

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии  
Департамент цифровых, робототехнических систем и электроники

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**  
**дисциплины**  
**«Основы кроссплатформенного программирования»**

Выполнил:  
Пугачев Кирилл Дмитриевич  
2 курс, группа ИТС-б-о-23-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи»,  
направленность (профиль)  
«Инфокоммуникационные системы и  
сети», очная форма обучения

---

(подпись)

Проверил:

Воронкин Р. А.

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2024 г.

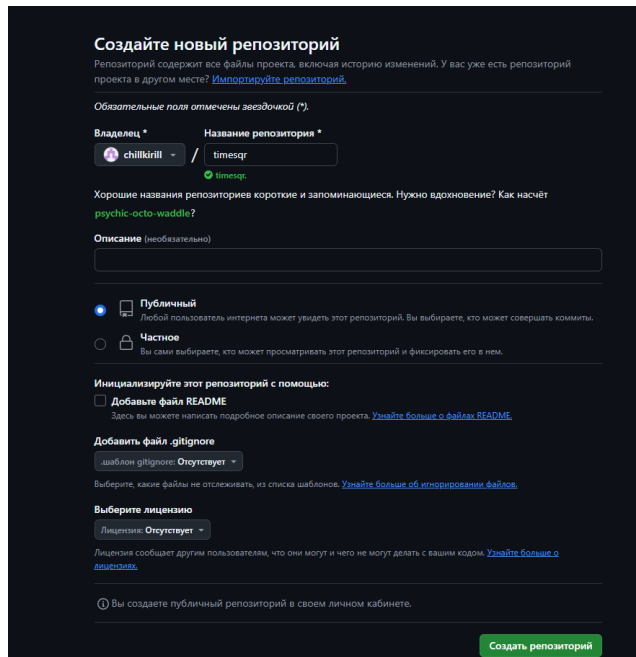
**Тема:** Исследование основных возможностей Git и GitHub

**Цель:** исследовать базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.

**Ссылка на репозиторий:** <https://github.com/chillkirill/LABA1>

### Порядок выполнения работы:

#### 1. Создание нового репозитория.



**Создайте новый репозиторий**  
Репозиторий содержит все файлы проекта, включая историю изменений. У вас уже есть репозиторий проекта в другом месте? [Импортируйте репозиторий](#).

Обязательные поля отмечены звездочкой (\*).

Владелец \* / Название репозитория \*

chillkirill / timesqr

Хорошие названия репозитория короткие и запоминающиеся. Нужно вдохновение? Как насчёт `psychic-octo-waddle`?

Описание (необязательно)

☒ Публичный  
Любой пользователь интернета может увидеть этот репозиторий. Вы выбираете, кто может совершать коммиты.

☐ Частное  
Вы сами выбираете, кто может просматривать этот репозиторий и фиксировать его в нем.

Инициализируйте этот репозиторий с помощью:

☒ Добавьте файл README  
Здесь вы можете написать подробное описание своего проекта. [Узнайте больше о файлах README](#).

Добавить файл .gitignore

шаблон .gitignore:

Выберите, какие файлы не отслеживать из списка шаблонов. [Узнайте больше об игнорировании файлов](#).

Выберите лицензию

Лицензия:

Лицензия сообщает другим пользователям, что они могут и чего не могут делать с вашим кодом. [Узнайте больше о лицензиях](#).

① Вы создадите публичный репозиторий в своем личном кабинете.

[Создать репозиторий](#)

Рисунок 1. Создание репозитория

#### 2. Клонирование репозитория на компьютер.

```
C:\Users\user\Desktop>git config --global Kirill <chillkirill>
Ошибка в синтаксисе команды.

C:\Users\user\Desktop>git config --global user.name chillkirill

C:\Users\user\Desktop>git config --global user.email quwlast@gmail.com

C:\Users\user\Desktop>git config user.name
chillkirill

C:\Users\user\Desktop>git config user.email
quwlast@gmail.com

C:\Users\user\Desktop>git clone https://github.com/chillkirill/LABA1.git
Cloning into 'LABA1'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (5/5), done.

C:\Users\user\Desktop>
```

