

Министерство науки и высшего образования РФ

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

Факультет информационных технологий и программирования

Лабораторная работа №1

Работа с системой версионного контроля GIT.

Выполнил студент группы № М3102

Лопатенко Георгий Валентинович

Подпись:

Проверил:

Приискалов Роман Андреевич

Санкт-Петербург

2021

Текст лабораторной работы

Лабораторная работа предназначена для углубленного ознакомления с возможностями Git. Предполагается, что у студента уже есть собственный GitHub аккаунт и репозиторий по предмету - программирование.

Произвести клонирование репозитория и выполнить задания на копии репозитория.

- 1) Организовать модель ветвления, где каждая лабораторная - отдельная ветка.
- 2) Отправка, принятие Pull Request - ваш код должен пройти код ревью вашего одногруппника. Обязательное наличие комментариев, правок. Разрешается, в целях обучения, допущение преднамеренных ошибок в коде.
- 3) Использование тегов для обозначения версий.
- 4) Установить приложение графического интерфейса для гит.
- 5) Новое клонирование репозитория из пункта 1. И изначального коммита, используя модель ветвления GitFlow, реорганизуем свои ветки. Для этого использовать только команды гита. Запрещается копировать и удалять в папках. В отчет необходимо предоставить последовательность и описание используемых команд.

Отчет

Будем пользоваться публичным репо <https://github.com/Lopa10ko/ITMO-algo-1sem>.

1) Организовать модель ветвления, где каждая лабораторная - отдельная ветка.

Глобальное изменение настроек Git, установка имени и почты пользователя:

```
git config --global user.name "George Lopatenko"
git config --global user.email lopa10ko239@mail.ru
git config --global core.editor vim
```

Перемещение в директорию на локальной машине:

```
C:\Users\Georgy>cd C:\Users\Georgy\Desktop\itmo-algo
```

Инициализация гит структуры:

```
git init
Initialized empty Git repository in C:/Users/Georgy/Desktop/itmo-algo/.git/
```

Создание базовых файлов:

```
echo > .gitignore
echo > README.md
```

Добавление всех файлов из директории и первый коммит:

```
git add .
git commit -m "initial commit"
```

Переименование основной ветки:

```
git branch -M master main
```

Добавление ссылки на удаленный репо:

```
git remote add origin
```

```
https://github.com/Lopa10ko/ITMO-algo-1sem.git
```

Пуш main:

```
git push -u origin main
```

Создание побочной ветки разработки:

```
git checkout -b lab1
```

```
git status
```

Untracked files:

```
1A.py
```

```
1B.py
```

```
1C.py
```

```
1D.py
```

```
1E.py
```

```
problems1.pdf
```

Добавление всех файлов из директории с готовой первой лабой, коммит и пуш:

```
git add .
```

```
git commit -m "added lab1 branch"
```

```
git push -u origin lab1
```

Создание новой ветки, добавление всех файлов из директории с готовой второй лабой, коммит и пуш:

```
git checkout -b lab2
```

```
git add .
```

warning: LF will be replaced by CRLF in E.cpp

```
git config core.autocrlf true
```

```
git commit -m "added lab2 branch"
```

```
git status
```

```
git push -u origin lab2
```

Просмотр истории коммитов:

```
git log --name-only
```

Тэги коммитов:

```
git tag
```

```
git checkout <branch_name>
```

```
git tag -a v1.0 -m "1.0"
```

```
git tag
```

```
git commit ...
```

```
git push ...
```

```
git push --tags
```

```
git pull = get fetch + get merge
```

2) Pull request

The screenshot shows a GitHub pull request for the file `1E.py`. The pull request is titled "1Menemi1 reviewed 4 days ago" and has a "View changes" button. The code diff shows the following changes:

```
1E.py Outdated
1 + file_input = open('turtle.in', 'r')
2 + file_output = open('turtle.out', 'w')
3 +
4 + # game board and prizes markup initialization
```

Below the code diff, there is a comment from 1Menemi1: "тут код довольно очевидный, можно было не комментировать (аналогично с другими 2 комментариями)". There is a "Reply..." input field and a "Resolve conversation" button.

Below the comment, there is a commit history showing that Lopa10ko requested a review from 1Menemi1 4 days ago. The pull request was then approved by 1Menemi1 4 days ago, and Lopa10ko merged the commit `eebf99f` into the `main` branch 4 days ago. There is a "Revert" button.

3) Tags

The screenshot shows the GitHub repository page for `Lopa10ko / ITMO-algo-1sem`. The repository is public and has 1 star and 0 forks. The "Tags" tab is selected, showing a list of tags:

Tag	Commit	Created	Download
4.0	b0452a8	4 days ago	zip tar.gz
3.0	fe7abb1	4 days ago	zip tar.gz
2.0	6e43a40	4 days ago	zip tar.gz
1.1	09c2833	4 days ago	zip tar.gz
1.0	754edc8	14 hours ago	zip tar.gz

The screenshot shows the GitHub repository page for `Lopa10ko / ITMO-algo-1sem`. The repository is public and has 1 star and 0 forks. The "Releases" tab is selected, showing a list of releases:

Release	Version	Created	Download
4.0	b0452a8	4 days ago	zip tar.gz
3.0	fe7abb1	4 days ago	zip tar.gz
2.0	6e43a40	4 days ago	zip tar.gz
1.1	09c2833	4 days ago	zip tar.gz
1.0	754edc8	14 hours ago	zip tar.gz

4) Установка графического интерфейса: Git Desktop

Who would win

Most advanced version control system used to keep track of changes in any set of files. Aimed at speed, data integrity, and support for distributed, non-linear workflows.



