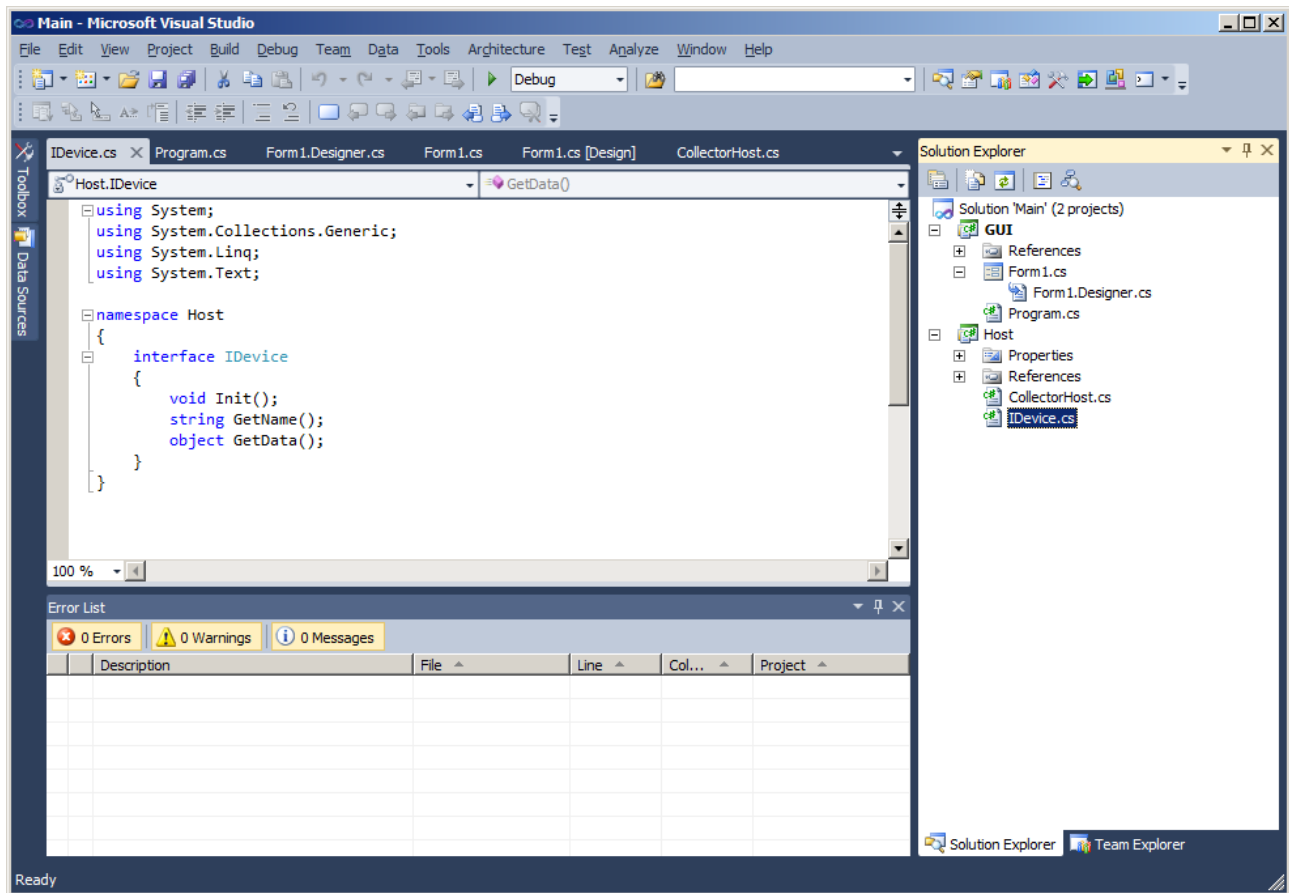
Это вторая часть инструкции для разработки. Здесь я полагаю что первая часть прочитана и вы разбираетесь в git — у вас он уже установлен, вы зарегистрировались на github.com, и базовые настройки как имя и эл. почта уже ввели.

