

Федеральное государственное автономное  
образовательное учреждение высшего  
образования  
«Национальный исследовательский университет  
ИТМО»

Факультет Информационных технологий и программирования

Работа: Лабораторная работа по Git 2

Выполнила студентка группы №М3103  
*Кравченкова Елизавета Александровна*

Проверил:  
*Повышев Владислав Вячеславович*

Санкт-Петербург

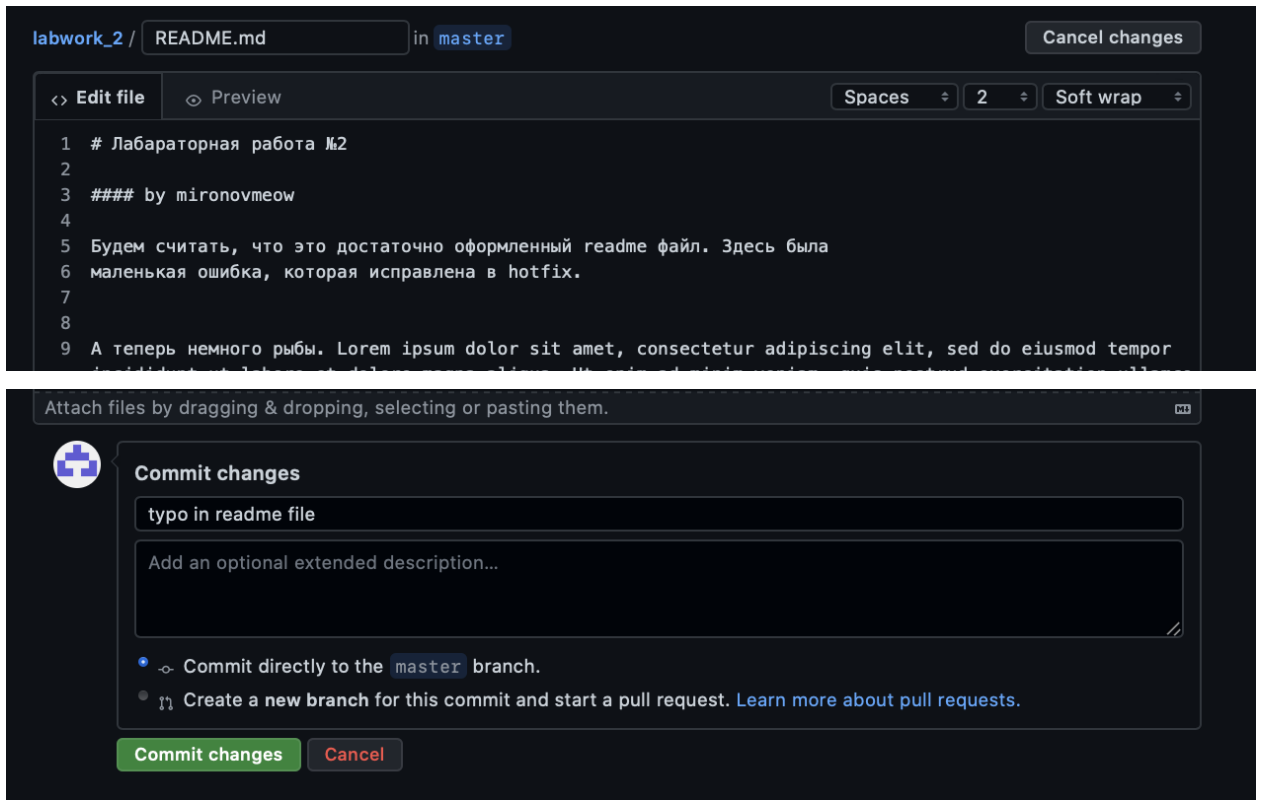
2022 г.