Рисунок 2. Клонирование репозитория

### 3. Добавление в файл README информации о группе и ФИО.

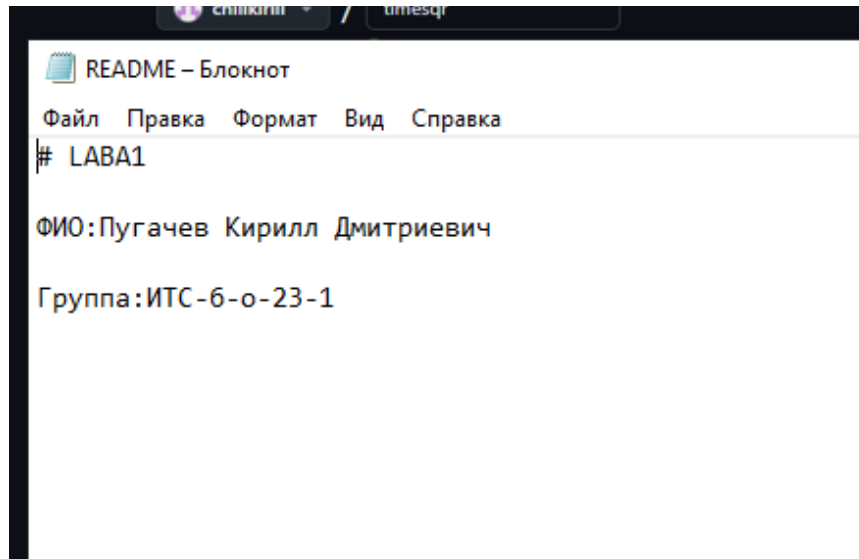


Рисунок 3. Изменение README

### 4. Добавление изменений в локальный репозиторий.

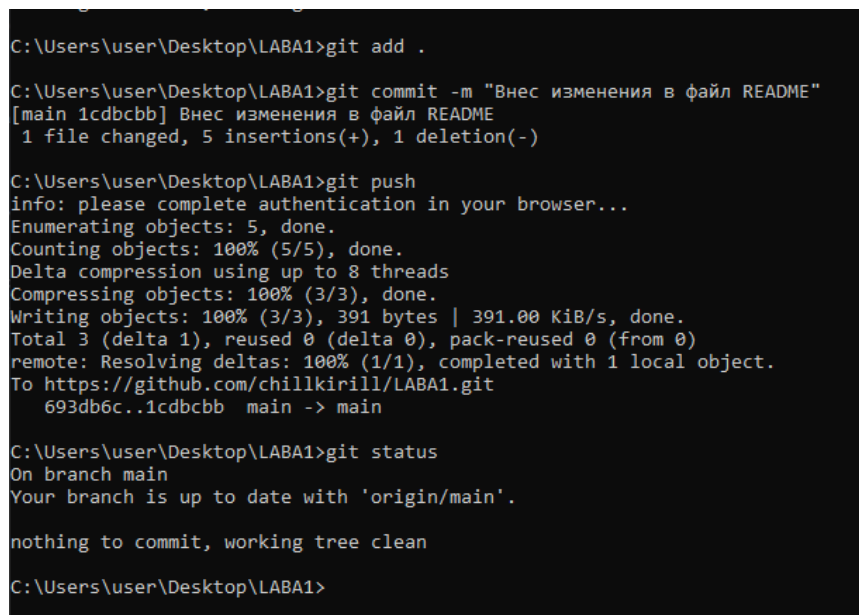


Рисунок 4. Добавление изменений

### 5. Создание коммита.

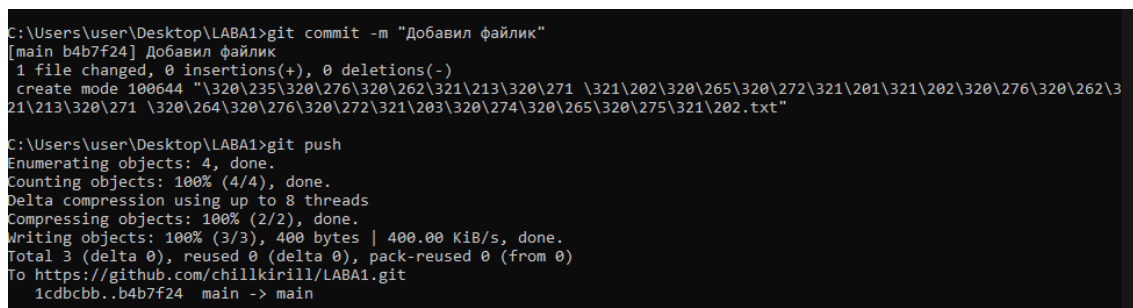


Рисунок 5. Создание коммита

## 6. Изменение репозитория и создание семи коммитов.

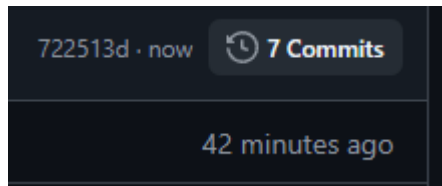


Рисунок 6. Создание семи коммитов

## 7. Распространение изменений на исходный репозиторий

```
C:\Users\user\Desktop\LABA1>git commit -m "Ошибка была исправлена"
[main 722513d] Ошибка была исправлена
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\user\Desktop\LABA1>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 328 bytes | 328.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/chillkirill/LABA1.git
 6fd9845..722513d  main -> main

C:\Users\user\Desktop\LABA1>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Рисунок 7. Распространение изменений

## 8. Подтверждение изменений в исходном репозитории.

|              |                               |                          |           |
|--------------|-------------------------------|--------------------------|-----------|
| chillkirill  | Ошибка была исправлена        | 722513d · 10 minutes ago | 7 Commits |
| .gitignore   | Initial commit                | 52 minutes ago           |           |
| LICENSE      | Initial commit                | 52 minutes ago           |           |
| README.md    | Внес изменения в файле README | 14 minutes ago           |           |
| Программа.py | Ошибка была исправлена        | 10 minutes ago           |           |

Рисунок 8. Изменённый репозиторий

## **Ответы на контрольные вопросы:**

### **1. Что такое СКВ и каково ее 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 чем недостатки локальных и централизованных СКВ?**

Локальные СКВ (системы контроля версий) менее удобны для совместной работы, поскольку требуют постоянной синхронизации изменений между участниками проекта. Централизованные СКВ, в свою очередь, могут быть уязвимы к отказам сервера, а также к потере данных, если главный репозиторий поврежден.

### **3. К какой СКВ относится Git?**

Git относится к децентрализованным системам контроля версий.

### **4. В чем концептуальное отличие Git от других СКВ?**

Git отличается от других СКВ тем, что он децентрализован: каждый разработчик имеет полную копию репозитория, а не только доступ к нему. Это позволяет работать автономно, делать коммиты локально и синхронизировать изменения с другими разработчиками.

### **5. Как обеспечивается целостность хранимых данных в Git?**

Git использует криптографическую хеш-функцию SHA-1 для вычисления уникального хэша для каждого файла и коммита. Изменения в файле изменяют его хэш, что позволяет Git отслеживать изменения и обеспечивать целостность данных.

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

Файлы в Git могут находиться в 3 состояниях:

- Untracked: файл не отслеживается Git, т.е. не включен в репозиторий.
- Staged: файл был изменен и подготовлен к коммиту, т.е. включен в следующий коммит.

## **7. Что такое профиль пользователя в GitHub?**

Профиль пользователя на GitHub — это публичная страница, которая представляет вас как разработчика. На ней отображаются ваши репозитории, активности, подписки, а также информация о вас: имя, местоположение, сайт, био, аватар.

