1. Создать локальный репозиторий:

Student@301\_14 MINGW64 ~

$ cd C:/Users/Student/source/repos/test

Student@301\_14 MINGW64 ~/source/repos/test

$ git init

Initialized empty Git repository in C:/Users/Student/source/repos/test/.git/

Student@301\_14 MINGW64 ~/source/repos/test (master)

$ git add .

warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it

Student@301\_14 MINGW64 ~/source/repos/test (master)

$ git commit -m "Добавлены новые файлы"

[master (root-commit) b2efe62] Добавлены новые файлы

6 files changed, 533 insertions(+)

create mode 100644 .gitignore

create mode 100644 test.sln

create mode 100644 test/App.config

create mode 100644 test/Program.cs

create mode 100644 test/Properties/AssemblyInfo.cs

create mode 100644 test/test.csproj

Student@301\_14 MINGW64 ~/source/repos/test (master)

$

1. Создать и работать с ветками
2. Student@301\_14 MINGW64 ~/source/repos/test (master)
3. $ git checkout -b feature-branch
4. Switched to a new branch 'feature-branch'
5. Student@301\_14 MINGW64 ~/source/repos/test (feature-branch)
6. $ git add .
7. Student@301\_14 MINGW64 ~/source/repos/test (feature-branch)
8. $ git commit -m "В скрипт Program.cs было добавлено пару команд"
9. [feature-branch b332de2] В скрипт Program.cs было добавлено пару команд
10. 1 file changed, 3 insertions(+)
11. Student@301\_14 MINGW64 ~/source/repos/test (feature-branch)
12. $ git checkout master
13. Switched to branch 'master'
14. Student@301\_14 MINGW64 ~/source/repos/test (master)
15. $ git merge feature-branch
16. Updating b2efe62..b332de2
17. Fast-forward
18. test/Program.cs | 3 +++
19. 1 file changed, 3 insertions(+)
20. Student@301\_14 MINGW64 ~/source/repos/test (master)
21. $

3. Создание и расширение конфликтов слияния

Student@301\_14 MINGW64 ~/source/repos/test (master)

$ git checkout -b conflict-branch

Switched to a new branch 'conflict-branch'

Student@301\_14 MINGW64 ~/source/repos/test (conflict-branch)

$ git checkout master

Switched to branch 'master'

M test/Program.cs

Student@301\_14 MINGW64 ~/source/repos/test (master)

$ git merge conflict-branch

Already up to date.

Student@301\_14 MINGW64 ~/source/repos/test (master)

$ git add .

Student@301\_14 MINGW64 ~/source/repos/test (master)

$ git commit -m "Разрешен конфликт в Program.cs"

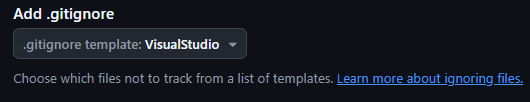
[master 6d3477f] Разрешен конфликт в Program.cs

1 file changed, 2 insertions(+)

Student@301\_14 MINGW64 ~/source/repos/test (master)

$

4. Настроить .gitignore



5. Просмотр истории коммитов и восстановление версий

Student@301\_14 MINGW64 ~/source/repos/test (master)

$ git log

commit 6d3477fea0f8cd1d4ffdf06768e1e0995444dff4 (HEAD -> master)

Author: Your Name <you@example.com>

Date: Thu Oct 24 17:07:28 2024 +0400

Разрешен конфликт в Program.cs

commit b332de2ef16dc0124ca801d76da6641c6fea5b5d (feature-branch, conflict-branch)

Author: Your Name <you@example.com>

Date: Thu Oct 24 16:58:31 2024 +0400

В скрипт Program.cs было добавлено пару команд

commit b2efe62815ad608108abcdeca8d6d170e3ec6f73

Author: Your Name <you@example.com>

Date: Thu Oct 24 16:54:51 2024 +0400

Добавлены новые файлы

Student@301\_14 MINGW64 ~/source/repos/test (master)

$ git checkout b332de2ef16dc0124ca801d76da6641c6fea5b5d

Note: switching to 'b332de2ef16dc0124ca801d76da6641c6fea5b5d'.

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 b332de2 В скрипт Program.cs было добавлено пару команд

Student@301\_14 MINGW64 ~/source/repos/test ((b332de2...))

$ git checkout master

Previous HEAD position was b332de2 В скрипт Program.cs было добавлено пару команд

Switched to branch 'master'

Student@301\_14 MINGW64 ~/source/repos/test (master)

$

6. Использовать Git Stash

Student@301\_14 MINGW64 ~/source/repos/test (master)

$ git stash

Saved working directory and index state WIP on master: 6d3477f Разрешен конфликт в Program.cs

Student@301\_14 MINGW64 ~/source/repos/test (master)

$ git stash pop

error: Your local changes to the following files would be overwritten by merge:

test/Program.cs

Please commit your changes or stash them before you merge.

Aborting

On branch master

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: test/Program.cs

no changes added to commit (use "git add" and/or "git commit -a")

The stash entry is kept in case you need it again.

Student@301\_14 MINGW64 ~/source/repos/test (master)

$

7. Создание тегов и работа с ними:

Student@301\_14 MINGW64 ~/source/repos/test (master)

$ git tag v1.0

Student@301\_14 MINGW64 ~/source/repos/test (master)

$ git tag

v1.0

Student@301\_14 MINGW64 ~/source/repos/test (master)

$ git checkout v1.0

Note: switching to 'v1.0'.

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 6d3477f Разрешен конфликт в Program.cs

M test/Program.cs

Student@301\_14 MINGW64 ~/source/repos/test ((v1.0))

$ git checkout master

Switched to branch 'master'

M test/Program.cs

Student@301\_14 MINGW64 ~/source/repos/test (master)

$