

Система контроля версий Git

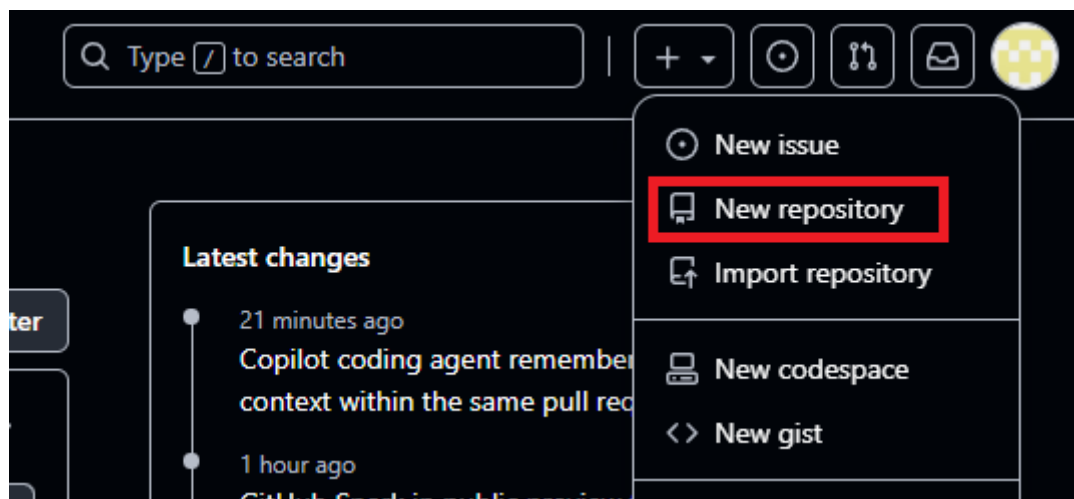
Для внесения изменений в кодовую базу разрабатываемых проектов мы используем систему версионного контроля Git. Если вы еще не знаете, что это такое, рекомендуется прочитать [1] и [2].

Взаимодействие с Git осуществляется посредством командной строки. В реальной работе как правило чаще всего используются несколько команд, такие как: создание ветки, слияние веток (merge), добавление в staged, создание коммита (commit), выгрузка изменений с локального на удаленный репозиторий, загрузка обновлений в локальный репозиторий и т. д.

В статье [3] представлено подробное руководство типичной работы с Git практически для любого проекта. Подробнее о ветках можно узнать в учебнике [4] (глава 3 полностью).

Вопросы и задания

1. Для чего нужна система контроля версий? Назовите известные вам системы контроля версий.
2. Придумайте минимум 5 сценариев, помимо разработки и поддержки кодовой базы проекта, в которых может быть полезно использовать систему контроля версий.
3. Простейшей реализацией системы контроля версий в процессе разработки проекта является простое копирование директории проекта в начале разработки нового функционала с указанием даты в качестве имени каталога. Назовите преимущества (минимум одно) и недостатки (минимум три) этого подхода.
4. Что представляет собой коммит? А что такое ветка?
5. Создайте свой репозиторий на GitHub [5]. Кнопка добавления нового репозитория находится в хэдере напротив Вашего аватара:



Вы можете назвать его как Вам угодно, но в конфигурации поле «visibility» оставьте в значении «public». Клонировать созданный Вами репозиторий на свой компьютер, выполните все необходимые настройки, создайте файл с инструкцией по работе с коммитами, сделайте коммит и отправьте изменения в удаленный репозиторий. Ссылку на Ваш репозиторий пришлите своему наставнику.

6. Напишите алгоритм для воспроизведения ситуации конфликта слияния двух веток для локального репозитория. Конфликт пусть происходит при изменении некоторого файла `file.txt`. Этот алгоритм должен состоять из шагов, которые можно воспроизвести через командную строку. Начало алгоритма можно взять следующее:

создать директорию -> инициализировать git -> создать файл `readme.txt` -> сделать первый коммит (initial commit) -> ...

Если удастся, дополните алгоритм одним из возможных сценариев разрешения конфликта.

Ссылки

- [1]. <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>
- [2]. <https://www.atlassian.com/ru/git/tutorials/what-is-version-control/>.
- [3]. <https://www.atlassian.com/ru/git/tutorials/setting-up-a-repository/>
- [4]. <https://git-scm.com/book/ru/v2/%d0%92%d0%b5%d1%82%d0%b2%d0%bb%d0%b1%d0%b2%d0%b3%d0%b4%d0%b5%d0%b6%d0%b7%d0%b8%d0%b9%d0%ba%d0%bb%d0%bc%d0%bd%d0%be%d0%bf%d0%c0%d0%c1%d0%c2%d0%c3%d0%c4%d0%c5%d0%c6%d0%c7%d0%c8%d0%c9%d0%ca%d0%cb%d0%cc%d0%cd%d0%ce%d0%cf%d0%d0%d0%d1%d0%d2%d0%d3%d0%d4%d0%d5%d0%d6%d0%d7%d0%d8%d0%d9%d0%da%d0%db%d0%dc%d0%dd%d0%de%d0%df%d0%e0%d0%e1%d0%e2%d0%e3%d0%e4%d0%e5%d0%e6%d0%e7%d0%e8%d0%e9%d0%ea%d0%eb%d0%ec%d0%ed%d0%ee%d0%ef%d0%f0%d0%f1%d0%f2%d0%f3%d0%f4%d0%f5%d0%f6%d0%f7%d0%f8%d0%f9%d0%fa%d0%fb%d0%fc%d0%fd%d0%fe%d0%ff>

%b5%d0%bd%d0%b8%d0%b5-%d0%b2-Git-%d0%9e-
%d0%b2%d0%b5%d1%82%d0%b2%d0%bb%d0%b5%d0%bd%d0%
b8%d0%b8-%d0%b2-%d0%b4%d0%b2%d1%83%d1%85-
%d1%81%d0%bb%d0%be%d0%b2%d0%b0%d1%85

[5]. <https://github.com/>