

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»

Выполнил:

Евдаков Евгений Владимирович

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., 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:

Задание 1.

Создал новый репозиторий и клонировал его на свой компьютер. Добавил некоторое правило в файл *gitignore*, чтобы Git игнорировал файлы в формате .png

```
C:\Users\Gaming-PC>git clone https://github.com/EvgenyEvdakov/LabaEvd1.2.git
Cloning into 'LabaEvd1.2'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
```

Рисунок 1. Новый репозиторий

Задание 2. Проработать примеры лабораторной работы. Отразить вывод на консоли при выполнении команд git в отчете для лабораторной работе.

Пример 1. Просмотр истории коммитов

```
C:\Users\Gaming-PC>git log
commit de40fc17d8fc284fcd90eebc4def7e163e7f223 (HEAD -> main, tag: v3, origin/main, origin/HEAD)
Author: Evgeny <kkrukov02@gmail.com>
Date: Sun Mar 12 15:54:56 2023 +0300

    return

commit e1f4a2b40c924ba2a2ea7a0de821df164e95fea9
Author: Evgeny <kkrukov02@gmail.com>
Date: Sun Mar 12 15:53:06 2023 +0300

    output of the result

commit ab920ab629e1bf44e2c39f6cec8637514878cdd5
Author: Evgeny <kkrukov02@gmail.com>
Date: Sun Mar 12 15:49:50 2023 +0300

    function

commit adc85208a1f71159eb8675195f2ddf0891a61a17
Author: Evgeny <kkrukov02@gmail.com>
Date: Sun Mar 12 15:47:15 2023 +0300

    int

commit e23fd8466a4e85634d9e295a3730987c57e69621
Author: Evgeny <kkrukov02@gmail.com>
Date: Sun Mar 12 15:46:21 2023 +0300

    scanf a, b, c, x

commit 45e645a2f1085ea961bb1127f59683d97650d79e
Author: Evgeny <kkrukov02@gmail.com>
Date: Sun Mar 12 15:42:27 2023 +0300

    output to the screen

commit 823f5b1b8b1e7f08b2a4fb053a999b17e284dc8b (tag: v2)
Author: Evgeny <kkrukov02@gmail.com>
Date: Sun Mar 12 15:37:42 2023 +0300

    Main and setkocale

commit 6efdf0862c73b39fb2c1067bca3ac81d584b454 (tag: v1)
Author: Evgeny <kkrukov02@gmail.com>
Date: Sun Mar 12 15:29:03 2023 +0300

    Add libraries

commit 1e72f3c0c2f21940e78595057af5655cee7915b2
Author: Evgeny <kkrukov02@gmail.com>
Date: Sun Mar 12 15:20:51 2023 +0300

    Add .gitignore

commit 92573a6ad64de78d4ea3bc45eb96bb0a3001d8e2
Author: Evgeny <kkrukov02@gmail.com>
Date: Sun Mar 12 15:04:44 2023 +0300

    Add .gitignore

commit eaa975c3e889a0e7c95b0139556364569ccee7b7
Author: Evgeny <123456789@EvgenyEvdakov@users.noreply.github.com>
Date: Sun Mar 12 14:46:50 2023 +0300

    update README.md

commit 0d179d4f66e751dbceb77b300a48635db10205db
Author: Evgeny <123456789@EvgenyEvdakov@users.noreply.github.com>
Date: Sun Mar 12 14:41:03 2023 +0300

    create README.md
```

Рисунок 2. Команда “git log”

Вывод только двух записей. Команда - `git log -p -2`.

```
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git log -p -2
commit de40fc17d8fc284fdc90aebc4def77e163e7f223 (HEAD -> main, tag: v3, origin/main, origin/HEAD)
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:54:56 2023 +0300

    return

diff --git a/programm.cpp b/programm.cpp
index 8987297..5b16ffa 100644
--- a/programm.cpp
+++ b/programm.cpp
@@ -25,5 +25,5 @@ int main()
     printf("b = %d\n", b);
     printf("c = %d\n", c);
     printf("x = %d\n", x);
+
+    return 0;
 }

commit e1f4a2b40c924ba2ea7a0de821df164e95fea9
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:53:06 2023 +0300

    output of the result

diff --git a/programm.cpp b/programm.cpp
index 350f381..8987297 100644
--- a/programm.cpp
+++ b/programm.cpp
@@ -21,5 +21,9 @@ int main()
     b = 0;
     if (a != x || a != c)
         b = 1;
+
+    printf("a = %d\n", a);
+    printf("b = %d\n", b);
+    printf("c = %d\n", c);
+    printf("x = %d\n", x);
 }
```

