



# Git και GitHub στην πράξη

**Αθ. Ανδρούτσος**



# Σενάριο 1

- Σε ένα project έχουμε δύο ομάδες: 1) η ομάδα του backend, που αναπτύσσει το project, και, 2) η ομάδα των clients, που χρησιμοποιούν τις υπηρεσίες που προσφέρει το project
- Η ομάδα του backend δουλεύει με σταδιακές προσαυξήσεις (increments) όπου σε κάθε προσαύξηση δημιουργεί νέο branch το οποίο και κάνει push (pushάρει) στο GitHub
- Οι clients λαμβάνουν αρχικά το project με clone, στη συνέχεια κάνουν pull σε κάθε increment και στη συνέχεια κάνουν checkout στο κάθε νέο branch
- Και μετά the magic happens! Με κάθε checkout αλλάζουν τα περιεχόμενα του active directory και παίρνουμε την υπηρεσία που θέλουμε σε όποιο branch κάνουμε checkout



# Project (1)

Version Control με Git / GitHub

- Ξεκινάει η ομάδα του backend και δημιουργεί το νέο project
- Έστω το όνομα του project είναι GitProj1

The screenshot shows the 'New Project' dialog in IntelliJ IDEA. The 'Name' field is 'GitProj1'. The 'Location' is '~\IdeaProjects\idea1023'. The 'Project will be created in' path is '~\IdeaProjects\idea1023\GitProj1'. The 'Language' is 'Java'. The 'Build system' is 'IntelliJ'. The 'JDK' is '11 Amazon Corretto version 11.0.10'. The 'Add sample code' checkbox is unchecked. The 'Generate code with onboarding tips' checkbox is unchecked. The 'Advanced Settings' section is expanded, showing 'Module name' as 'GitProj1', 'Content root' as '~\IdeaProjects\idea1023\GitProj1', and 'Module file location' as '~\IdeaProjects\idea1023\GitProj1'. The 'Create' button is highlighted in blue.



# Project (2)

Version Control με Git / GitHub

The screenshot shows an IDE window titled "GitProj1 - Main.java". The left sidebar displays the project structure: "GitProj1" (C:\Users\A8ana\IdeaProjects\idea1023\G) with subfolders ".idea", "src", and "gr.aueb.cf". The "Main" file is selected under "gr.aueb.cf". The main editor area shows the following Java code:

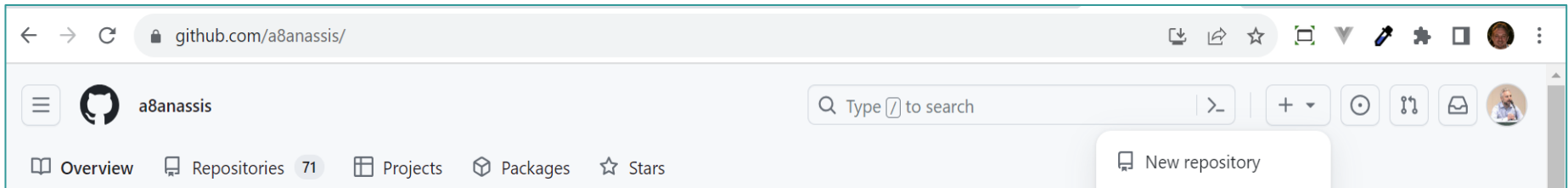
```
1 package gr.aueb.cf;  
2  
3 public class Main {  
4  
5     public static void main(String[] args) {  
6         System.out.println("Hello Git");  
7     }  
8 }  
9
```

- Το project είναι απλό και περιέχει μία κλάση Main και την μέθοδο main, αλλά θα μπορούσε να είναι και πολύ μεγάλο και σύνθετο project



# New Repository – GitHub (1)

Version Control με Git / GitHub



**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

*Required fields are marked with an asterisk (\*).*

Owner \* a8anassis / Repository name \* git-proj1

✓ git-proj1 is available.

Great repository names are short and memorable. Need inspiration? How about [cuddly-octo-lamp](#) ?

Description (optional)

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

- Στο GitHub δημιουργούμε νέο Repository
- Έστω git-proj1
- Σε πραγματικό case το repository θα ήταν private, και θα ορίζαμε collaborators



# New Repository – GitHub (2)

Version Control με Git / GitHub

Initialize this repository with:

☒ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: Java ▼

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▼

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

You are creating a public repository in your personal account.

Create repository

- Προαιρετικά μπορούμε να ορίσουμε και το *.gitignore* αλλά αυτό είναι κάτι που μπορούμε να ορίσουμε και αργότερα



# Collaborators

Version Control με Git / GitHub

- Σε πραγματικό use-case θα επιλέγαμε *Settings* και *Add people* για να προσθέσουμε Collaborators στο private repository

github.com/a8anassis/git-proj1/settings/access

a8anassis / git-proj1

Code Issues Pull requests Actions Projects Wiki Security Insights **Settings**

General

Access

**Collaborators**

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

Manage

DIRECT ACCESS

0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

Add a collaborator to git-proj1

Sofoklis1

Sofoklis  
Sofoklis1 • Invite collaborator

You haven't invited any collaborators yet

**Add people**



# GitHub .git

Version Control με Git / GitHub

The screenshot displays the GitHub interface for a repository named 'git-proj1' owned by 'a8anassis'. The repository is public and has 1 branch and 0 tags. The 'Code' button is highlighted, and a dropdown menu is open showing options to clone the repository using HTTPS, SSH, or GitHub CLI. The SSH URL 'git@github.com:a8anassis/git-proj1.git' is selected and highlighted. The 'About' section on the right shows the repository is a 'Hello World Project' with 0 stars, 1 watching, and 0 forks. The 'Releases' section shows no releases published with a link to 'Create a new release'.

