

Цель работы:

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

3.3. Задание.

– Сделайте отчёт по предыдущей лабораторной работе в формате Markdown. – В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md

Выполнение задания

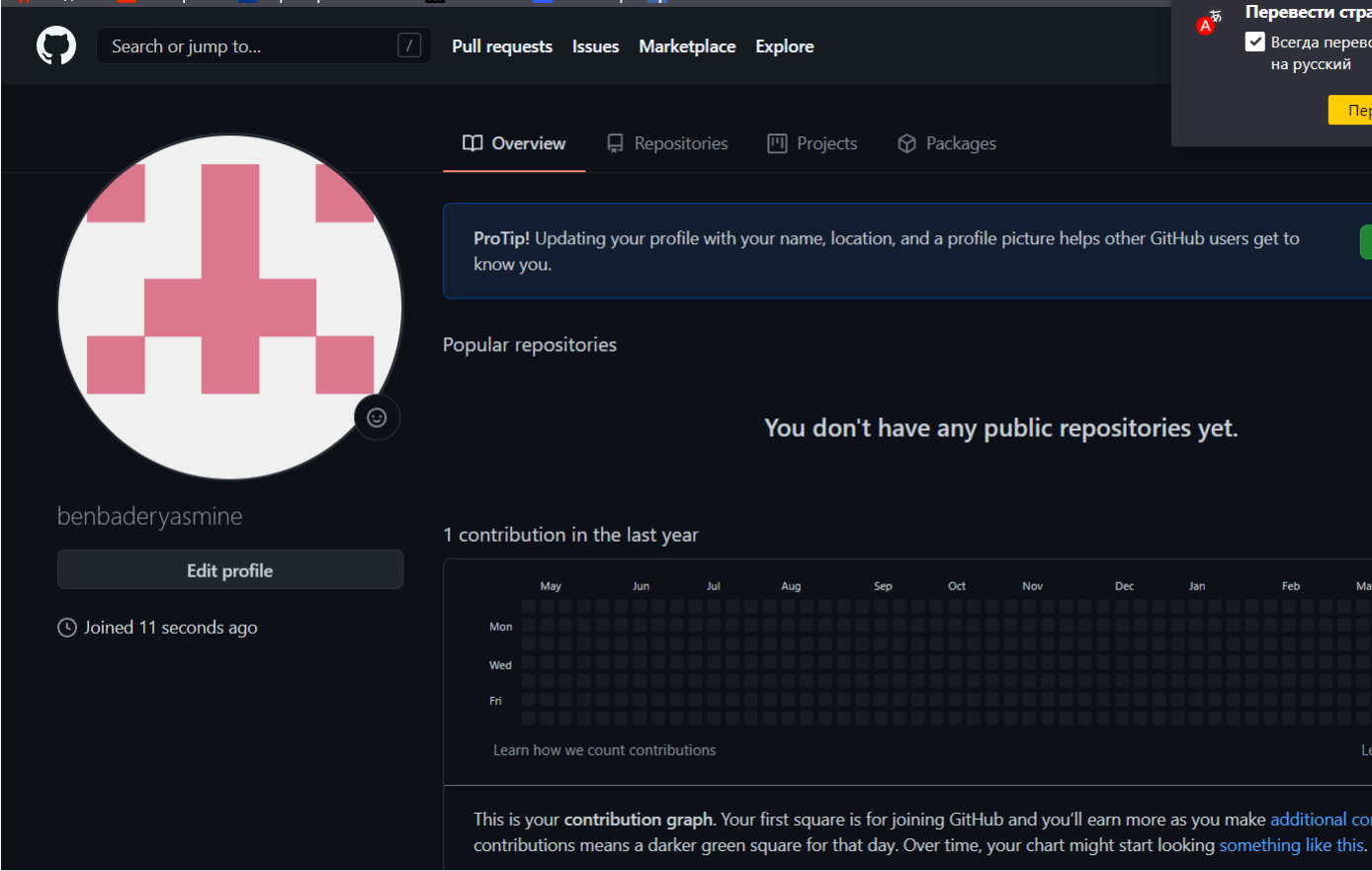
Оформление лабораторной работы №2 на Markdown.

Цель:

Изучить идеологию и применение средств контроля версий.

Ход работы:

- 1. Создал учетную запись на github.



- 2. Обозначил рабочий каталог как test, создав ее командой mkdir. После перешел в данный каталог командой cd. Инициализировал систему git командой git init. Создал заготовку для файла README.md: echo "# lab02" >> README.md git add README.md – Делаю первый коммит и выкладываю на github: git commit -m "first commit" git remote add origin git@github.com:sciproc-intro.git git push -u origin master
- 3. Добавил файл лицензий:
- 4. Для начала посмотрим список имеющихся шаблонов
 - 5. Теперь скачиваем шаблон для
- 5. Добавил новые файлы и выполнил коммит
- 6. Отправим на github
- 7. Инициализировал git-flow, установив префикс для ярлыков в v.
- 8. Проверяю, что я нахожусь на ветке develop.
- 9. Создаю релиз с версией 1.0.0
- 10. Записал версию
- 11. Добавил в индекс
- 12. Залил релизную ветку в основную ветку
- 13. Отправил данные на github
- 14. Проверил

Вывод:

Изучил и понял как работать с системой контроля версий, с помощью командной строки, а именно с Git. Разобрал команды.

Контрольные вопросы:

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

Система контроля версий (Version Control System, VCS) представляет собой программное обеспечение для облегчения работы с изменяющейся информацией. VCS нужны для хранения полной истории изменений; Описания причин всех производимых изменений; Отката изменений, если что-то пошло не так; Поиска причины и ответственного за появления ошибок в программе; Совместной работы группы над одним проектом; Возможности изменять код, не мешая работе других пользователей.

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Хранилище (repository), или репозиторий, — место хранения файлов и их версий, служебной информации. Версия (revision), или ревизия, — состояние всего хранилища или отдельных файлов в момент времени («пункт истории»). Commit («[трудовой] вклад», не переводится) — процесс создания новой версии; иногда синоним версии. Рабочая копия (working copy) — текущее состояние файлов проекта (любой версии), полученных из хранилища и, возможно, измененных.

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида. Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществляется через специальное клиентское приложение. Пример: CVS-одна из первых систем второго поколения (1986г.). Обладает множеством недостатков и считается устаревшей.

Децентрализованные системы контроля версий, в отличие от централизованной модели, может существовать несколько экземпляров репозитория, которые время от времени синхронизируются между собой. Пример: Git-распределенная система управления версиями, созданная Л. Торвальдсом для управления разработкой ядра Linux.

Отличия между централизованными и децентрализованными VCS. Централизованные: • Простота использования. • Вся история — всегда в едином общем хранилище. • Нужно подключение к сети. • Резервное копирование нужно только одному хранилищу. • Удобство разделения прав доступа к хранилищу. • Почти все изменения навсегда попадают в общее хранилище. Децентрализованные: • Двухфазный commit: 1) запись в локальную историю, 2) пересылка изменений другим. • Подключение к сети не нужно. • Локальные хранилища могут служить резервными копиями. • Локальное хранилище контролирует его владелец, • но общее — администратор. • Возможна правка локальной истории перед отправкой на сервер.

4. Опишите действия с VCS при единоличной работе с хранилищем. Участник проекта (пользователь) перед началом работы посредством определенных команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент.