Лабораторная работа №1

Контрольные вопросы Git

1. Система управления версиями, или контроля версий, записывает историю изменения файла или набора файлов, чтобы в будущем была возможность вернуться к конкретной версии.
2. Распределенные системы контроля версий (Distributed Version Control System, DVCS).
3. Файлы в Git могут находиться в трех основных состояниях: зафиксированном, модифицированном и индексированном.
4. Индексированное (staged) состояние означает, что вы пометили текущую версию модифицированного файла как предназначенную для следующей фиксации. Команда git add.
5. Зафиксированное (committed) состояние означает, что данные надежно сохранены в локальной базе. git commit -m Сообщение'
6. git config –list
7. Есть два подхода к созданию Git-проекта. Можно взять существующий проект или папку и импортировать в Git. А можно клонировать уже существующий репозиторий с другого сервера
8. git init
9. Для этого потребуется несколько команд git add, добавляющих файлы, за которыми вы хотите следить, а затем команда git commit.
10. git status
11. Файл .gitignore со списком соответствующих паттернов создается, если некоторый класс файлов вы не хотите ни автоматически добавлять в репозиторий, ни видеть в списке неотслеживаемых.
12. git diff показывает добавленные и удаленные строки — то есть все вставки в программу.
13. git commit осуществляет фиксацию
14. git log -p -2 отображает журнал всех внесенных поправок. Одним из самых полезных является параметр -p, показывающий разницу, внесенную каждым коммитом. А дополнительный параметр -2 ограничивает выводимый результат последними двумя записями.
15. Для повторного сохранения версии в такой ситуации можно воспользоваться git commit –amend
16. git remote
17. git fetch
18. git push origin master
19. git branch имя. Ветвление (branching) означает отклонение от основной линии разработки, после которого работа перестает затрагивать эту самую основную линию. HEAD представляет собой символическую ссылку на вашу текущую ветку.
20. Переход на существующую ветку реализует команда git checkout.
21. git merge
22. Если в двух ветках, которые вы собираетесь слить, вы внесли разные изменения в один и тот же файл, Git не сможет просто взять и объединить их.Вручную проверить файлы и отредактировать причину конфликта а потом зафиксировать результат.
23. GitLab — сайт и система управления [репозиториями](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%B9) кода для [Git](https://ru.wikipedia.org/wiki/Git). Из дополнительных возможностей: собственная вики и [система отслеживания ошибок](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BE%D1%82%D1%81%D0%BB%D0%B5%D0%B6%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BE%D1%88%D0%B8%D0%B1%D0%BE%D0%BA).
24. <https://ru.stackoverflow.com/questions/431520/%D0%9A%D0%B0%D0%BA-%D0%B2%D0%B5%D1%80%D0%BD%D1%83%D1%82%D1%8C%D1%81%D1%8F-%D0%BE%D1%82%D0%BA%D0%B0%D1%82%D0%B8%D1%82%D1%8C%D1%81%D1%8F-%D0%BA-%D0%B1%D0%BE%D0%BB%D0%B5%D0%B5-%D1%80%D0%B0%D0%BD%D0%BD%D0%B5%D0%BC%D1%83-%D0%BA%D0%BE%D0%BC%D0%BC%D0%B8%D1%82%D1%83>