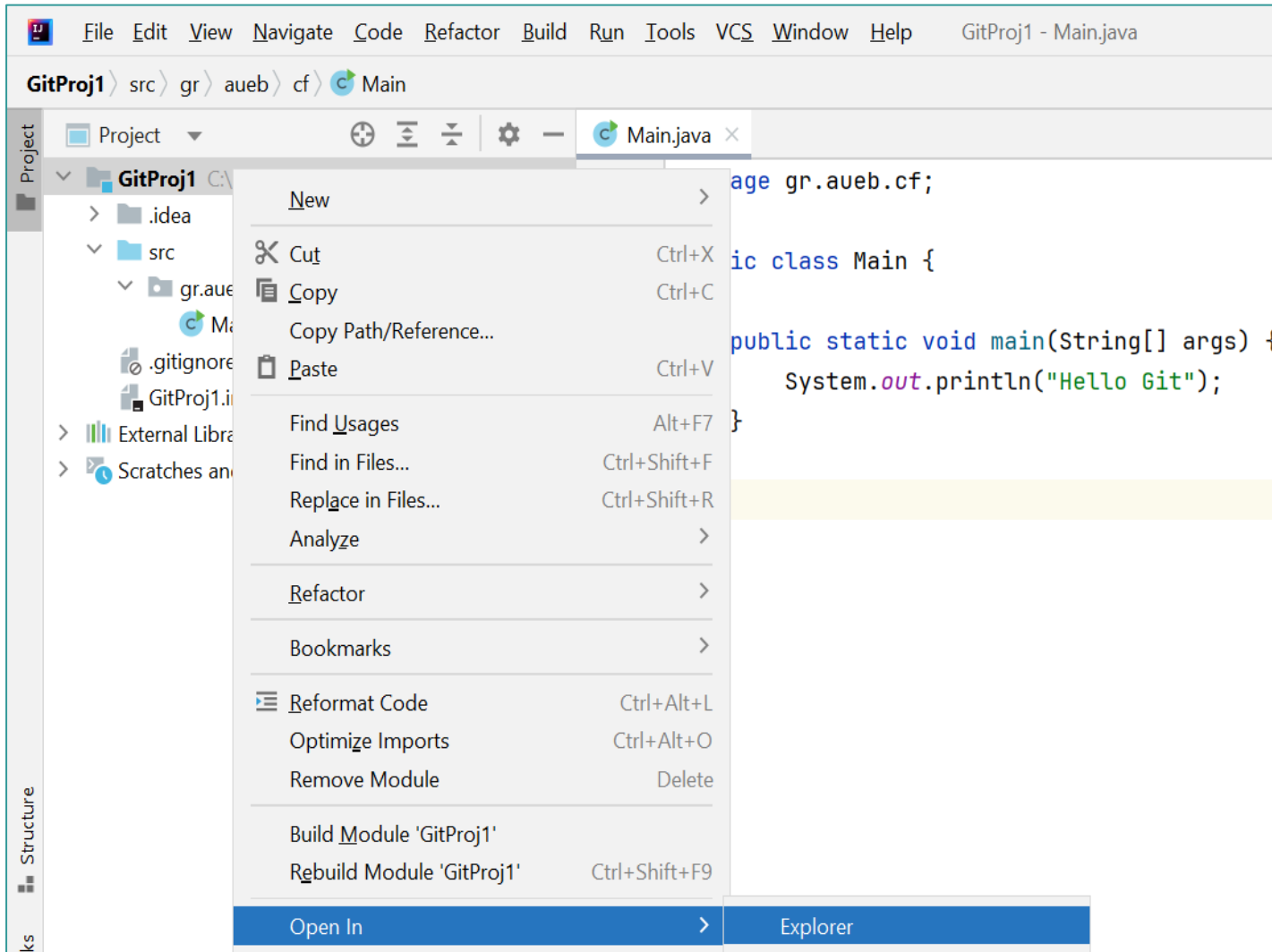
- Επιλέγουμε το Repository URL (remote repository) ώστε στη συνέχεια να το συνδέσουμε με το local repository





