

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №1.1**  
**дисциплины «Основы кроссплатформенного программирования»**

Выполнил:  
Арзютов Иван Владиславович  
1 курс, группа ИТС-б-о-22-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи»,  
направленность (профиль)  
«Инфокоммуникационные системы и  
сети», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р. А., доцент кафедры  
инфокоммуникаций

---

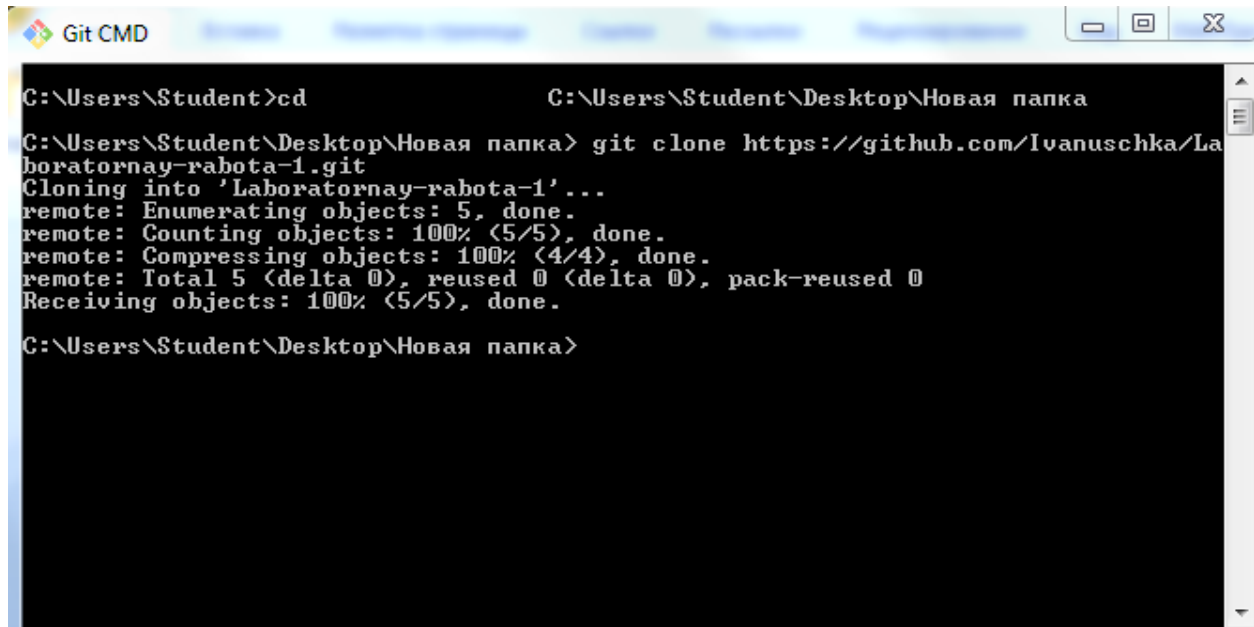
(подпись)

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

Ставрополь, 2023 г.

## Исследование основных возможностей Git и GitHub.

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

A screenshot of a Windows command prompt window titled "Git CMD". The window shows the execution of a git clone command. The user is in the directory C:\Users\Student\Desktop\Новая папка. The command executed is git clone https://github.com/Ivanuschka/Laboratory-work-1.git. The output shows the cloning process, including enumerating objects, counting objects, and compressing objects, all completed successfully. The final prompt is C:\Users\Student\Desktop\Новая папка>.

```
C:\Users\Student>cd C:\Users\Student\Desktop\Новая папка
C:\Users\Student\Desktop\Новая папка>git clone https://github.com/Ivanuschka/Laboratory-work-1.git
Cloning into 'Laboratory-work-1'...
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
Receiving objects: 100% (5/5), done.
C:\Users\Student\Desktop\Новая папка>
```

### Копирование репозитория.

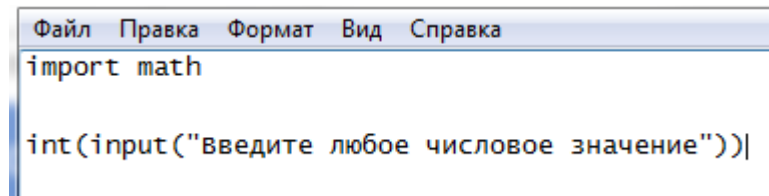
Клонировал репозиторий в нужную папку на рабочий компьютер, через Git CMD, а не через кривой GitBASH, но ровный если знать команды Linux-а.