Рисунок 3. Команда “`git log -p -2`”

Чтобы увидеть сокращенную статистику для каждого коммита, можно использовать опцию `--stat` :

```
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git log --stat
commit de40fc17d8fc284fdc90aebc4def77e163e7f223 (HEAD -> main, tag: v3, origin/main, origin/HEAD)
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:54:56 2023 +0300

    return

    programm.cpp | 2 +-
    1 file changed, 1 insertion(+), 1 deletion(-)

commit e1f4a2b40c924ba2ea7a0de821df164e95fea9
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:53:06 2023 +0300

    output of the result

    programm.cpp | 4 +++
    1 file changed, 4 insertions(+)

commit a8920ab625e1bf44e2c59f6cec8637514878cdd5
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:49:50 2023 +0300

    function

    programm.cpp | 7 ++++++
    1 file changed, 7 insertions(+)

commit a0c85208a1f71159eb8675195f2ddf0891a61a17
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:47:15 2023 +0300
```

Рисунок 4. Команда “`git log --stat`”

Опция `oneline` выводит каждый коммит в одну строку, что может быть очень удобным если вы просматриваете большое количество коммитов.

```
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git log --pretty=oneline
de40fc17d8fc284fdc90eebc4def77e163e7f223 (HEAD -> main, tag: v3, origin/main, origin/HEAD) return
e1f4a2b40c924ba2a2ea7a0de821df164e95fea9 output of the result
a8920ab625e1bf44e2c59f6cec8637514878cdd5 function
a0c85208a1f71159eb8675195f2ddf0891a61a17 int
e23fd8466a4e85634d9e295a3730987c57e69621 scanf a, b, c, x
45e645a2f1085ea961bb1127f59683d97650d79e output to the screen
823f5b1b8b1e7f08b2a4fb055a999b17e284dc8b (tag: v2) Main and setkocale
6efdf0882c73b19fb2c1067bcaa3ac81d584b454 (tag: v1) Add libraries
1e72f3c0c2f21940e78595057af5655cee7915b2 Add .gitignore
92573a6ad64de78d4ea3bc45eb96bb0a3001d8e2 Add .gitignore
eaa975c3e889a0e7c95b0139556364569cceedfb7 Update README.md
0d179d4f66e751dbceb77b300a48635db10295db Create README.md
```

Рисунок 5. Команда “git log –pretty=oneline”

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

```
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git log --pretty=format:"%h - %an, %ar : %s"
de40fc1 - Evgeny, 2 hours ago : return
e1f4a2b - Evgeny, 2 hours ago : output of the result
a8920ab - Evgeny, 2 hours ago : function
a0c8520 - Evgeny, 2 hours ago : int
e23fd84 - Evgeny, 2 hours ago : scanf a, b, c, x
45e645a - Evgeny, 2 hours ago : output to the screen
823f5b1 - Evgeny, 2 hours ago : Main and setkocale
6efdf08 - Evgeny, 2 hours ago : Add libraries
1e72f3c - Evgeny, 2 hours ago : Add .gitignore
92573a6 - Evgeny, 2 hours ago : Add .gitignore
eaa975c - Евгений, 3 hours ago : Update README.md
0d179d4 - Евгений, 3 hours ago : Create README.md
```

Рисунок 6. Команда “git log --pretty=format:"%h - %an, %ar : %s"”

Опции `oneline` и `format` являются особенно полезными с опцией `--graph` команды `log` .

```
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git log --pretty=format:"%h %s" --graph
* de40fc1 return
* e1f4a2b output of the result
* a8920ab function
* a0c8520 int
* e23fd84 scanf a, b, c, x
* 45e645a output to the screen
* 823f5b1 Main and setkocale
* 6efdf08 Add libraries
* 1e72f3c Add .gitignore
* 92573a6 Add .gitignore
* eaa975c Update README.md
* 0d179d4 Create README.md
```

Рисунок 7. Команда “git log --pretty=format:"%h %s" --graph”

