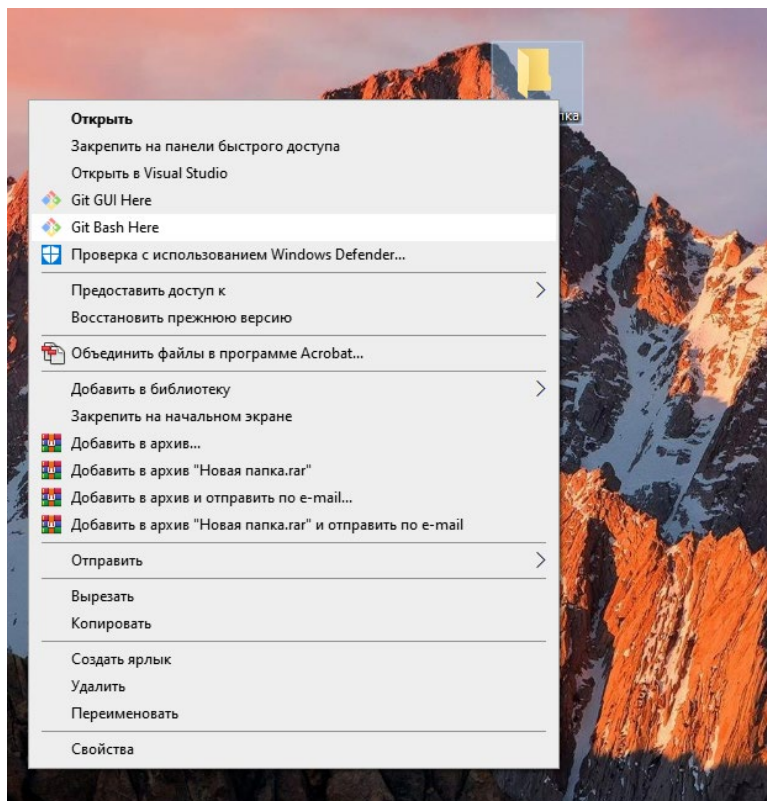


Иванов Илья Ивт-19

Лабораторная №1. Работа с Git

Создать репозиторий.

Чтобы создать репозиторий нужно выбрать папку и после нажать правой кнопкой мыши и выбрать **Git Bush Here**.



Git init- создание репозитория

```
Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (master)
$ git init
Initialized empty Git repository in C:/Users/Илья/Desktop/Новая папка/.git/

Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (master)
$ |
```

Добавить файлы к отслеживанию.

`git add <имя файла>`

```
Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (master)
$ git add File.txt

Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (master)
$ |
```

Просмотреть состояние репозитория.

`git status`

```
Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (master)
$ git status
On branch master

No commits yet

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

Зафиксировать изменения.

`git commit -am "<Сообщение коммита>"` - сделать коммит

```
Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (master)
$ git commit -am "First commit"
[master (root-commit) 9bf4401] First commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 File.txt
```

Исправить сообщение предыдущего коммита.

`git commit --amend -m <Новое сообщение коммита>`

```
Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (master)
$ git commit --amend -m "New commit"
[master e882ab8] New commit
Date: Fri Feb 5 05:42:43 2021 +0900
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 File.txt
```

Сделать коммит, посмотреть разницу между коммитами.

Внесём изменения в файл

`git log -p` - посмотреть изменения

```

commit 287f6ac0e8edaf53c7cbfcc95463e251b4fad5c6 (HEAD -> master)
Author: lxrdrnbn <ilyaivanov12345@gmail.com>
Date:   Fri Feb 5 05:52:44 2021 +0900

    Changed file

diff --git a/File.txt b/File.txt
index e69de29..f4acf0c 100644
--- a/File.txt
+++ b/File.txt
@@ -0,0 +1 @@
+New line
\ No newline at end of file

commit 87dd96bc900cae9524ce97fd4c3c3fe7653c7cd9
Author: lxrdrnbn <ilyaivanov12345@gmail.com>
Date:   Fri Feb 5 05:51:54 2021 +0900

    New commit

diff --git a/File.txt b/File.txt
new file mode 100644
index 0000000..e69de29

```

Создать новую ветку.

`git branch <имя ветки>` - создание ветки

```

Илья@DESKTOP-QB57ON5 MINGW64 ~/Desktop/Новая папка (master)
$ git branch New_branch

```

Переключиться на новую ветку.

`git checkout <Имя ветки>`