“Laboratornaya rabota \_1”, а эта папка находится в папке “Новая папка” .

Вставил новый путь к файлам своего репозитория, для проверки статуса данных. Сперва мой путь выглядел так - C:\Users\Student\Desktop\Новая папка. Затем так - C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1.

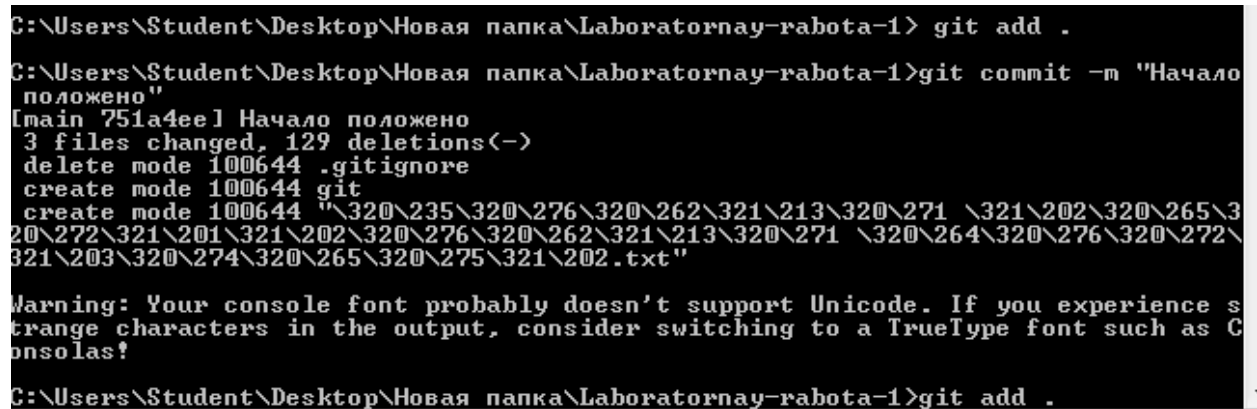
И все заработало.



```
Файл  Правка  Формат  Вид  Справка
import math

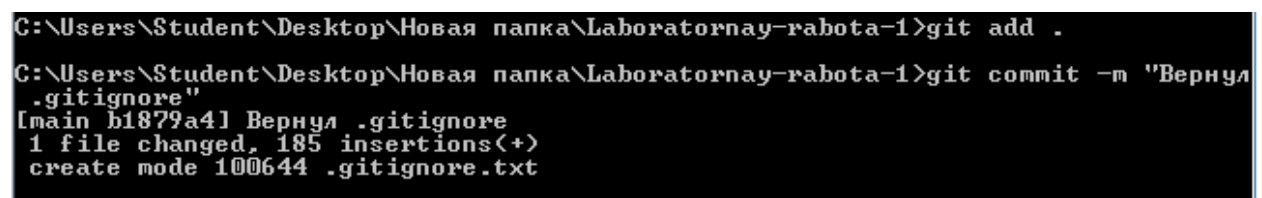
int(input("Введите любое числовое значение"))|
```

Создание программы с последующими Commit-ами.



