

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №1.2

Дисциплина: «Основы кроссплатформенного программирования»

Тема: «Исследование возможностей Git для работы с локальными
репозиториями»

Выполнил: студент 1 курса

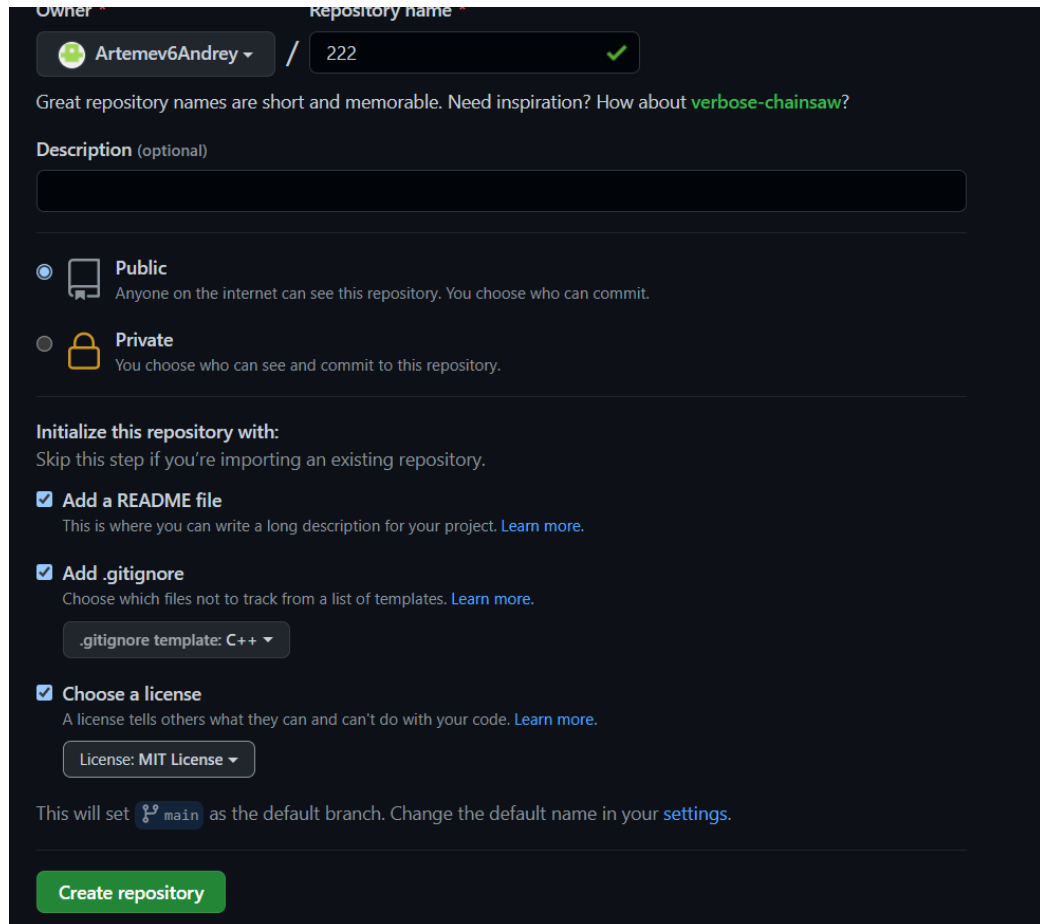
группы ИВТ-б-о-21-1

Артемьев Андрей Витальевич

Ставрополь 2022

Выполнение работы:

1. Создал общедоступный репозиторий 222 на GitHub в котором будет использована лицензия MIT и выбранный мной язык программирования.



Owner: Artemev6Andrey / Repository name: 222 ✓

Great repository names are short and memorable. Need inspiration? How about [verbose-chainsaw?](#)

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☒ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)
.gitignore template: C++ ▾

☒ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)
License: MIT License ▾

This will set `main` as the default branch. Change the default name in your [settings](#).