Оглавление

Структура проекта в MS Visual Studio.....	2
Разработка проекта.....	3
Прежде чем начать работу.....	3
Процесс разработки.....	3
Сохраняем изменения в хранилище.....	4
Загружаем наши изменения в общее хранилище.....	5
Слияние локальной ветви с удалённой	6
Как создать свою библиотеку.....	6

Структура проекта в MS Visual Studio

Основное приложение проекта находится в папке **RoadDataCollector/Main**. Внутри есть файл **Main.sln**. Открываем его в MS Visual Studio.



Структура сравнительно простая: есть решение Main и в нем входят много проектов. Проектов представляют все задания нашей курсовой.

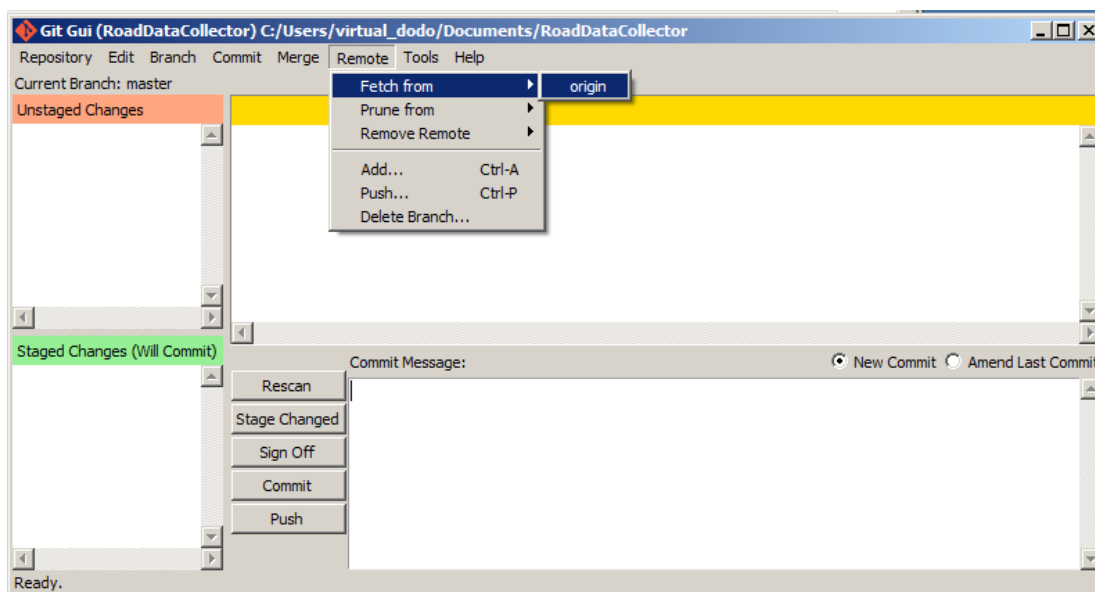
Если ваше задание уже существует под формой проекта(заданной старосты), то можете начинать разработки. Если нет, то от вас требуется создать проект в решение Main(См. Глава «Как создать свою библиотеку»)

Разработка проекта

Прежде чем начать работу

Прежде чем начать работать, обновляем состояние проекта, чтобы у нас была *свежая* версия проекта. (Это нужно делать каждый раз когда начинаете делать курсовой, чтобы всегда работали над последую версию)

Открываем **Git Gui** для **RoadDataCollector**. В меню выбираем **Remote** → **Fetch from** → **origin**:



Программа потребует логин и пароль от гитхаба.

Если все прошло хорошо, зелёный надпись загорелся, это означает что у нас последняя версия проекта и мы можем приступить к работе.

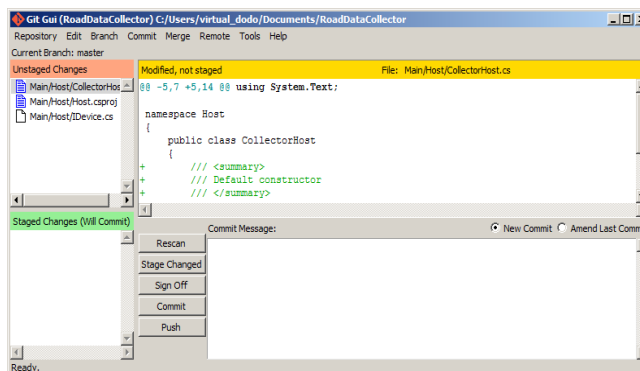
Процесс разработки

Делаем то что учились последние нескольких курсов. Добавляем проекты в Visual Studio; добавляем файлы; редактируем другие; удаляем файлы. т.е. все как обычно.