# 1.Pull request

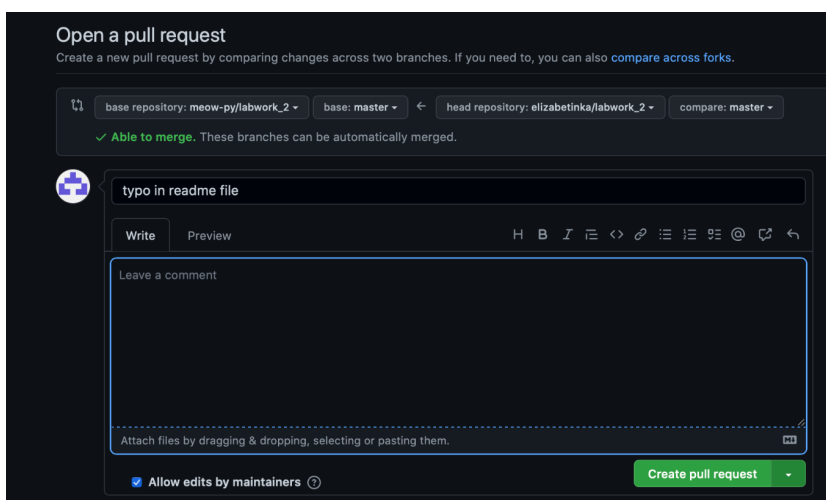
1.1. Форкаем проект. Это создаст копию репозитория в моем аккаунте



1.2 Изменяем README.md файл.И делаем коммит

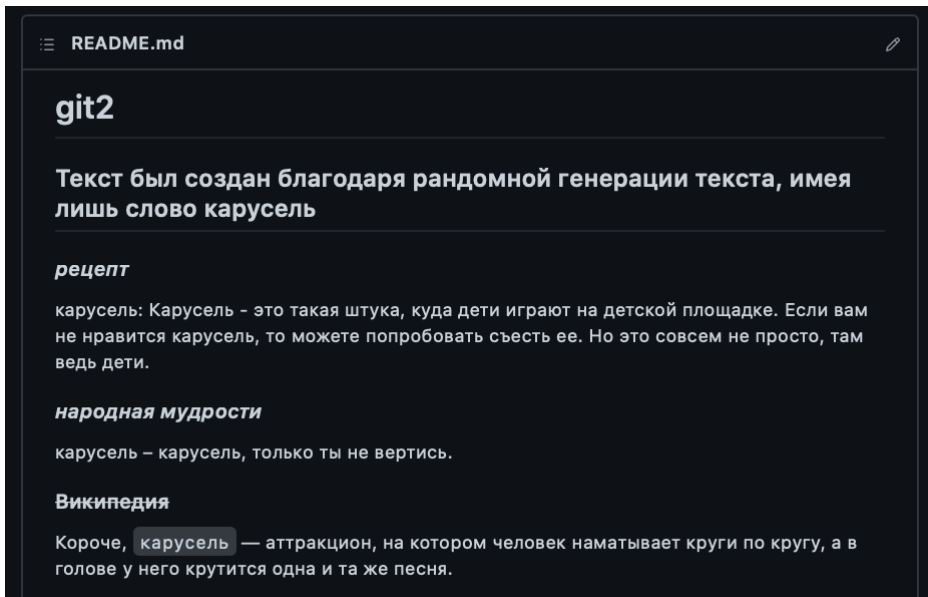


1.3 Создаем pull request



## 2. GitFlow

### 2.1 Создаем репозиторий. Оформляем README.md. Клонировем репозиторий



```
indo.evri.momint@MacBook-Pro-Elizaveta ~ % git clone git@github.com:elizabetinka/git2.git
Клонирование в «git2»...
```

### 2.2 Для начала использования git-flow. Скачаем и инициализируем его. Создается ветка develop

```
brew install git-flow-avh
```

```
% git flow init
```

### 2.3 Работа с ветками фич.(Разработка новых фич для последующих релизов)

#### 2.3.1 Создадим ветку MYFEATURE2(из ветки develop) и переключимся на нее

```
indo.evri.momint@MacBook-Pro-Elizaveta git2 % git flow feature start MYFEATURE2
Переключились на новую ветку «feature/MYFEATURE2»
```

#### 2.3.2 Добавим файл.Сделаем коммит. Завершим работу с веткой фич.(произойдет слияние с develop. Переключение на develop Ветка фич удалится)

```
git add new_tortik.txt
git add tortik.txt
git commit -m "two_cakes"
```

```
indo.evri.momint@MacBook-Pro-Elizaveta git2 % git flow feature finish MYFEATURE2
Переключились на ветку «develop»
```