Create repository

Рисунок 1.1 Созданный репозиторий в GitHub

```
32 lines (26 sloc) | 270 Bytes

1 # Prerequisites
2 *.d
3
4 # Compiled Object files
5 *.slo
6 *.lo
7 *.o
8 *.obj
9
10 # Precompiled Headers
11 *.gch
12 *.pch
13
14 # Compiled Dynamic libraries
15 *.so
16 *.dylib
17 *.dll
18
19 # Fortran module files
20 *.mod
21 *.smod
22
23 # Compiled Static libraries
24 *.lai
25 *.la
26 *.a
27 *.lib
28
29 # Executables
30 *.exe
31 *.out
32 *.app
```

Рисунок 1.2 Изменения в файле .gitignore

2. Клонировал созданный репозиторий на раб. компьютер:

```
C:\Users\aa715\Desktop\222>git clone https://github.com/Artemev6Andrey/222.git
Cloning into '222'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 3), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), done.
Resolving deltas: 100% (3/3), done.

C:\Users\aa715\Desktop\222>
```

Рисунок 2. Клонирование репозитория

3. Добавил информацию в README и закоммитил:

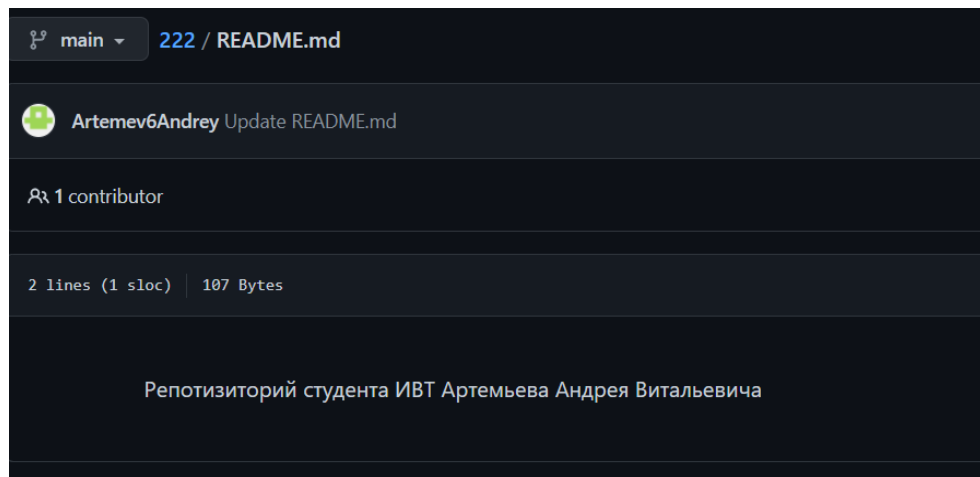


Рисунок 3.1 Добавление информации в README.md

4. Написал в репозитории небольшую программу, сделал коммит и пуш:

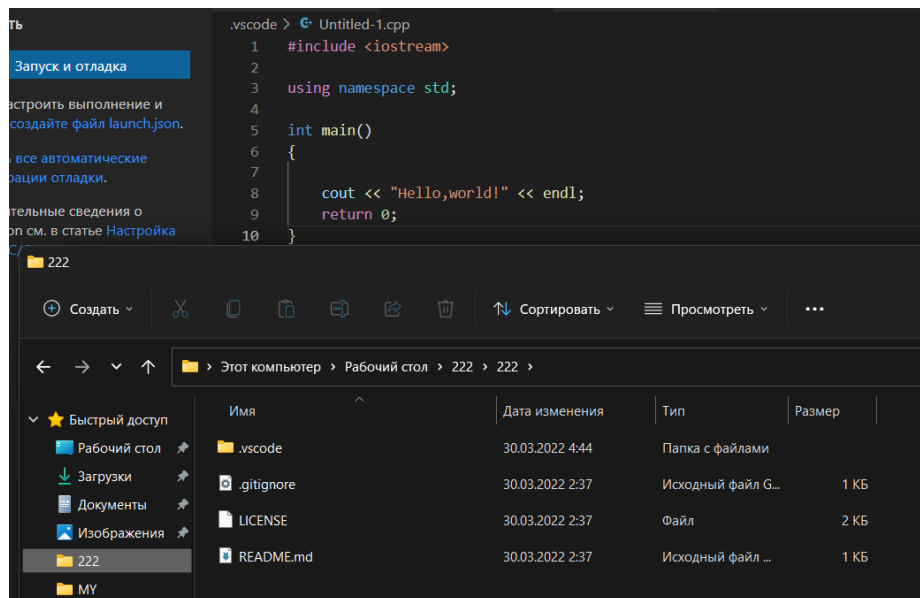


Рисунок 4.1 Изменения в программе

```

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git commit -m "Programm Hello world"
[main 42b7ba1] Programm Hello world
3 files changed, 44 insertions(+)
create mode 100644 .vscode/Untitled-1.cpp
create mode 100644 .vscode/launch.json
create mode 100644 .vscode/tasks.json

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git pull
Already up to date.

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.21 KiB | 1.21 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Artemev6Andrey/222.git
6276d66..42b7ba1 main -> main

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$

```

Рисунок 4.3 Коммит и пуш программы на уд. сервер

Artemev6Andrey Programm Hello world 42b7ba1 3 minutes ago 4 commits		
📁 .vscode	Programm Hello world	3 minutes ago
📄 .gitignore	Initial commit	6 hours ago
📄 LICENSE	Initial commit	6 hours ago
📄 README.md	Update README.md	5 hours ago

Рисунок 4.4 Изменения на уд. сервере

5. Делал коммиты в процессе изменения программы, отметил их тегами и запустил на уд. сервер коммиты затем теги:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7
8      cout << "Hello,world!" << endl;
9      cout<<1+1;
10     return 0;
11 }

```

Рисунок 5.1 Изменения в программе

```

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git add .

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git commit -m "2"
[main 865a486] 2
1 file changed, 1 insertion(+)

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$

```

Рисунок 5.2 Коммит изменений

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7
8      cout << "Hello,world!" << endl;
9      cout<<1+1;
10     cout <<"Hello world!\n";
11     return 0;
12 }

```

Рисунок 5.3 Изменения в программе

```

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git add .

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git commit -m "3"
[main 6cff68d] 3
1 file changed, 1 insertion(+)

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$

```

Рисунок 5.4 Коммит изменений

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7
8      cout << "Hello,world!" << endl;
9      cout<<1+1;
10     cout <<"Hello world!\n";
11     cout<<"World,23!";
12     return 0;
13 }

```

Рисунок 5.5 Изменения в программе

```

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git add .

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git commit -m "4"
[main 8d8b07b] 4
1 file changed, 1 insertion(+)

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$

```

Рисунок 5.6 Коммит изменений

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7
8      cout << "Hello,world!" << endl;
9      cout<<1+1;
10     cout <<"Hello world!\n";
11     cout<<"\nWorld,23!";
12     int a, b;
13     return 0;
14 }

```

Рисунок 5.7 Изменения в программе

```

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git add .

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git commit -m "5"
[main 25b3ade] 5
1 file changed, 2 insertions(+), 1 deletion(-)

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$

```

Рисунок 5.8 Коммит изменений

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7
8      cout << "Hello,world!" << endl;
9      cout <<1+1;
10     cout <<"Hello world!\n";
11     cout <<"\nWorld,23!";
12     int a, b;
13     return 0;
14     cout <<2+3;
15 }

```

Рисунок 5.9 Изменения в программе

```

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git add .