# Project folder/dir (1)

Version Control με Git / GitHub

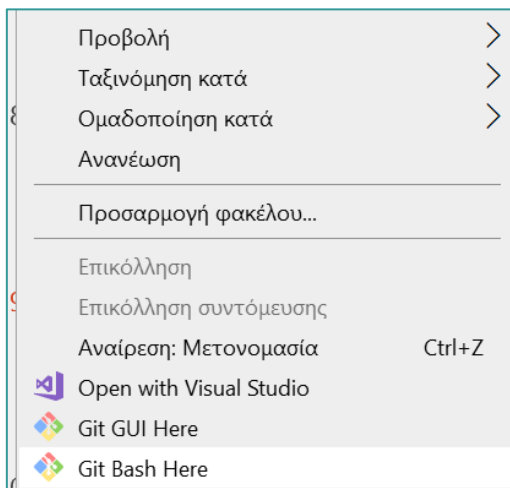
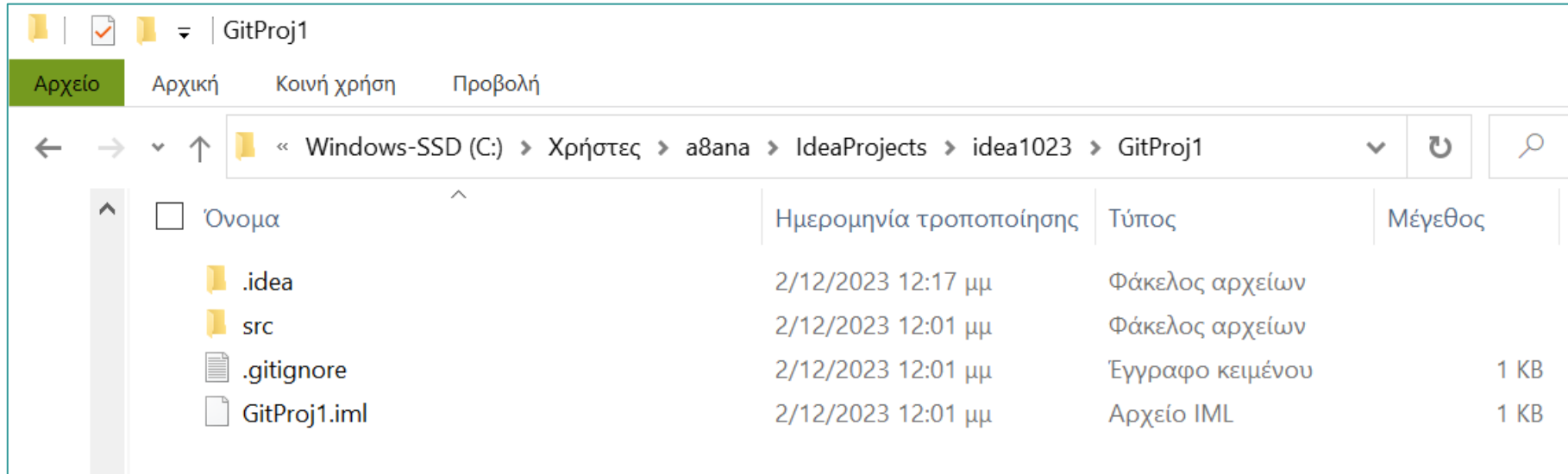


- Ανοίγουμε τον project folder ώστε να ανοίξουμε ένα Git Bash στον folder



# Project folder/dir (2)

Version Control με Git / GitHub



- Ή θα μπορούσαμε και κατευθείαν από το Git Bash ή από το ΛΣ (MacOs, Linux)



# Git Bash - IntelliJ

Version Control με Git / GitHub

```
Terminal: Local x Git Bash x + v
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (hello)
$ git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
          [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
          [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
          [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
          [--super-prefix=<path>] [--config-env=<name>=<envvar>]
          <command> [<args>]
```

- Επίσης, μπορούμε να ανοίξουμε το Git Bash στον folder του project και από το terminal του IntelliJ κατευθείαν, χωρίς να κάνουμε την προηγούμενη διαδικασία



# .gitignore

Version Control με Git / GitHub

```
File Edit View Navigate Code Refactor Build Run Tools Git Window Help GitProj1 - Main.java

GitProj1 > src > gr > aueb > cf > Main

Project
  GitProj1 C:\Users\aa8ana\IdeaProjects\idea1023\G
    .idea
    out
    src
    GitProj1.iml
  External Libraries
  Scratches and Consoles

Commit

Pull Requests

1 package gr.aueb.cf;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         System.out.println("Hello Git");
7     }
8 }
```

- Μιας και το GitHub repo έχει .gitignore, διαγράφουμε το .gitignore του IntelliJ



# Git Bash – Git init

Version Control με Git / GitHub

```
MINGW64:/c/Users/a8ana/IdeaProjects/idea1023/GitProj1
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1
$ git init
Initialized empty Git repository in C:/Users/a8ana/IdeaProjects/idea1023/GitProj1/.git/
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (main)
$ git remote add origin git@github.com:a8anassis/git-proj1.git
```

- Από το Git Bash, στον project folder (και όχι σε κάποιον υποφάκελο ή έξω από το project) κάνουμε **git init** ώστε να μετατρέψουμε το project folder σε git folder
- Επίσης, συνδέουμε remote repo / local repo με **git remote add** αφού πάρουμε copy-paste το GitHub repo URL



# Git pull origin main

Version Control με Git / GitHub

```
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (main)
$ git pull origin main
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 861 bytes | 50.00 KiB/s, done.
From github.com:a8anassis/git-proj1
 * branch                main          -> FETCH_HEAD
 * [new branch]          main          -> origin/main
```

- Με `git pull origin main` η ομάδα του backend παίρνει ότι έχει το remote repo (readme.md και .gitignore) και τα κάνει fetch και merge



