**Лабораторная работа №20**

# Тема Изучение систем контроля версий GIT, GITHUB, GitLAB и др.

# Ход работы:

**Практическая часть:**

**Контрольные вопросы:**

1. CVS (Concurrent Versions System) - система контроля версий, которая позволяет управлять изменениями в исходном коде программного обеспечения. Она позволяет отслеживать изменения, вносимые в проект, и возвращаться к предыдущим версиям файлов.

2. Существует несколько видов CVS, таких как CVSNT, CVS Suite и другие. Они обладают различными функциональными возможностями и особенностями, но все они предназначены для управления версиями файлов и совместной работы над проектом.

3. GIT - система контроля версий, которая использует распределенную модель управления версиями. Она позволяет отслеживать изменения в исходном коде, работать с несколькими ветками одновременно и совместно работать над проектом. Одним из преимуществ GIT является быстрое выполнение операций, даже при работе с большими проектами.

4. Для установки GIT необходимо загрузить установочный файл с официального сайта и запустить его на компьютере. После установки необходимо настроить имя пользователя и электронную почту, которые будут использоваться при коммитах. Для этого можно использовать команды git config --global user.name "имя пользователя" и git config --global user.email "адрес электронной почты".

GITHUB

1. GitHub - сервис, который предоставляет возможность хранить, управлять и совместно работать над проектами с использованием системы контроля версий GIT. Он позволяет разработчикам загружать свои проекты на сервер, делиться ими с другими пользователями и получать обратную связь.

2. GitHub Desktop - приложение для настольных компьютеров, которое позволяет управлять проектами на GitHub с помощью графического интерфейса. Оно позволяет загружать проекты на сервер, создавать ветки, коммиты и выполнять другие операции с проектом.

3. Термины:

1. Репозиторий - хранилище файлов проекта, которое содержит все версии файлов и информацию об изменениях.

2. Рабочая область и хранилище - два основных состояния файлов проекта в системе контроля версий.

Рабочая область - локальная копия файлов проекта на компьютере разработчика.

Хранилище - удаленный сервер, на котором хранятся все версии файлов проекта.

3. Форк - создание копии проекта на GitHub, которую можно изменять и загружать обратно на сервер.

4. Ветка - параллельная линия разработки, которая может содержать свои версии файлов проекта.

5. Мастер - основная ветка разработки, которая содержит стабильную версию проекта.

6. Коммит - сохранение изменений в файлы проекта в системе контроля версий.

7. Пул - запрос на объединение изменений из одной ветки в другую.

8. Пуш - загрузка изменений в файлы проекта на сервер.

9. Пулреквест - запрос на объединение изменений из одной ветки в другую, который отправляется другому разработчику для рассмотрения и подтверждения.

10. Мердж - объединение изменений из одной ветки в другую.

11. Кодревью - процесс проверки кода другими разработчиками для выявления ошибок и улучшения качества кода.