Сохраняем изменения в хранилище

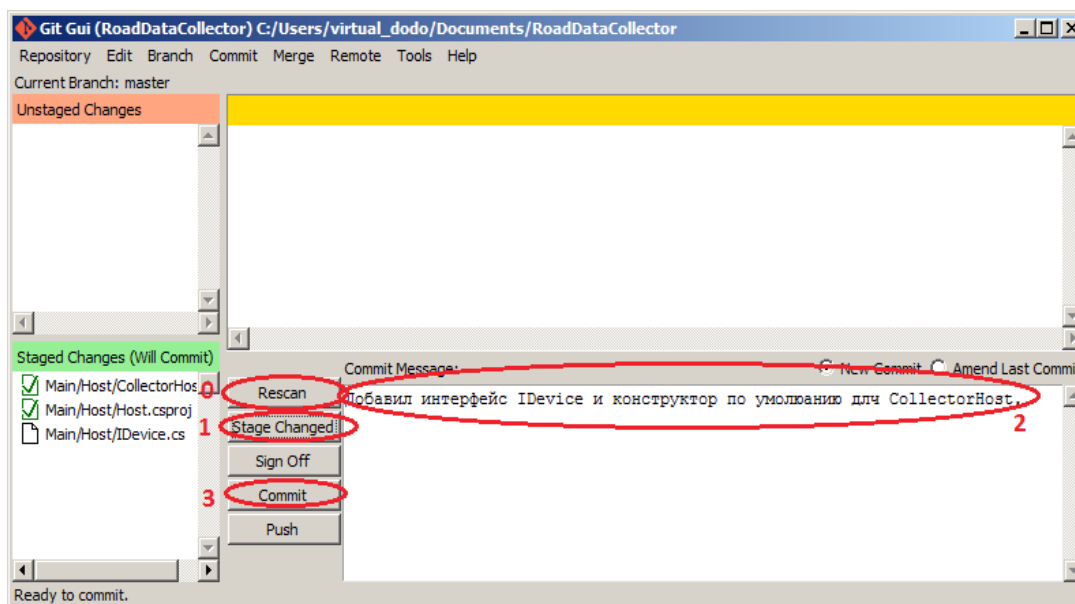
Когда завершили с разработкой какого-то модуля сохраняем работу.

Чтобы сохранить работу в локальном хранилище открываем Git Gui в директории RoadDataCollector. В моем примере я создал 1 файл и отредактировал 2 других. У меня окно выглядело вот так:



Сейчас необходимо добавить изменения в локальном хранилище:

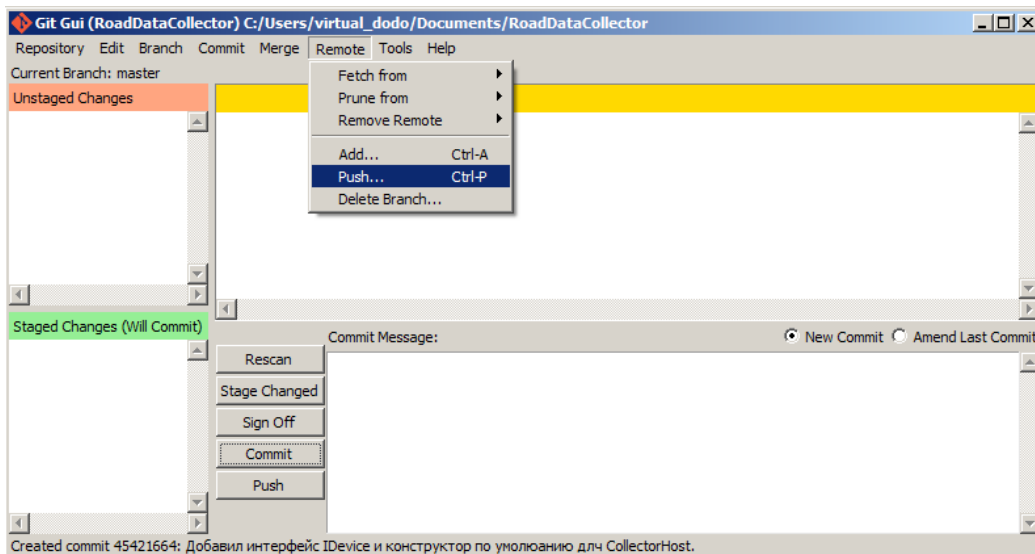
0. Нажимаем кнопку **Rescan**, чтобы убедиться что все изменения отображены в Git Gui;
1. Нажимаем **Stage Changed**, это подготавливает все изменения к сохранению. Если есть новые файлы, тогда программа спросит «Хотите ли добавить новых файлов в список с изменениями». Отвечаем с «Йес».
2. Вводим короткое описание наших изменений в поле **Commit Message**. 1-3 предложения.
3. Нажимаем кнопку **Commit**.