Ограничение вывода

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

```

C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git log --since=1.day
commit de40fc17d8fc284fdc90eebc4def77e163e/f223 (HEAD -> main, tag: v3, origin/main, origin/HEAD)
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:54:56 2023 +0300

    return

commit e1f4a2b40c924ba2a2ea7a0de821df164e95fea9
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:53:06 2023 +0300

    output of the result

commit a8920ab625e1bf44e2c59f6cec8637514878cdd5
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:49:50 2023 +0300

    function

commit a0c85208a1f71159eb8675195f2ddf0891a61a17
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:47:15 2023 +0300

    int

commit e23fd8466a4e85634d9e295a3730987c57e69621
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:46:21 2023 +0300

    scanf a, b, c, x

commit 45e645a2f1085ea961bb1127f59683d97650d79e
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:42:27 2023 +0300

```

Рисунок 8. Команда “git log --since=1.day”

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.

```

C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git log -S int
commit e1f4a2b40c924ba2a2ea7a0de821df164e95fea9
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:53:06 2023 +0300

    output of the result

commit a0c85208a1f71159eb8675195f2ddf0891a61a17
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:47:15 2023 +0300

    int

commit 45e645a2f1085ea961bb1127f59683d97650d79e
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:42:27 2023 +0300

    output to the screen

commit 823f5b1b8b1e7f08b2a4fb055a999b17e284dc8b (tag: v2)
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:37:42 2023 +0300

    Main and setkocale

```

Рисунок 9. Команда “git log -S int”

Операции отмены

Если вы хотите переделать коммит — то внесите необходимые изменения, добавьте их в индекс и сделайте коммит ещё раз, указав параметр --amend :

```
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git commit -m "test --amend"
[main 6c1dc45] test --amend
1 file changed, 4 insertions(+), 6 deletions(-)

C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git add forgotten_file.cpp

C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git commit -m "test zav --amend" --amend
[main 1135046] test zav --amend
Date: Sun Mar 12 18:36:45 2023 +0300
1 file changed, 6 deletions(-)
```

Рисунок 10. Команда “git --amend”

Просмотр удалённых репозиторийев

Для того, чтобы просмотреть список настроенных удалённых репозиторийев, вы можете запустить команду `git remote` .

```
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git remote -v
origin https://github.com/EvgenyEvdakov/LabaEvd1.2.git (fetch)
origin https://github.com/EvgenyEvdakov/LabaEvd1.2.git (push)
```

Рисунок 11. Команда “git remote -v”

Просмотр удаленного репозитория

Если хотите получить побольше информации об одном из удалённых репозиторийев, вы можете использовать команду `git remote show <remote>`.

```
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git remote show origin
* remote origin
Fetch URL: https://github.com/EvgenyEvdakov/LabaEvd1.2.git
Push URL: https://github.com/EvgenyEvdakov/LabaEvd1.2.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 (local out of date)
```

Рисунок 12. Команда “git show origin”

Удаление и переименование удалённых репозиторийев

Для переименования удалённого репозитория можно выполнить `git remote rename` .

```
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git remote rename LabaratornayaCross2 Laba1.2
Renaming remote references: 100% (1/1), done.
```

Рисунок 13. Команда “git remote rename”

Работа с тегами

Просмотр списка тегов

Просмотреть список имеющихся тегов в Git можно очень просто. Достаточно набрать команду `git tag` (параметры `-l` и `--list` опциональны):

```
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git tag
v1
v2
v3
```

Рисунок 14. Команда “git tag”

Создание аннотированного тега в Git выполняется легко. Самый простой способ — это указать `-a` при выполнении команды `tag`

```
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git tag -a v4 -m "v4"
```

Рисунок 15. Создание тега.

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>`

```
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git push origin --tags
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 12 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (13/13), 1.30 KiB | 1.30 MiB/s, done.
Total 13 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/EvgenyEvdakov/LabaEvd1.2.git
 * [new tag]         v1 -> v1
 * [new tag]         v2 -> v2
 * [new tag]         v3 -> v3
 * [new tag]         v4 -> v4
```

Рисунок 16. Команда “git push origin –tags”

Задание 3.

Написал небольшую программу в новом файле `programm.cpp`, сделал не менее 7-ми коммитов с 3-мя тегами. Использование команды “`git log –graph –pretty=oneline –abbrev-commit`”.

```

C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git log --graph --pretty=oneline --abbrev-commit
* de40fc1 (HEAD -> main, tag: v3) return
* e1f4a2b output of the result
* a8920ab function
* a0c8520 int
* e23fd84 scanf a, b, c, x
* 45e645a output to the screen
* 823f5b1 (tag: v2) Main and setkocale
* 6efdf08 (tag: v1) Add libraries
* 1e72f3c (origin/main, origin/HEAD) Add .gitignore
* 92573a6 Add .gitignore
* eaa975c Update README.md
* 0d179d4 Create README.md

```

Рисунок 17. История хранилища.