## **8. Какие бывают репозитории в GitHub?**

- Публичные: доступны всем пользователям GitHub. Их код можно просматривать, скачивать и использовать бесплатно.

- Приватные: доступны только владельцу репозитория и приглашенным сотрудникам. Для приватных репозиторий нужна платная подписка GitHub.

- Также существуют fork-репозитории: это копии публичных репозиторий, которые можно модифицировать и использовать для собственных проектов. Fork-репозитории могут быть как публичными, так и приватными.

## **9. Укажите основные этапы модели работы с GitHub.**

Основные этапы работы с GitHub:

1. Создание репозитория: Создается новый репозиторий на GitHub, где будет храниться код проекта.

2. Клонирование репозитория: Локальная копия репозитория скачивается на компьютер с помощью команды `git clone`.

3. Работа с кодом: В локальной копии вносятся изменения в код проекта.

4. Добавление изменений в индекс: Измененные файлы добавляются в индекс с помощью команды `git add`.

5. Коммит: Изменения в индексе сохраняются в локальной истории с помощью команды `git commit`.

6. Пуш: Изменения из локального репозитория отправляются на сервер GitHub с помощью команды `git push`.

7. Pull: Изменения из репозитория GitHub скачиваются в локальную копию с помощью команды `git pull`.

## **10. Как осуществляется первоначальная настройка Git после установки?**

`git config` - это команда, которая позволяет вам настроить Git под свой стиль работы и предпочтения. Она используется для установки глобальных, локальных и системных параметров Git.

Глобальные параметры: применяются ко всем репозиториям на вашем компьютере. Их установка осуществляется с флагом `--global`.

Локальные параметры: применяются только к текущему репозиторию. Их установка осуществляется без флага `--global`.

Системные параметры: применяются ко всем пользователям системы. Их установка осуществляется с флагом `--system`.

## **11. Опишите этапы создания репозитория в GitHub.**

1. Авторизоваться на сайте.
2. Нажать кнопку "New" и ввести название, описание и тип доступа для нового репозитория.
3. Нажать кнопку "Create repository".

## **12. Какие типы лицензий поддерживаются GitHub при создании репозитория?**

- MIT: Позволяет свободное использование, модификацию и распространение, включая коммерческие цели.
- Public Domain: Отказывается от всех прав на программное обеспечение, позволяя свободное использование, модификацию и распространение, без каких-либо ограничений.
- Также GitHub предоставляет возможность выбрать "None" (без лицензии), что означает, что вы не предоставляете никаких прав на использование вашего кода.

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

Клонирование репозитория GitHub осуществляется с помощью команды `git clone`: `git clone <URL_репозитория>`

Клонирование репозитория создает локальную копию репозитория на вашем компьютере, что позволяет вам:

Сохранять историю изменений: Локальная копия сохраняет полную историю всех изменений в репозитории.

Совместная работа: Клонирование позволяет нескольким разработчикам работать над проектом одновременно и синхронизировать свои изменения.

#### **14. Как проверить состояние локального репозитория Git?**

Чтобы проверить состояние локального репозитория Git, используется команда `git status`.

