

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 ПРАКТИЧЕСКОЙ РАБОТЕ №2**  
**дисциплины «Программирование на Python»**

Выполнил:  
Данилецкий Дмитрий Витальевич  
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:  
Воронкин Р. А., канд. технических  
наук, 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

### Краткие теоритические сведения

Чтобы просмотреть историю коммитов в Git, используйте команду: `git log`

`-n <количество>`: Ограничение вывода по количеству коммитов.

`--since=<дата>`: Ограничение вывода по дате начиная с указанной даты.

`--until=<дата>`: Ограничение вывода по дате до указанной даты.,

Для внесения изменений в уже сделанный коммит в Git, используйте команду:

`sqlCopy code`

`git commit --amend`

Чтобы отменить индексацию (убрать файл из стейджа) файла в Git, используйте команду:

`git reset <имя файла>`

Чтобы отменить изменения в файле и вернуть его к состоянию последнего коммита, используйте команду:

`git checkout -- <имя файла>`

Чтобы просмотреть список удаленных репозиториев, связанных с вашим локальным репозиторием Git, используйте команду:

`git remote -v`

Удаленный репозиторий Git (remote repository) представляет собой хранилище для Git-репозитория, которое находится на удаленном сервере или в другом месте, отличном от вашей локальной рабочей директории. Удаленные репозитории используются для совместной работы над проектами и обмена кодом между разработчиками.

Чтобы просмотреть список удаленных репозиториев, связанных с вашим локальным репозиторием Git, используйте команду: `git remote -v`

Чтобы добавить удаленный репозиторий к вашему локальному репозиторию Git, используйте команду `git remote add` с указанием имени и URL-адреса удаленного репозитория. Вот как это сделать:

```
git remote add <имя-удаленного-репозитория> <URL-адрес-репозитория>
```

Для отправки (пуша) изменений на удаленный репозиторий используйте команду `git push`. Для получения (пулла) изменений из удаленного репозитория используйте команду `git pull`. Вот как это делается:

Отправка изменений на удаленный репозиторий (пуш):

```
git push <имя-удаленного-репозитория> <ветка-локальная>:<ветка-удаленная>
```

Например, чтобы отправить изменения из текущей локальной ветки "main" на удаленный репозиторий с именем "origin" и ветку "main", выполните:

```
git push origin main
```

Получение изменений из удаленного репозитория (пулл):

```
git pull <имя-удаленного-репозитория> <ветка-удаленная>:<ветка-локальная>
```

Например, чтобы получить изменения из удаленной ветки "main" на локальную ветку "main" из удаленного репозитория "origin", выполните:

```
git pull origin main
```

Если вы не указываете ветку-локальную и ветку-удаленную, Git автоматически попытается сопоставить ветки с одинаковыми именами.

Важно, чтобы перед пушем и пуллом вы сначала были уверены, что ваш локальный репозиторий синхронизирован с удаленным (чтобы избежать конфликтов).

Для просмотра информации о удаленном репозитории в Git, такой как URL-адрес удаленного репозитория и другие детали, используйте команду `git remote show`:

```
git remote show <имя-удаленного-репозитория>
```

## Ход работы

1. Изучил теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub с выбранным языком программирования C++.
3. Проработал примеры лабораторной работы:

```
slime@DESKTOP-TIA0FVK MINGW64 ~/LAB-1 (main)
$ cd

slime@DESKTOP-TIA0FVK MINGW64 ~ (master)
$ git clone https://github.com/schacon/simplegit-progit
Cloning into 'simplegit-progit'...
remote: Enumerating objects: 13, done.
remote: Total 13 (delta 0), reused 0 (delta 0), pack-reused 13
Receiving objects: 100% (13/13), done.
Resolving deltas: 100% (3/3), done.
```

Рисунок 1. Клонирование репозитория

```
slime@DESKTOP-TIA0FVK MINGW64 ~ (master)
$ cd C://Users//slime//simplegit-progit

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git log
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master, origin/HEAD)
Author: Scott Chacon <schacon@gmail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 10:31:28 2008 -0700

    first commit

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ |
```