Задание 5.

Посмотрел содержимое коммитов командой `git show <ref>`, где <ref>:

- 1) HEAD : последний коммит;

```

C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git show HEAD
commit de40fc17d8fc284fdc90eebc4def77e163e7f223 (HEAD -> main, tag: v3)
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:54:56 2023 +0300

    return

diff --git a/programm.cpp b/programm.cpp
index 8987297..5b16ffa 100644
--- a/programm.cpp
+++ b/programm.cpp
@@ -25,5 +25,5 @@ int main()
     printf("b = %d\n", b);
     printf("c = %d\n", c);
     printf("x = %d\n", x);
-
+    return 0;
}

```

Рисунок 18. Последний коммит.

- 2) HEAD~1 : предпоследний коммит (и т. д.);

```

C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git show HEAD~1
commit e1f4a2b40c924ba2ea7a0de821df164e95fea9
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:53:06 2023 +0300

    output of the result

diff --git a/programm.cpp b/programm.cpp
index 350f381..8987297 100644
--- a/programm.cpp
+++ b/programm.cpp
@@ -21,9 +21,9 @@ int main()
     b = 0;
     if (a != x || a != c)
         b = 1;
+    printf("a = %d\n", a);
+    printf("b = %d\n", b);
+    printf("c = %d\n", c);
+    printf("x = %d\n", x);
}

```

Рисунок 19. Предпоследний коммит.

- 3) b34a0e : коммит с указанным хэшем.


```

C:\Users\Gaming-PC\LabaEvd1.2>git show a8920ab
commit a8920ab625e1bf44e2c59f6cec8637514878cdd5
Author: Evgeny <kkrutkov02@gmail.com>
Date: Sun Mar 12 15:49:50 2023 +0300

    function

diff --git a/programm.cpp b/programm.cpp
index f89de79..350f381 100644
--- a/programm.cpp
+++ b/programm.cpp
@@ -15,4 +15,11 @@ int main()
     scanf_s("%d", &c);
     printf("C2<E2><E5><E4><E8><F2><E5> <E7><ED><E0><F7><E5><ED><E8><E5> d: ");
     scanf_s("%d", &x);
+    if (a == x)
+        b = a;
+    if (a != x && a == c)
+        b = 0;
+    if (a != x || a != c)
+        b = 1;
+}

```

Рисунок 20. Коммит с указанным хэшем.

Задание 6. Откат к заданной версии.

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

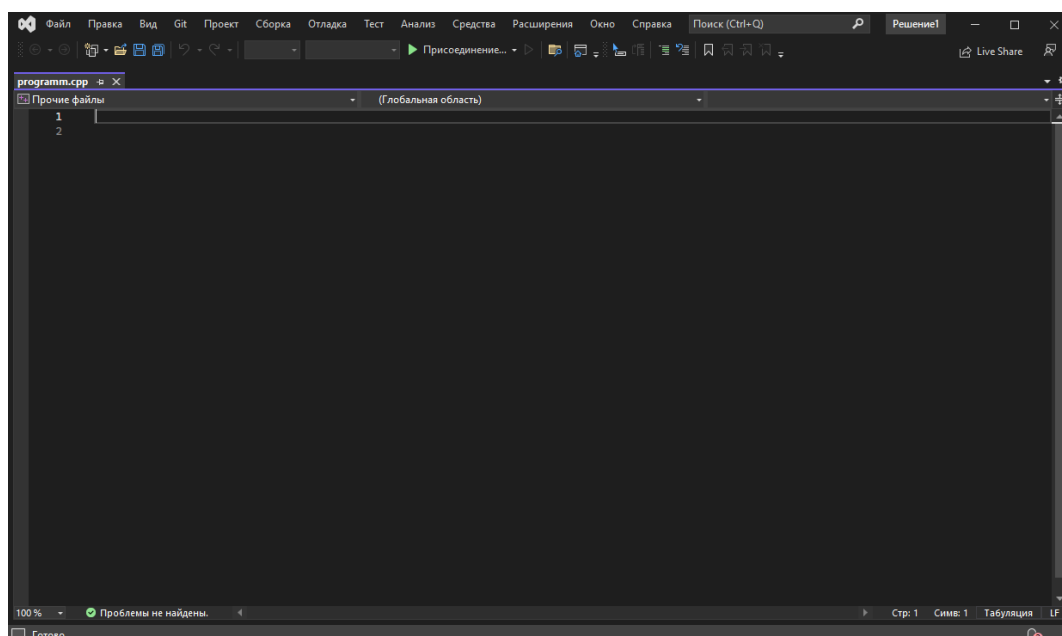


Рисунок 21. Удаление программ

1.2. Удалил это изменение с помощью команды `git checkout -- main.cpp.docx`.

```

C:\Users\Gaming-PC\LabaEvd1.2>git checkout -- programm.cpp
C:\Users\Gaming-PC\LabaEvd1.2>

```

Рисунок 22. Команда “git checkout – programm.cpp”

Код вернулся к прежнему состоянию.