```

Илья@DESKTOP-QB57ON5 MINGW64 ~/Desktop/Новая папка (master)
$ git checkout New_branch
Switched to branch 'New_branch'

```

Внести изменения. Посмотреть разницу между ветками.

В новой ветке внесем изменения в файл и сделаем коммит

`git diff <ветка1> <ветка2>` - разница между ветками

```
Илья@DESKTOP-QB57ON5 MINGW64 ~/Desktop/Новая папка (New_branch)
$ git commit -am "New branch commit"
[New_branch b0b2fe3] New branch commit
1 file changed, 2 insertions(+), 1 deletion(-)

Илья@DESKTOP-QB57ON5 MINGW64 ~/Desktop/Новая папка (New_branch)
$ git diff master New_branch
diff --git a/File.txt b/File.txt
index f4acf0c..5f0d30b 100644
--- a/File.txt
+++ b/File.txt
@@ -1,2 @@
-New line
\ No newline at end of file
+New line
+Second line
\ No newline at end of file
```

Зафиксировать изменения.

```
Илья@DESKTOP-QB57ON5 MINGW64 ~/Desktop/Новая папка (New_branch)
$ git commit -am "New branch commit"
[New_branch b0b2fe3] New branch commit
1 file changed, 2 insertions(+), 1 deletion(-)
```

Посмотреть что находится в файлах на ветке master.

`git show <ветка> :<файл>`

```
Илья@DESKTOP-QB57ON5 MINGW64 ~/Desktop/Новая папка (New_branch)
$ git show master File.txt
commit 287f6ac0e8edaf53c7cbfcc95463e251b4fad5c6 (master)
Author: lxrdrnbnr <ilyaivanov12345@gmail.com>
Date: Fri Feb 5 05:52:44 2021 +0900

    Changed file

diff --git a/File.txt b/File.txt
index e69de29..f4acf0c 100644
--- a/File.txt
+++ b/File.txt
@@ -0,0 +1 @@
+New line
\ No newline at end of file
```

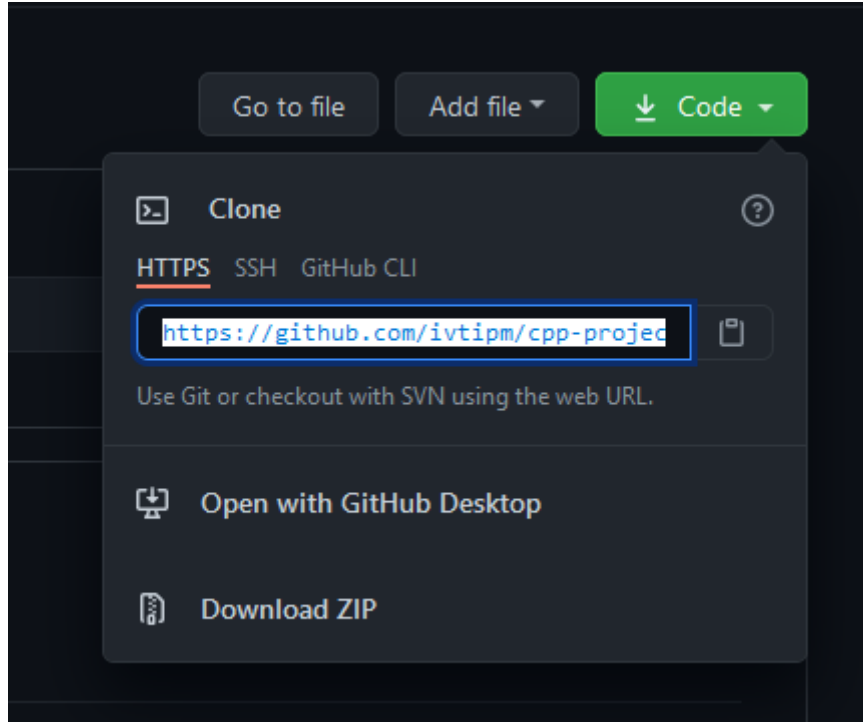
Объединить ветки.

`git merge <имя ветки с которой хотим объединиться>`

```
Илья@DESKTOP-QB57ON5 MINGW64 ~/Desktop/Новая папка (master)
$ git merge New_branch
Already up to date.
```

Клонировать удаленный репозиторий.

Нужно выбрать репозиторий на GitHub и нажать на кнопку Code, в списке скопировать ссылку



`git clone <Ссылка на удалённый репозиторий>`