#### 2.3.3 Делаем аналогичные действия с новой веткой MYFEATURE3. Не будем завершать с ней работу, а опубликуем ее

```
% git flow feature publish MYFEATURE3
```

## 2.4 Создание релиза(Поддержка подготовки нового релиза продукта)

### 2.4.1 Создание ветки RELEASE(из ветки develop)

```
[indo.evri.momint@MacBook-Pro-Elizaveta git2 % git flow release start RELEASE  
Переключились на новую ветку «release/RELEASE»
```

2.4.2 Закончиваем работу с веткой RELEASE(Ветка релиза сливается в main и develop. В main она помечается тегом). После этого пушим теги

```
% git flow release finish RELEASE
```

```
% git push --tags
```

2.4.3 Сделаем тоже самое. Только не будем завершать работу с веткой. А опубликуем ее

```
git flow release publish RELEASE
```

2.5 Ветка исправлений(Исправления нужны в том случае, когда нужно незамедлительно устранить нежелательное состояние продакшн-версии продукта)

### 2.5.1 Создание ветки HOTFIX (из ветки main)

```
git flow hotfix start VERSION
```

2.5.2 Завершение работы с HOTFIX(сливается в develop и в main. В main отмечается тегом)После этого пушим теги

```
git flow hotfix finish VERSION
```

```
% git push --tags
```

2.5.3 Сделаем тоже самое. Только не будем завершать работу с веткой. А опубликуем ее

```
git flow hotfix publish VERSION2
```

2.6 Теперь ветки и теги в репозитории выглядит так

The screenshot displays a Git repository interface with two main sections: 'Your branches' and 'Tags'.

**Your branches:**

- Default branch:** main (Updated 7 hours ago by elizabetinka)
- Other branches:**
  - hotfix/VERSION2 (Updated 8 hours ago by elizabetinka)
  - release/RELEASE3 (Updated 8 hours ago by elizabetinka)
  - feature/MYFEATURE3 (Updated 8 hours ago by elizabetinka)
  - develop (Updated 8 hours ago by elizabetinka)

**Tags:**

- VERSION:** 8 hours ago, commit 87437ed, available as zip or tar.gz
- RELEASE2:** 8 hours ago, commit 0fdf483, available as zip or tar.gz

## 3.Submodules


3.1 Добавляем в виде подмодуля репозиторий из лабораторной по Git 1.  
Коммитим изменения. Делаем push в свой репозиторий

```
[indo.evri.momint@MacBook-Pro-Elizaveta git2 % git submodule add /Users/indo.evri.momint/geometric_lib
Клонирование в «/Users/indo.evri.momint/git2/geometric_lib»...
готово.
[indo.evri.momint@MacBook-Pro-Elizaveta git2 % git status
Текущая ветка: main
Эта ветка соответствует «origin/main».

Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    новый файл:   .gitmodules
    новый файл:   geometric_lib

[indo.evri.momint@MacBook-Pro-Elizaveta git2 % git commit -m "add submodules"
```

3.2 Так выглядит ветка main в github. Подмодуль добавлен!

	elizabetinka add submodules	95d8ef4 6 hours ago	🕒 10 commits
📁	geometric_lib @ d078c8d	add submodules	6 hours ago
📄	.gitmodules	add submodules	6 hours ago
📄	README.md	Update README.md	7 hours ago
📄	bestrecipes.txt	best_recipes	6 hours ago
📄	new_tortik.txt	two_cakes	7 hours ago
📄	tortik.txt	fixing	6 hours ago

## 4.LFS

Git LFS заменяет большие файлы текстовыми указателями внутри Git'a, в то время как само содержимое этих файлов сохраняется на удалённом сервере

4.1 Скачаем LFS-расширение. Настроим Git LFS для своей учетной записи

```
% brew install git-lfs
```

```
[indo.evri.momint@MacBook-Pro-Elizaveta git2 % git lfs install
Updated Git hooks.
Git LFS initialized.
```

