

**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 РАБОТЕ №1.2  
дисциплины «Основы кроссплатформенного программирования»**

Выполнил:  
Сластёнов Андрей Сергеевич  
1 курс, группа ИТС-б-о-21-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи»,  
направленность (профиль) Сети связи  
и системы коммутации,  
очная форма обучения

---

(подпись)

Руководитель практики от  
университета:  
Воронкин Р.А., канд. тех. наук, доцент,  
доцент кафедры инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г

**Тема:** исследование возможностей Git для работы с локальными репозиториями.

**Цель работы:** исследовать базовые возможности системы контроля версий Git для работы с локальными репозиториями.

### Порядок выполнения работы:

1. Создал новый репозиторий и клонировал его на свой компьютер.

```
C:\Users\Andre>git clone https://github.com/ReinhardSailo/lab1.2.git
Cloning into 'lab1.2'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
C:\Users\Andre> cd C:\Users\Andre\lab1.2
```

Рисунок 1. Новый репозиторий

2. Добавил некоторое правило в файл gitignore, чтобы Git игнорировал файлы в формате .idea

```
# Prerequisites
*.d

# Compiled Object files
*.slo
*.lo
*.o
*.obj

# Precompiled Headers
*.gch
*.pch

# Compiled Dynamic libraries
*.so
*.dylib
*.dll

# Fortran module files
*.mod
*.smod

# Compiled Static libraries
*.lai
*.la
*.a
*.lib

# Executables
*.exe
*.out
*.app

# idea File
.idea/
.idea
```

Рисунок 2. Работа с gitignore

3. Добавил информацию в файл README.md о дисциплине, группе и ФИО.

```
1 # laba1.2
2 Slastyonov Andrey, ITS-b-o-22-1, crossplatfom programming
3
```

Рисунок 3. Работа с README

4. Написал программу в новом файле main.cpp, сделал не менее 7-ми коммитов с 4-мя тегами.

```
commit 97c914e1568a5bc1f56b1416c0cf69baacab5924 (HEAD -> main)
Author: Andrey <mrsailoris@gmail.com>
Date: Thu Jun 29 22:18:58 2023 +0300
```

Изменены строки приветствия

```
commit b5a3b353d4b98c9fd7e3d2342fd9c9c302c75f2f
Author: Andrey <mrsailoris@gmail.com>
Date: Thu Jun 29 22:18:21 2023 +0300
```

Изменена функция calculate\_sum

```
commit 4cccd52e29ff73ba59e2650d266ef9900bf09d346
Author: Andrey <mrsailoris@gmail.com>
Date: Thu Jun 29 22:17:45 2023 +0300
```

Добавлена новая функция

```
commit f76300d19a5f04517bb199067946c1a93981df61
Author: Andrey <mrsailoris@gmail.com>
Date: Thu Jun 29 22:17:13 2023 +0300
```

Изменена функция print\_name

```
commit 32752e9de4a9dfa6efc2ebd9f729233409f94fe3
Author: Andrey <mrsailoris@gmail.com>
Date: Thu Jun 29 22:16:25 2023 +0300
```

Изменена функция print\_hello

```
:|
```

Рисунок 4. История хранилища

Задание 5. Посмотрел содержимое коммитов командой `git show`, где : 1) HEAD : последний коммит;

```

C:\Users\Andre\laba1.2>git show HEAD
commit 97c914e1568a5bc1f56b1416c0cf69baacab5924 (HEAD -> main)
Author: Andrey <mrsailoris@gmail.com>
Date: Thu Jun 29 22:18:58 2023 +0300

    Изменены строки приветствия

diff --git a/README.md b/README.md
index b5468be..22ec527 100644
--- a/README.md
+++ b/README.md
@@ -4,4 +4,5 @@ Slastyonov Andrey, ITS-b-o-22-1, crossplatfom programming
     def calculate_sum(a, b):
         return a + b + 1

-print(calculate_sum(2, 3))
\ No newline at end of file
+print("Привет, мир!")
+print("Hello, world!")
\ No newline at end of file

C:\Users\Andre\laba1.2>

```

Рисунок 5. Последний коммит

2) HEAD~1 : предпоследний коммит.

```

C:\Users\Andre\laba1.2>git show HEAD~1
commit b5a3b353d4b98c9fd7e3d2342fdcf9c302c75f2f
Author: Andrey <mrsailoris@gmail.com>
Date: Thu Jun 29 22:18:21 2023 +0300

    Изменена функция calculate_sum

diff --git a/README.md b/README.md
index b1355e2..b5468be 100644
--- a/README.md
+++ b/README.md
@@ -2,6 +2,6 @@
 Slastyonov Andrey, ITS-b-o-22-1, crossplatfom programming

     def calculate_sum(a, b):
-        return a + b
+        return a + b + 1

     print(calculate_sum(2, 3))
\ No newline at end of file

C:\Users\Andre\laba1.2>

```

Рисунок 6. Предпоследний коммит.

5. Откат к заданной версии.

1.1. Удалил весь программный код с файла main.cpp и сохранил его.



Рисунок 7. Удаление программы

Код вновь вернулся.

1.3. Вновь повторил пункт 1.1. и сделал коммит.

1.4. Откатить состояние хранилища к предыдущей версии командой: `git reset --hard HEAD~1` .

```
C:\Work\lab1.2>git commit -m "Yedalenie koda"
[main 9cf5a9c] Yedalenie koda
1 file changed, 31 deletions(-)

C:\Work\lab1.2>git reset --hard HEAD~1
HEAD is now at 49e4dce 7 commit

C:\Work\lab1.2>
```

Рисунок 8. Возвращение к предпоследней версии коммита

Код вновь вернулся.

**Ссылка:**

#### Ответы на контрольные вопросы:

1) Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов? Историю коммитов можно выполнить с помощью команды `git log`. Дополнительные

опции для просмотра истории: %H, %h, %T, %t, %P, %p тд. -p, --stat, --shortstat, --name-only, --name-status и тд.

2) Как ограничить вывод при просмотре истории коммитов?

Ограничить вывод при просмотре истории коммитов можно с помощью команды `git log -n`, где `n` – число последних коммитов.

3) Как внести изменения в уже сделанный коммит? Если вы хотите переделать коммит — внесите необходимые изменения, добавьте их в индекс и сделайте коммит ещё раз, указав параметр `--amend` : `git commit --amend`.

4) Как отменить индексацию файла в Git? Отменить индексацию файла можно с помощью команды: `git reset HEAD` .

5) Как отменить изменения в файле? Отменить изменения в файле можно с помощью команды: `git checkout -` -

6) Что такое удаленный репозиторий Git? Удалённые репозитории представляют собой версии вашего проекта, сохранённые в интернете или ещё где-то в сети.

7) Как выполнить просмотр удаленных репозиториях данного локального репозитория? Выполнить просмотр удаленных репозиториях данного локального репозитория можно с помощью команды: `git remote`.

8) Как добавить удаленный репозиторий для данного локального репозитория? Для того, чтобы добавить удалённый репозиторий и присвоить ему имя (shortname), просто выполните команду `git remote add` .

9) Как выполнить отправку/получение изменений с удаленного репозитория? Для получения данных из удалённых проектов, следует выполнить: `git fetch [remote-name]`. Для отправки изменений в удаленный репозиторий используется команда: `git push`

10) Как выполнить просмотр удаленного репозитория? Если хотите получить побольше информации об одном из удалённых репозиториях, вы можете использовать команду: `git remote show` .

11) Каково назначение тэгов Git? Git имеет возможность пометить определённые моменты в истории как важные. Для таких случаев были придуманы тэги.

12) Как осуществляется работа с тэгами Git? Просмотреть список имеющихся тегов в Git можно очень просто. Достаточно набрать команду `git tag`. Создание аннотированного тега в Git выполняется легко. Самый простой способ — это указать -a при выполнении команды `tag`. С помощью команды `git show` вы можете посмотреть данные тега вместе с коммитом. По умолчанию, команда `git push` не отправляет теги на удалённые сервера. После создания теги нужно отправлять явно на удалённый сервер. Процесс аналогичен отправке веток — достаточно выполнить команду `git push origin`. Для удаления тега в локальной репозитории достаточно выполнить команду `git tag -d`. Если вы хотите получить версии файлов, на которые указывает тег, то вы можете сделать `git checkout` для тега.

13) Самостоятельно изучите назначение флага `--prune` в командах `git fetch` и `git push`. Каково назначение этого флага? `Git prune` — это команда, которая удаляет все файлы, недоступные из текущей ветки. Команда `prune` полезна, когда в вашем рабочем каталоге много файлов, которые вы не хотите хранить. `git fetch --prune` делает то же самое: удалит ссылки на ветки, которые не существуют на удалённом компьютере. Опция `--prune` в команде `git push` удалит ветку из удалённого репозитория, если в локальной репозитории не существует ветки с таким именем.

**Вывод:** исследовал базовые возможности системы контроля версий Git для работы с локальными репозиториями