```
Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (New_branch)
$ git clone https://github.com/ivtipm/cpp-projects-lxrdrgnbrn.git
Cloning into 'cpp-projects-lxrdrgnbrn'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 8 (delta 1), reused 1 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.
Resolving deltas: 100% (1/1), done.
```

Отправить изменения в удаленный репозиторий, забрать изменения из удаленного репозитория.

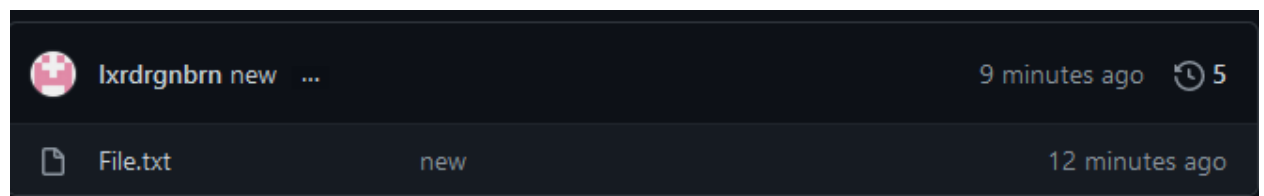
`git remote add origin <ссылка на репозиторий>`

```
Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (New_branch)
$ git remote add origin https://github.com/ivtipm/cpp-projects-lxrdrgnbrn.git
```

“git push <ссылка на репозиторий>”

```
Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (master)
$ git push https://github.com/ivtipm/cpp-projects-lxrdrgnbrn.git
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (11/11), 864 bytes | 864.00 KiB/s, done.
Total 11 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/ivtipm/cpp-projects-lxrdrgnbrn/pull/new/master
remote:
To https://github.com/ivtipm/cpp-projects-lxrdrgnbrn.git
 * [new branch]      master -> master

Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (master)
```



git pull <ссылка на удаленный репозиторий> <название ветки, на которую хотим загрузить изменения>||.

```
1 lines (1 sloc) | 19 Bytes
1 I am from GitHub!
```

```
Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (master)
$ git pull https://github.com/ivtipm/cpp-projects-lxrdrgnbrn.git master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 636 bytes | 70.00 KiB/s, done.
From https://github.com/ivtipm/cpp-projects-lxrdrgnbrn
 * branch      master      -> FETCH_HEAD
Updating 5cdbc7f..0eae230
Fast-forward
 File.txt | 1 +
1 file changed, 1 insertion(+)
```



File – Блокнот

Файл Правка Формат Вид

I am from GitHub!

Создать worktree для ветки master.

Git worktree add -b <название новой ветки> <название рабочего дерева><путь к новому рабочему дереву>

```
Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (master)
$ git worktree add newTree
Preparing worktree (new branch 'newTree')
HEAD is now at 0eae230 Update File.txt
```

Имя	Дата изменения	Тип	Размер
cpp-projects-lxrdrnbn	05.02.2021 6:12	Папка с файлами	
File	05.02.2021 6:20	Текстовый докум...	1 КБ
newTree	05.02.2021 6:23	Папка с файлами	

Моделирование конфликта. Решение данного конфликта.

Смоделируем конфликт двух веток, когда в файлах зафиксированы изменения в одном и том же месте.

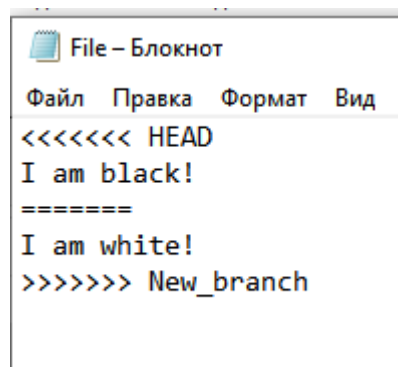
```
Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (master)
$ git commit -am "conflict"
[master a7d5d29] conflict
1 file changed, 1 insertion(+), 1 deletion(-)

Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (master)
$ git checkout New_branch
Switched to branch 'New_branch'

Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (New_branch)
$ git commit -am "Conflict"
[New_branch 38c3c0a] Conflict
1 file changed, 1 insertion(+), 2 deletions(-)

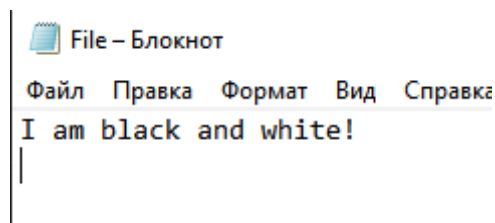
Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (New_branch)
$ git checkout master
Switched to branch 'master'

