Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное   
Образовательное учреждение высшего образования   
«Комсомольский-на-Амуре государственный университет»

Факультет компьютерных технологий

Кафедра «Проектирование, управление и разработка информационных систем»

ОТЧЕТ

По лабораторной работе №6

Git и GitHub.

Вариант №3

Студент группы 2ИБ-1 М.К. Шолохов

Преподаватель Е.Э. Шаповалов

2023

**Содержание**

[Введение 3](#_Toc132461908)

[Теоритическая часть 4](#_Toc132461909)

[Практическая часть 6](#_Toc132461910)

[Список использованных источников](#_Toc132461911)

# Введение

GitHub — это платформа размещения кода для управления версиями и совместной работы. Она позволяет вам и другим пользователям работать над проектами вместе из любого места.

На GitHub вы учитесь и участвуете в других проектах, храните код для работы или учебы, берете код других проектов и вникаете во все детали. А еще вы можете создавать сайты бесплатно напрямую из репозитория!

Если вы хотите работать на GitHub, то вовсе не обязательно быть гуру в программировании, ведь все самое основное делается прямо на сайте.

Безусловно, GitHub нужен не всем. Допустим, вы ещё только учитесь кодить или неспешно делаете небольшой проект для личного пользования — и вас устраивает хранение проекта на локальной машине. Может, сейчас вы просто учите язык, который вам нравится, и на данном этапе не хотите хвататься за всё сразу.

В первую очередь GitHub необходим проектам с частыми обновлениями, множеством версий, большим количеством файлов, необходимостью синхронизации разработки и удобного развёртывания.

**Теоретическая часть**

Система контроля версий (СКВ) - это инструмент, который используется для управления изменениями в исходном коде программного обеспечения. Он позволяет разработчикам отслеживать изменения в коде, возвращаться к предыдущим версиям, сравнивать изменения и объединять их.

Основные принципы Git:- Git хранит историю изменений в виде дерева файлов и каталогов.- Git работает локально, что позволяет быстро переключаться между версиями кода.- Git использует хеш-суммы для идентификации изменений, что обеспечивает целостность данных.- Git позволяет создавать ветки для параллельной работы над различными версиями кода.- Git использует слияние для объединения изменений из разных веток.

Ветвление - это процесс создания новой ветки из существующей. Слияние - это процесс объединения изменений из одной ветки в другую. Существуют различные стратегии слияния, такие как fast-forward, recursive, octopus и др.

Для создания репозитория на GitHub нужно зарегистрироваться на сайте, создать новый репозиторий и загрузить свой код в удаленный репозиторий с помощью команд Git.

Для работы с Git и GitHub в своей среде разработки нужно настроить соответствующие плагины или расширения. Например, для работы с Visual Studio Code можно установить расширение GitLens.

Команды Git, которые можно использовать для отслеживания изменений, создания веток, слияния, удаления и т.д., включают git add, git commit, git branch, git merge, git push, git pull, git clone и др.

При слиянии веток могут возникать конфликты, когда один и тот же файл изменяется в обеих ветках. Для их решения нужно проанализировать изменения, выбрать нужные и сохранить результат.

Pull request - это запрос на объединение изменений из одной ветки в другую. Он позволяет другим разработчикам просмотреть изменения и оставить комментарии перед объединением.

Для автоматической сборки и развертывания приложения на удаленном сервере можно использовать сервисы Continuous Integration/Continuous Deployment (CI/CD), такие как Travis CI или CircleCI.

При работе с Git и GitHub необходимо следить за правильным документированием и комментированием кода. Каждый коммит должен иметь понятное описание изменений, а комментарии к коду должны быть информативными и понятными для других разработчиков.

# Практическая часть Индивидуальное задание

В ходе выполнения лабораторной работы, мы создали файл README.md в своем репозитории. В начале мы создали новый репозитор на GitHub и в нем файл README.md (Рисунок 1), затем мы его сохранили и сделали публичным. После мы начали его редактировать и добавлять информацию (Рисунок 2).