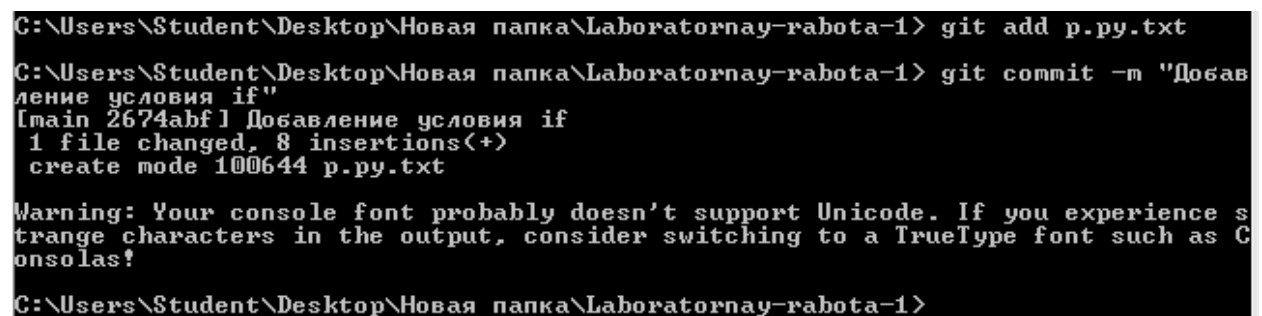
```
C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1> git add .
C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1> git commit -m "Начало
положено"
[main 751a4ee] Начало положено
3 files changed, 129 deletions(-)
delete mode 100644 .gitignore
create mode 100644 git
create mode 100644 "\320\235\320\276\320\262\321\213\320\271 \321\202\320\265\3
20\272\321\201\321\202\320\276\320\262\321\213\320\271 \320\264\320\276\320\272\
321\203\320\274\320\265\320\275\321\202.txt"
Warning: Your console font probably doesn't support Unicode. If you experience s
trange characters in the output, consider switching to a TrueType font such as C
onsole!
C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1> git add .
```

Первый commit –m.



```
C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1> git add .
C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1> git commit -m "Вернул
.gitignore"
[main b1879a4] Вернул .gitignore
1 file changed, 185 insertions(+)
create mode 100644 .gitignore.txt
```

Второй commit –m.



```
C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1> git add p.py.txt
C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1> git commit -m "Добав
ление условия if"
[main 2674abf] Добавление условия if
1 file changed, 8 insertions(+)
create mode 100644 p.py.txt
Warning: Your console font probably doesn't support Unicode. If you experience s
trange characters in the output, consider switching to a TrueType font such as C
onsole!
C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1>
```

Третий commit –m.

```

C:\Users\Student>cd C:\Users\Student\Desktop\Новая папка

C:\Users\Student\Desktop\Новая папка> git clone https://github.com/Ivanuschka/Laboratornay-rabota-1.git
Cloning into 'Laboratornay-rabota-1'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 20 (delta 4), reused 12 (delta 3), pack-reused 0
Receiving objects: 100% (20/20), 5.52 KiB | 2.76 MiB/s, done.
Resolving deltas: 100% (4/4), done.

C:\Users\Student\Desktop\Новая папка> git add .
fatal: not a git repository (or any of the parent directories): .git

C:\Users\Student\Desktop\Новая папка> cd C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1

C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1> git add .

C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1> git commit -m "Конец программы"
'dit' не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1> git commit -m "Конец программы"
[main 9b03400] Конец программы
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 "\320\235\320\276\320\262\321\213\320\271 \321\202\320\265\320\272\321\201\321\202\320\276\320\262\321\213\320\271 \320\264\320\276\320\272\321\203\320\274\320\265\320\275\321\202.txt"

Warning: Your console font probably doesn't support Unicode. If you experience strange characters in the output, consider switching to a TrueType font such as Consolas!

C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1> git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 250 bytes | 250.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Ivanuschka/Laboratornay-rabota-1.git
5e24f8c..9b03400 main -> main

C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1>

```

Четвертый и пятый commit-ы.

```

C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1> git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 250 bytes | 250.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Ivanuschka/Laboratornay-rabota-1.git
   5e24f8c..9b03400  main -> main

C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1>git add .

C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1>git commit -m "Послед
ние изменения"
[main 2aa5202] Последние изменения
   1 file changed, 5 insertions(+)

Warning: Your console font probably doesn't support Unicode. If you experience s
trange characters in the output, consider switching to a TrueType font such as C
onsolas!

C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1>git push
To https://github.com/Ivanuschka/Laboratornay-rabota-1.git
   ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/Ivanuschka/Laboratornay-r
abota-1.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1>git add .

C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1>git pull
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 2.62 KiB | 11.00 KiB/s, done.
From https://github.com/Ivanuschka/Laboratornay-rabota-1
   9b03400..a587b2e  main      -> origin/main
Merge made by the 'ort' strategy.
 .gitignore.txt => .gitignore | 2 +-
   1 file changed, 1 insertion(+), 1 deletion(-)
 rename .gitignore.txt => .gitignore (99%)

C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1>git commit -m "Конец"

On branch main
Your branch is ahead of 'origin/main' by 2 commits.
   (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1>git push
Enumerating objects: 9, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 715 bytes | 715.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Ivanuschka/Laboratornay-rabota-1.git
   a587b2e..63696cc  main -> main

C:\Users\Student\Desktop\Новая папка\Laboratornay-rabota-1>

```

Остальные commit-ы.

**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 этих файлов.

### 2. В чем недостатки локальных и централизованных СКВ?

Это единая точка отказа, представленная централизованным сервером. Если этот сервер выйдет из строя на час, то в течение этого времени никто не сможет использовать контроль версий для сохранения изменений, над которыми работает, а также никто не сможет обмениваться этими изменениями с другими разработчиками.

### 3. К какой СКВ относится Git?

К распределённым системам контроля версий.

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

Git не хранит и не обрабатывает данные таким же способом как другие СКВ.

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

В Git для всего вычисляется хеш-сумма, и только потом происходит сохранение. В дальнейшем обращение к сохранённым объектам происходит по этой хеш-сумме.

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

1) Зафиксированный значит, что файл уже сохранён в вашей локальной базе;

2) К изменённым относятся файлы, которые поменялись, но ещё не были зафиксированы;

3) Подготовленные файлы — это изменённые файлы, отмеченные для включения в следующий коммит.

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

Профиль - это наша публичная страница на GitHub, как и в социальных сетях. В нем другие пользователи могут посмотреть ваши работы.

8. Какие бывают репозитории в GitHub?

Локальные и удаленные репозитории.

9. Укажите основные этапы модели работы с GitHub.

- 1) Регистрация;
- 2) Создание репозитория;
- 3) Клонирование репозитория;
- 4) Добавление новых файлов.

10. Как осуществляется первоначальная настройка Git после установки?

Убедимся, что Git установлен используя команду: `git version`. Перейдём в папку с локальным репозиторием используя команду: `cd /d <Расположения папки на компьютере>`. Свяжем локальный репозиторий и удалённый командами: `git config --global user.name <YOUR_NAME>` `git config --global user.email <EMAIL>`.

11. Опишите этапы создания репозитория в GitHub.

- 1) В правом верхнем углу, рядом с аватаром есть кнопка с плюсиком, нажимая которую мы переходим к созданию нового репозитория;
- 2) В результате будет выполнен переход на страницу создания репозитория. Наиболее важными на ней являются следующие поля: Имя репозитория. Описание (Description). Public/private. “Initialize this repository with a README” .gitignore и LICENSE.

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

Microsoft Reciprocal License, The Code Project Open License (CPOL), The Common Development and Distribution License (CDDL), The Microsoft Public License (Ms-PL), The Mozilla Public License 1.1 (MPL 1.1), The Common Public License Version 1.0 (CPL), The Eclipse Public License 1.0, The MIT License, The BSD License, The Apache License, Version 2.0, The Creative Commons Attribution-ShareAlike 2.5 License, The zlib/libpng License, A Public Domain dedication, The Creative Commons Attribution 3.0 Unported License, The Creative Commons).



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

После создания репозитория его необходимо клонировать на ваш компьютер. Для этого на странице репозитория необходимо найти кнопку Clone или Code и щелкнуть по ней, чтобы отобразить адрес репозитория для клонирования.

Откройте командную строку или терминал и перейдите в каталог, куда вы хотите скопировать хранилище. Затем напишите `git clone` и введите адрес.

14. Как проверить состояние локального репозитория Git?

`git status`

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` ?

Файлы обновятся на 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.

`git clone.`

`git pull.`

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

1) GitLab — альтернатива GitHub номер один. GitLab предоставляет не только веб-сервис для совместной работы, но и программное обеспечение с открытым исходным кодом;

2) BitBucket — это служба хостинга репозитория и управления версиями от Atlassian. Она тесно интегрирована с другими инструментами Atlassian — Jira, HipChat и Confluence.

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 это совершенно бесплатное приложение с открытым исходным кодом, разработанное GitHub. С его помощью можно взаимодействовать с GitHub (что и не удивительно), а также с другими платформами (включая Bitbucket и GitLab).

**Вывод:** В ходе работы усвоил и освоил первые и самые важные команды для пользования репозиторием. Исследовал и буду исследовать базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.