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git commit -m "6"
[main eed115e] 6
1 file changed, 3 insertions(+), 2 deletions(-)

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ |

```

Рисунок 5.10 Коммит изменений

```

#include <iostream>

using namespace std;

int main()
{

    cout << "Hello,world!" << endl;
    cout <<1+1;
    cout <<"Hello world!\n";
    cout <<"\nWorld,23!";
    int a, b;
    cout <<2+3;
    cout <<5+5;
    return 0;
}

```

Рисунок 5.9 Изменения в программе


```

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git add .

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git commit -m "7"
[main 697487b] 7
1 file changed, 2 insertions(+), 1 deletion(-)

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$

```

Рисунок 5.11 Коммит изменений


```






aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git push
Enumerating objects: 27, done.
Counting objects: 100% (27/27), done.
Delta compression using up to 8 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (24/24), 2.24 KiB | 2.24 MiB/s, done.
Total 24 (delta 11), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (11/11), completed with 1 local object.
To https://github.com/Artemev6Andrey/222.git
42b7ba1..697487b main -> main

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ |

```

Рисунок 5.12 Пуш изменений


Artemev6Andrey doc
262f6a2 14 days ago

	.vscode	7
	doc	doc
	.gitignore	Initial commit
	LICENSE	Initial commit
	README.md	Update README.md

README.md

Репозиторий студента ИВТ Артемьева Андрея Витальевича

Рисунок 5.13 Изменения на уд. Сервере







7	Artemev6Andrey committed 14 days ago	 697487b <>
6	Artemev6Andrey committed 14 days ago	 eed115e <>
5	Artemev6Andrey committed 14 days ago	 25b3ade <>
4	Artemev6Andrey committed 14 days ago	 8d8b07b <>
3	Artemev6Andrey committed 14 days ago	 6cFf68d <>
2	Artemev6Andrey committed 14 days ago	 865a486 <>

Рисунок 5.14 История коммитов на уд. Сервере

```
MINGW64:/c/Users/aa715/Desktop/222/222

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git tag -a 1.0 -m "7"

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git tag -a 2/0 -m "2"

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git tag -a 3/0 -m "3"

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git tag
1.0
2/0
3/0

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ |
```

Рисунок 5.15 Присваивание тега коммиту

```
aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git push origin --tags
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 219 bytes | 219.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
Remote: Resolving deltas: 100% (2/2), done.
To https://github.com/Artemev6Andrey/222.git
 * [new tag]         1.0 -> 1.0
 * [new tag]         2/0 -> 2/0
 * [new tag]         3/0 -> 3/0