4.2 Выберем тип файлов, который будем отслеживать(psd). Добавим файл с этим расширением.Добавим все файлы в индекс. Коммитим. Делаем push в репозиторий

```
[indo.evri.momint@MacBook-Pro-Elizaveta git2 % git lfs track "*.psd"
Tracking "*.psd"
```

```
indo.evri.momint@MacBook-Pro-Elizaveta git2 % git add .gitattributes
indo.evri.momint@MacBook-Pro-Elizaveta git2 % touch "file.psd"
indo.evri.momint@MacBook-Pro-Elizaveta git2 % git add "file.psd"
indo.evri.momint@MacBook-Pro-Elizaveta git2 % git commit -m "Add design file use lfs"
```

## 5. Справочник по основным командам Git

### Глобальные изменения репозитория

git push (отправляет на удаленный репозиторий текущее состояние проекта)  
git pull (забирает изменения с удаленного репозитория в свой локальный)  
git merge "name branch" (втягиваем ветку в ветку, в которой мы сейчас находимся)  
git init (создать репозиторий)  
git clone "path" (клонировать репозиторий)  
git remote -v (выведет все доступные репозитории)  
git remote add origin "ssh репозитория" (добавляем удаленный репозиторий)  
git rebase master (объединить текущую ветку с веткой мастер)

### Работа с ветками

git checkout «name branch» (для переключения между ветками)  
git checkout -b «name branch» (создать новую ветку и переключится на нее)  
git checkout - (переключится на ветку с которой ушли)  
git branch(выводит все доступные ветки)  
git branch -d «name» (удаляет ветку, надо перед этим перейти в master)  
git branch -D «name» (удаляет ветку с коммитами, надо перед этим перейти в master)

### Коммиты

git commit -m «first commit» (Коммит)  
git log (История коммитов)  
git log -p (Более полная история коммитов)  
git diff «хеш-сумма» «хеш сумма» (Выведет изменения между 2 коммитами)  
git status (Выведет состояние файлов в ветке)  
git add. (Добавить файлы в индекс)  
git rm (Удалить файлы из индекса)  
git restore "name"(откатывает изменения в файле до состояния последнего коммита)  
git checkout "хеш сумма" (перемещает на какой-то коммит в истории, но не перемещает указатель на ветку)  
git reset --hard "хеш сумма" (перемещает на какой-то коммит в истории, перемещает указатель на ветку)

### Работа над документами(не git, просто полезно)

cd "name" (войти в папку)  
ls -la (выведет все доки в папке)  
touch "name" (создать документ)  
git mv "name" "name" (переименовывает)  
mkdir "name" (создание папки)

### Git Flow

git flow feature(release/hotfix) start(finish/publish) "name\_branch"

## 5.1 Базовый пример работы с основными командами

```
W [indo.evri.momint@MacBook-Pro-Elizaveta ~ % git clone git@github.com:elizabetinka/git2.git
Клонирование в «git2»...

[indo.evri.momint@MacBook-Pro-Elizaveta git2 % git checkout main
Уже на «main»
Эта ветка соответствует «origin/main».
[indo.evri.momint@MacBook-Pro-Elizaveta git2 % touch "test_file.txt"
[indo.evri.momint@MacBook-Pro-Elizaveta git2 % git status
Текущая ветка: main
Эта ветка соответствует «origin/main».

Неотслеживаемые файлы:
  (используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)
    test_file.txt

индекс пуст, но есть неотслеживаемые файлы
(используйте «git add», чтобы проиндексировать их)
[indo.evri.momint@MacBook-Pro-Elizaveta git2 % git add test_file.txt
[indo.evri.momint@MacBook-Pro-Elizaveta git2 % git commit -m "add test_file"
[main f635c33] add test_file
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 test_file.txt
[indo.evri.momint@MacBook-Pro-Elizaveta git2 % git push
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (2/2), 255 байтов | 255.00 КиБ/с, готово.
Всего 2 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано 0 пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:elizabetinka/git2.git
   7452122..f635c33  main -> main
[indo.evri.momint@MacBook-Pro-Elizaveta git2 %
```