Рисунок 2. Просмотр коммитов в обратном к хронологическому порядке

```

MINGW64:/c/Users/slime/simplegit-progit
$ git log -p -2
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master, origin/HEAD)
Author: Scott Chacon <schacon@gmail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

diff --git a/Rakefile b/Rakefile
index a874b73..8f94139 100644
--- a/Rakefile
+++ b/Rakefile
@@ -5,7 +5,7 @@ require 'rake/gempackagetask'
 spec = Gem::Specification.new do |s|
   s.platform = Gem::Platform::RUBY
   s.name = "simplegit"
-  s.version = "0.1.0"
+  s.version = "0.1.1"
   s.author = "Scott Chacon"
   s.email = "schacon@gmail.com"
   s.summary = "A simple gem for using Git in Ruby code."

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

diff --git a/lib/simplegit.rb b/lib/simplegit.rb
index a0a60ae..47c6340 100644
--- a/lib/simplegit.rb
+++ b/lib/simplegit.rb
@@ -18,8 +18,3 @@ class SimpleGit
   end
end
end

```

Рисунок 3. Просмотр внесенных изменений в каждый коммит (для последних 2 коммитов)

```

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git log --stat
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master, origin/HEAD)
Author: Scott Chacon <schacon@gmail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

Rakefile | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

lib/simplegit.rb | 5 ----
1 file changed, 5 deletions(-)

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 10:31:28 2008 -0700

    first commit

README      | 6 ++++++
Rakefile    | 23 ++++++++++++++++++++++
lib/simplegit.rb | 25 ++++++++++++++++++++++
3 files changed, 54 insertions(+)

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ |

```

Рисунок 4. Просмотр краткой статистики коммитов

```

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git log --pretty=oneline
ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master, origin/HEAD) changed the verison number
085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7 removed unnecessary test code
a11bef06a3f659402fe7563abf99ad00de2209e6 first commit

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git log --pretty=short
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master, origin/HEAD)
Author: Scott Chacon <schacon@gmail.com>

    changed the verison number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>

    removed unnecessary test code

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gmail.com>

    first commit

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git log --pretty=full
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master, origin/HEAD)
Author: Scott Chacon <schacon@gmail.com>
Commit: Scott Chacon <schacon@gmail.com>

    changed the verison number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Commit: Scott Chacon <schacon@gmail.com>

    removed unnecessary test code

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gmail.com>
Commit: Scott Chacon <schacon@gmail.com>

    first commit

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ |

```

Рисунок 5. Различные опции программы git log

```

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git log --pretty=format:"%h - %an, %ar : %s - %cn"
ca82a6d - Scott Chacon, 16 years ago : changed the verison number - Scott Chacon
085bb3b - Scott Chacon, 16 years ago : removed unnecessary test code - Scott Chacon
a11bef0 - Scott Chacon, 16 years ago : first commit - Scott Chacon

```

Рисунок 6. Еще одна опция git log

```

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git log --pretty=format:"%h %s" --graph
* ca82a6d changed the verison number
* 085bb3b removed unnecessary test code
* a11bef0 first commit

```

Рисунок 7. Опция git log с графом в формате ASCII, который показывает текущую ветку и историю слияний

```

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git log --relative-date
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master, origin/HEAD)
Author: Scott Chacon <schacon@gmail.com>
Date: 16 years ago

    changed the verison number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date: 16 years ago

    removed unnecessary test code

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gmail.com>
Date: 16 years ago

    first commit

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git log --name-only
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master, origin/HEAD)
Author: Scott Chacon <schacon@gmail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

Rakefile

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

lib/simplegit.rb

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 10:31:28 2008 -0700

    first commit

README
Rakefile
lib/simplegit.rb

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$

```

Рисунок 8. Опции git log

```

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git log --since=2008-01-15
commit ca82a6dff817ec66f44342007202690a93763949 (HEAD -> master, origin/master, origin/HEAD)
Author: Scott Chacon <schacon@gmail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 10:31:28 2008 -0700

    first commit

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$

```

Рисунок 9. Опция git log с ограничение по дате

```

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git log -S SimpleGit
commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 10:31:28 2008 -0700

    first commit

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ |

```

Рисунок 10. git log с опцией показывающей только коммиты с изменением определённой функции

```

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git log -- C:\\Users\\slime\\simplegit-progit\\lib
commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gmail.com>
Date: Sat Mar 15 10:31:28 2008 -0700

    first commit

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$

```

Рисунок 11. git log с опцией показывающей только коммиты с изменением определённого файла

```

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git log --pretty="%h - %s" --author='Scott Chacon' --since="2008-01-01" --until="2008-12-01" --no-merges
a11bef0 - first commit

```

Рисунок 12. Git log с набором опций для создания определённого формата



```

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git add *

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   AAA.txt
    new file:   README.md

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    README

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git reset HEAD README.md
Unstaged changes after reset:
D   README

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   AAA.txt

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    README

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md

```

Рисунок 14. Отмена индексации файла

```

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git checkout -- README

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   AAA.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md

slime@DESKTOP-TIA0FVK MINGW64 ~/simplegit-progit (master)
$

```

Рисунок 15. Отмена изменений в файле

```

slime@DESKTOP-TIA0FVK MINGW64 ~ (master)
$ git clone https://github.com/schacon/ticgit
Cloning into 'ticgit'...
remote: Enumerating objects: 1857, done.
remote: Total 1857 (delta 0), reused 0 (delta 0), pack-reused 1857
Receiving objects: 100% (1857/1857), 334.06 KiB | 302.00 KiB/s, done.
Resolving deltas: 100% (837/837), done.

slime@DESKTOP-TIA0FVK MINGW64 ~ (master)
$ cd ticgit

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git remote
origin

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$

```

Рисунок 16. Просмотр удалённого репозитория

```

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git remote
origin

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git remote add pb https://github.com/paulboone/ticgit

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git remote -v
origin https://github.com/schacon/ticgit (fetch)
origin https://github.com/schacon/ticgit (push)
pb https://github.com/paulboone/ticgit (fetch)
pb https://github.com/paulboone/ticgit (push)

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git fetch pb
remote: Enumerating objects: 43, done.
remote: Counting objects: 100% (22/22), done.
remote: Total 43 (delta 22), reused 22 (delta 22), pack-reused 21
Unpacking objects: 100% (43/43), 5.99 KiB | 46.00 KiB/s, done.
From https://github.com/paulboone/ticgit
* [new branch]      master      -> pb/master
* [new branch]      ticgit      -> pb/ticgit

```

Рисунок 17. Добавление удалённого репозитория

```
slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git remote show origin
* remote origin
Fetch URL: https://github.com/schacon/ticgit
Push URL: https://github.com/schacon/ticgit
HEAD branch: master
Remote branches:
  master tracked
  ticgit tracked
Local branch configured for 'git pull':
  master merges with remote master
Local ref configured for 'git push':
  master pushes to master (up to date)
```

Рисунок 18. Просмотр удаленного репозитория

```
slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git remote rename pb Dm
Renaming remote references: 100% (2/2), done.

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git remote
Dm
origin

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git remote remove Dm

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git remote
origin
```

Рисунок 19. Удаление и переименование удалённого репозитория

```

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git tag -a v1.4 -m "1.4"

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git tag
v1.4

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git show v1.4
tag v1.4
Tagger: Daniletskiy-D <slimedodoser@gmail.com>
Date: Sun Sep 17 18:36:57 2023 +0300

1.4

commit 847256809a3d518cd36b8f81859401416fe8d945 (HEAD -> master, tag: v1.4, origin/master, origin
AD)
Author: Jeff Welling <Jeff.Welling@Gmail.com>
Date: Tue Apr 26 17:29:17 2011 -0700

    Added note to clarify which is the canonical TicGit-ng repo

diff --git a/README.mkd b/README.mkd
index ab92035..9ea9ff9 100644
--- a/README.mkd
+++ b/README.mkd
@@ -1,3 +1,6 @@
+Note: the original TicGit author has pulled all the TicGit-ng changes into his repository, creat
a potentially confusing situation. The schacon TicGit repo, this one, is not consistently mainta
d. For up to date TicGit-ng info and code, check the canonical TicGit-ng repository at
+https://github.com/jeffwelling/ticgit
+
## TicGit-ng ##

This project provides a ticketing system built on Git that is kept in a

