

Жданов Виктор
Кушнир Владислав
TI164

Поддукин Владимир

Теория

Git - это распределенная система контроля версий. Это главное отличие **git** от **svn**. Каждый разработчик создает на своем компьютере отдельный, полноценный **репозиторий**.

В рамках этого репозитория можно делать все тоже самое, что и обычно в **svn** - **создавать ветки, просматривать изменения, выполнять коммиты**. Для того, чтобы можно было работать с удаленными репозиториями и обмениваться изменениями с другими разработчиками, в **git** есть две команды, не имеющие аналогов в **svn** - **git push** и **git pull**.

git push - вливание локальных изменений в удаленный репозиторий. **git pull** - вливание изменений из удаленного репозитория в локальный. Обмен данными обычно происходит с использованием протокола **SSH**.

Git поддерживают несколько крупных репозиториев - **GitHub, SourceForge, BitBucket** и **Google Code**. Удобно использовать один из них в качестве основного хранилища для корпоративных проектов.

Как выписать репозиторий с github

Создаем новую директорию для проекта `project_name`, переходим в нее.

Выполняем команду:

```
git clone git@github.com:devlabuser/sharp.git ./
```

"./" означает, что создать репозиторий нужно в текущей директории.

Результат: каталог с выписанной веткой `master`. Теперь можно создавать новые ветки, или выписывать с **github** существующие.

Как выписать ветку с github

С помощью команды "**checkout**" можно выписать уже существующую ветку с **github**:

```
$ git checkout -b dev origin/dev
```

```
$ git checkout -b project_branch origin/project_branch
```

Как создать новую ветку в локальном репозитории:

Создаем новую ветку в локальном репозитории:

```
$ git checkout -b dev
```

```
Switched to a new branch 'dev'
```

Публикуем ее на github:

```
$ git push origin dev
```

```
Total 0 (delta 0), reused 0 (delta 0)
```

```
To git@github.com:devlabuser/sharp.git
```

```
* [new branch]    dev -> dev
```

Как переключиться на другую ветку в git:

```
$ git checkout project2_branch
```

Как посмотреть список веток:

Команда "**branch**" позволяет посмотреть список веток в локальном репозитории. Текущая ветка будет помечена звездочкой:

```
$ git branch
```

```
* dev
```

```
master
```

Как сделать commit:

Создаем новую ветку, выполняем в ней нужные изменения.

Подготавливаем коммит, добавляя в него файлы командой:

```
$ git add <file1> <file2> ...
```

Выполняем коммит:

```
$ git commit -m 'Комментарий к коммиту'
```

Как правило, в репозитории существует две основные ветки - dev и master. Dev - общая ветка разработчиков и тестировщиков. Именно в нее добавляются все новые разработки перед очередным релизом. Master - ветка для выкладки продукта на боевые сервера.

После коммита надо влить в нашу ветку изменения из ветки dev и master:

```
$ git pull origin dev
```

```
$ git pull origin master
```

Теперь наша ветка содержит изменения для проекта, и все последние изменения по другим задачам, которые успела внести команда.

Переключаемся на ветку dev:

```
$ git checkout dev
```

Вливаем в dev изменения из ветки проекта:

```
$ git merge project_branch
```

Заливаем последнюю версию ветки dev на удаленный сервер:

```
$ git push origin dev
Counting objects: 4, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 286 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:devlab/sharp.git
d528335..9a452d9 dev -> dev
```

Как выполнить слияние с другой веткой:

git merge выполняет слияние текущей и указанной ветки. Изменения добавляются в текущую ветку.

```
$ git merge origin/ticket_1001_branch
```

git pull забирает изменения из ветки на удаленном сервере и проводит слияние с активной веткой.

```
$ git pull origin ticket_1001_branch
```

