

1) Создан пустой репозиторий

```
David@HOME-PC MINGW64 ~/Desktop/test_1
$ git init
Initialized empty Git repository in C:/Users/David/Desktop/test_1/.git/
```

2) Добавление файлов к отслеживанию

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git add code.cpp
```

3) Просмотр состояния репозитория

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   code.cpp
```

4) Фиксируем изменения

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git commit -am "add file"
[master (root-commit) 86310b8] add file
1 file changed, 208 insertions(+)
create mode 100644 code.cpp
```

5) Исправляем сообщение предыдущего коммита

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git commit --amend -m "add code.cpp"
[master 083d03a] add code.cpp
Date: Tue Oct 27 10:14:57 2020 +0300
1 file changed, 208 insertions(+)
create mode 100644 code.cpp
```

6) Делаем изменения в файле, еще один коммит и смотрим разницу

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git commit -am "add change"
[master deb2b90] add change
1 file changed, 1 insertion(+)

David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git log
commit deb2b905d46baa273a6b6555ad2bebe6e20bbb2d (HEAD -> master)
Author: OOPDavidka <nikitabodr123@mail.ru>
Date: Tue Oct 27 10:27:49 2020 +0300

    add change

commit 083d03ae9fa9465c16ffbbba13c61546fe8e6c25a
Author: OOPDavidka <nikitabodr123@mail.ru>
Date: Tue Oct 27 10:14:57 2020 +0300

    add code.cpp
```

7),8) Создаем ещё одну ветку и переключаемся на неё

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git branch "v2"
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git checkout "v2"
Switched to branch 'v2'
David@HOME-PC MINGW64 ~/Desktop/test_1 (v2)
```

9) Вносим некоторые изменения и смотрим разницу между ветками

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (v2)
$ git diff master
diff --git a/code.cpp b/code.cpp
index 1a5ee15..578a281 100644
--- a/code.cpp
+++ b/code.cpp
@@ -3,6 +3,7 @@
#include <iomanip>
using namespace std;
//ГЛОБАЛЬНЫЕ ИЗМЕНЕНИЯ
+//добавил ещё один комментарий
//новый комментарий
//создание и заполнение СЛАУ
float** create_matrix(unsigned n) {
```

10) фиксируем изменения

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (v2)
$ git commit -am "fixed"
[v2 e02f7f5] fixed
1 file changed, 1 insertion(+)
```

11) Переключаемся на ветку мастер и смотрим что находится в файлах.

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (v2)
$ git checkout master
Switched to branch 'master'
```

(блокнот замечает что файл был изменён сторонней программой, и можно сразу же просмотреть содержимое файлов в этой ветке)

12) Объединим ветки master и v2

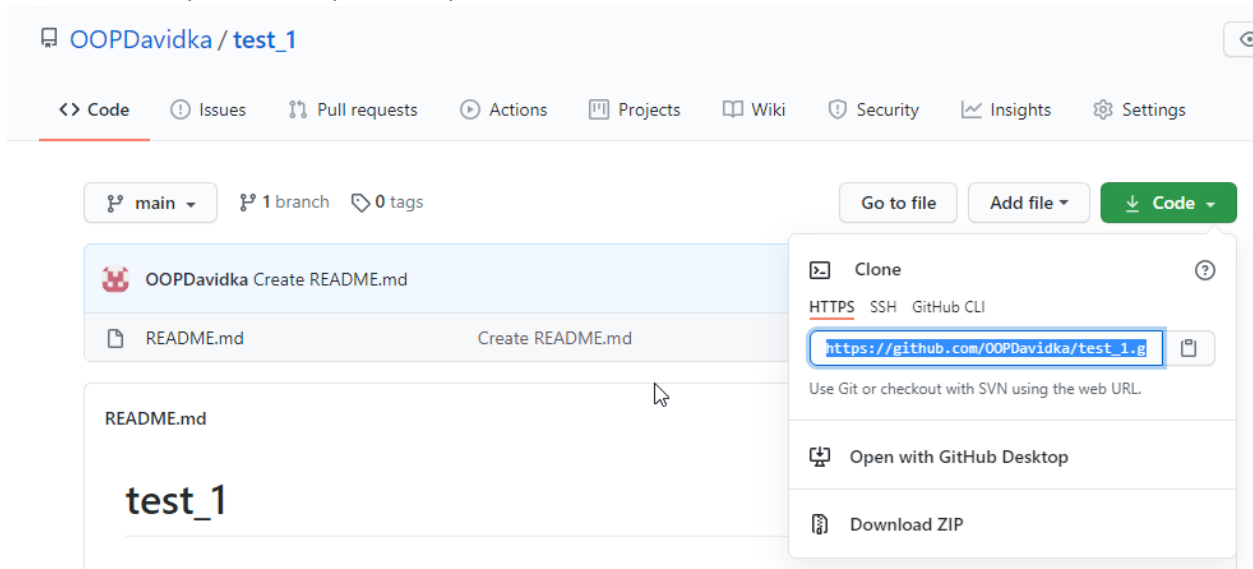
```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git merge v2
Updating deb2b90..e02f7f5
Fast-forward
 code.cpp | 1 +
1 file changed, 1 insertion(+)
```

13) Клонировем удалённый репозиторий

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git clone https://github.com/OOPDavidka/test_1.git
Cloning into 'test_1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 595 bytes | 3.00 KiB/s, done.
```