```

Рисунок 20. Создание тега и просмотр списка тегов

```

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git tag -d v1.4
Deleted tag 'v1.4' (was 4a72bc7)

```

Рисунок 21. Удаление тега

```

slime@DESKTOP-TIA0FVK MINGW64 ~/ticgit (master)
$ git checkout v1.4
Note: switching to 'v1.4'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 8472568 Added note to clarify which is the canonical TicGit-ng repo

```

Рисунок 22. Переход на тег

4. Выполнил клонирование созданного репозитория на рабочий компьютер.

```
slime@DESKTOP-TIA0FVK MINGW64 ~ (master)
$ git clone https://github.com/Daniletskiy-D/LAB2.git
Cloning into 'LAB2'...
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.

slime@DESKTOP-TIA0FVK MINGW64 ~ (master)
$ cd LAB2

slime@DESKTOP-TIA0FVK MINGW64 ~/LAB2 (main)
$
```

Рисунок 23. Клонирование репозитория

5. Дополнил файл .gitignore необходимыми правилами для языка программирования C++.

6. Добавил в файл README.md информацию о дисциплине, группе и ФИО студента, выполняющего лабораторную работу.

7. Написал небольшую программу на C++. Фиксируя изменения при написании программы в локальном репозитории семью коммитами и тремя тэгами.

