

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчет о лабораторной работе № 1 по дисциплине
«Основы программной инженерии»

Выполнил студент
2 курса, группы ПИЖ-б-о-20-1
Емельянова Я.А.
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

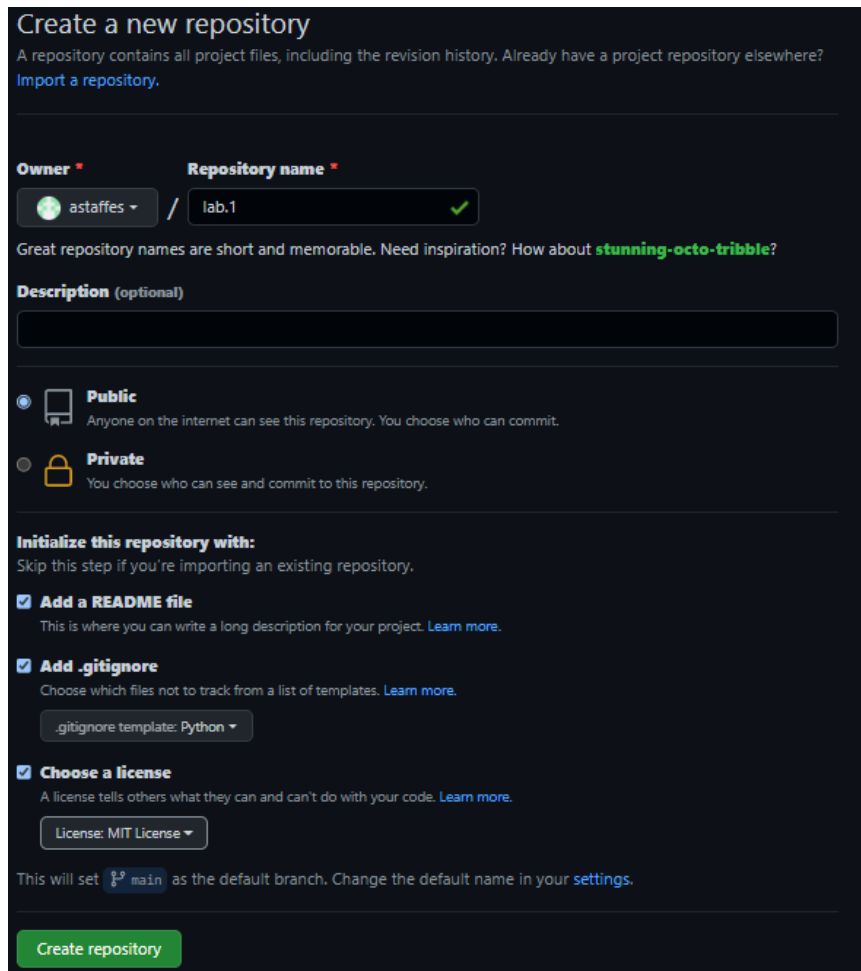
Ставрополь, 2021 г.

1. Ход работы

```
C:\Users\Яна>git version  
git version 2.35.1.windows.2
```

```
C:\Users\Яна>git config --global user.name EmelyanovaYana  
C:\Users\Яна>git config --global user.email astaffes@yandex.ru
```


Рис. 1 – установка Git



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner ^{*} Repository name ^{*}

 astaffes / lab1 ✓

Great repository names are short and memorable. Need inspiration? How about [stunning-octo-tribble?](#)

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.


☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☒ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)
.gitignore template: Python ▾

☒ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)
License: MIT License ▾

This will set  main as the default branch. Change the default name in your [settings](#).

Create repository

Рис. 2 – создание репозитория GitHub

```
C:\Users\Яна>git clone https://github.com/astaffes/lab1  
Cloning into 'lab1'...  
remote: Enumerating objects: 9, done.  
remote: Counting objects: 100% (9/9), done.  
remote: Compressing objects: 100% (9/9), done.  
remote: Total 9 (delta 1), reused 4 (delta 0), pack-reused 0  
Receiving objects: 100% (9/9), done.  
Resolving deltas: 100% (1/1), done.
```

Рис. 3 – клонирование репозитория

```

C:\Users\Яна\lab1>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md
        deleted:    rrr.txt

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\Яна\lab1>git commit -m "Add information about local repository in readme file"
[main 3c9741f] Add information about local repository in readme file
1 file changed, 8 deletions(-)
delete mode 100644 rrr.txt

C:\Users\Яна\lab1>git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)

```

Рис. 4 – локальное изменение содержимого

```

C:\Users\Яна\lab1>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 278 bytes | 278.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/astaffes/lab1
   7ce8336..bab6361  main -> main

C:\Users\Яна\lab1>git pull
Already up to date.

