

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 №2
дисциплины «Программирование на Python»
Вариант ____

Выполнил:
Дзуев Альбрт Мухаметоич
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных
систем», очная форма обучения

(подпись)

Руководитель практики:
Руководитель практики: Воронкин
Р.А., канд. техн. наук, доцент кафедры
инфокоммуникации

(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. Создание нового репозитория и проработка команд показанных в методических материалах.

```
dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол
$ git clone https://github.com/dzeall00/Python2.git
Cloning into 'Python2'...
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), 4.18 KiB | 4.18 MiB/s, done.
```

Картинка 1. Клонирование нового репозитория.

```
dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git log
commit 82281b673f701d9ef70dea13993b3969aa43d055 (HEAD -> main)
Author: dzeall00 <dzuevalbert0@gmail.com>
Date: Sat Nov 4 17:50:39 2023 +0300

    Введене недостающих компанентов

commit f23f82bfb652b3a5c1f5c9ce2f7535dd7aae8794
Author: dzeall00 <dzuevalbert0@gmail.com>
Date: Sat Nov 4 17:47:56 2023 +0300

    Введение переменных

commit 1555dce42a24aefcd263c2c72268a39adf98483f (tag: v0.1)
Author: dzeall00 <dzuevalbert0@gmail.com>
Date: Sat Nov 4 17:44:37 2023 +0300

    Начало программы

commit 9467bd01c68afaceccb1b77b2a3e736a322e1cc0
Author: dzeall00 <dzuevalbert0@gmail.com>
Date: Sat Nov 4 16:28:03 2023 +0300
```

Картинка 2. Команда git log.

```

dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git log -p -2
commit 9467bd01c68afaceccb1b77b2a3e736a322e1cc0 (HEAD -> main)
Author: dzeall100 <dzuevalbert0@gmail.com>
Date: Sat Nov 4 16:28:03 2023 +0300

    Оформление README

diff --git a/README.md b/README.md
index ac6724c..ae8b427 100644
--- a/README.md
+++ b/README.md
@@ -1,1 @@
-# Python2
\ No newline at end of file
+Дзюев Альберт Мухаметович ИВТ-6-о-22-1 вторая лабораторная работа по дисциплине
  Программирование на Python
\ No newline at end of file

commit fd4875b11e85bb6358847639031dc1ff2c0fa353 (origin/main, origin/HEAD)
Author: dzeall100 <145676686+dzeall100@users.noreply.github.com>
Date: Sat Nov 4 16:21:05 2023 +0300

    Initial commit

...skipping...
commit 9467bd01c68afaceccb1b77b2a3e736a322e1cc0 (HEAD -> main)
Author: dzeall100 <dzuevalbert0@gmail.com>
Date: Sat Nov 4 16:28:03 2023 +0300

    Оформление README

diff --git a/README.md b/README.md
index ac6724c..ae8b427 100644
--- a/README.md
+++ b/README.md
@@ -1,1 @@
-# Python2
\ No newline at end of file
+Дзюев Альберт Мухаметович ИВТ-6-о-22-1 вторая лабораторная работа по дисциплине Программирование на Python
\ No newline at end of file

commit fd4875b11e85bb6358847639031dc1ff2c0fa353 (origin/main, origin/HEAD)
Author: dzeall100 <145676686+dzeall100@users.noreply.github.com>
Date: Sat Nov 4 16:21:05 2023 +0300

    Initial commit

diff --git a/.gitignore b/.gitignore
new file mode 100644
index 0000000..8a30d25
--- /dev/null
+++ b/.gitignore
@@ -0,0 +1,398 @@