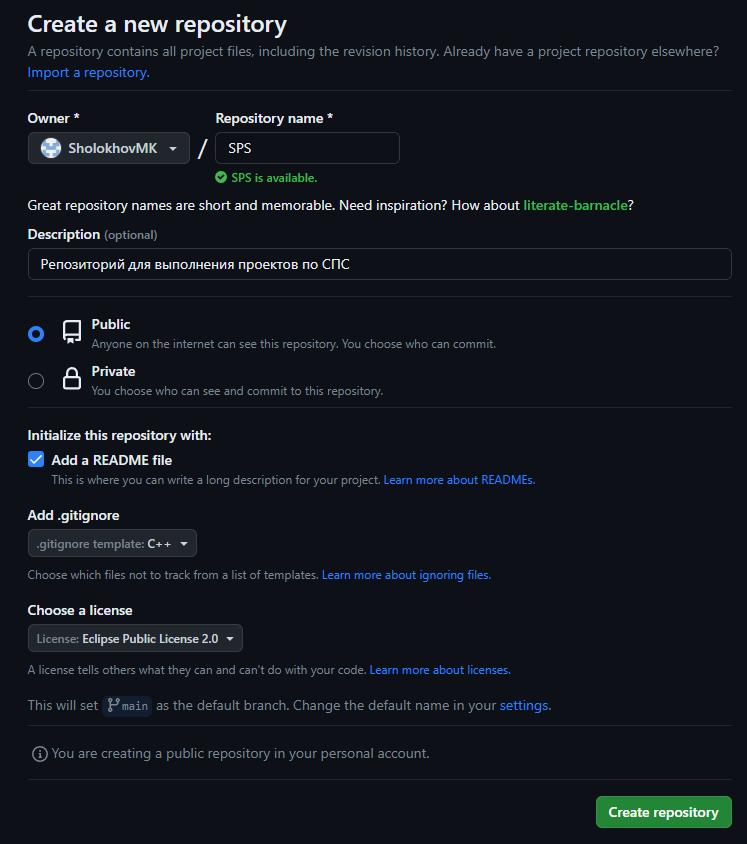


Рисунок 1 – Создание репозитория

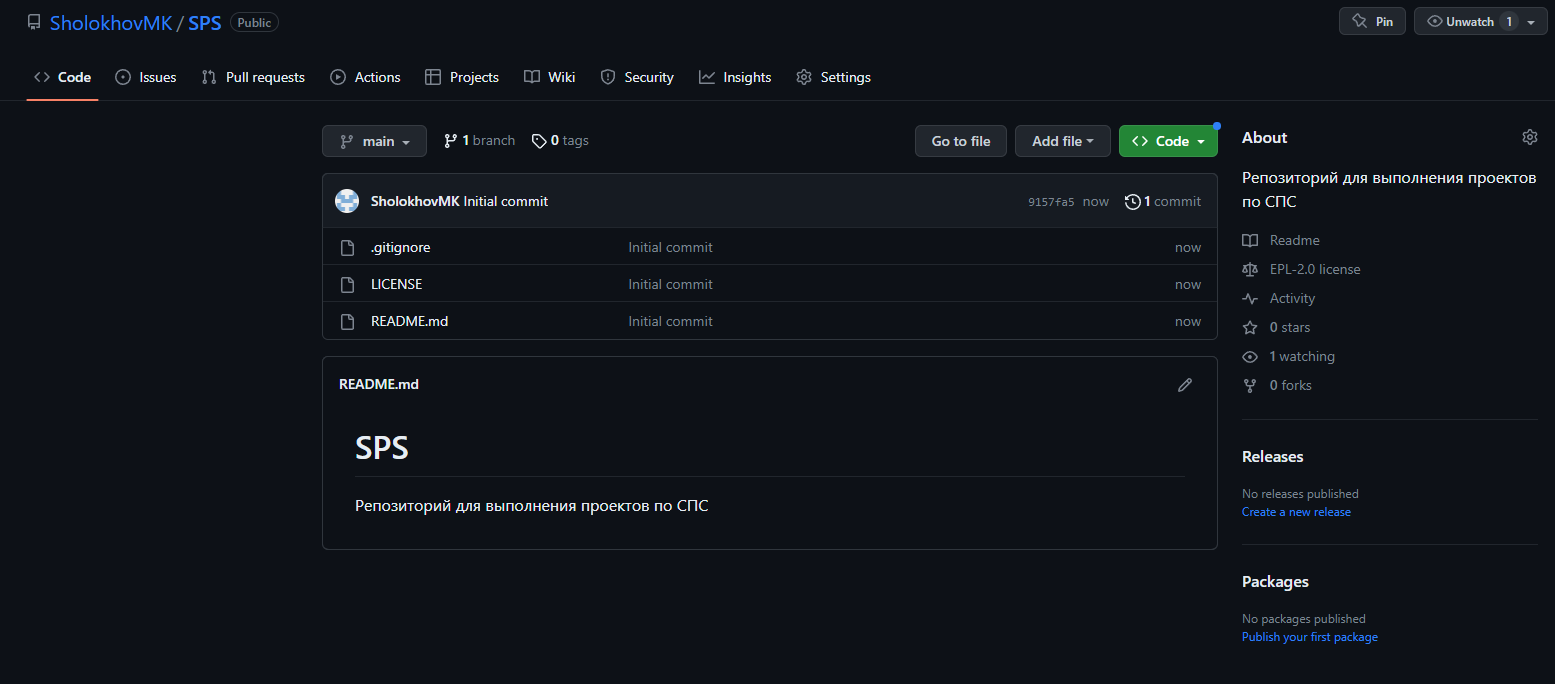


Рисунок 2 – Добавление информации

# Практическая часть Групповое задание

# Заключение

GitHub предлагает множество инструментов для работы с кодом и позволяет сделать ее проще. Вы можете объединяться со специалистами из разных стран и совместно работать над проектом, помогая друг другу с исправлением ошибок и ускоряя процесс разработки параллельно решая одну задачу.

# Список использованных источников

1 Острейковский, В.А. Информатика: учебник для вузов / В. А. Острейковский. – М.: Высшая школа, 2001, 2000. – 512 с.

2 Алексеев, А.П. Информатика 2001 / А.П. Алексеев. – М.: СОЛОН-Р, 2001. – 364 с.

3 Основы современных компьютерных технологий: учеб. пособие для вузов / под ред. А.Д. Хомоненко. – 2-е изд. – СПб.: КОРОНА-принт, 2002. – 446 с.