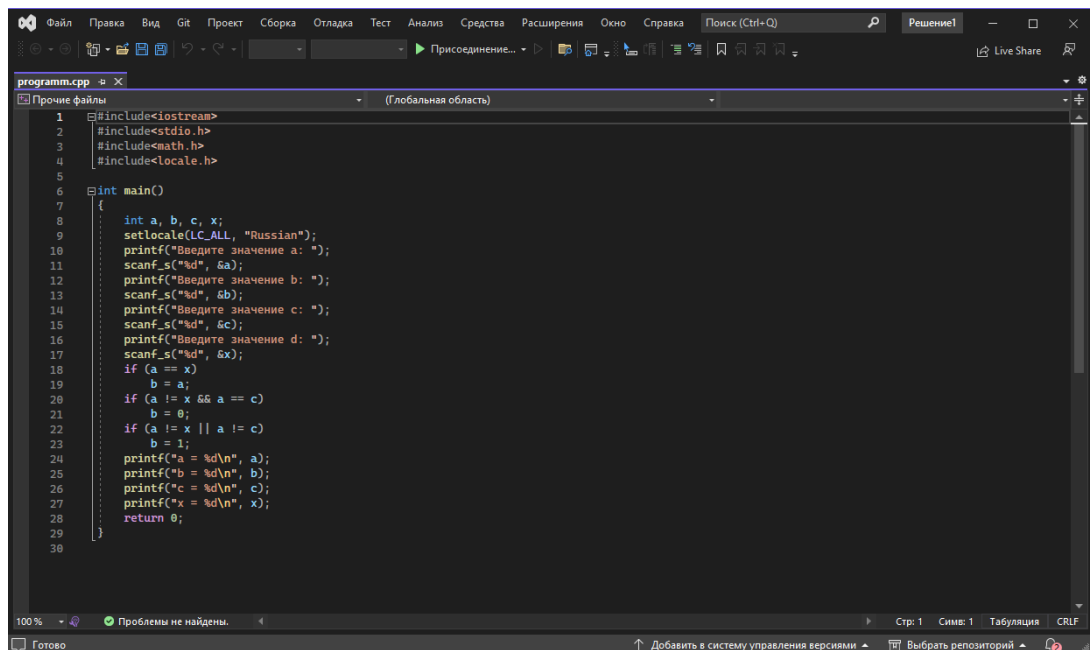


Рисунок 23. Восстановление программы.

1.3. Вновь повторил пункт 1.1. и сделал коммит.

```
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git commit -m "delete"
[main acb4ece] delete
1 file changed, 29 deletions(-)

C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>
```

Рисунок 24. Коммит.

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

```
C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>git reset --hard HEAD~1
HEAD is now at de40fc1 return

C:\Users\Gaming-PC\LabaEvd1.2\LabaEvd1.2>
```

Рисунок 25. Возвращение к предпоследней версии коммита.

Ссылка на репозиторий: <https://github.com/EvgenyEvdakov/LabaEvd1.2>

После проделанных пунктов можно сделать вывод, что можно отменять ненужные изменения, в случае если коммит не произошел и изменения не нужны. Также, если коммит был уже сделан, то можно вернуться к предпоследней версии коммита (где было сохранение до сохранения с ненужными изменениями).

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 <shortname> <url>`.

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 <remote-name> <branch-name>*

10) Как выполнить просмотр удаленного репозитория?

Если хотите получить побольше информации об одном из удалённых репозиториях, вы можете использовать команду: *git remote show <remote>*.

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 <tagname>*.

Для удаления тега в локальном репозитории достаточно выполнить команду *git tag -d <tagname>*.

Если вы хотите получить версии файлов, на которые указывает тег, то вы можете сделать *git checkout <tagname>* 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: исследовав базовые возможности системы контроля версий Git для работы с локальными репозиториями.