Илья@DESKTOP-QB570N5 MINGW64 ~/Desktop/Новая папка (master)
$ git merge New_branch
Auto-merging File.txt
CONFLICT (content): Merge conflict in File.txt
Automatic merge failed; fix conflicts and then commit the result.
```

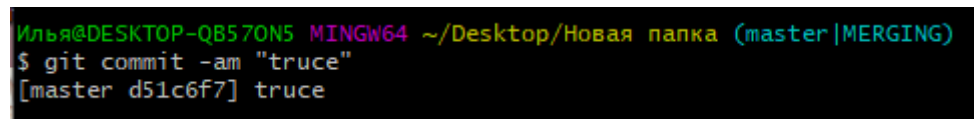


```
File - Блокнот
Файл  Правка  Формат  Вид
<<<<<< HEAD
I am black!
=====
I am white!
>>>>>> New_branch
```

Теперь чтобы его решить нужно в ручную залезть и переделать файл, после провести коммит.



```
File - Блокнот
Файл  Правка  Формат  Вид  Справка
I am black and white!
```



```
Илья@DESKTOP-QB57ON5 MINGW64 ~/Desktop/Новая папка (master|MERGING)
$ git commit -am "true"
[master d51c6f7] true
```

Использование тегов.

Git имеет возможность помечать определённые моменты в истории как важные. Чтобы посмотреть список имеющихся тегов нужно прописать команду **"git tag"** (-l, --list).

Создание меток.

В Git есть 2 типа меток: легковесные и аннотированные.

Легковесный тег — это что-то очень похожее на ветку, которая не изменяется — просто указатель на определённый коммит.

А вот аннотированные теги хранятся в базе данных Git как полноценные объекты. Они имеют контрольную сумму, содержат имя автора, его e-mail и дату создания, имеют

комментарий и могут быть подписаны и проверены с помощью GNU Privacy Guard (GPG).

Аннотированные метки

Чтобы создать аннотированную метку нужно прописать команду **"git tag -a "<Название тега>" -m "<сообщение к тегу>"**

```
Илья@DESKTOP-QB57ON5 MINGW64 ~/Desktop/Новая папка (master)
$ git tag -a v1.0 -m "version_1.0"

Илья@DESKTOP-QB57ON5 MINGW64 ~/Desktop/Новая папка (master)
$ git tag
v1.0
```

Легковесные метки

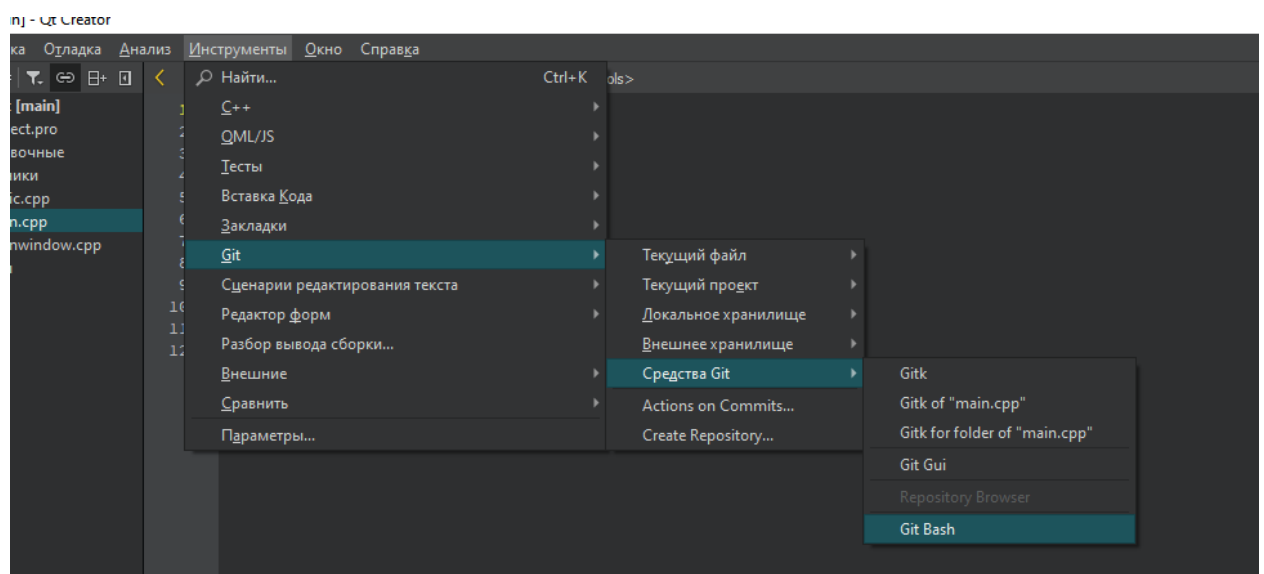
Чтобы создать легковесную метку нужно прописать команду **"git tag <название тега>"**