# Περιεχόμενα του main branch

Version Control με Git / GitHub

```
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (main)
$ ls -la
total 19
drwxr-xr-x 1 a8ana 197609  0 Dec  2 12:40 ./
drwxr-xr-x 1 a8ana 197609  0 Dec  2 12:01 ../
drwxr-xr-x 1 a8ana 197609  0 Dec  2 12:40 .git/
-rw-r--r-- 1 a8ana 197609 314 Dec  2 12:40 .gitignore
drwxr-xr-x 1 a8ana 197609  0 Dec  2 12:30 .idea/
-rw-r--r-- 1 a8ana 197609 433 Dec  2 12:01 GitProj1.iml
-rw-r--r-- 1 a8ana 197609  34 Dec  2 12:40 README.md
drwxr-xr-x 1 a8ana 197609  0 Dec  2 12:36 out/
drwxr-xr-x 1 a8ana 197609  0 Dec  2 12:01 src/
```

- Με ls-la βλέπουμε τα αναλυτικά περιεχόμενα του main branch



# .gitignore

```
MINGW64:/c/Users/a8ana/IdeaProjects/idea1023/GitProj1
out/
.idea/
GitProj1.iml
```

- Κάνουμε edit το .gitignore (με vi .gitignore) και εισάγουμε τα παραπάνω
- **out/** κάνει ignore τον φάκελο out (ότι τελειώνει με / θεωρείται φάκελος στο .gitignore )
- **.idea/** κάνει ignore τον φάκελο .idea
- **GitProj1.iml** κάνει ignore το αρχείο GitProj.iml (ότι δεν τελειώνει σε / στο .gitignore θεωρείται αρχείο)





# Backend main – Staging

Version Control με Git / GitHub

```
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (main)
$ git add .

a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   .gitignore
        new file:   src/gr/aueb/cf/Main.java

a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (main)
$ |
```

- Εισάγουμε τα νέα αρχεία του branch στο staging area (με **git add .**) ώστε στο επόμενο βήμα να τα κάνουμε commit. Με **git status** επιβεβαιώνουμε
- Αν θέλαμε να κάνουμε unstage, μπορούμε με **git restore --cached** . ή **git restore --staged** . ( . Σημαίνει unstage όλα τα αρχεία του active directory και των subdirectories )



# Commit and push

Version Control με Git / GitHub

```
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (main)
$ git commit -m "Main class initial"
[main ca3515e] Main class initial
2 files changed, 11 insertions(+), 23 deletions(-)
create mode 100644 src/gr/aueb/cf/Main.java
```

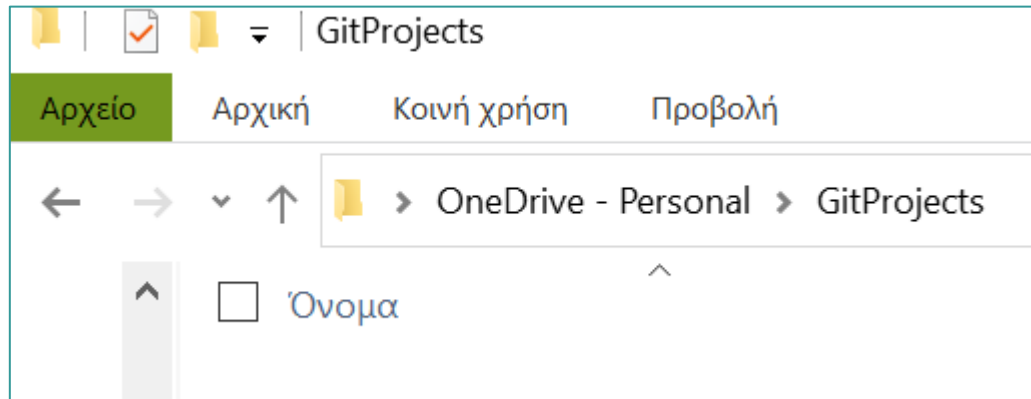
```
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (main)
$ git push -u origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (8/8), 620 bytes | 310.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:a8anassis/git-proj1.git
   3d3200c..ca3515e  main -> main
branch 'main' set up to track 'origin/main'.
```

- To back-end κάνει commit και push



# Clients (collaborators) clone

Version Control με Git / GitHub



```
a8ana@thanassis-pc MINGW64 ~/OneDrive/GitProjects
$ git clone git@github.com:a8anassis/git-proj1.git
```

- Οι collaborators δημιουργούν ένα folder GitProjects και μέσα σε αυτό τον folder κάνουν clone το repository από τον GitHub (το όνομα του folder θα είναι git-proj1)
- Θα μπορούσαν να επιλέξουν διαφορετικό όνομα για το local folder με ***git clone*** [git@github.com:a8anassis/git-proj1.git](https://github.com/a8anassis/git-proj1.git) ***new-folder-name***



# Cloned dir

```
a8ana@thanassis-pc MINGW64 ~/OneDrive/GitProjects
$ ls
git-proj1/

a8ana@thanassis-pc MINGW64 ~/OneDrive/GitProjects
$ cd git-proj1/

a8ana@thanassis-pc MINGW64 ~/OneDrive/GitProjects/git-proj1 (main)
$ ls -la
total 10
drwxr-xr-x 1 a8ana 197609  0 Dec  2 13:24 ./
drwxr-xr-x 1 a8ana 197609  0 Dec  2 13:24 ../
drwxr-xr-x 1 a8ana 197609  0 Dec  2 13:24 .git/
-rw-r--r-- 1 a8ana 197609 30 Dec  2 13:24 .gitignore
-rw-r--r-- 1 a8ana 197609 34 Dec  2 13:24 README.md
drwxr-xr-x 1 a8ana 197609  0 Dec  2 13:24 src/
```