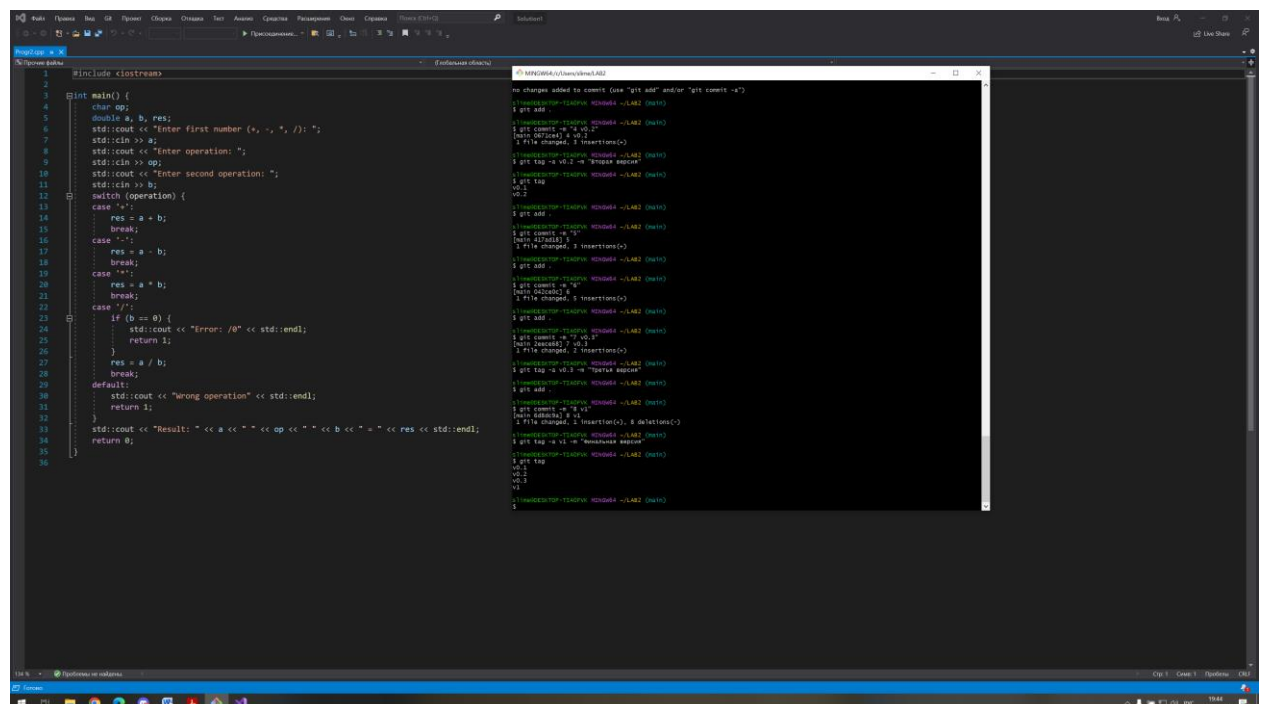


Рисунок 24. Написание программы

8. Просмотрел историю (журнал) хранилища.

```
slime@DESKTOP-TIA0FVK MINGW64 ~/LAB2 (main)
$ git log --graph --pretty=oneline --abbrev-commit
* 6d8dc9a (HEAD -> main, tag: v1) 8 v1
* 2eece68 (tag: v0.3) 7 v0.3
* 042ce0c 6
* 417ad18 5
* 0671ce4 (tag: v0.2) 4 v0.2
* 47f12f5 (tag: v0.1) 3 v0.1
* 387c5fb 2
* 8198cc1 1
* 2bc78d3 Начало
* 0d45673 (origin/main, origin/HEAD) Initial commit

slime@DESKTOP-TIA0FVK MINGW64 ~/LAB2 (main)
$
```

Рисунок 25. Просмотр истории

9. Просмотрел содержимое коммитов.

```
slime@DESKTOP-TIA0FVK MINGW64 ~/LAB2 (main)
$ git show HEAD
commit 6d8dc9abe76d4c4ab4996947fd958b72359206e9 (HEAD -> main, tag: v1)
Author: Daniletskiy-D <slimedodoser@gmail.com>
Date: Sun Sep 17 19:44:02 2023 +0300

    8 v1

diff --git a/Progr2.cpp b/Progr2.cpp
index 873d351..1ee9537 100644
--- a/Progr2.cpp
+++ b/Progr2.cpp
@@ -3,16 +3,12 @@
 int main() {
     char op;
     double a, b, res;

-    std::cout << "Enter first number: ";
+    std::cout << "Enter first number (+, -, *, /): ";
     std::cin >> a;

     std::cout << "Enter operation: ";
     std::cin >> op;

     std::cout << "Enter second operation: ";
     std::cin >> b;

     switch (operation) {
     case '+':
         res = a + b;
@@ -34,9 +30,6 @@ int main() {
     std::cout << "Wrong operation" << std::endl;
     return 1;
 }

-    std::cout << "Result: " << a << " " << op << " " << b << " = " << res << std::endl;
-    return 0;
-}
}
```

Рисунок 26. Содержимое последнего коммита

```

slime@DESKTOP-TIA0FVK MINGW64 ~/LAB2 (main)
$ git show HEAD~1
commit 2eece682d0fb2088d04a8892dbc2e70e5fd34a0b (tag: v0.3)
Author: Daniletskiy-D <slimedodoser@gmail.com>
Date:   Sun Sep 17 19:42:27 2023 +0300

    7 v0.3

diff --git a/Progr2.cpp b/Progr2.cpp
index 0222a3b..873d351 100644
--- a/Progr2.cpp
+++ b/Progr2.cpp
@@ -28,6 +28,8 @@ int main() {
     std::cout << "Error: /0" << std::endl;
     return 1;
 }
+   res = a / b;
+   break;
 default:
     std::cout << "Wrong operation" << std::endl;
     return 1;

slime@DESKTOP-TIA0FVK MINGW64 ~/LAB2 (main)
$ git show 042ce0c
commit 042ce0c7a774db01f177c1e1f5c74eaf49ef8f72
Author: Daniletskiy-D <slimedodoser@gmail.com>
Date:   Sun Sep 17 19:40:44 2023 +0300

    6

diff --git a/Progr2.cpp b/Progr2.cpp
index 10deb6c..0222a3b 100644
--- a/Progr2.cpp
+++ b/Progr2.cpp
@@ -23,6 +23,11 @@ int main() {
     case '*':
         res = a * b;
         break;
+   case '/':
+       if (b == 0) {
+           std::cout << "Error: /0" << std::endl;
+           return 1;
+       }
 default:
     std::cout << "Wrong operation" << std::endl;
     return 1;

```

Рисунок 27. Содержимое предпоследнего коммита и коммита с выбранным хэшем

## 10. Откат к заданной версии.

10.1. Удалил весь код из файла программы репозитория и сохранил этот файл.



