**https://github.com/07Rinat07**

**Лабораторная работа №11**

**Тема:** Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий)

**Цель занятия:**  
Научиться устанавливать, настраивать и работать с репозиторием GitHub.

**Оборудование и ПО:**

* Персональный компьютер
* Visual Studio 2019
* Расширение GitHub Extension for Visual Studio

**Время выполнения:** 90 мин

**Краткая теория**

Система управления версиями (VCS) — инструмент, позволяющий хранить историю изменений файлов и совместно работать над проектом.

* **Subversion (SVN)** — централизованная система, все данные хранятся на одном сервере.
* **Git** — распределённая система, у каждого разработчика есть полный клон репозитория.

**Репозиторий** — центральное хранилище данных.  
**Рабочая копия** — локальные файлы, с которыми работает программист.  
**Коммит** — фиксация изменений в истории проекта.  
**Ветви** — линии разработки, которые можно создавать, объединять или удалять.

**Ход выполнения работы**

1. Установлено расширение **GitHub Extension for Visual Studio**.
2. В панели Visual Studio выбрано «Добавить в систему управления версиями» → проект добавлен в Git.
3. Основное окно:
   * **Изменения (commit)** — сохранение правок,
   * **Ветви (branch)** — создание и объединение ветвей,
   * **Синхронизация (sync)** — обмен данными с GitHub.
4. Создан небольшой проект (форма + кнопки + текст). Каждый шаг сопровождался commit.
5. Создана новая ветвь **Addfunction** → реализован новый функционал.
6. Проверена история изменений (View History).
7. Выполнено переключение между ветками и наблюдение различий.
8. Произведено слияние ветвей (**Addfunction** в **master**) → получен объединённый проект.
9. Смоделирована ситуация конфликта при слиянии и выполнено его разрешение вручную.
10. Настроена работа с GitHub:
    * зарегистрирован аккаунт,
    * выполнен вход через Visual Studio,
    * проект опубликован на GitHub,
    * проверена синхронизация.

**Выполненные задания**

1. Удаление репозиториев проверено.
2. Загрузка (клонирование) чужих репозиториев выполнена.
3. Совместная работа в паре: создан общий репозиторий, выполнено разделение задач, протестировано слияние ветвей с GitHub.

**Контрольные вопросы**

**1. Что такое система управления версиями?**  
Система, которая хранит историю изменений и позволяет совместную разработку.

**2. В чем различия между Git и Subversion?**

* Git — распределённая система, клон репозитория есть у каждого.
* SVN — централизованная система, всё хранится на сервере.

**3. Что такое репозиторий и рабочая копия?**

* Репозиторий — хранилище данных и истории проекта.
* Рабочая копия — локальные файлы на компьютере.

**4. Для чего используются коммиты?**  
Для фиксации изменений с комментариями.

**5. Что такое ветвь в Git?**  
Отдельная линия разработки проекта.

**6. Как решаются конфликты при слиянии ветвей?**  
Вручную выбирается или объединяется нужный вариант кода.

**7. Как синхронизировать локальный репозиторий с GitHub?**  
Использовать команды git pull и git push или кнопку «Синхронизация» в Visual Studio.

**8. Для чего нужен сервис GitHub?**  
Для хранения репозиториев онлайн, командной работы и автоматизации.

**9. Какие команды Git используются чаще всего?**  
git init, git clone, git add, git commit, git status, git branch, git merge, git push, git pull.

**10. Как удалить локальный и удаленный репозиторий?**

* Локальный — удалить папку проекта.
* Удалённый (GitHub) — через настройки репозитория → Delete repository.