

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

Выполнила:

Дмитриченко Марат Заурович  
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.

**Тема:** Исследование возможностей Git для работы с локальными репозиториями

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

### Ход работы:

**Задание 1:** Просмотреть историю (журнал) хранилища командой git log

```
shellmor@shellmor-virtual-machine:~/PycharmProjects/lab-1.2$ git log --graph --pretty=oneline --abbrev-commit
* 8bfcd80 (HEAD -> main) add test
* 61f88a7 add return
* ede15b6 end def
* 867bca1 add for range
* a3a2dcd add def
* 23adf65 add file program
* 803774b add README
* bcbd1da (origin/main, origin/HEAD) Initial commit
```

Рисунок 1. Просмотр истории хранилища командой git log

**Задание 2:** Просмотреть содержимое коммитов командой git show

```
shellmor@shellmor-virtual-machine:~/PycharmProjects/lab-1.2$ git show HEAD~1
commit 61f88a72726c67ef9b4658f4dc34e370340cf581
Author: maratdmirichenki <maratdmirichenki@gmail.com>
Date: Wed Apr 26 14:42:04 2023 +0300

    add return

diff --git a/program.py b/program.py
index 0474819..587a404 100644
--- a/program.py
+++ b/program.py
@@ -4,3 +4,6 @@ def b_sorter(array):
     for j in range(0, length-i-1):
         if array[j] > array[j+1]:
             array[j], array[j+1] = array[j+1], array[j]
+    return array
+
+
shellmor@shellmor-virtual-machine:~/PycharmProjects/lab-1.2$ git show HEAD
commit 8bfcd802b734f725cacfd6405ecb2cdea784b0 (HEAD -> main)
Author: maratdmirichenki <maratdmirichenki@gmail.com>
Date: Wed Apr 26 14:42:13 2023 +0300

    add test

diff --git a/program.py b/program.py
index 587a404..c25d045 100644
--- a/program.py
+++ b/program.py
@@ -7,3 +7,6 @@ def b_sorter(array):
     return array

+array_for_sort = [4, 1, 3, 2, 6, 5, 7]
+
+print(b_sorter(array=array_for_sort))
\ No newline at end of file
shellmor@shellmor-virtual-machine:~/PycharmProjects/lab-1.2$ git show 61f88a7
commit 61f88a72726c67ef9b4658f4dc34e370340cf581
Author: maratdmirichenki <maratdmirichenki@gmail.com>
Date: Wed Apr 26 14:42:04 2023 +0300

    add return

diff --git a/program.py b/program.py
index 0474819..587a404 100644
--- a/program.py
+++ b/program.py
@@ -4,3 +4,6 @@ def b_sorter(array):
     for j in range(0, length-i-1):
         if array[j] > array[j+1]:
             array[j], array[j+1] = array[j+1], array[j]
+    return array
+
+