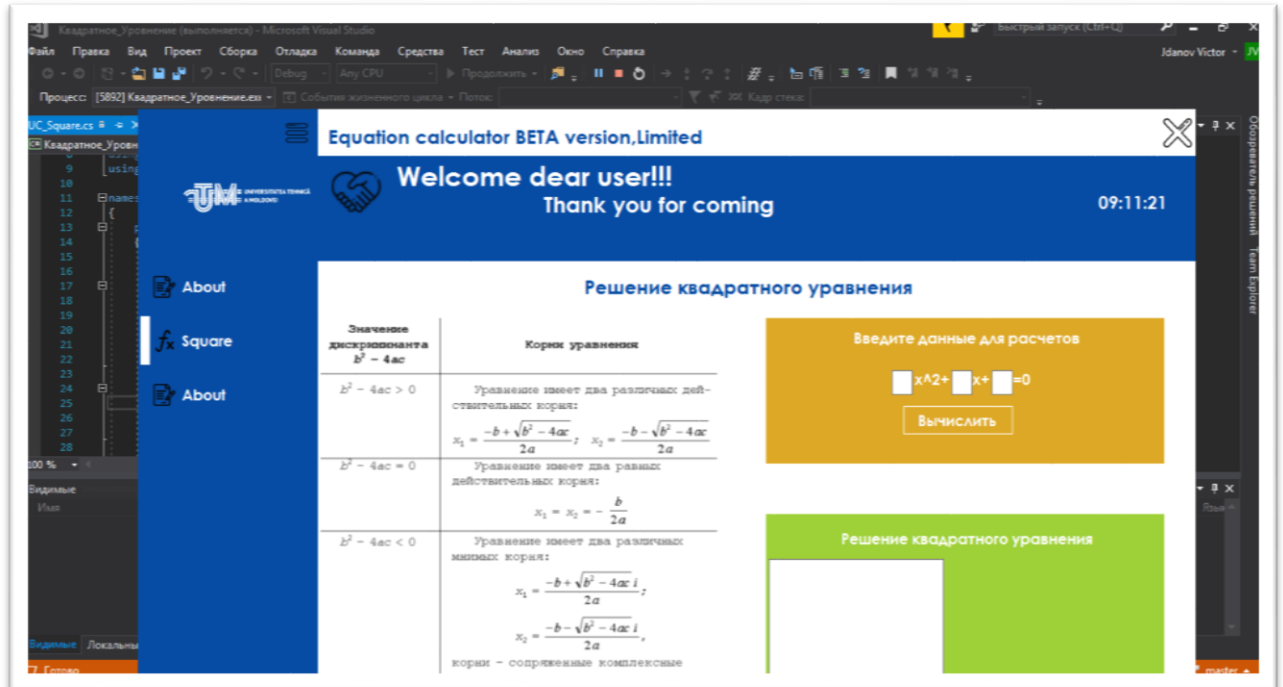
git pull отличается от **git merge** тем, что **merge** только выполняет слияние веток, а **pull** прежде чем выполнить слияние - закатывает изменения с удаленного сервера. **merge** удобно использовать для слияния веток в локальном репозитории, **pull** - слияния веток, когда одна из них лежит на **github**.

Создание нового локального репозитория

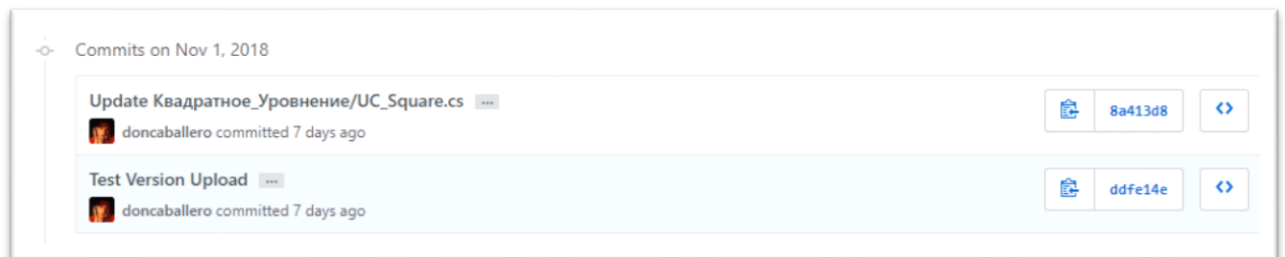
```
$ mkdir project_dir
$ cd project_dir
$ git init
```