```

Картинка 3. Команда git log с аргументом -p и параметром -2.

```

dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git log --stat
commit 9467bd01c68afaceccb1b77b2a3e736a322e1cc0 (HEAD -> main)
Author: dzeall100 <dzuevalbert0@gmail.com>
Date: Sat Nov 4 16:28:03 2023 +0300

    Оформление README

README.md | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit fd4875b11e85bb6358847639031dc1ff2c0fa353 (origin/main, origin/HEAD)
Author: dzeall100 <145676686+dzeall100@users.noreply.github.com>
Date: Sat Nov 4 16:21:05 2023 +0300

    Initial commit

.gitignore | 398 +++++
LICENSE | 21 +
README.md | 1 +
3 files changed, 420 insertions(+)

```

Картинка 4. Команда git log с опцией --stat.

```
dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git log --pretty=oneline
9467bd01c68afaceccb1b77b2a3e736a322e1cc0 (HEAD -> main) Оформление README
fd4875b11e85bb6358847639031dc1ff2c0fa353 (origin/main, origin/HEAD) Initial commit
```

Картинка 5. Команда git log --pretty=oneline.

```
dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git log --pretty=format:"%h - %an, %ar : %s"
9467bd0 - dzeall00, 7 minutes ago : Оформление README
fd4875b - dzeall00, 14 minutes ago : Initial commit
```

Картинка 6. Команда git log --pretty=format.

```
dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git log --pretty=format:"%h %s" --graph
* 9467bd0 Оформление README
* fd4875b Initial commit
```

Картинка 7. Команда git log --pretty=format:"%h %s" --graph.

```
dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git log --since=2.weeks
commit 9467bd01c68afaceccb1b77b2a3e736a322e1cc0 (HEAD -> main)
Author: dzeall00 <dzuevalbert0@gmail.com>
Date: Sat Nov 4 16:28:03 2023 +0300

    Оформление README

commit fd4875b11e85bb6358847639031dc1ff2c0fa353 (origin/main, origin/HEAD)
Author: dzeall00 <145676686+dzeall00@users.noreply.github.com>
Date: Sat Nov 4 16:21:05 2023 +0300

    Initial commit
```

Картинка 8. Команда git log --since=2.weeks.

```
dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git remote
origin

dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git remote -v
origin https://github.com/dzeall00/Python2.git (fetch)
origin https://github.com/dzeall00/Python2.git (push)
```

Картинка 9. Команда git remote и с ключем -v.

```
dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git remote show origin
* remote origin
Fetch URL: https://github.com/dzeall00/Python2.git
Push URL: https://github.com/dzeall00/Python2.git
HEAD branch: main
Remote branch:
  main tracked
Local branch configured for 'git pull':
  main merges with remote main
Local ref configured for 'git push':
  main pushes to main (up to date)
```


Картинка 10. Команда git remote show origin.

```
dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git tag

dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git tag -l "v1.8.5*"

dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git tag -a v1.4 -m "my version 1.4"

dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git tag
v1.4
```

Картинка 11. Команда git tag с различными ключами.

```
dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git show v1.4
tag v1.4
Tagger: dzeall00 <dzuevalbert0@gmail.com>
Date: Sat Nov 4 16:51:06 2023 +0300

my version 1.4

commit 9467bd01c68afaceccb1b77b2a3e736a322e1cc0 (HEAD -> main, tag: v1.4, origin
/main, origin/HEAD)
Author: dzeall00 <dzuevalbert0@gmail.com>
Date: Sat Nov 4 16:28:03 2023 +0300

    Оформление README

diff --git a/README.md b/README.md
index ac6724c..ae8b427 100644
--- a/README.md
+++ b/README.md
@@ -1,1 @@
-# Python2
\ No newline at end of file
+Дзюев Альберт Мухаметович ИВТ-6-о-22-1 вторая лабораторная работа по дисциплине
+ Программирование на Python
\ No newline at end of file
:..skipping...
tag v1.4
Tagger: dzeall00 <dzuevalbert0@gmail.com>
Date: Sat Nov 4 16:51:06 2023 +0300

my version 1.4

commit 9467bd01c68afaceccb1b77b2a3e736a322e1cc0 (HEAD -> main, tag: v1.4, origin/main, origin/HEAD)
Author: dzeall00 <dzuevalbert0@gmail.com>
Date: Sat Nov 4 16:28:03 2023 +0300

    Оформление README

diff --git a/README.md b/README.md
index ac6724c..ae8b427 100644
--- a/README.md
+++ b/README.md
@@ -1,1 @@
-# Python2
\ No newline at end of file
+Дзюев Альберт Мухаметович ИВТ-6-о-22-1 вторая лабораторная работа по дисциплине Программирование на Python
\ No newline at end of file
```