- Δημιουργήθηκε το git-proj1 folder και πρέπει να κάνουμε cd git-proj1 για να μπούμε μέσα στον folder. Με ls-la στη συνέχεια βλέπουμε τα περιεχόμενα του folder



# Collaborators run the project

Version Control με Git / GitHub

```
a8ana@thanassis-pc MINGW64 ~/OneDrive/GitProjects/git-proj1/src/gr/aueb/cf (main)
$ cat Main.java
package gr.aueb.cf;

public class Main {

    public static void main(String[] args) {
        System.out.println("Hello Git");
    }
}

a8ana@thanassis-pc MINGW64 ~/OneDrive/GitProjects/git-proj1/src/gr/aueb/cf (main)
$ java Main.java
Hello Git
```

- Με ***cat main.java*** βλέπουμε τα περιεχόμενα του Main.java και με ***java Main.java*** εκτελούμε (execute) μιας και *java* είναι το runtime που κάνει build και run το project, δηλαδή εκτελεί την main method



# Back-end makes changes

Version Control με Git / GitHub

```
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (main)
$ git branch hello

a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (main)
$ git checkout hello
Switched to branch 'hello'

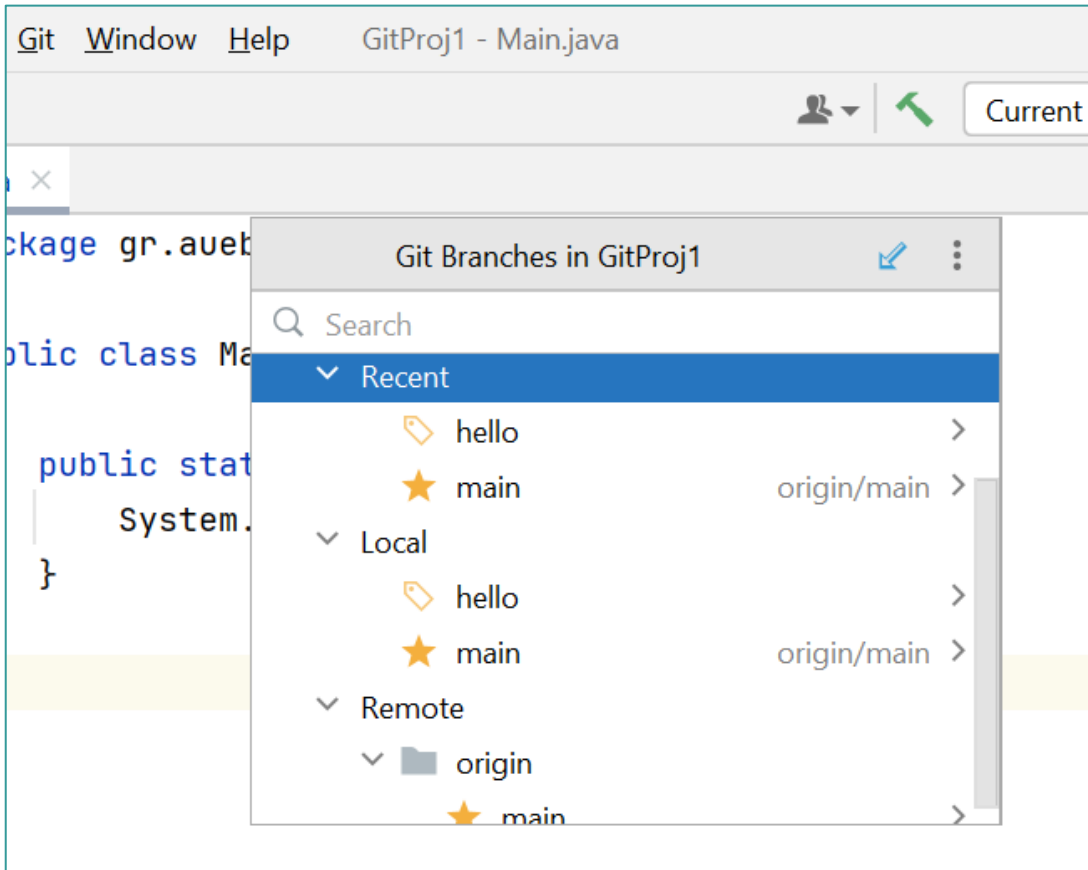
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (hello)
$
```

- Στην άλλη πλευρά η ομάδα του backend κάνει αλλαγές ή προσθέτει features στο project
- Τις αλλαγές/προσθήκες τις κάνει σε νέο branch, γιατί για αυτό υπάρχει το version management, για να έχουμε το history των versions του Project μας
- Αφού έχει γίνει release ένα version μετά τροποποιήσεις κάνουμε σε branches



# Git Branches

Version Control με Git / GitHub



- Και μέσα από το IntelliJ μπορούμε να κάνουμε checkout στο branch 'hello'
- Ή θα μπορούσαμε και από εδώ μέσα να δημιουργήσουμε νέο branch από το main branch με δεξί κλικ στο main