```
Илья@DESKTOP-QB57ON5 MINGW64 ~/Desktop/Новая папка (master)
$ git tag v1.1

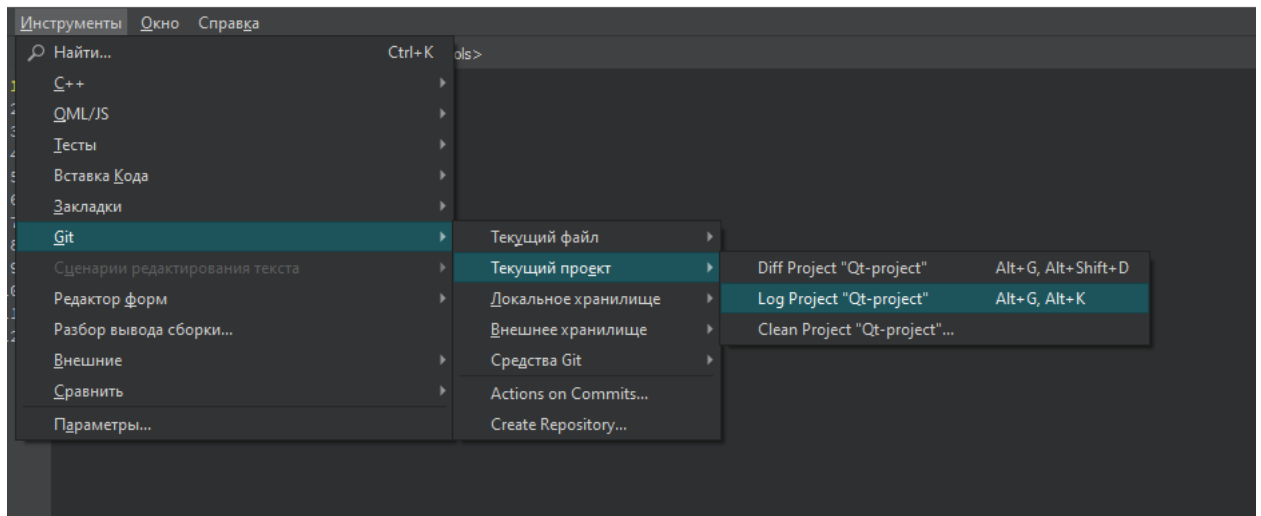
Илья@DESKTOP-QB57ON5 MINGW64 ~/Desktop/Новая папка (master)
$ git tag
v1.0
v1.1
```

Git в Qt Creator.

В Qt Creator можно работать со встроенным Git, для этого нужно перейти **Инструменты->Git->Средства Git->Git Bash**. Это откроет консоль Git прямо в Qt Creator.



Также в Qt Creator можно посмотреть состояние, лог и другие свойства текущего файла или проекта, для этого надо также перейти в подменю Git и оттуда в текущий файл или проект.



Вопросы:

1. Что такое система управления версиями? Для чего она используется?

Система управления версиями – программа, которая запоминает все модификации данных, и при необходимости позволяет выполнить откат. Используется для облегчения хранения версий вашего проекта, а также для удобства передачи этих версий между участниками данного проекта.

2. Что такое репозиторий (локальный и удалённый)?

Репозиторий это место, в котором хранятся данные вашего проекта, за которыми следит система управления версиями.

Локальный репозиторий это репозиторий, который находится на вашем компьютере, к примеру обычная папка и есть репозиторий.

Удалённый репозиторий это место для хранения ваших данных в облаке, например на GitHub.

3. Какие файлы следует добавлять к отслеживанию, а какие нет.

В отслеживании должны быть файлы исходных кодов и другие файлы, необходимые для компиляции и запуска программы. Исполняемые файлы не отслеживаются, потому что, их всегда можно получить после компиляции и чтобы не засорять репозиторий.

4. В каких случаях создавать ветку?

Если планируются обширные изменения, то создаются отдельные ветви, чтобы параллельно существовала исходная версия программы и версия, в которую вносятся изменения – рабочая версия.

5. Что такое конфликт? Как исправить?

Конфликт слияния это ситуация, когда в двух объединяемых ветках (master и любая другая) в одном и том же файле в одном и том же месте разные данные. В этот момент возникает конфликт, который необходимо разрешить. Чтобы разрешить его, необходимо исправить возникший конфликт в файле и сделать коммит.