Картинка 12. Команда просмотра данных выбранного тега.

```
dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git push origin --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 164 bytes | 164.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/dzeall00/Python2.git
 * [new tag]          v1.4 -> v1.4
```

Картинка 13. Команда для отправки всех тегов в удалённый репозиторий.

```
dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git tag -d v1.4
Deleted tag 'v1.4' (was ec4a73b)
```

Картинка 14. Команда для удаления тега.

```
dzeu@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git push origin :refs/tags/v1.4
To https://github.com/dzeall00/python2.git
- [deleted]          v1.4
```

Картинка 15. Команда для удаления тега с внешних серверов.

```
dzuev@LAPTOP-QEQHF3D5 MINGW32 ~/OneDrive/Рабочий стол/Python2 (main)
$ git log --graph --pretty=oneline --abbrev-commit
* 2d1708c (HEAD -> main, tag: v1.0, tag: v0.5, origin/main, origin/HEAD) Заверше
ние программы
* 82281b6 Введены недостающих компонентов
* f23f82b Введение переменных
* 1555dce (tag: v0.1) Начало программы
* 9467bd0 Оформление README
* fd4875b Initial commit
```

Картинка 16. Просмотр имеющихся коммитов

[illegible]

Картинка 17. Просмотр последнего коммита

```
commit 82281b23f701d8ef70deaf39b3b369aa43d055
Author: dzeal100 <dzeal100@gmail.com>
Date: Sat Nov 4 17:50:39 2023 +0300

Введение недостающих компонентов

diff --git "a/320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\320\237\320\270\321\202\320\276\320\2752\320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\320\237\320\270\321\202\320\276\320\2752.cpp" "b/320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\320\237\320\270\321\202\320\276\320\2752.cpp"
+++ b/320\237\321\200\320\276\320\263\321\200\320\260\320\274\320\274\320\260\320\237\320\270\321\202\320\276\320\2752.cpp
@@ -18,4 +18,17 @@ double min(double x, double y)
 }
 return y;
 }
 return x;
 }
No newline at end of file
bool thc(double x, double y, double z, double w, double a, double b)
{
    double k, c;
    if (x == 0) {
        return (a == x && b == min(y, z) && x == max(y, w));
    }
    k = (w - y) / (x - w);
    c = y - k * w;
    return b == a * k + c;
}
```

Картинка 18. Просмотр пред последнего коммита.

Ответы на контрольные вопросы:

1. Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов? После того, как вы создали несколько коммитов или же клонировали репозиторий с уже существующей историей коммитов, вероятно Вам понадобится возможность посмотреть, что было сделано — историю коммитов. Одним из основных и наиболее мощных инструментов для этого является команда `git log`. Команда `git log` имеет очень большое количество опций для поиска коммитов по разным критериям. Рассмотрим наиболее популярные из них. Одним из самых полезных аргументов является `-p` или `--patch`, который показывает разницу (выводит патч), внесенную в каждый коммит. Если вы хотите увидеть сокращенную статистику для каждого коммита, вы можете использовать опцию `--stat`. Следующей действительно полезной опцией является `--pretty`. Эта опция меняет формат вывода. Существует несколько встроенных вариантов отображения. Опция `oneline` выводит каждый коммит в одну строку, что может быть очень удобным если вы просматриваете большое количество коммитов. К тому же, опции `short`, `full` и `fuller` делают вывод приблизительно в том же формате, но с меньшим или большим количеством информации соответственно. Наиболее

интересной опцией является `format`, которая позволяет указать формат для вывода информации.