Она покажет:

- Измененные файлы: Файлы, которые были изменены, но еще не добавлены в индекс.
- Файлы, добавленные в индекс: Файлы, которые были изменены и добавлены в индекс для следующего коммита.
- Неотслеживаемые файлы: Файлы, которые не отслеживаются Git (не входят в репозиторий).

**15. Как изменяется состояние локального репозитория Git после выполнения следующих операций: добавления/изменения файла в локальный репозиторий Git; добавления нового/ измененного 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` ?**

Команда `git push origin main` отправляет изменения из локальной истории на сервер GitHub. Локальные коммиты теперь доступны в удаленном репозитории.

**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`.

1. Клонировать репозиторий на оба компьютера: `git clone <URL_репозитория>`.
2. Работать над проектом на первом компьютере, фиксировать изменения (`git add`, `git commit`).
3. Отправить изменения на GitHub: `git push origin main`.
4. Обновить локальный репозиторий на втором компьютере: `git pull origin main`.
5. Повторить шаги 2-4 для второго компьютера.

**17. GitHub является не единственным сервисом, работающим с Git. Какие сервисы еще Вам известны? Приведите сравнительный анализ одного из таких сервисов с GitHub.**

Bitbucket: Разработан Atlassian, известен своей интеграцией с другими продуктами Atlassian, такими как Jira и Confluence. Предлагает как бесплатные, так и платные планы с поддержкой частных репозиторий.

GitHub и Bitbucket - популярные платформы для хостинга Git-репозиторий, но у них есть некоторые различия. GitHub предлагает бесплатные публичные репозитории и платные частные, в то время как Bitbucket предоставляет бесплатные частные репозитории для небольших команд и платные планы с расширенными функциями.

GitHub интегрируется с различными сервисами, такими как Travis CI и CircleCI, в то время как Bitbucket отлично интегрируется с продуктами Atlassian, такими как Jira и Confluence. GitHub предлагает более широкий набор функций для управления кодом, сотрудничества и развертывания, в то время как Bitbucket предлагает набор функций, достаточный для большинства проектов. Сообщество разработчиков на GitHub значительно

больше, чем на Bitbucket, но Bitbucket обеспечивает хорошую поддержку Atlassian.

**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 реализуются описанные в лабораторной работе операции Git с помощью одного из таких программных средств.**

GitHub Desktop: Разработан самой компанией GitHub, интегрирован с веб-платформой GitHub и предлагает простой и интуитивно понятный интерфейс.

- GitKraken: Известен своим стильным дизайном, большим количеством функций и поддержкой многих систем контроля версий, включая Git, Mercurial и SVN.

- Sourcetree: Разработан Atlassian, производителем Jira и Bitbucket, имеет хорошую интеграцию с этими сервисами и предлагает широкий набор функций для управления репозиториями.

- TortoiseGit: Бесплатная программа с открытым исходным кодом для Windows, известна своей интеграцией в контекстное меню Проводника Windows и простой использованием.

Как реализуются операции Git в GitHub Desktop

В GitHub Desktop необходимо нажать кнопку "Clone a repository" и ввести URL репозитория. После чего выбрать папку для клонирования.

**Вывод:** в ходе этой лабораторной работы я изучил основные возможности системы контроля версий Git и платформы GitHub.