.git	27.10.2020 10:36	Папка с файлами	
test_1	27.10.2020 10:38	Папка с файлами	
code.cpp	27.10.2020 10:36	C++ Source file	7 КБ

14) Создадим удалённый репозиторий на гитхабе



Связываем локальный репозиторий с удалённым и публикуем ветку мастер в удалённом репозитории


```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git remote add origin https://github.com/OOPDavidka/test_1.git
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
```


```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git push --set-upstream origin master
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 4 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 2.87 KiB | 490.00 KiB/s, done.
Total 12 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/OOPDavidka/test_1/pull/new/master
remote:
To https://github.com/OOPDavidka/test_1.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```


Делаем локальные изменения в файле , коммитим и отправляем в удалённый репозиторий


```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git commit -am "14"
[master 2618f58] 14
1 file changed, 1 insertion(+)
```

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git push
Everything up-to-date
```

 master had recent pushes 4 minutes ago [Compare & pull request](#)

 master ▾

 2 branches


 0 tags


[Go to file](#)

[Add file ▾](#)

[Code ▾](#)

This branch is 4 commits ahead, 1 commit behind main. [Pull request](#) [Compare](#)

 OOPDavidka 14 2618f58 17 minutes ago 4 commits

 code.cpp

14

17 minutes ago

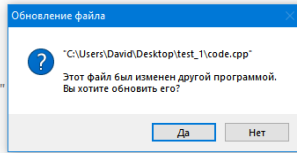
Help people interested in this repository understand your project by adding a README. [Add a README](#)

Далее сделаем изменения в файле удалённого репозитория и отправим их в локальный

```
1 #include <iostream>
2 #include <math.h>
3 #include <iomanip>
4 using namespace std;
5 //коммит перед отправкой изменений с удалённого репозитория на локальный
6 //коммит перед отправкой
7 //ГЛОБАЛЬНЫЕ ИЗМЕНЕНИЯ)
8 //добавил ещё один комментарий
9 //новый комментарий
10 //создание и заполнение СЛАУ
11 float** create_matrix(unsigned n) {
12     float** m = new float*[n]; //выделение памяти
```

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git pull
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 1.29 KiB | 2.00 KiB/s, done.
From https://github.com/OOPDavidka/test_1
  2618f58..b7596e1 master    -> origin/master
* [new branch]      main    -> origin/main
Updating 2618f58..b7596e1
Fast-forward
 code.cpp | 1 +
1 file changed, 1 insertion(+)
```