Реализация

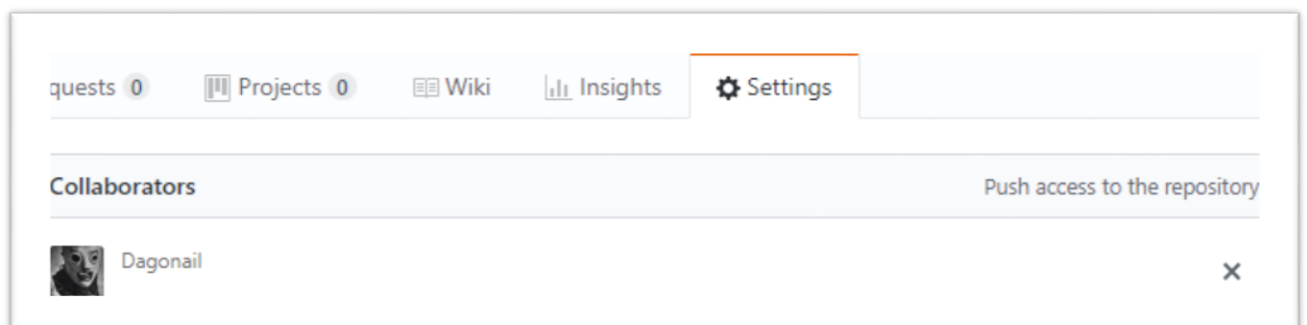
- 1) Создаём репозиторий на GitHub
https://github.com/NoManEXE/AlliedTesting_squareequation
- 2) Создание программы в **Visual Studio** для решения квадратного уравнения



- 3) **Git add ***
- 4) **Git commit -m "Test Version Upload"**
- 5) **Git push** – заливка первой версии программы в ветку **master**



- 6) Добавление в репозиторий другого разработчика



7) **git brench dev** – создание новой ветки для нового разработчика

8) **git checkout dev** – переход на новую ветку

9) Добавление изменений в ветку **dev**

```
C:\Users\User\Desktop\git_test\AlliedTesting_squareequation (dev -> origin)
λ git status
On branch dev
Your branch is up to date with 'origin/dev'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)


    readme.txt


nothing added to commit but untracked files present (use "git add" to track)
C:\Users\User\Desktop\git_test\AlliedTesting_squareequation (dev -> origin)
λ git add readme.txt
C:\Users\User\Desktop\git_test\AlliedTesting_squareequation (dev -> origin)
λ git commit -m "added the readme file with useful information for users"
[dev cab658e] added the readme file with useful information for users
1 file changed, 1 insertion(+)
create mode 100644 readme.txt
C:\Users\User\Desktop\git_test\AlliedTesting_squareequation (dev -> origin)
λ git push
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 339 bytes | 339.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/NoManEXE/AlliedTesting_squareequation.git
   8a413d8..cab658e  dev -> dev
```

9) **git merge dev** – слияние ветки **dev** и **master**

```
C:\Users\User\Desktop\git_test\AlliedTesting_squareequation (dev -> origin)
λ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
C:\Users\User\Desktop\git_test\AlliedTesting_squareequation (master -> origin)
λ git merge dev
Updating 8a413d8..cab658e
Fast-forward
 readme.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 readme.txt
C:\Users\User\Desktop\git_test\AlliedTesting_squareequation (master -> origin)
λ git pull
Already up to date.
```

Your recently pushed branches:

 dev (8 minutes ago)

 Compare & pull request

Default branch

master Updated 7 days ago by doncaballero


Default

Change default branch

Your branches

dev Updated 34 seconds ago by NoManEXE

0 | 1


 New pull request



Active branches

dev Updated 34 seconds ago by NoManEXE

0 | 1

 New pull request