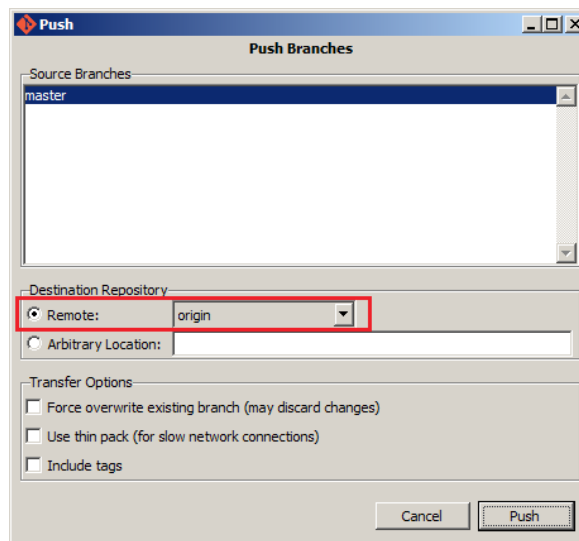
Таким образом вы сохранили изменения в вашем локальном хранилище. Рекомендую каждый раз когда завершите часть вашего задания, сохранять изменения. Каждое такое изменение называется **Коммитом**.

Загружаем наши изменения в общее хранилище

Чтобы поделиться изменением открываем Git Gui и выбираем из меню **Remote** → **Push...** (на русском: **Отправить**)



В диалоговом окне пункт **Remote** должен быть выбран и его значение должно быть **origin**.

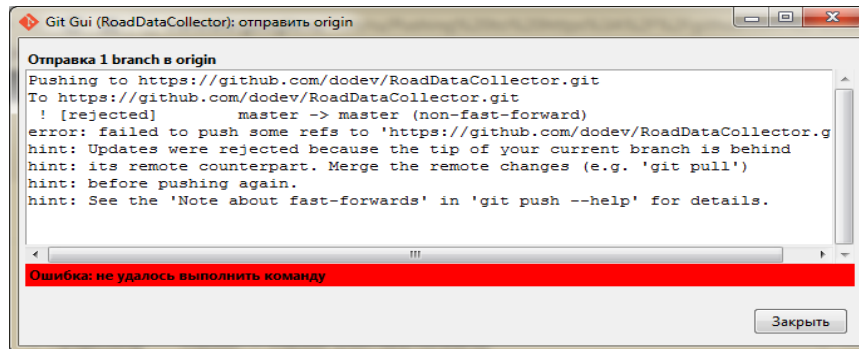


Если все в порядке, нажимаем кнопку **Push**. Программа потребует пароль ключа (та которая мы задали при генерации ключей в предыдущем главе).

Если надпись загорелся зелёным цветом, все прошло нормально и наши изменения уже находятся в гитхабе.

Слияние локальной ветви с удалённой

Если ваши изменения не загружаются нормально при **Remote > Push (Отправить)**:



Тогда вам придётся сделать следующие шаги:

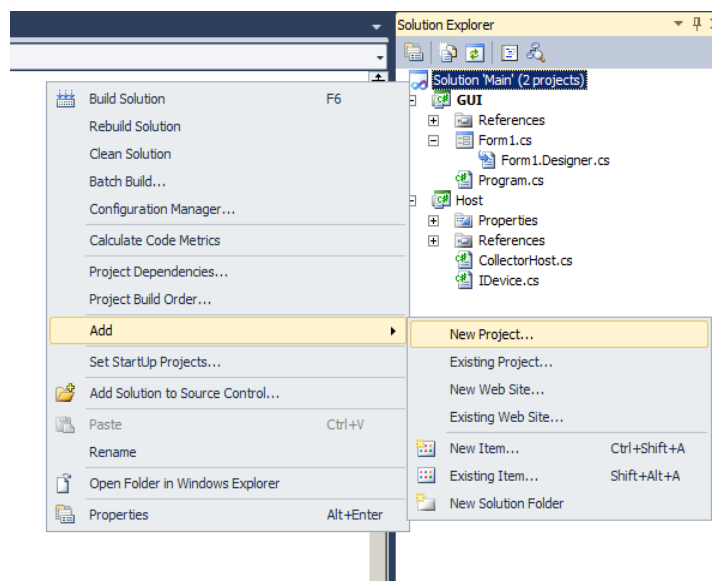
- 0) Закрываем все открытые файлы из репозитории;
- 1) В **Git Gui** нажимаем **Ctrl+M (Branch >> Local merge)**;
- 2) Выбираем **Tracking branches (~Отслеживание ветви)** и там **origin/master**;
- 3) Нажимаем **Merge (Слияние)**
- 4) Пробуем загрузить изменения заново (**Remote > Push**) :)

Как создать свою библиотеку

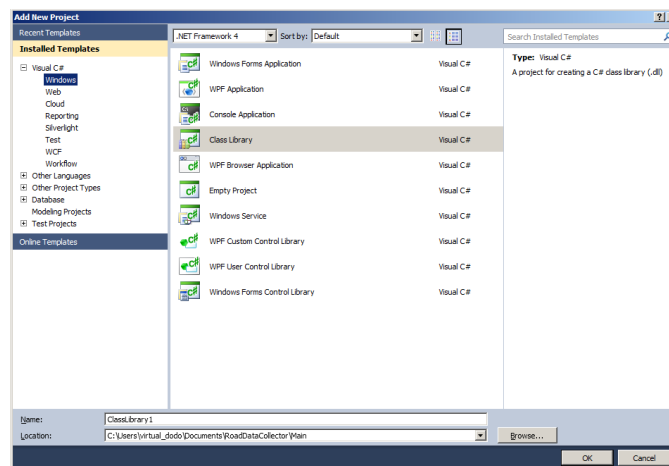
Если ваше задание не существует как отдельный проект, то от вас требуется создать подключаемую библиотеку для работы в проекте.

На примере MS Visual Studio 2010:

0. Открываем наш проект и в колонке Solution Explorer нажимаем правую кнопку и выбираем **Add → New Project...**

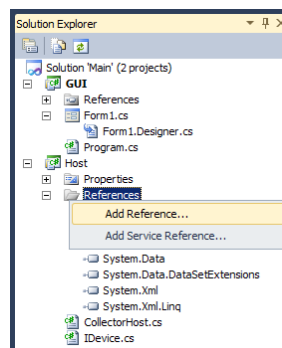


1. Выбираем тип проекта **Class Library**. Вводим адекватное имя библиотеки. **Location** не трогаем.

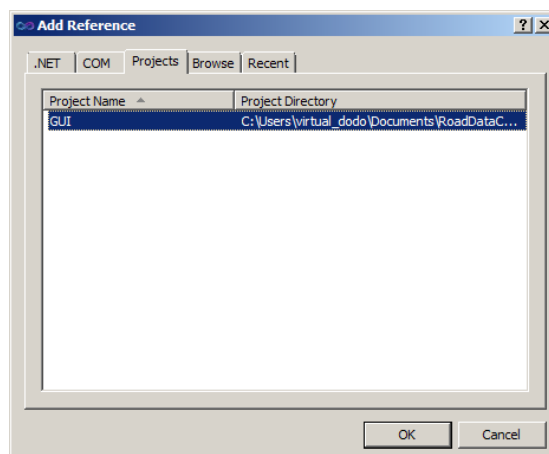


2. Пишем код :)

Если будет необходимо подключить другая часть курсовой в нашем задании: в **Solution Explorer** нажимаем правую кнопку на **References** нашего проекта и выбираем **Add Reference...**



Потом выбираем вкладка **Projects** и соотв. проект курсовой которой нам нужен.



Жмём ОК :)