```

Рисунок 2. Просмотр содержимого коммитов командой git show

### Задание 3: Освойте возможность отката к заданной версии

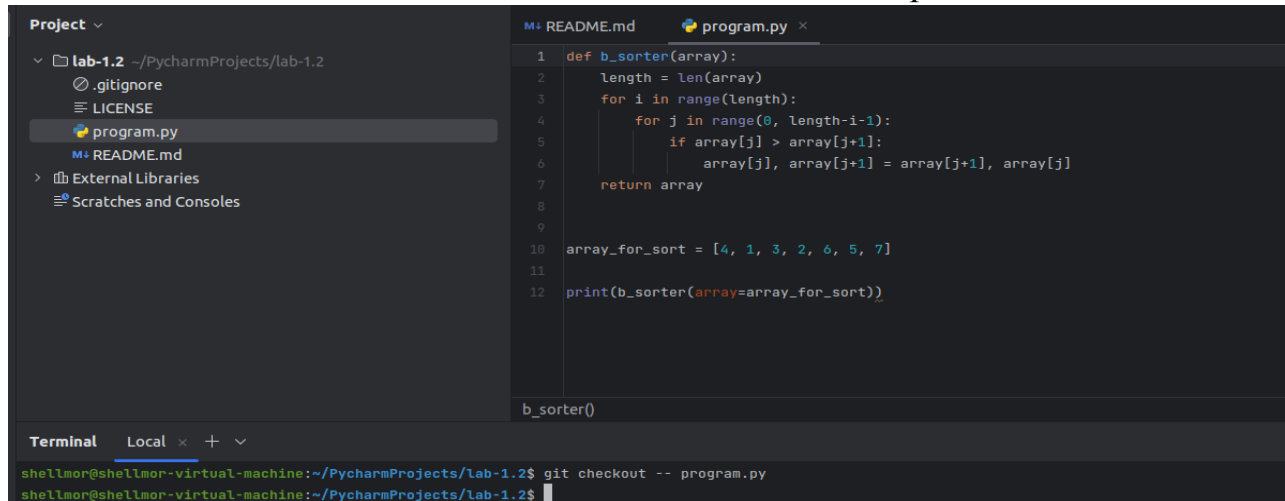


Рисунок 3. Удалите все несохраненные изменения

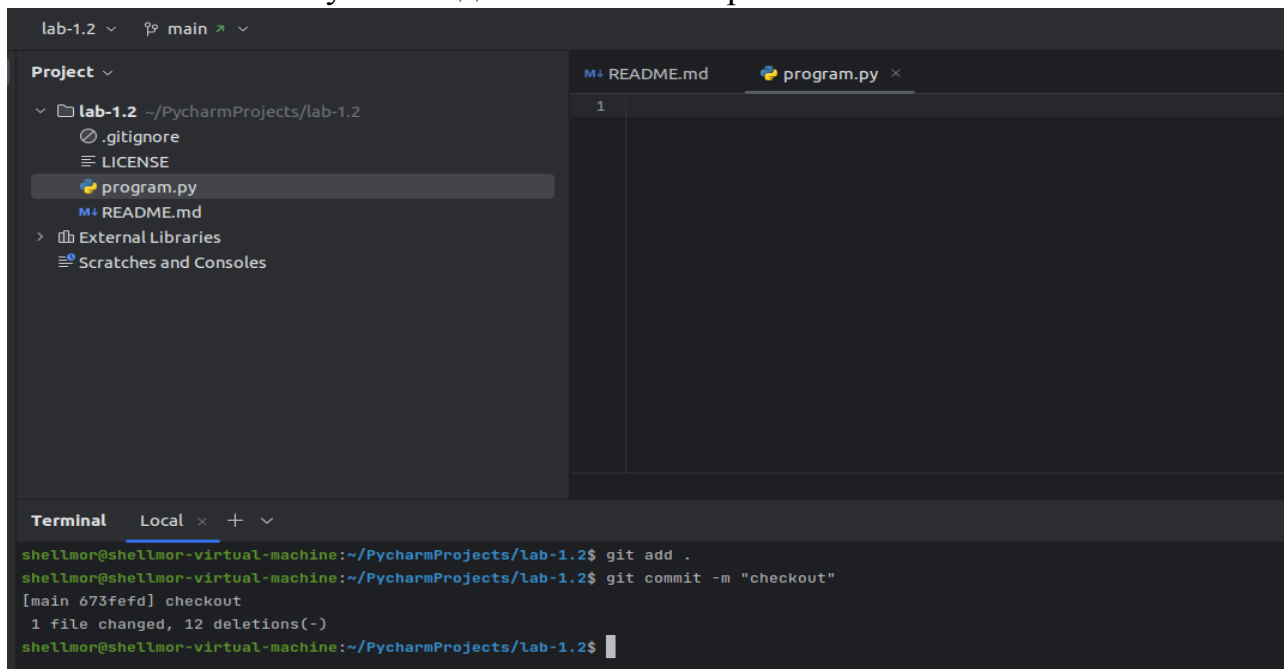
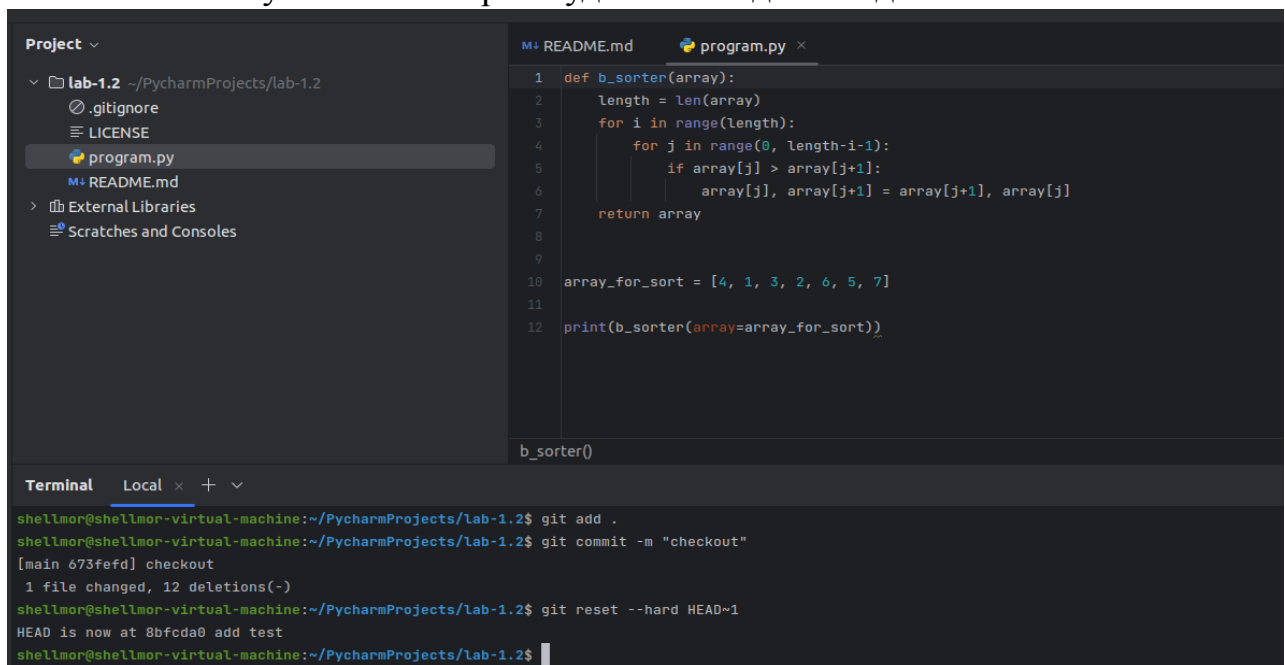