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
```

Рисунок 5.16 Пуш тегов

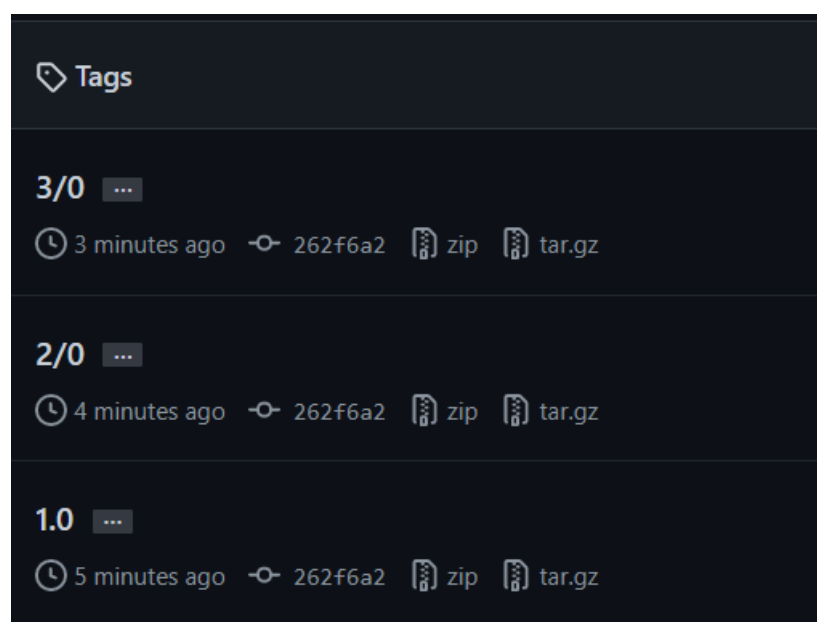
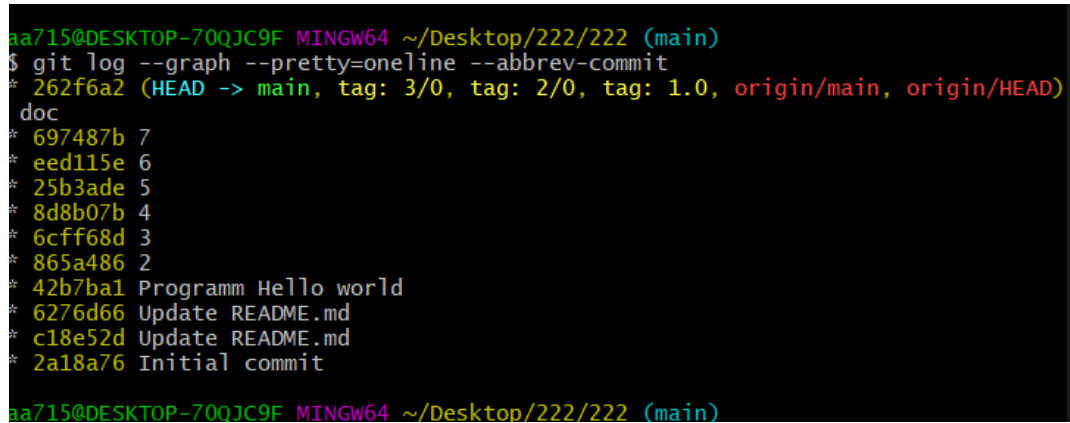


Рисунок 5.17 История тегов на уд. сервере

6. Просмотрел историю хранилища командой git log:



```
aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git log --graph --pretty=oneline --abbrev-commit
* 262f6a2 (HEAD -> main, tag: 3/0, tag: 2/0, tag: 1.0, origin/main, origin/HEAD)
  doc
* 697487b 7
* eed115e 6
* 25b3ade 5
* 8d8b07b 4
* 6cff68d 3
* 865a486 2
* 42b7ba1 Programm Hello world
* 6276d66 Update README.md
* c18e52d Update README.md
* 2a18a76 Initial commit
aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
```

Рисунок 6. История коммитов

7. Просмотрел содержимое коммитов командой git show HEAD, git show HEAD~, git show 865a4864e98aff1865c7ec1b593ceba1ef34a2b3:

```

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git show HEAD
commit 262f6a2276f6e303f06ae51ad39017b96e4ea8f7 (HEAD -> main, tag: 3/0, tag: 2/0, tag: 1.0, origin/main, origin/HEAD)
Author: Andrey <aa7155443@gmail.com>
Date: Wed Mar 30 10:55:42 2022 +0300

    doc

diff --git "a/doc/\320\220\321\200\321\202\320\265\320\274\321\214\320\265\320\2621.pdf" "b/doc/\320\220\321\200\321\202\320\265\320\274\321\214\320\265\320\2621.pdf"
new file mode 100644
index 0000000..69f53d5
--- /dev/null
+++ "b/doc/\320\220\321\200\321\202\320\265\320\274\321\214\320\265\320\2621.pdf"
@@ -0,0 +1,248 @@
+    МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
+    РОССИЙСКОЙ ФЕДЕРАЦИИ
+
+Федеральное государственное автономное образовательное учреждение
+высшего образования «СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
+
+    УНИВЕРСИТЕТ»
+...skipping...
commit 262f6a2276f6e303f06ae51ad39017b96e4ea8f7 (HEAD -> main, tag: 3/0, tag: 2/0, tag: 1.0, origin/main, origin/HEAD)
Author: Andrey <aa7155443@gmail.com>
Date: Wed Mar 30 10:55:42 2022 +0300

