

ЛАБОРАТОРНАЯ РАБОТА №1

РАБОТА В СИСТЕМЕ КОНТРОЛЯ ВЕРСИЙ GIT

ЦЕЛЬ РАБОТЫ

Приобрести умения и практические навыки организации хранения разработанных программных модулей в системе контроля версий Git.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Задание №1

Установка и настройка Git¹

1. Скачать установщик с официального сайта Git².
2. Запустить его.
3. Выбрать подходящие опции из предлагаемых мастером установки.

Задание №2

Дана библиотека³, вычисляющая некоторые тригонометрические функции.

1. Инициализируйте репозиторий в папке `mat_lib`.
2. Задайте имя пользователя и email глобально.
 - 2.1. Проверьте, что изменения внесены в файл глобальных настроек `.gitconfig`.
 - 2.2. Проверьте, что файл локальных настроек `.git/config` не был изменен.
3. Изучите содержимое папки `.git/`.
 - 3.1. Узнайте, на что сейчас указывает HEAD.
 - 3.2. Просмотрите файл, на который указывает HEAD (в этом пункте есть подвох).
4. Добавьте все файлы в индекс.
5. Сделайте первый коммит.
 - 5.1. Просмотрите объект коммита, найдите хэш объекта-дерева корня репозитория.
 - 5.2. Просмотрите объект дерева корня репозитория.
 - 5.3. Проверьте, на что указывает HEAD сейчас.
 - 5.4. Просмотрите файл, на который указывает HEAD.
 - 5.5. Ответьте на вопрос: на что указывает текущая ветка? Для этого просмотрите на объект, на который указывает ветка.
6. Выполните файл `test.py`.

¹ <https://git-scm.com/book/ru/v2/Введение-Установка-Git>

² <https://git-scm.com/download/>

³ https://drive.google.com/uc?export=download&confirm=no_antivirus&id=1wRy1Qcn2Zy_XQuDcdt_qSIWtOO3xVsZi

- 6.1. Просмотрите статус файлов, чтобы обнаружить, что появились файлы кэша.
7. Сделайте второй коммит.
- 7.1. Просмотрите объект коммита, найдите хэш родительского коммита.
- 7.2. Посмотрите, на что сейчас указывает HEAD.
- 7.3. Проверьте файл, на который указывает HEAD.
- 7.4. Узнайте, на что указывает текущая ветка. Для этого просмотрите на объект, на который указывает ветка.

Задание №3

Дана библиотека, написанная на Python – Geometric Lib⁴. В качестве задания необходимо выполнить ряд действий над репозиторием `geometric_lib`: создадим новую ветку, сделаем коммиты, посмотрим историю и изучим внесенные изменения средствами Git.

- 1) Выполните команду

```
git clone https://github.com/smartigaorg/geometric_lib.
```

Она скопирует наш репозиторий к вам на компьютер.

- 2) Создайте новую ветку с названием **new_features** и переключитесь на нее.

- 3) Добавьте новый файл в эту ветку. Например, с вычислениями для фигуры

Прямоугольник.

Его название: **rectangle.py**

Его содержимое:

```
def area(a, b):  
    ____return a * b  
  
def perimeter(a, b):  
    ____return a + b
```

- 4) Сделайте коммит с сообщением "L-03: Added rectangle.py".

- 5) Добавьте еще один файл в эту ветку. Например, с вычислениями для фигуры

Треугольник.

Его название: **triangle.py**

Его содержимое:

⁴ https://github.com/smartigaorg/geometric_lib

```
def area(a, h):  
    ____return a * h / 2  
  
def perimeter(a, b, c):  
    ____return a + b + c
```

6) Исправьте ошибку в вычислении периметра в файле `rectangle.py`, теперь он должен стать таким:

```
def area(a, b):  
    ____return a * b  
  
def perimeter(a, b):  
    ____return 2 * (a + b)
```

7) Создайте еще один коммит внутри этой же ветки, его сообщение: "L-03: Added triangle.py and fixed rectangle perimeter bug".

8) Постройте граф истории всего репозитория с однострочным выводом коммитов.

9) Постройте граф истории только текущей ветки. В ней должно быть два последних коммита.

10) Возьмите хэши двух последних коммитов из истории и посмотрите, какие изменения были внесены.

11) Обычно, так не делают на практике, но мы только учимся, поэтому давайте удалим ветку `new_features` без слияния. Не забудьте, что нельзя удалить ветку, на которой находится HEAD.

Задание №4

1. Создайте репозиторий в GitHub с наименованием, определенным преподавателем. Через Git Bash клонируйте пустой репозиторий в папку на Вашем локальном устройстве.
2. Добавьте в папку созданный проект (например, WPF-приложение). Изучите вывод команды `git status`. Проиндексируйте файлы командой `git add`. Снова посмотрите вывод `git status`.
3. Зафиксируйте изменения командой `git commit`.
4. Сделайте и зафиксируйте следующие изменения (каждый подпункт - одна фиксация):

- Добавьте одно новое окно в приложение;
- Добавьте предложенные преподавателем элементы управления (например, текстовое поле и кнопку) в стартовое окно;
- Добавьте к созданным на предыдущем этапе элементам управления обработчики событий.

5. Изучите вывод команды git log.

Задание №5

1. Создайте проект в Microsoft Visual Studio (например, WPF- приложение).
2. С помощью интерфейса для работы с GitHub в Microsoft Visual Studio создайте репозиторий с наименованием, определенным преподавателем.
3. Сделайте и зафиксируйте следующие изменения (каждый подпункт - одна фиксация) с помощью интерфейса для работы с GitHub в Microsoft Visual Studio:
 - Добавьте одно новое окно в приложение;
 - Добавьте предложенные преподавателем элементы управления (например, текстовое поле и кнопку) в стартовое окно;
 - Добавьте к созданным на предыдущем этапе элементам управления обработчики событий.
4. Изучите историю изменений через веб-интерфейс.

Задание №6

1. Создаем свой форк на GitHub от репозитория, адрес которой указан в таблице.

Группа	Адрес репозитория
4333	https://github.com/CITTest/Template4333.git
4335	https://github.com/CITTest/Template4335.git
4337	https://github.com/CITTest/Template4337.git
4338	https://github.com/CITTest/Template4338.git

2. Клонировем к себе свой форк репозитория.
3. Измените файл README.md. Сделайте коммит.
4. Выполните пуш ваших изменений в свой удаленный репозиторий. Заметьте, что вам не нужно добавлять удаленный репозиторий: он уже был добавлен под именем origin во время клонирования.

5. Откройте страницу вашего репозитория на GitHub. Можете убедиться, что изменения были успешно загружены в удаленный репозиторий, просмотрев историю коммитов на главной странице репозитория.
6. Ознакомьтесь с файлом /Template43XX/MainWindow.xaml в репозитории. Необходимо вставить в определенную строку элемента компоновки Grid (согласно своему порядковому номеру в списке группы) кнопку, которая будет выводить окно с информацией об авторе изменения (достаточно указать имя и возраст). Окно необходимо также предварительно создать с названием **«НомерГруппы_ФамилияСтудента»**, как и обработчик события.
7. Повторите шаги 4-5 для выполненных изменений.
8. Создайте пулл-реквест. В нем подробно опишите внесенные изменения. На странице предварительного просмотра пулл-реквеста проверьте, что вы не затронули файлы на других ветках и еще раз перепроверьте код.