

## Лабораторная работа №25

### Работа с системой контроля версий

#### 1 Цель работы

1.1 Научиться применять систему контроля версий git в процессе разработки программного обеспечения.

#### 2 Литература

2.1 <https://proglib.io/p/git-for-half-an-hour>

2.2 <https://learn.microsoft.com/ru-ru/visualstudio/version-control/git-with-visual-studio?view=vs-2022>

2.3 <https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository>

#### 3 Задание

Для локального хранения репозитория создать на ПК папку, в которой будут храниться локальные копии репозитория:

C:/Temp/isppNN/repos/

Вместо NN – номер группы

##### 3.1 Создание удаленного репозитория

3.1.1 В браузере перейти на страницу 172.17.0.30:3001 и зарегистрироваться (имя пользователя: ispp11NN (вместо NN – номер рабочего места. Например: ispp1101, ispp1114). Email: ispp11NN@arcotel.ru

3.1.2 Нажать на «+» и «Новый репозиторий» для создания репозитория

Указать настройки репозитория:

- имя репозитория: mdk0101\_NN (вместо NN – номер рабочего места. Например: mdk0101\_01, mdk0101\_14)

- видимость: сделать репозиторий частным

- описание: указать МДК.01.01, группу, фамилии авторов

- readme: default

- инициализировать репозиторий: да

- ветка по умолчанию: master

Нажать «Создать репозиторий». На открывшейся странице отобразится содержимое основной ветки проекта (master) и текст, находящийся в файле README.md.

##### 3.1.3 Изменение readme.md

Чтобы открыть файл README.md на редактирование, нужно нажать на него.

На открывшейся странице нажать кнопку карандаш («Отредактируйте этот файл») и заменить на следующий текст содержимое открывшегося на редактирование файла, используя инструменты форматирования Gitea (вместо текста, выделенного желтым цветом, указать свои данные):

**# Лабораторные и практические работы по МДК.01.01**

**## Автор**

**Фамилия Имя, группа**

## ## Используемое для разработки ПО

1. язык программирования **F#**
2. **.Net Framework 4.0**
3. среда разработки **Microsoft Visual Studio Community 2010**

## ## Требования к запуску приложения на стороне клиента:

- \* **.Net Framework 4.0**

3.1.4 При нажатии на кнопку «Предварительный просмотр» можно увидеть, как будет выглядеть файл. Если все корректно (образец показан на рисунке далее), то следует нажать на кнопку «Сохранить правки» (или «Фиксация изменений»). Если некорректно (нет заголовков разных уровней, нумерованного и маркированного списков), то нажать на «Редактировать файл» и внести изменения.

При нажатии на «Сохранить правки» отображается текст комментария к изменению. При необходимости его можно изменить перед отправкой изменения в репозиторий.

После нажатия на «История» и выбор коммита отображается сравнение старой и новой версий файла.

## 3.2 Работа с репозиторием в браузере

### 3.2.1 Создание файла

Перейти на страницу ветки master, нажать на «Новый файл» в верхней части страницы.

В открывшемся окне указать имя файла Program.cs в поле ввода и содержимое файла в многострочном поле ввода:

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello, world!");
        Console.ReadLine();
    }
}
```

В разделе «Фиксация изменений» указать описание «Добавление файла Program.cs, выводящего hello world».

### 3.2.2 Загрузить файл с компьютера.

### 3.2.3 Удалить созданный и загруженный файлы.

### 3.2.4 Просмотреть выполненные изменения. Изучить выполненные коммиты.

### 3.2.5 Создать ветвь на основе основной ветви в браузере и затем удалить ее (выполняется в разделе «Ветки»).

## 3.3 Создание рабочей копии на своем ПК путем клонирования

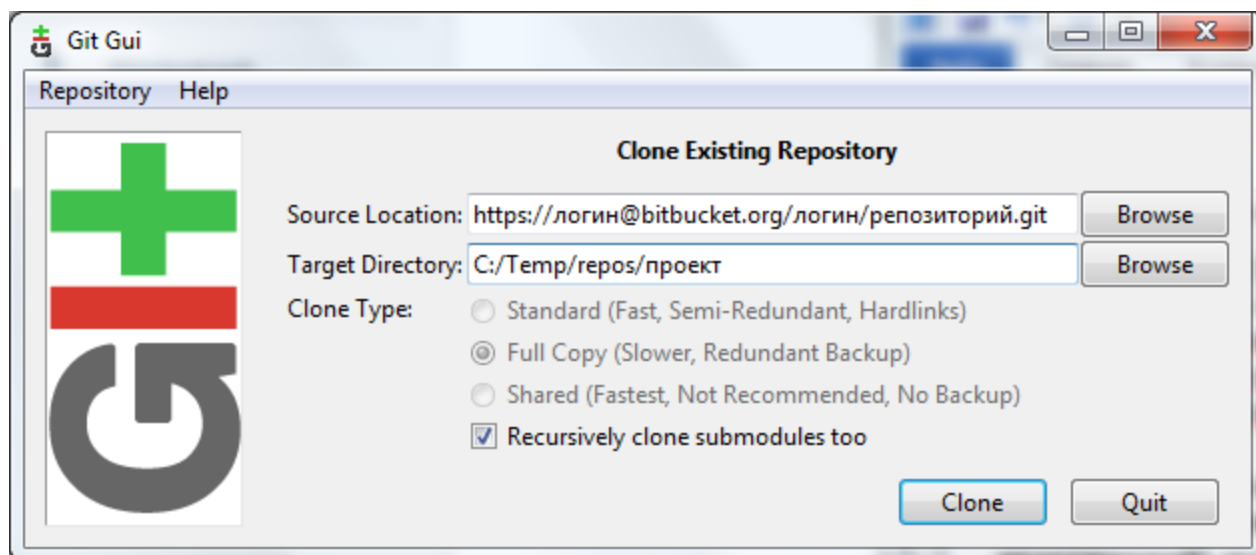
3.3.1 Создать на ПК папку, в которой будут храниться локальные копии репозитория (например C:/Temp/isppNN/repos/). Вместо NN – номер группы

### 3.3.2 Создать локальную копию репозитория

## В папке для репозитория в контекстном меню выбрать Git GUI Here – **Clone Existing Repository**

Заполнить поля источник (Source Location) и целевая папка (Target Directory):

- Source Location (репозиторий): указать адрес репозитория на сайте (можно скопировать на сайте, указав вместо localhost требуемый ip)
- Target Directory (локальная копия): нажать на Browse, в открывшемся окне нажать «Выбор папки» и дописать в поле ввода Target Directory после / имя папки LabWork25\_Task3 (в ней будут работы, выполненные на ЛР №25): C:/Temp/isppNN/repos/LabWork25\_Task3



При нажатии на кнопку «Clone» нужно указать логин и пароль и нажать Continue (данные предыдущих пользователей удаляются через Диспетчер учетных данных).

После этого в папку загрузится то, что находится в ветке master, и откроется Git GUI

Имя	Тип
.git	Папка с файлами
README.md	Файл "MD"

3.3.3 Скопировать в папку, в которую клонировали репозиторий, все файлы разработанного приложения. Содержимое скрытой папки .git не заменять (в ней настройки локальной копии репозитория)

### 3.3.4 Настройка логина и email в Git GUI:

Выбрать в меню Edit – Options...

Указать свои данные в полях User Name и Email Address (указать локально, эти данные будут использоваться для текущего проекта)

Нажать на кнопку «Save».

## **3.4 Внесение изменений в рабочую копию и загрузка изменений в новую ветку**

3.4.1 Выбрать в меню Git GUI: Branch – Create... В открывшемся окне указать Name LabWork25\_Task3, выбрать в Local Branch – master и нажать Create.

*Название папки с локальным репозиторием и ветки должно совпадать (проще будет искать работу и вносить в нее изменения).*

3.4.2 Создать в папке с рабочей копией консольный проект на С#. Содержимое скрытой папки .git не заменять (в ней настройки локальной копии репозитория).