```

Рис. 5 – pull & push

Ответы на вопросы:

1. Что такое СКВ и какого её назначение?

Система контроля версий (СКВ) – это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.

2. В чём недостатки локальных и централизованных СКВ?

Недостаток локального в отсутствии связи программистов между собой, а централизованного – в ненадёжности физических носителей и элементов системы (серверов, дисков и прочего).

3. К какой СКВ относится Git?

К распределительной СКВ.

4. В чём концептуальное отличие Git от других СКВ?

В методе хранения и обработке данных, которые больше похожи на снимки, чем на обычное хранилище.

5. Как обеспечивается целостность хранимых данных в Git?

Через хеш-сумму

6. В каких состояниях могут находиться файлы в Git? Как связаны эти состояния?

У Git есть три основных состояния, в которых могут находиться файлы:

- Зафиксированный, значит, что файл уже сохранен в локальной базе.
- К измененным относятся файлы, которые поменялись, но ещё не были зафиксированы.
- Подготовленные файлы – это изменённые файлы, отмеченные для включения в следующий коммит.

7. Что такое профиль пользователя в GitHub?

Публичная страница пользователя

8. Какие бывают репозитории в GitHub?

Общие и частные

9. Укажите основные этапы модели работы с GitHub.

Стандартный подход к работе с проектом состоит в том, чтобы иметь локальную копию репозитория и фиксировать изменения в этой копии, а не в удалённом репозитории, размещённом на GitHub. Этот локальный репозиторий

имеет полную историю версий проекта, которая может быть полезна при разработке без подключения к интернету.

10. Как осуществляется первоначальная настройка Git после установки?

Необходимо сначала проверить, был ли Git установлен успешно, с помощью команды `git version` в терминале, а затем следует указать имя и адрес электронной почты, связанный с аккаунтом GitHub с помощью следующих команд:

```
Git config --global user.name "your.name"
```

```
Git config --global user.email "your.email"
```

11. Опишите этапы создания репозитория в GitHub.

После нажатия кнопки "New repository" необходимо указать некоторые важные поля:

- Имя репозитория. Оно необязательно должно быть уникальным во всём github, но уникальным в рамках репозитория одного аккаунта.
- Описание, которое можно оставить пустым.
- Public/private если необходимо, чтобы репозиторий был публичным либо закрытым.
- .gitignore и license для игнорируемых файлов и для лицензирования репозитория

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

Apache License 2.0

GNU General Public License v3.0

MIT License

BSD 2-Clause "Simplified" License

BSD 3-Clause "New" or "Revised" License

Boost Software License 1.0

Creative commons Zero v1.0 Universal

Eclipse Public License 2.0

GNU Affero General Public License v3.0

GNU General Public License v2.0

GNU Lesser General Public License v2.1

Mozilla Public License 2.0

The Unlicense

13. Как осуществляется клонирование репозитория GitHub? Зачем нужно клонировать репозиторий?

С помощью команды “git clone”. Клонирование необходимо для работы с файлами репозитория на разных устройствах – изменения файлов, их добавления и удаления и т. д.

14. Как проверить состояние локального репозитория Git?

Командой “git status”

15. Как изменяется состояние локального репозитория Git после выполнения следующих операций: добавления/изменения файла в локальный репозиторий Git; добавления нового/изменённого файла под версионный контроль с помощью команды git add; фиксации (коммита) изменений с помощью команды git commit и отправки изменений на сервер с помощью команды git push?

Появятся незафиксированные файлы (выделены красным шрифтом); новые файлы будут зафиксированы в локальном репозитории (выделены зелёным шрифтом); изменения станут зафиксированными в хранилище, репозиторий обновится.

16. У вас имеется репозиторий на GitHub и два рабочих компьютера, с помощью которых Вы можете осуществлять работу над некоторым проектом с использованием этого репозитория. Опишите последовательность команд, с помощью которых оба локальных репозитория, связанных с репозиторием GitHub будут находиться в синхронизированном состоянии. Примечание: описание необходимо начать с команды git clone.

git clone

git add / git commit – m / git push

git pull

17. GitHub является не единственным сервисом, работающим с Git. Какие сервисы ещё Вам известны?

Bitbucket, GUI Clients, GitKraken и пр.

18. Интерфейс командной строки является не единственным и далеко не самым удобным способом работы с Git. Какие Вам известны программные средства с графическим интерфейсом пользователя для работы с Git? Приведите как реализуется описанные в лабораторной работе операции Git с помощью одного из таких программных средств.

- GitHub Desktop

- Fork
- Tower
- Sourcetree
- SmartGit
- Sublime Merge
- GitKraken
- GitUp