**Лабораторная работа №20**

**Тема: «Изучение систем контроля версий GIT, GITHUB, GitLAB»**

**Цель работы:** Изучить системы контроля версий. Научиться устанавливать и пользоваться GitHub.

**Ход работы:**

**Контрольные вопросы:**

1. Опишите назначение систем CVS.

* Система CVS (Concurrent Versions System) предназначена для контроля версий и координации работы разработчиков, обеспечивая отслеживание изменений в исходном коде, возможность коллективной разработки, управление доступом, отслеживание авторства изменений, слияние изменений и поддержку ветвления и тэгирования.

1. Какие виды CVS вам известны? Какими особенностями они обладают?

* Два распространенных вида систем контроля версий (СКВ) – централизованные (например, CVS, Subversion) и распределённые (например, Git, Mercurial).
* Особенности:

1. Централизованные СКВ: Хранятся на центральном сервере. Работают с "чекаутами" и "чекинами". Ограниченная независимость.
2. Распределённые СКВ: Каждый клиент имеет полную копию репозитория. Гибкость, возможность работы оффлайн, более эффективное ветвление и слияние.
3. На примере системы контроля версиями GIT опишите особенности подхода к работе с данными. Какими преимуществами данная система обладает?

* Git:
* Распределённость: Каждый клиент имеет полную копию репозитория, что обеспечивает гибкость и возможность работы оффлайн.
* Ветвление и слияние: Позволяет эффективно создавать, объединять и управлять ветками, облегчая параллельную разработку.
* Скорость: Эффективные операции благодаря локальному хранению данных.
* Гибкие тэги: Возможность помечать конкретные версии проекта.
* Отслеживание изменений: Контроль версий для файлов, эффективное отслеживание изменений и авторства.
* Широкие возможности конфигурации: Гибкие настройки для адаптации к различным рабочим процессам.
* Поддержка больших проектов: Эффективно работает с проектами любого размера.
* Преимущества:
* Быстрота и эффективность.
* Отличная поддержка ветвления и слияния.
* Работа в оффлайн-режиме.
* Гибкость и расширяемость.

1. Опишите процесс установки И настройки GIT на свой ПК .
2. Установка:

* Скачать Git с официального сайта (https://git-scm.com/).
* Запустить установщик и следовать инструкциям по установке.

1. Настройка:

* Открыть терминал (командную строку).
* Установить имя пользователя: `git config --global user.name "Ваше Имя"`.
* Установить email: `git config --global user.email "ваш@example.com"`.
* Проверить настройки: `git config --list`.

**GitHub:**

1. Назначение сервиса GitHub:

* GitHub представляет собой веб-платформу для хостинга и совместной работы над проектами с использованием системы контроля версий Git. Он облегчает коллаборацию, отслеживание изменений, и предоставляет инструменты для управления проектами и код-ревью.

1. Назначение сервиса GitHub Desktop:

* GitHub Desktop - это графический интерфейс для Git, упрощающий взаимодействие с репозиториями на GitHub. Он предоставляет интуитивный интерфейс для выполнения базовых операций без использования командной строки.

1. Термины Git и GitHub:

* Репозиторий: Хранилище для проекта, где хранятся все файлы и история изменений.
* Рабочая область и хранилище: Рабочая область - место, где происходит активная разработка. Хранилище (или индекс) - промежуточное хранилище перед фиксацией изменений.
* Форк: Создание копии чужого репозитория с целью внесения изменений.
* Ветка: Ответвление от основной линии разработки для независимой работы над определенным функционалом.
* Мастер: Основная ветка, обычно используемая для хранения стабильного кода.
* Коммит: Фиксация изменений с описанием изменений.
* Пул: Запрос на включение изменений из одной ветки в другую.
* Пуш: Отправка локальных изменений в удаленный репозиторий.
* Пулреквест: Предложение изменений в репозиторий через форк с последующим запросом на их включение.
* Мердж: Объединение изменений из одной ветки в другую.
* Кодревью: Процесс обзора и анализа кода, часто используемый при внесении изменений в проект.

Ход работы: [ссылка](https://github.com/Suprunyuk-Nikita/BirthdayCalendar.git)