Лабораторная работа №10

Сервис хранения исходного кода GitHub

Введение

GitHub (https://github.com) – крупнейший сервис для хранения исходных текстов, но помимо этого имеет массу возможностей для публикации проектов, документирования, совместной работы и так далее.

Основная цель данной лабораторной работы — получить базовые навыки работы с GitHub и настройки среды для комфортной работы с ним. Тем не менее, использовать именно GitHub необязательно. Вы вправе выбрать и альтернативы: GitLab, BitBucket и т. п.

Данная лабораторная работа так же требует знание git, но работа с git из консоли уже не обязательна, допускается создание коммитов, отправка изменений и из IDE (Visual Studio, Visual Studio Code или любая другая).

Задания отмеченные звёздочкой (*) не являются обязательными.

1. Регистрация на GitHub

Регистрация на ресурсе тривиальна и не требует дополнительных уточнений. Если у вас уже есть учётная запись на GitHub, создание ещё одной не требуется, можно использовать существующую.

Хочется отметить, что учётная запись на GitHub потребуется и в дальнейшей учёбе.

2. Настройка локального доступа

Оптимальным вариантом работы с удалёнными git-репозиториями является протокол SSH.

Подробная инструкция доступна в документации GitHub:
https://docs.github.com/ru/authentication/connecting-to-github-with-ssh

Это относительно сложный в настройке способ, но в дальнейшем вам будет гораздо удобнее взаимодействовать с удалёнными репозиториями GitHub без постоянного ввода пароля.

ВНИМАНИЕ: Следите за тем, чтобы созданный приватный ключ был доступен только вам. Возможно, стоит сделать его резервную копию. Публикация его в каком-либо виде в т. ч. и в отчёте по данной лабораторной работе категорически не приветствуется!

3. Создание репозитория, работа с удалённым репозиторием

GitHub позволяет хранить неограниченное количество публичных или же приватных проектов в системе контроля версий git.

1. Создайте новый репозиторий на GitHub.com

2. Если у вас сохранился локальный репозиторий с лабораторной работы №9, то можете использовать его, в противном случае создайте новый локальный репозиторий с несколькими файлами и несколькими коммитами.

3. Настройте удалённый репозиторий для вашего локального.

4. Загрузите все изменения из локального репозитория на GitHub.

5. Убедитесь, что на GitHub теперь есть полная копия вашего локального репозитория.

4. Работа с Markdown

В репозитории отсутствует файл README.md. Содержимое этого файла GitHub выводит на главной странице репозитория как описание проекта.

Изучите синтаксис Markdown (он весьма прост) и создайте в вашем репозитории файл README.md с произвольным текстовым содержимым. Документ должен содержать:

1. Варианты форматирования текста (*курсив*, **полужирный** и т. п.).

2. Нумерованные или ненумерованные списки.

3. Несколько подзаголовков.

4*. Подзаголовки разного уровня вложенности.

5*. Ссылки на другие интернет-ресурсы

6*. Изображения

7*. Пример исходного кода

5. Лицензия

Как правило, любой публично доступный проект имеет файл с лицензией, описывающий юридические требования и права на доступ к проекту. Добавьте в свой репозиторий лицензию согласно варианта (табл. 1). Текст лицензии в отчёт о лабораторной работе вставлять не нужно (он, как правило слишком длинный), но описание ваших действий — нужно. Как люди могут использовать ваш код, согласно выбранной лицензии?

Таблица 1: Варианты лицензий

Номер варианта	Лицензия
1, 5, 9, 13	Apache License 2.0
2, 6, 10, 14	GNU GPL 3.0
3, 7, 11, 15	MIT Lincense
4, 8, 12	As Is (unlicense)

6. Создание личной копии чужого репозитория (fork)

Для участия в общих проектах GitHub предлагает свой сценарий работы: создать копию чужого репозитория (fork), работать с ним и при необходимости предлагать свои изменения в оригинальный репозиторий (создавать pull request).

Создайте на GitHub личную копию репозитория, указанного в таблице 2. И на локальном компьютере тоже, разумеется.

Таблица 2: Список репозиториев

Вариант	Адрес репозитория
1, 4, 7, 10, 13	https://github.com/bzzzil/tiiit10_1
2, 5, 8, 11, 14	https://github.com/bzzzil/tiiit10_2
3. 6, 9, 12, 15	https://github.com/bzzzil/tiiit10_3

7. Создание pull-request

Как вы уже могли заметить, программа в репозитории, что вы склонировали, имеет ошибки.

Ваша задача:

1. Исправить эти ошибки

2. Сохранить изменения в локальном репозитории git

3. Отправить изменения в ваш личный репозиторий на GitHub

4. Предложить эти изменения в оригинальный репозиторий (создать pull request)

Созданный вами pull request, понятное дело, будет отвергнут: ведь в отличие от нормальных проектов, тот, что вы клонировали создан специально для обучения и ошибки в нём должны сохраниться на будущее.

8. И снова README.md

В этом репозитории также отсутствует описание. Что эта программа делает? Создайте описание используя возможности сайта GitHub.com. Ну и эти изменения, что уже есть на GitHub, надо забрать в локальный репозиторий.

9. Секретная копия репозитория (*)

Тот репозиторий, что вы создали в п.6, является ответвлением (fork) от оригинального репозитория. И это сразу заметно на странице репозитория (см. рис. 1).

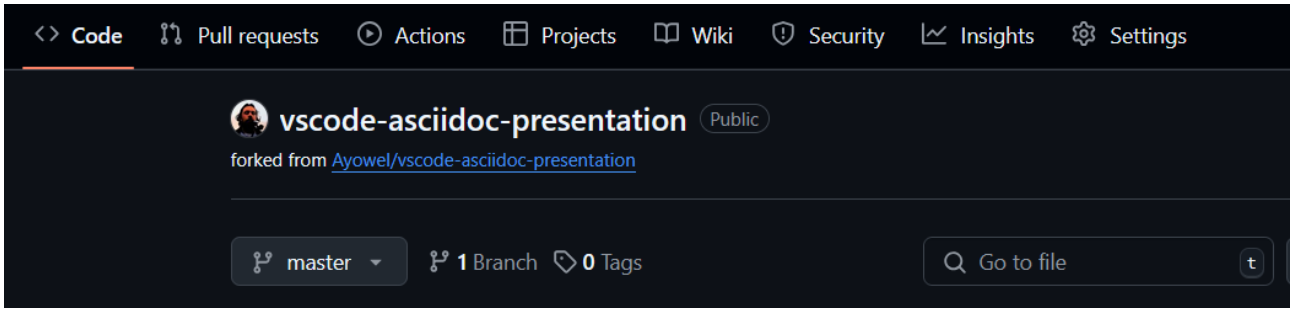

Рис. 1: Репозиторий vscode-asciidoc-presentation является ответвлением от Ayowel/vscode-asciidoc-presentation

А как сделать клон репозитория с сохранением всей истории коммитов, но без этой провокационной надписи «forked from ...»?

10. GitHub Pages (*)

Большим проектам мало файла README.md для документирования и описания всех возможностей проекта. Поэтому GitHub предлагает создать целый сайты документации, которая хранятся в репозитории, так же как и файлы проекта — GitHub Pages (https://pages.github.com/). Создайте **для вашего проекта** такой сайт.

Понятно, что писать документацию для несуществующего проекта нет никакого смысла, поэтому можно использовать любой произвольный текст. Или сгенерировать его.