2. Как ограничить вывод при просмотре истории коммитов? В дополнение к опциям форматирования вывода, команда `git log` принимает несколько опций для ограничения вывода — опций, с помощью которых можно увидеть определенное подмножество коммитов. Одна из таких опций — это опция `-2`, которая показывает только последние два коммита. В действительности вы можете использовать `-n`, где `n` — это любое натуральное число и представляет собой `n` последних коммитов. На практике вы не будете часто использовать эту опцию, потому что Git по умолчанию использует постраничный вывод, и вы будете видеть только одну страницу за раз. Опции для ограничения вывода по времени, такие как `--since` и `--until`, являются очень удобными. Опция `--author` дает возможность фильтровать по автору коммита, а опция `--grep` искать по ключевым словам в сообщении коммита. Следующим действительно полезным фильтром является опция `-S`, которая принимает аргумент в виде строки и показывает только те коммиты, в которых изменение в коде повлекло за собой добавление или удаление этой строки. Последней полезной опцией, которую принимает команда `git log` как фильтр, является путь. Если вы укажете каталог или имя файла, вы ограничите вывод только теми коммитами, в которых были изменения этих файлов. Эта опция всегда указывается последней после двойного тире (`--`), чтобы отделить пути от опций.

3. Как внести изменения в уже сделанный коммит? Отмена может потребоваться, если вы сделали коммит слишком рано, например, забыв добавить какие-то файлы или комментарий к коммиту. Если вы хотите переделать коммит — внесите

необходимые изменения, добавьте их в индекс и сделайте коммит ещё раз, указав параметр `–amend`.

4. Как отменить индексацию файла в Git? Использовать `git reset HEAD ...` для исключения из индекса.

5. Как отменить изменения в файле? Использовать `git checkout --` для возвращения к версии из последнего коммита.

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? Для того, чтобы просмотреть список настроенных удалённых репозиториях, вы можете запустить команду `git remote`. Она выведет названия доступных удалённых репозиториях. Если вы клонировали репозиторий, то увидите как минимум `origin` — имя по умолчанию, которое Git даёт серверу, с которого производилось клонирование.

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]`. Когда вы хотите поделиться своими наработками, вам необходимо отправить их в удалённый репозиторий. Команда для этого действия простая: `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` .

Она выдаёт URL удалённого репозитория, а также информацию об отслеживаемых ветках.

11. Каково назначение тэгов Git? Как и большинство СКВ, Git имеет возможность пометить определённые моменты в истории как важные. Как правило, эта функциональность используется для отметки моментов выпуска версий (v1.0, и т. п.). Такие пометки в Git называются тегами.

12. Как осуществляется работа с тэгами Git? Просмотреть список имеющихся тегов в Git можно очень просто. Достаточно набрать команду `git tag` (параметры `-l` и `--list` опциональны). Создание аннотированного тега в Git выполняется легко. Самый простой способ — это указать `-a` при выполнении команды `tag`. По умолчанию, команда `git push` не отправляет теги на удалённые сервера. После создания тега нужно отправлять явно на удалённый сервер. Процесс аналогичен отправке веток — достаточно выполнить команду `git push origin`. Для удаления тега в локальном 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. Однако, это переведёт репозиторий в состояние «detached HEAD», которое имеет ряд неприятных побочных эффектов.

13. Самостоятельно изучите назначение флага `--prune` в командах `git fetch` и `git push`. Каково назначение этого флага? Исходя из описания, предоставленного `git help fetch`: `--prune` используется для удаления ссылок удалённого отслеживания, которые больше не существуют в удалённом репозитории, а из описания, предоставленного `git help push`: `--prune` используется для удаления ветвей на удалённом репозитории, для которых нет аналога в локальном репозитории.

Вывод: в результате выполнения работы были исследованы возможности Git для работы с локальными репозиториями.