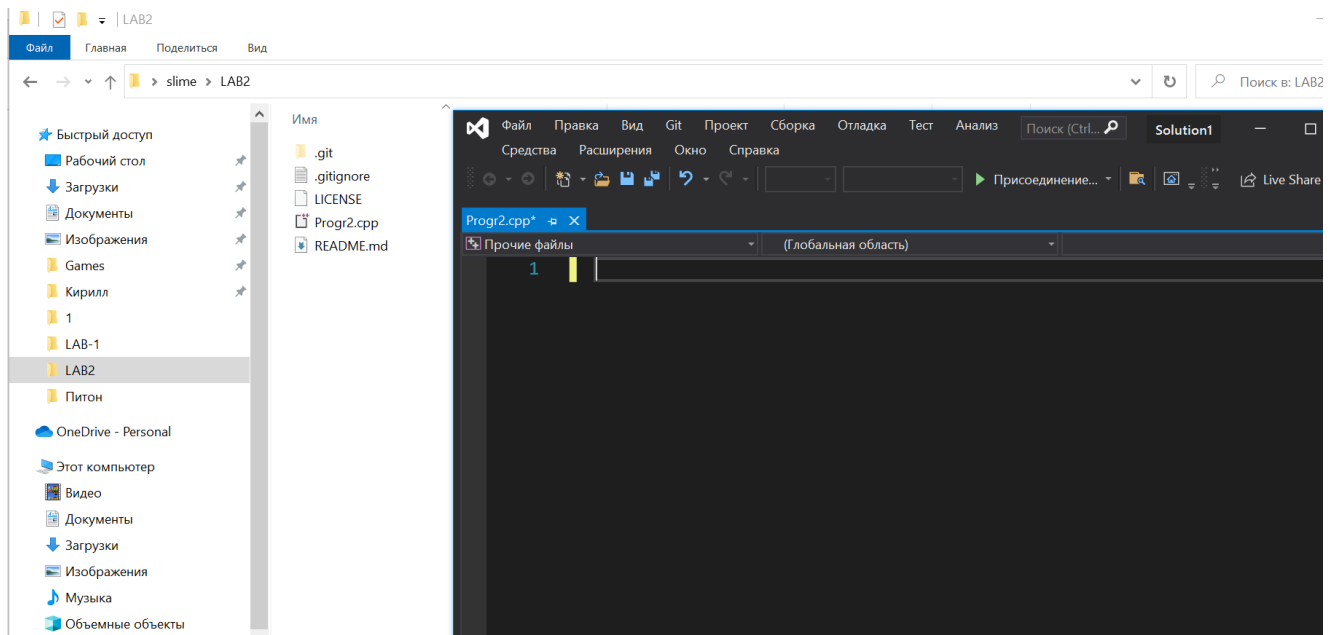


Рисунок 28. Удаление кода

10.2. Удалил все несохраненные изменения в файле

```
slime@DESKTOP-TIA0FVK MINGW64 ~/LAB2 (main)
$ git checkout -- Progr2.cpp
```

Рисунок 29. Удаление всех несохраненных изменений в файле

10.3. Повторил пункт 10.1 и сделал коммит.

10.4. Откатил состояние хранилища к предыдущей версии.

```
slime@DESKTOP-TIA0FVK MINGW64 ~/LAB2 (main)
$ git commit -m "Удаление кода"
[main aa9d962] удаление кода
1 file changed, 35 deletions(-)

slime@DESKTOP-TIA0FVK MINGW64 ~/LAB2 (main)
$ ^[[200~git reset --hard
bash: $'\E[200~git': command not found

slime@DESKTOP-TIA0FVK MINGW64 ~/LAB2 (main)
$ git reset --hard HEAD~1
HEAD is now at 6d8dc9a 8 v1

slime@DESKTOP-TIA0FVK MINGW64 ~/LAB2 (main)
$ |
```

Рисунок 30. Создание коммита и откат к пред. версии



## Контрольные вопросы

1. Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов?