    doc

diff --git "a/doc/\320\220\321\200\321\202\320\265\320\274\321\214\320\265\320\2621.pdf" "b/doc/\320\220\321\200\321\202\320\265\320\274\321\214\320\265\320\2621.pdf"
new file mode 100644
index 0000000..69f53d5
--- /dev/null
+++ "b/doc/\320\220\321\200\321\202\320\265\320\274\321\214\320\265\320\2621.pdf"
@@ -0,0 +1,248 @@
+    МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
+    РОССИЙСКОЙ ФЕДЕРАЦИИ
+
+Федеральное государственное автономное образовательное учреждение
+высшего образования «СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
+
+    УНИВЕРСИТЕТ»
+    Кафедра инфокоммуникаций
+    Институт цифрового развития
+
+    ОТЧЕТ

```

Рисунок 7.1 Содержимое коммитов командами

```

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git show HEAD~
commit 697487bb8871583e1845c0e710d767290d9d4e35
Author: Andrey <aa7155443@gmail.com>
Date: Wed Mar 30 06:39:28 2022 +0300

    7

diff --git a/.vscode/Untitled-1.cpp b/.vscode/Untitled-1.cpp
index 178f101..f416e1d 100644
--- a/.vscode/Untitled-1.cpp
+++ b/.vscode/Untitled-1.cpp
@@ -10,6 +10,7 @@ int main()
     cout <<"Hello world!\n";
     cout <<"\nWorld,23!";
     int a, b;
+    cout <<2+3;
+    cout <<5+5;
     return 0;
+    cout <<2+3;
}
\ No newline at end of file

```

```

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$ git show 865a4864e98aff1865c7ec1b593ceba1ef34a2b3
commit 865a4864e98aff1865c7ec1b593ceba1ef34a2b3
Author: Andrey <aa7155443@gmail.com>
Date:   Wed Mar 30 06:21:45 2022 +0300

    2

diff --git a/.vscode/Untitled-1.cpp b/.vscode/Untitled-1.cpp
index 1cf3bda..45c65c1 100644
--- a/.vscode/Untitled-1.cpp
+++ b/.vscode/Untitled-1.cpp
@@ -6,5 +6,6 @@ int main()
 {
     cout << "Hello,world!" << endl;
+    cout<<1+1;
     return 0;
 }
\ No newline at end of file

aa715@DESKTOP-70QJC9F MINGW64 ~/Desktop/222/222 (main)
$

```

Рисунок 7.2 Содержание коммитов командами

8. Удалил весь код в файле Untitled-1.cpp и сохранил его, затем удалил все несохраненные изменения командой, после этого еще раз удалил весь код в файле и сделал коммит, после чего откатил состояние файла к предыдущей версии.

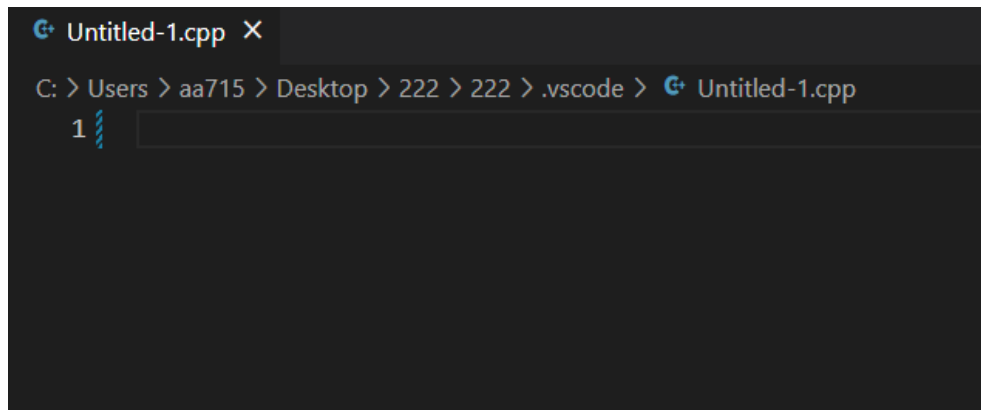


Рисунок 8.1 Удаление кода в файле Untitled-1.cpp