```
1 #include <iostream>
2 #include <math.h>
3 #include <iomanip>
4 using namespace std;
5 //коммит перед отправкой
6 //ГЛОБАЛЬНЫЕ ИЗМЕНЕНИЯ
7 //добавил ещё один комментарий
8 //новый комментарий
9 //создание и заполнение СЛАУ
10 float** create_matrix(unsigned n) {
11     float** m = new float*[n]; //указатель на указатели
12     for (unsigned i = 0; i < n; i++) {
13         m[i] = new float [n+1];
14     }
15     //Цикл заполнения
16     for (unsigned i = 0; i < n; i++) {
17         for (unsigned j = 0; j <= n; j++) {
18             if (j < n) {
19                 cout << "a[" << i+1 << "][" << j+1 << "] = "
20                     << " "; cin >> m[i][j];
21             }
22             else {
23                 cout << "y[" << i+1 << "] = ";
24                 cin >> m[i][j];
25             }
26         }
27     }
28     return m;
29 }
30
31 //выделение памяти под СЛАУ
32 float** create_matrix_NULL(unsigned n) {
33     float** m = new float* [n];
34 }
```



```
1 #include <iostream>
2 #include <math.h>
3 #include <iomanip>
4 using namespace std;
5 //коммит перед отправкой изменений с удалённого репозитория на локальный
6 //коммит перед отправкой
7 //ГЛОБАЛЬНЫЕ ИЗМЕНЕНИЯ
8 //добавил ещё один комментарий
9 //новый комментарий
10 //создание и заполнение СЛАУ
11 float** create_matrix(unsigned n) {
12     float** m = new float*[n]; //указатель на указатели
13     for (unsigned i = 0; i < n; i++) {
14         m[i] = new float [n+1];
15     }
16     //Цикл заполнения
17 }
```

Текстовый редактор заметил изменения и принял их.

15.1) Создаем worktree для ветки мастер(для этого нужно переключиться на другую ветку)

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git checkout v2
Switched to branch 'v2'

David@HOME-PC MINGW64 ~/Desktop/test_1 (v2)
$ git worktree add ../folder master
Preparing worktree (checking out 'master')
HEAD is now at b7596e1 Update code.cpp
```



folder



OOP



test_1

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (v2)
$ git worktree list
C:/Users/David/Desktop/test_1 e02f7f5 [v2]
C:/Users/David/Desktop/folder b7596e1 [master]
```

15.2) Моделируем конфликт и решаем его, для этого в каждой ветке закоммитим изменения в одной и той же строке кода и попытаемся объединить ветки.
Но сначала нужно сделать очистку worktree

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (v2)
$ git worktree prune
```

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (v2)
$ git commit -am "conflict"
[v2 a36d0ac] conflict
1 file changed, 1 insertion(+)
```

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (v2)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git commit -am "conflicts"
[master 4589ca0] conflicts
1 file changed, 1 insertion(+)
```

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git merge v2
Auto-merging code.cpp
CONFLICT (content): Merge conflict in code.cpp
Automatic merge failed; fix conflicts and then commit the result.
```

При попытке объединить ветки консоль выдала ошибку а в текстовом файле появилась следующая конструкция:

```
<<<<<<< HEAD
//десятая строка кода
=====
//конфликт)
>>>>>>> v2
```

Решаем конфликт выбором необходимого изменения и снова делаем слияние веток

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master|MERGING)
$ git commit -am "solve problem"
[master fb8b33b] solve problem

David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git merge v2
Already up to date.
```

15.3)Использование тэгов

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git tag -a v2.0 -m "version 2.0"

David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git tag
v2.0

David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git show
commit fb8b33b5d35658656de202e717f05b55d1e7460d (HEAD -> master, tag: v2.0)
Merge: 4589ca0 a36d0ac
Author: OOPDavidka <nikitabodr123@mail.ru>
Date: Tue Oct 27 12:04:14 2020 +0300

    solve problem
```

```
David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git push origin --tags
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 767 bytes | 383.00 KiB/s, done.
Total 8 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/OOPDavidka/test_1.git
 * [new tag]          v2.0 -> v2.0

David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git tag -d v2.0

David@HOME-PC MINGW64 ~/Desktop/test_1 (master)
$ git tag
v2.0
-d
```