Рисунок 4. Повторное удаление кода и создание коммита.



```
commit 8bfcd02b734f725cacfd6405ecb2cdea784b0 (HEAD -> main)
Author: maratdmitchenko <maratdmitchenko@gmail.com>
Date: Wed Apr 26 14:42:13 2023 +0300

    add test

diff --git a/program.py b/program.py
index 587a404..c25d045 100644
--- a/program.py
+++ b/program.py
@@ -7,3 @@ def b_sorter(array):
     return array

+array_for_sort = [4, 1, 3, 2, 6, 5, 7]
+
+print(b_sorter(array=array_for_sort))
\ No newline at end of file
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
(END)
```

main


1 branch

0 tags

Go to file

Add file

<> Code


 martini69lol add test

8bfcda0 in 2 hours

🕒 8 commits

📁 .idea	add README	now
📄 .gitignore	Initial commit	now
📄 LICENSE	Initial commit	now
📄 README.md	add README	now
📄 program.py	add test	now

README.md



## Лабораторная работа 1.2 Исследование возможностей Git для работы с локальными репозиториями

Выполнил Дмитриченко Марат Заурович ИТС-6-о-22-1

### Рисунок 7. Фиксация изменений

Ссылка GitHub:

<https://github.com/martini69lol/lab-1.2>

Ссылка GitLab:

<https://gitlab.com/lab235617/lab-1-2>

### **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 <file>`

*5) Как отменить изменения в файле?*

Отменить изменения в файле можно с помощью команды:  
`git checkout --<file>`

*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 пометать 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 -а 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 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.

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 удаленном компьютере. Опция `-prune` в команде `git push` удалит ветку из удаленного репозитория, если в локальном репозитории не существует ветки с таким именем.

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