# Αλλαγές / Προσθήκες

Version Control με Git / GitHub

```
Main.java x
1 package gr.aueb.cf;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         System.out.println("Hello Coding Factory");
7     }
8 }
```

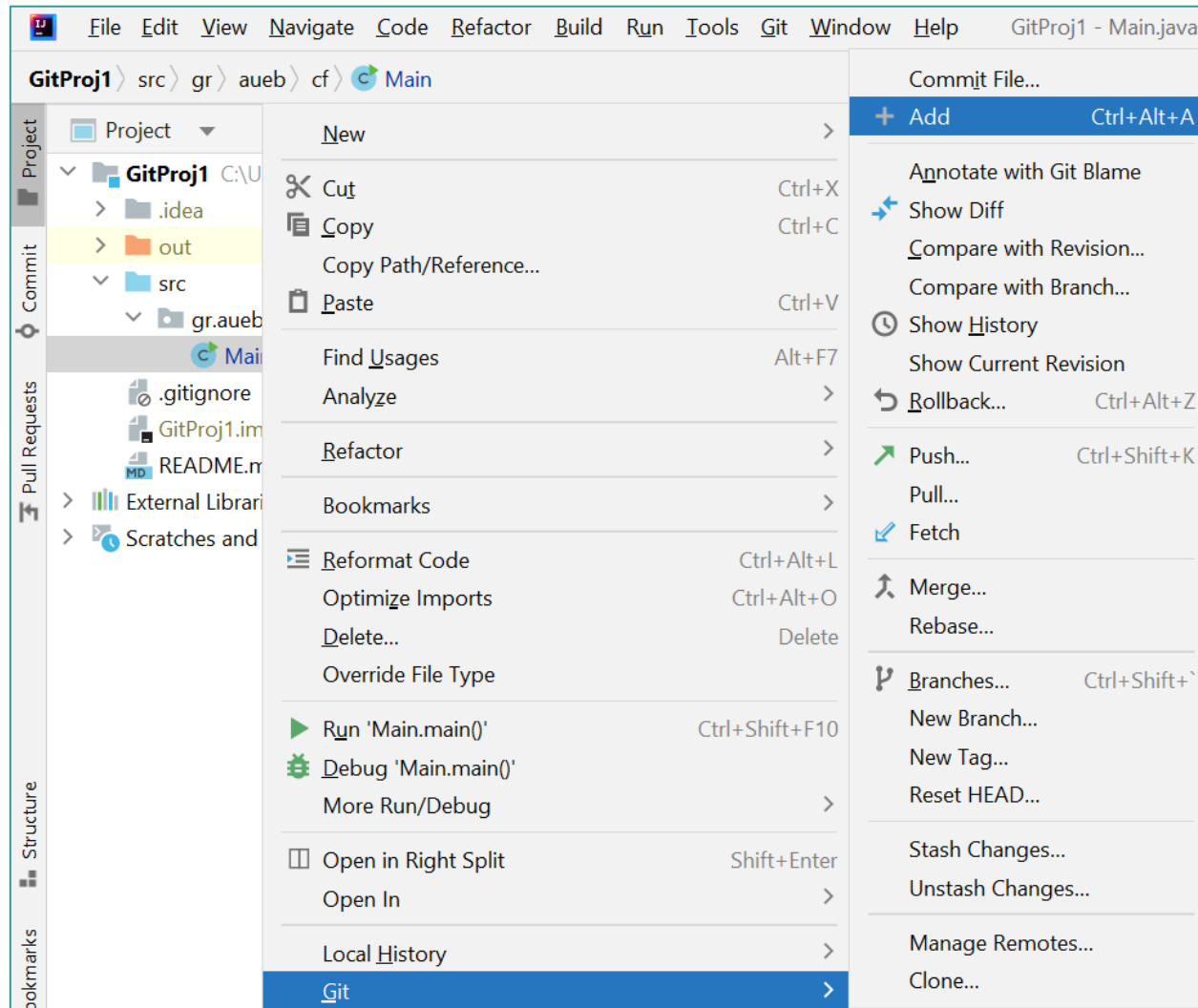
- Αλλάξαμε το project μας. Η αλλαγή είναι ότι άλλαξε η println και εμφανίζει κάτι άλλο (Hello Coding Factory)
- Γενικά, θα μπορούσαν να έχουν γίνει πολλές αλλαγές / προσθήκες