```

C:\Users\aa715\Desktop\222\222\.vscode>git checkout -- Untitled-1.cpp
C:\Users\aa715\Desktop\222\222\.vscode>

```

Рисунок 8.2 checkout изменений файла Untitled-1.cpp

```
Untitled-1.cpp X
C: > Users > aa715 > Desktop > 222 > 222 > .vscode > G+ Untitled-1.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7
8      cout << "Hello,world!" << endl;
9      cout <<1+1;
10     cout <<"Hello world!\n";
11     cout <<"\nWorld,23!";
12     int a, b;
13     cout <<2+3;
14     cout <<5+5;
15     return 0;
16 }
```

Рисунок 8.3 Изменения в файле с программой после команды

```
Untitled-1.cpp X
C: > Users > aa715 > Desktop > 222 > 222 > .vscode > G+ Untitled-1.cpp
1  
```

Рисунок 8.4 Удаление кода в файле с программой

```
C:\Users\aa715\Desktop\222\222\.vscode>git commit -m "1.0"
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Untitled-1.cpp

no changes added to commit (use "git add" and/or "git commit -a")
C:\Users\aa715\Desktop\222\222\.vscode>
```

Рисунок 8.5 Коммит изменений

Вывод: команда `git -checkout <FileName>` удаляет изменения произошедшие с файлом в репозитории до коммита.

Контрольные вопросы и ответы на них:

Вопросы для защиты работы.

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

Наиболее простой и в то же время мощный инструмент для этого — команда `git log`. По умолчанию, без аргументов, `git log` выводит список коммитов созданных в данном репозитории в обратном хронологическом порядке. То есть самые последние коммиты показываются первыми.

Одна из опций, когда вы хотите увидеть сокращенную статистику для каждого коммита, вы можете использовать опцию `-stat`.

Вторая опция (одна из самых полезных аргументов) является `-p` или `--patch`, который показывает разницу (выводит патч), внесенную в каждый коммит. Так же вы можете ограничить количество записей в выводе команды; используйте параметр `-2` для вывода только двух записей (пример команды `git log -p -2`).

Третья действительно полезная опция это `--pretty`. Она 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. Особенно это может быть полезным, когда вы хотите сгенерировать вывод для автоматического анализа — так как вы указываете формат явно, он не будет изменен даже после обновления Git.

Для опции `git log --pretty=format` существуют различного рода опции для изменения формата отображения.

2. Как ограничить вывод при просмотре истории коммитов?

Для ограничения может использоваться функция `git log <n>`, где `n` число записей.

Также, существуют опции для ограничения вывода по времени, такие как `--since` и `--until`, они являются очень удобными. Например, следующая команда покажет список коммитов, сделанных за последние две недели:

```
git log --since=2.weeks
```

Это команда работает с большим количеством форматов — вы можете указать определенную дату вида `2008-01-15` или же относительную дату, например `2 years 1 day 3 minutes ago`.

Также вы можете фильтровать список коммитов по заданным параметрам. Опция `--author` дает возможность фильтровать по автору коммита, а опция `--grep` (показывает только коммиты, сообщение которых содержит указанную строку) искать по ключевым словам в сообщении коммита. Функция `-S` показывает только коммиты, в которых изменение в коде повлекло за собой добавление или удаление указанной строки.

3. Как внести изменения в уже сделанный коммит?

Внести изменения можно с помощью команды `git commit --amend`

Эта команда берёт индекс и применяет его к последнему коммиту. Если после последнего коммита не было никаких проиндексированных изменений (например, вы запустили приведённую команду сразу после предыдущего коммита), то состояние проекта будет абсолютно таким же и всё, что мы изменим, это комментарий к коммиту.

Для того, чтобы внести необходимые изменения - нам нужно проиндексировать их и выполнить команду `git commit --amend`.


