**Лабораторная работа №2**

**Управление версиями в Git**

**Цель работы**

Получение практических навыков управления версиями в Git. Знакомство с основными командами Git. Знакомство с интеграцией GitHub с системой управления проектами и системой отслеживания ошибок YouTrack.

**Задание**

Создать пользователя. Создать локальный репозиторий. Выполнить преобразования с локальным репозиторием, используя операции Git. Добавить удалённый репозиторий. Выполнить преобразования с удалённым репозиторием. Осуществить интеграцию GitHub с системой управления проектами и системой отслеживания ошибок YouTrack. Выполнить преобразования с ветками.

**Порядок выполнения**

1. Настройте пользовательские параметры.
   1. Введите команду **git config –global user.name ‘admin’**
   2. Введите команду **git config –global user.email ‘admin@admin.com’**
2. Проверьте пользовательскую конфигурацию.
   1. Введите команду **git config -list**
3. Создайте локальный Git-репозиторий.
   1. Создайте рабочий каталог на локальном диске.
   2. Введите команду **cd d://repository**
   3. Введите команду **git init**
4. Добавьте файлы под версионный контроль.
   1. Поместите файл в каталог.
   2. Введите команду **git add .**
5. Выполните коммит изменений.
   1. Введите команду **git commit –m ‘comment’**
6. Проверьте статус репозитория.
   1. Введите команду **git status**
7. Осуществите игнорирование файлов.
   1. Создайте в рабочем каталоге файл **.gitignore.**
   2. В файле укажите шаблоны файлов, которые не надо брать под версионный контроль. Введите **\*.[oa]**. Сохраните изменения.
   3. Введите команду **cat .gitignore**.
8. Выведите информацию об изменениях в файле.
   1. Введите команду **git diff**
9. Выполните удаление файла.
   1. Введите команду **git rm Test.txt**
10. Выполните перемещение файла.
    1. Введите команду **git mv Test.txt Test**
11. Выведите историю коммитов.
    1. Введите команду **git log**
12. Отмените изменения коммита.
    1. Введите команду **git commit –amend –m ‘comment’**
13. Исключите файл из области подготовки.
    1. Введите команду **git reset HEAD Test.txt**
14. Отмените изменения для файла, который был изменён.
    1. Введите команду **git checkout –Test.txt**
15. Выполните клонирование удалённого репозитория с сервера GitHub.
    1. Введите команду **git clone** [**https://github.com/user/project.git**](https://github.com/user/project.git)
16. Посмотрите настроенные удалённые репозитории.
    1. Введите команду **git remote**
17. Добавьте удалённый репозиторий.
    1. Введите команду **git remote add origin <https://github.com/user/project.git>**
18. Добавьте данные в удалённый репозиторий.
    1. Введите команду **git push –u origin master**
19. Получите данные об удалённом репозитории.
    1. Введите команду **git remote show origin**
20. Смените имя текущего удалённого репозитория.
    1. Введите команду **git remote rename test test2**
21. Выполните удаление удалённого репозитория.
    1. Введите команду **git rm test2**
22. Узнайте информацию о существующих метках.
    1. Введите команду **git tag**
23. Создайте аннотированную метку.
    1. Введите команду **git tag –a v1.0 –m ‘version 1.0’**
24. Узнайте информацию об авторе метки.
    1. Введите команду **git show v1.0**
25. Добавьте метку в репозиторий.
    1. Введите команду **git push orivin v1.0**
26. Осуществите интеграцию GitHub с YouTrack.
    1. Перейдите на вкладку **Администрирование.**
    2. Выберите пункт **Интеграция с репозиториями.**
    3. Заполните форму.
    4. Подключитесь к серверу VCS.
    5. Выбрать проект в YouTrack к которому необходимо подключиться.
    6. Выбрать репозиторий в GitHub.
    7. Указать коммитеров.
    8. Указать отслеживаемые ветви.
    9. Указать группы пользователей, которые будут иметь доступ к работе над проектом.
27. Выполните создание ветки.
    1. Введите команду **git branch test**
28. Выполните переключение ветки.
    1. Введите команду **git checkout test**
29. Выполните слияние веток.
    1. Введите команду **git checkout master**
    2. Введите команду **git merge test**
30. Выполните удаление ветки.
    1. Введите команду **git branch –d test**

**Содержание отчёта**

1. Титульный лист
2. Цель работы
3. Индивидуальное задание
4. Снимки экрана выполнения задания лабораторной работы
5. Выводы