# Add, Commit, Push to GitHub

Version Control με Git / GitHub

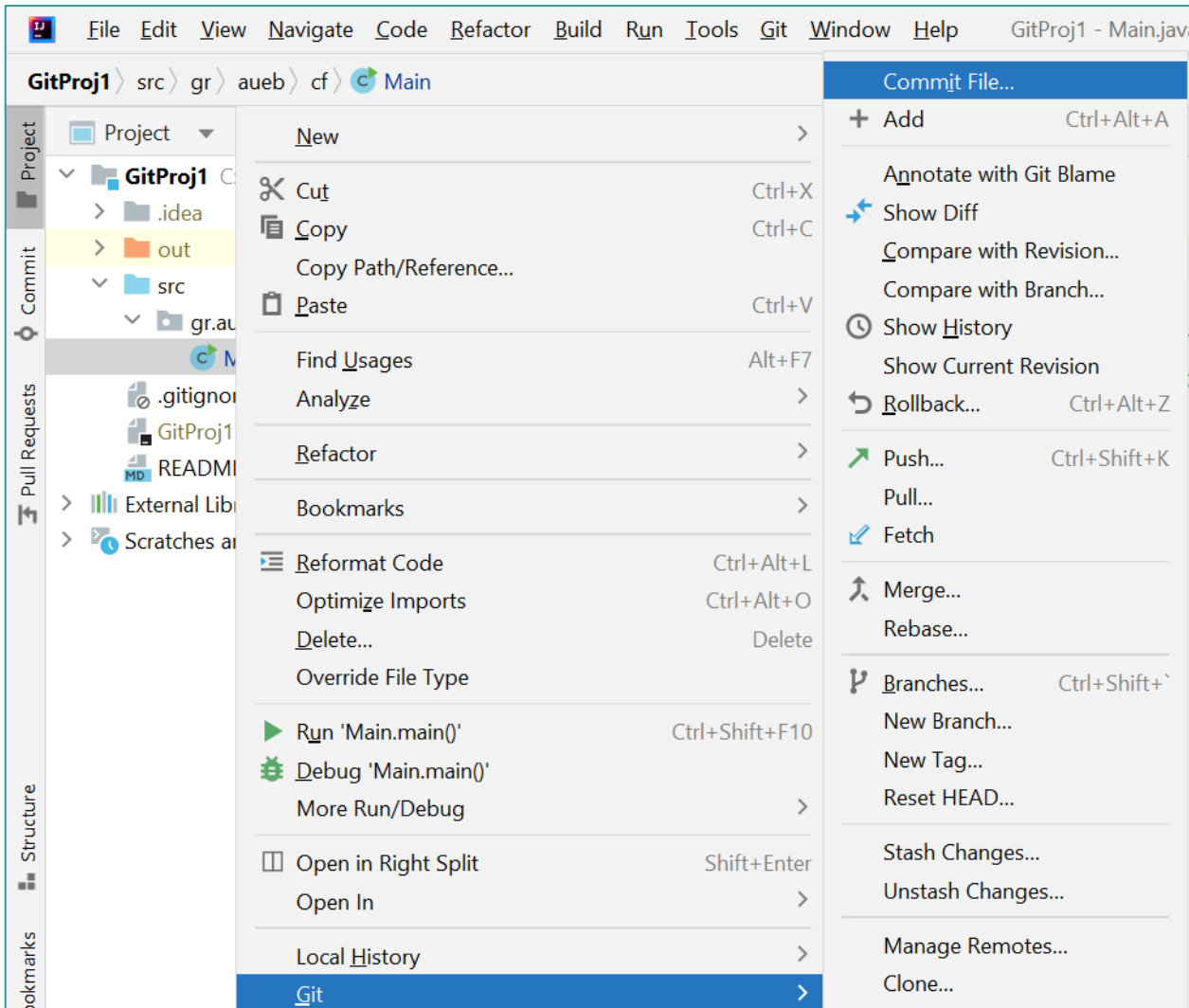


- Μπορεί να γίνει add και commit ή κατευθείαν commit και μετά push



# Commit (1)

Version Control με Git / GitHub

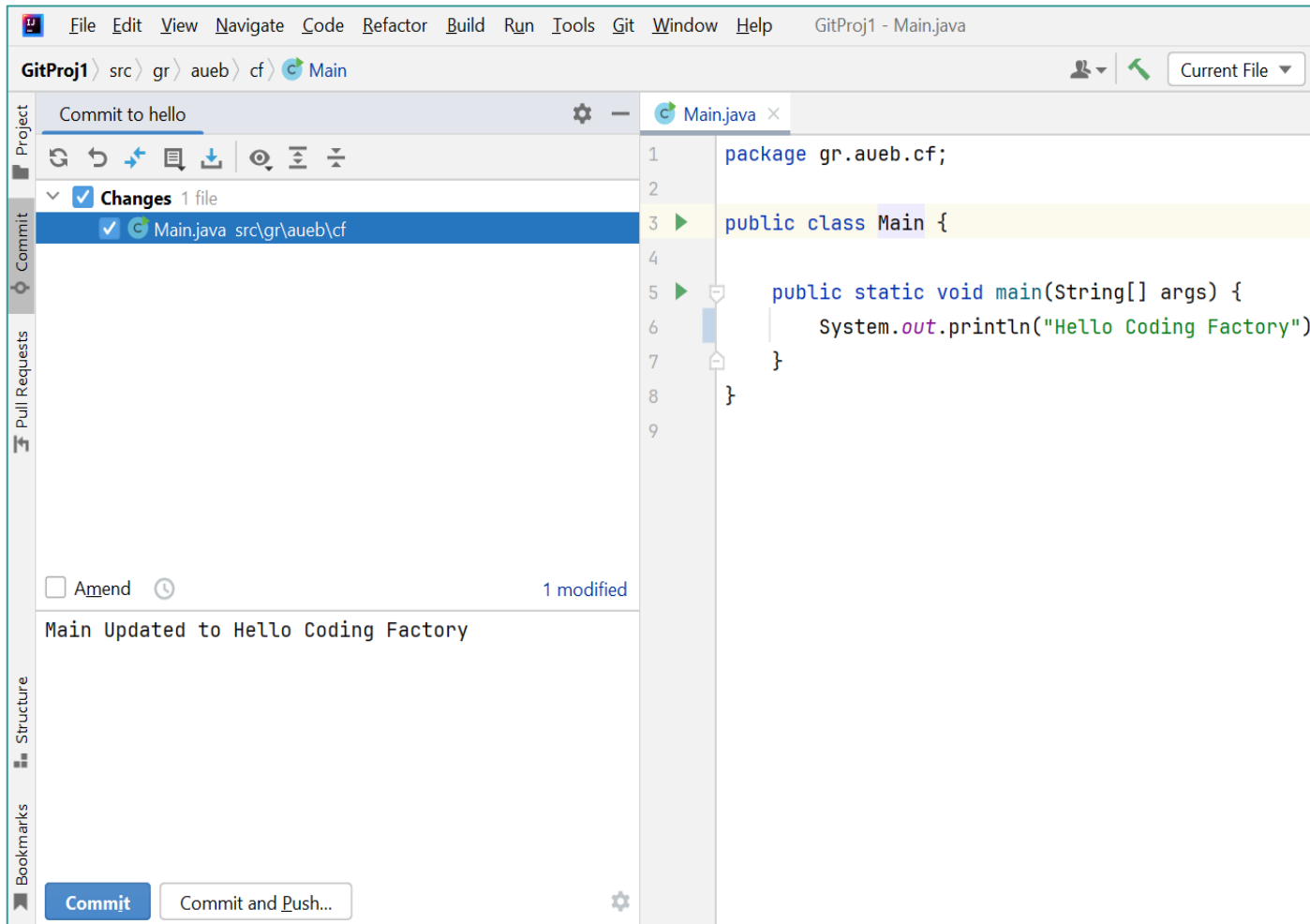