Чтобы просмотреть историю коммитов в Git, используется команда: `git log`

Опция `--pretty` меняет формат вывода. Некоторые дополнительные опции для просмотра истории коммитов:

- 1) `--oneline`: Отображает каждый коммит в одной строке. К тому же, опции `short`, `full` и `fuller` делают вывод приблизительно в том же формате, но с меньшим или большим количеством информации соответственно
- 2) `--graph`: Рисует графическое представление веток и слияний.
- 3) `--author=<имя>`: Фильтрует коммиты по автору.
- 4) `--since=<дата>` и `--until=<дата>`: Фильтрует коммиты по датам.

2. Как ограничить вывод при просмотре истории коммитов?

Можно ограничить количество записей в выводе команды; для этого используется параметр `-x` для вывода только `x` записей, например:

```
$ git log -p -2
```

3. Как внести изменения в уже сделанный коммит?

Для внесения изменений в уже сделанный коммит в Git, используется команда:

```
git commit --amend
```

Затем можно внести необходимые изменения и перезаписать последний коммит.

4. Как отменить индексацию файла в Git?

Чтобы отменить индексацию файла в Git, используется команда:

```
git reset <имя файла>
```

5. Как отменить изменения в файле?

Чтобы отменить изменения в файле и вернуть его к состоянию последнего коммита, используется команда:

```
git checkout -- <имя файла>
```

#### 6. Что такое удаленный репозиторий Git?

Удаленный репозиторий Git (remote repository) - это репозиторий Git, который расположен на удаленном сервере или в другом месте, отличном от вашей локальной рабочей директории. Удаленные репозитории используются для совместной работы над проектами с другими разработчиками. Они служат для обмена кодом и отслеживания изменений между разными участниками.

#### 7. Как выполнить просмотр удаленных репозиториях данного локального репозитория?

Чтобы просмотреть список удаленных репозиториях, связанных с вашим локальным репозиторием Git, используется команда:

```
git remote -v
```

#### 8. Как добавить удаленный репозиторий для данного локального репозитория?

Чтобы добавить удаленный репозиторий к вашему локальному репозиторию Git, используется команда `git remote add <имя-удаленного-репозитория> <URL-адрес-репозитория>`

#### 9. Как выполнить отправку/получение изменений с удаленного репозитория?

Отправка изменений на удаленный репозиторий (пуш): `git push <имя-удаленного-репозитория> <ветка>`

Получение изменений из удаленного репозитория (пулл): `git pull <имя-удаленного-репозитория> <ветка>`

#### 10. Как выполнить просмотр удаленного репозитория?

Для просмотра информации о удаленном репозитории в Git, используется команда `git remote show <имя-удаленного-репозитория>`

#### 11. Каково назначение тэгов Git?

Тэги Git используются для пометки определенных коммитов как важных моментов в истории проекта, что облегчает их идентификацию и

версионирование. Тэги обычно используются для обозначения релизов и стабильных версий проекта.

## 12. Как осуществляется работа с тэгами Git?

Просмотр списка тегов: `$ git tag` (параметры `-l` и `--list` опциональны)

Создание тегов: `git tag -a <имя-тэга> -m "Описание"`

Обмен тегами: `git push origin <имя-тэга>`

Удаление тегов: `git tag -d <имя-тэга>`

Переход на тег: `git checkout <имя-тэга>`

## 13. Самостоятельно изучите назначение флага `--prune` в командах `git fetch` и `git push`. Каково назначение этого флага?

Флаг `--prune` в командах `git fetch` и `git push` выполняет следующие назначения:

В команде `git fetch` с флагом `--prune` Git будет удалять (подчищать) удаленные ветки и тэги из вашего локального репозитория, если они были удалены из удаленного репозитория. Это позволяет вам поддерживать локальный репозиторий в актуальном состоянии и избегать хранения устаревших ссылок на удаленные объекты.

В команде `git push` с флагом `--prune` Git будет удалять удаленные ветки из удаленного репозитория, которые больше не существуют в вашем локальном репозитории. Это полезно, чтобы синхронизировать удаленный репозиторий с вашим локальным состоянием и удалить удаленные ветки, которые были удалены вами или другими разработчиками.

Использование флага `--prune` позволяет поддерживать удаленный и локальный репозитории в чистом состоянии и избегать "мусора" от удаленных веток и тэгов.

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