```
git commit -m 'initial commit'
```

```
git add forgotten_file
```

```
git commit --amend
```

Эффект от выполнения этой команды такой, как будто мы не выполнили предыдущий коммит, а еще раз выполнили команду `git add` и выполнили коммит.

4. Как отменить индексацию файла в Git?

Например, вы изменили два файла и хотите добавить их в разные коммиты, но случайно выполнили команду `git add *` и добавили в индекс оба. Как исключить из индекса один из них? Команда `git status` напомним вам:

Прямо под текстом «Changes to be committed» говорится: используйте `git reset HEAD <file>` для исключения из индекса.

5. Как отменить изменения в файле?

С помощью команды `git checkout -- <file>`.

6. Что такое удаленный репозиторий Git?

Удалённый репозиторий это своего рода наше облако, в которое мы сохраняем те или иные изменения в нашей программе/коде/файлах.

7. Как выполнить просмотр удаленных репозитория данного локального репозитория?

Для того, чтобы просмотреть список настроенных удалённых репозитория, необходимо запустить команду `git remote`.

Также можно указать ключ `-v`, чтобы просмотреть адреса для чтения и записи, привязанные к репозиторию. Пример: `git remote -v`

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 <shortname> <url>`.

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

Если необходимо получить изменения, которые есть у Пола, но нету у вас, вы можете выполнить команду `git fetch <Название репозитория>`. Важно отметить, что команда `git fetch` забирает данные в ваш локальный репозиторий, но не сливает их с какими-либо вашими наработками и не модифицирует то, над чем вы работаете в данный момент. Вам необходимо вручную слить эти данные с вашими, когда вы будете готовы.

Если ветка настроена на отслеживание удалённой ветки, то вы можете использовать команду `git pull` чтобы автоматически получить изменения из удалённой ветки и слить их со своей текущей. Выполнение `git pull`, как правило, извлекает (fetch) данные с сервера, с которого вы изначально клонировали, и автоматически пытается слить (merge) их с кодом, над которым вы в данный момент работаете.

Чтобы отправить изменения на удалённый репозиторий необходимо отправить их в удалённый репозиторий. Команда для этого действия простая: `git push <remote-name> <branch-name>`.

10. Как выполнить просмотр удаленного репозитория?

Для просмотра удалённого репозитория, можно использовать команду `git remote show <remote>`.

11. Каково назначение тэгов Git?

Теги - это ссылки указывающие на определённые версии кода/написанной программы. Они удобны чтобы в случае чего вернуться к нужному моменту. Также при помощи тегов можно помечать важные моменты.

12. Как осуществляется работа с тэгами Git?

Просмотреть наличие тегов можно с помощью команды: `git tag`.

А назначить (указать, добавить тег) можно с помощью команды `git tag -a v1.4(версия изначальная) -m "Название"`.

С помощью команды `git show` вы можете посмотреть данные тега вместе с коммитом: `git show v1.4`.

Отправка тегов, 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 <tagname>`. Для отправки всех тегов можно использовать команду `git push origin tags`.

Для удаления тега в локальном репозитории достаточно выполнить команду `git tag -d <tagname>`. Например, удалить созданный ранее лёгкий тег можно следующим образом: `git tag -d v1.4-lw`

Для удаления тега из внешнего репозитория используется команда `git push origin --delete <tagname>`.

Если вы хотите получить версии файлов, на которые указывает тег, то вы можете сделать `git checkout` для тега пример: `git checkout -b version2 v2.0.0`.

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

`Git fetch --prune` команда получения всех изменений с репозитория GitHub.

В команде `git push --prune` удаляет удаленные ветки, у которых нет локального аналога.