3.4.3 Изменить содержимое файла readme.md:

- вместо строки # Лабораторные и практические работы по МДК.01.01
- внести строку # Лабораторная работа №25 по МДК.01.01

3.4.4 Загрузить файлы из локальной копии в ветку на сайте

1. Нажать «Rescan» (во вкладке Unstaged Changes отобразятся все добавленные файлы и папки)

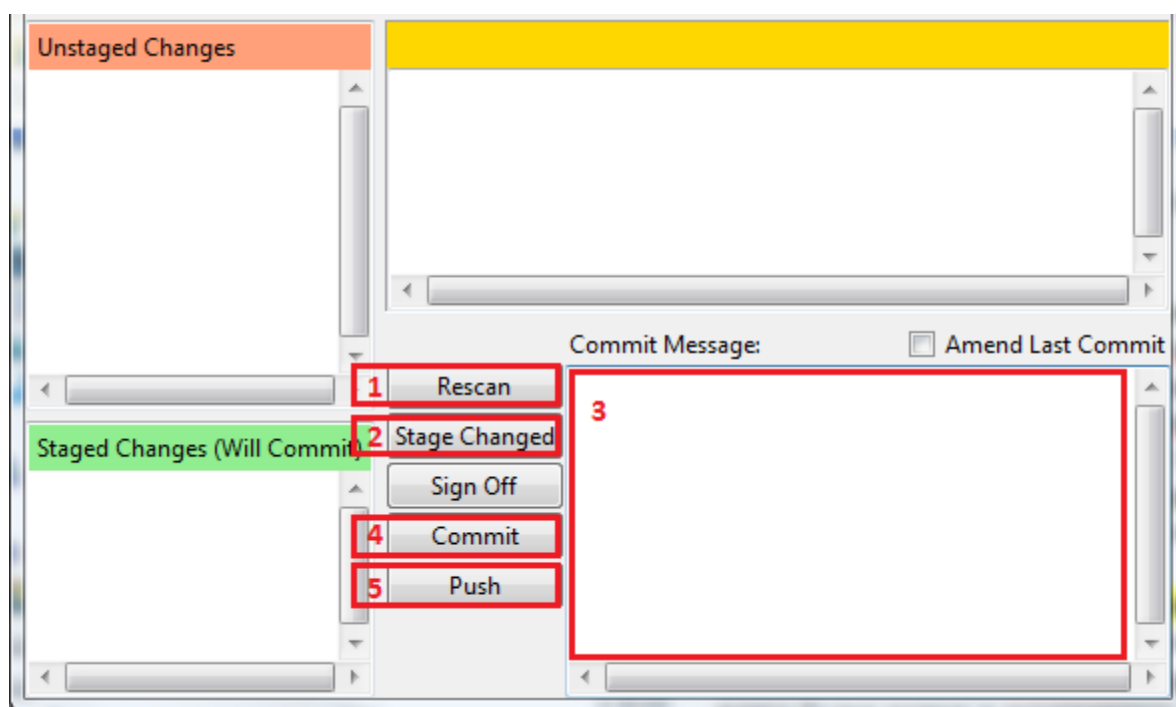
2. Нажать «Stage Changed» и «Да» в диалоговом окне. Файлы перенесутся из вкладки «Unstaged Changes» во вкладку «Staged Changes»

3. В Commit Message внести описывающий внесенные изменения текст коммита «Загрузка файлов лабораторной работы №25» (текст всегда должен указывать, что загружается)

4. Нажать «Commit», все файлы пропадут

5. Нажать «Push» для отправки изменений на сервер

6. В открывшемся окне выбрать ту ветку, в которую требуется загрузить изменения: LabWork25\_Task3 (в master ничего не загружать), нажать Push. В случае корректного выполнения в окне появится зеленая полоса с надписью Success.



### 3.5 Получение и отправка изменений

3.5.1 Закрывать Git GUI (нужно, чтобы научиться открывать локальный репозиторий)

3.5.2 Получение изменений из удаленного репозитория

Открыть в браузере ветку LabWork25\_Task3 удаленного репозитория. Изменить содержимое файла README.md, заменив «Лабораторная работа №25 по МДК.01.01» на «Лабораторная работа №25» и сохранить изменения.

Перейти в папку с рабочей копией LabWork25\_Task3 и нажать ПКМ – Git GUI Here.

В меню Git GUI нажать Remote – Fetch from – origin для получения изменений из удаленного репозитория.

меню Git GUI нажать Merge – Local merge. В открывшемся окне нажать Merge.

### 3.5.3 Отправка изменений в удаленный репозитория

Внести изменение в файл приложения в рабочей копии (например, добавить комментарии, переименовать переменные).

Сохранить изменения в удаленном репозитории: Rescan – Stage Changed – Commit Message – Commit – Push.

## 3.6 Работа с командной строкой git bash

Все использованные команды записать в лекционную тетрадь с комментариями

### 3.6.1 Клонировать удаленный репозиторий, расположенный по адресу:

<http://172.17.0.30:3001/administrator/LabWork25.git>

Для выполнения в контекстном меню папки repos вызвать командную строку git bash и выполнить следующие команды:

git clone адрес

Копирование выполняется в папку с тем же названием, что у репозитория

git clone адрес папка

Копирование выполняется в папку указанным названием (указать Task361)

### 3.6.2 Создать локальный репозиторий:

Создать в папке repos папку Task362