Me

Development > Creations > Android > My Color Manager All Saves >				
File with	Burn	New folder		
Name	Date modified	Type	Size	
ColorManager	13-03-2017 12:47	File folder		
ColorManager 9-5-2016	13-03-2017 12:47	File folder		
ColorManager 11-5-2016	13-03-2017 12:48	File folder		
ColorManager save 1	13-03-2017 12:48	File folder		
MyColorManager 03-06-16	13-03-2017 12:48	File folder		
MyColorManager 4-7-2016	13-03-2017 12:49	File folder		
MyColorManager 4-7-2016 2nd	13-03-2017 12:49	File folder		
MyColorManager 5-6-16	13-03-2017 12:50	File folder		
MyColorManager 5-8-16	13-03-2017 12:51	File folder		
MyColorManager 6-7-16	13-03-2017 12:51	File folder		
MyColorManager 10-6-2016	13-03-2017 12:52	File folder		
MyColorManager 10-7-2016	13-03-2017 12:52	File folder		
MyColorManager 10-8-2016	13-03-2017 12:53	File folder		
MyColorManager 13-7-2016	13-03-2017 12:54	File folder		
MyColorManager 15-5-16	13-03-2017 12:54	File folder		
MyColorManager 17-5-16	13-03-2017 12:55	File folder		
MyColorManager 18-7-16	13-03-2017 12:56	File folder		
MyColorManager 19-05-16	13-03-2017 12:56	File folder		
MyColorManager 20-5-16 13-24	13-03-2017 12:57	File folder		
MyColorManager 20-8-2016	13-03-2017 12:58	File folder		
MyColorManager 21-7-2016	13-03-2017 12:58	File folder		
MyColorManager 23-5-16 2	13-03-2017 12:59	File folder		
MyColorManager 23-7-2016	13-03-2017 12:59	File folder		
MyColorManager 24-5-16	13-03-2017 13:00	File folder		
MyColorManager 25-8-2016	13-03-2017 13:01	File folder		
MyColorManager 25-9-2016	13-03-2017 13:02	File folder		
MyColorManager 26-5-16	13-03-2017 13:02	File folder		
MyColorManager 26-7-2016	13-03-2017 13:03	File folder		
MyColorManager 27-7-2016	13-03-2017 13:04	File folder		
MyColorManager 27-09-2016	13-03-2017 13:05	File folder		
MyColorManager 30-7-2016	17-09-2017 01:06	File folder		
MyColorManager 2016-9-21	13-03-2017 13:06	File folder		

5) Git-Flow (<https://github.com/Lopa10ko/ITMO-algo-2021-2022>)

Инициализация git flow модели:

```
git flow init
```

Which branch should be used for bringing forth production releases?

```
- main
```

Branch name for production releases: [main] main

Branch name for "next release" development: [develop] develop

How to name your supporting branch prefixes?

Feature branches? [feature/]

Bugfix branches? [bugfix/]

Release branches? [release/]

Hotfix branches? [hotfix/]

Support branches? [support/]

Version tag prefix? []

Hooks and filters directory?

```
[C:/Users/Georgy/Desktop/ITMO-algo--21-22/ITMO-algo-2021-2022/.git/hooks]
```

Пушим ветку develop:

```
Georgy@LAPTOP-GDEFFIGV MINGW64 ~/Desktop/ITMO-algo--21-22/ITMO-algo-2021-2022 (develop)
```

```
$ git push --set-upstream origin develop
```

Добавляем ветку с фичами:

```
$ git flow feature start feature_branch
```

Смотрим, какие ветки есть:

```
$ git branch -a
```

```
develop
```

```
* feature/feature_branch
```

```
main
```

```
remotes/origin/HEAD -> origin/main
```

```
remotes/origin/develop
```

```
remotes/origin/main
```

Создаем папку в ветке с фичами:

```
(feature/feature_branch) $ mkdir problems1
```

Добавляем файлы и коммитим:

```
$ cd ./problems1
$ touch 1A.py 1B.py 1C.py 1D.py 1E.py
$ cd ~/Desktop/ITMO-algo--21-22/ITMO-algo-2021-2022
$ git add problems1
$ git add problems1/*
$ git commit -m "added lab1"
```

Закрываем работу ветки с фичами:

```
$ git flow feature finish feature_branch
```

Summary of actions:

- The feature branch 'feature/feature_branch' was merged into 'develop'
- Feature branch 'feature/feature_branch' has been locally deleted
- You are now on branch 'develop'

Пушим:

```
(develop) $ git push origin
```

Создаем ветку с релизом:

```
(develop) $ git flow release start '0.1.0'
```

Пушим ее:

```
(release/0.1.0) $ git push --set-upstream origin release/0.1.0
```

Добавляем все необходимые файлы и метаданные:

```
$ git add .
$ git push origin
Everything up-to-date
```

Закрываем работу ветки с релизом:

```
(release/0.1.0) $ git flow release finish '0.1.0'
```

Summary of actions:

- Release branch 'release/0.1.0' has been merged into 'main'
- The release was tagged '0.1.0'
- Release tag '0.1.0' has been back-merged into 'develop'
- Release branch 'release/0.1.0' has been locally deleted; it has been remotely deleted from 'origin'
- You are now on branch 'develop'

Окончательный пуш:

```
(develop) $ git checkout main
(main) $ git push origin
```

Пуш тэгов:

```
(main) $ git tag -l
git push --tags
```

MISSION PASSED!
RESPECT +99