В папке Task362 выполнить следующее:

- инициализировать репозиторий: git init
- добавить в папку файл readme.md и заполнить его
- проверить статус репозитория: git status
- добавить файл в отслеживаемые: git add readme.md
- добавить коммит: git commit -m 'текст коммита1'
- проверить статус репозитория: git status
- добавить папку с любой из лабораторных в паку Task362
- добавить все содержимое репозитория в отслеживаемые: git add .
- добавить коммит: git commit -m 'текст коммита2'
- просмотреть коммиты: git log
- изменить текст последнего коммита: git commit --amend -m 'Новый комментарий'

### 3.6.3 Отправка изменений в удаленный репозиторий

Создать на сайте 172.17.0.30:3001 репозиторий Task363 без инициализации.

- в папке Task362 в контекстном меню вызвать Git Bash


- подключиться к удаленному репозиторию:

git remote add origin адрес удаленного репозитория Task363

- отправить изменения на сервер: `git push origin master`

### **3.7 Работа с репозиторием в Visual Studio**

3.7.1 Запустить Visual Studio. Нажать «Клонирование репозитория». Указать адрес удаленного репозитория, путь сохранения: `C:/Temp/isppNN/repos/LabWork25_7`.

3.7.2 Нажать Файл — Добавить проект. Добавить оконное приложение в ту же папку, в которой создана рабочая копия. Новые файлы отображаются с пиктограммой .

3.7.3 Отправить изменения на сервер

Нажать на пиктограмму с карандашом в нижнем правом углу строки состояния.

В выпадающем списке выбрать после открытия окна нажать «Создать ветвь» и создать ветвь `LabWork25_7`. Ввести текст сообщения для коммита «Загрузка приложения» нажать «Зафиксировать все» и кнопку «Отправить» (пиктограмма со стрелкой вверх).

Проверить в браузере, что проект загрузился.

3.7.4 Получить изменения и отправить изменения

Отредактировать в браузере один из файлов проекта из ветви `LabWork25_7`, например, добавить однострочный комментарий в код.

Для получения изменений в Visual Studio нужно нажать кнопку «Вытянуть» (пиктограмма со стрелкой вниз).

## **4 Порядок выполнения работы**

4.1 Выполнить все задания из п.3.

4.2 Ответить на контрольные вопросы.

## **5 Содержание отчета**

5.1 Титульный лист

5.2 Цель работы

5.3 Ответы на контрольные вопросы

5.4 Вывод

## **6 Контрольные вопросы**

6.1 Что такое «репозиторий»?

6.2 Что указывается в файле `readme.md`?

6.3 Что указывается в файле `.gitignore`?

6.4 Какое программное обеспечение может применяться для управления `git`-репозиторием?

6.5 Где может располагаться репозиторий?