- Οι αλλαγές στην Main class πρέπει να γίνουν commit
- Μπορούμε μέσα από το IntelliJ να επιλέξουμε το αρχείο/αρχεία που θέλουμε να κάνουμε commit και να επιλέξουμε Commit File...



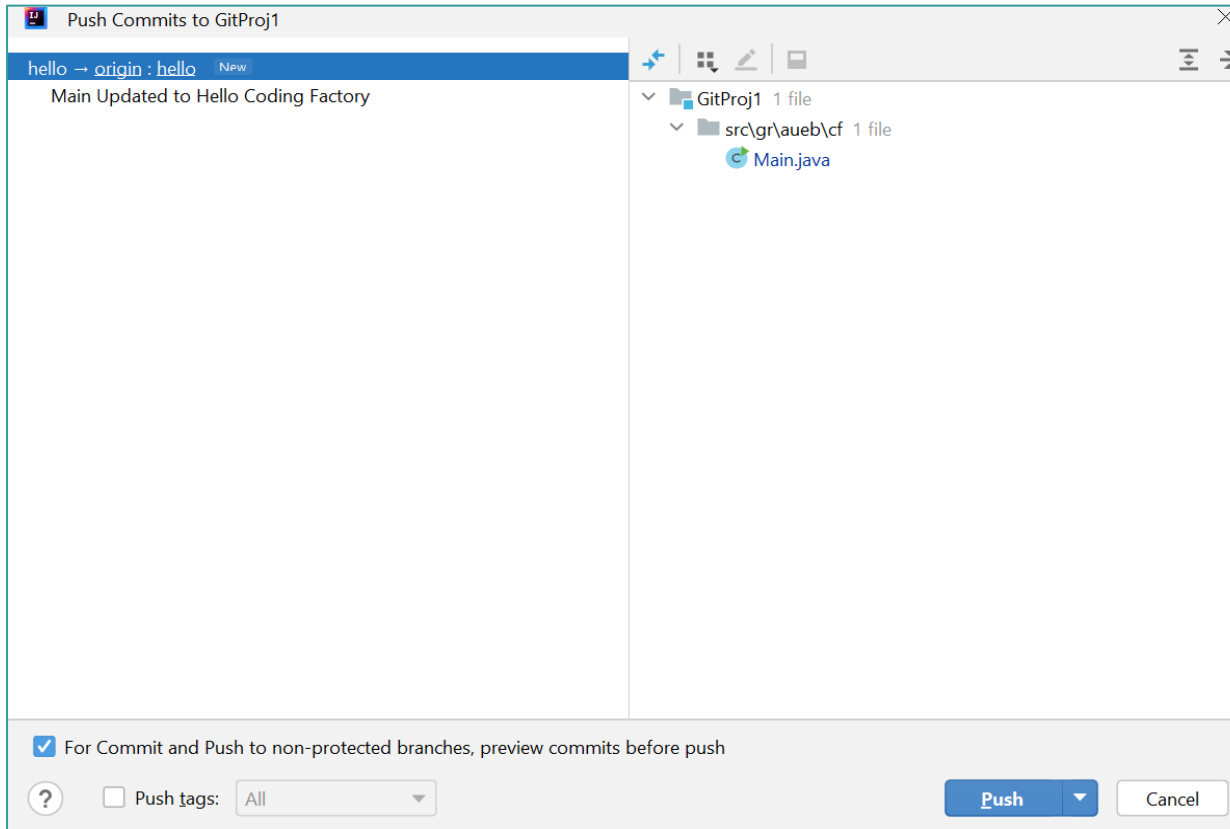
# Commit (2)

- Δίνουμε το μήνυμα για το Commit και κάνουμε **Commit και Push**





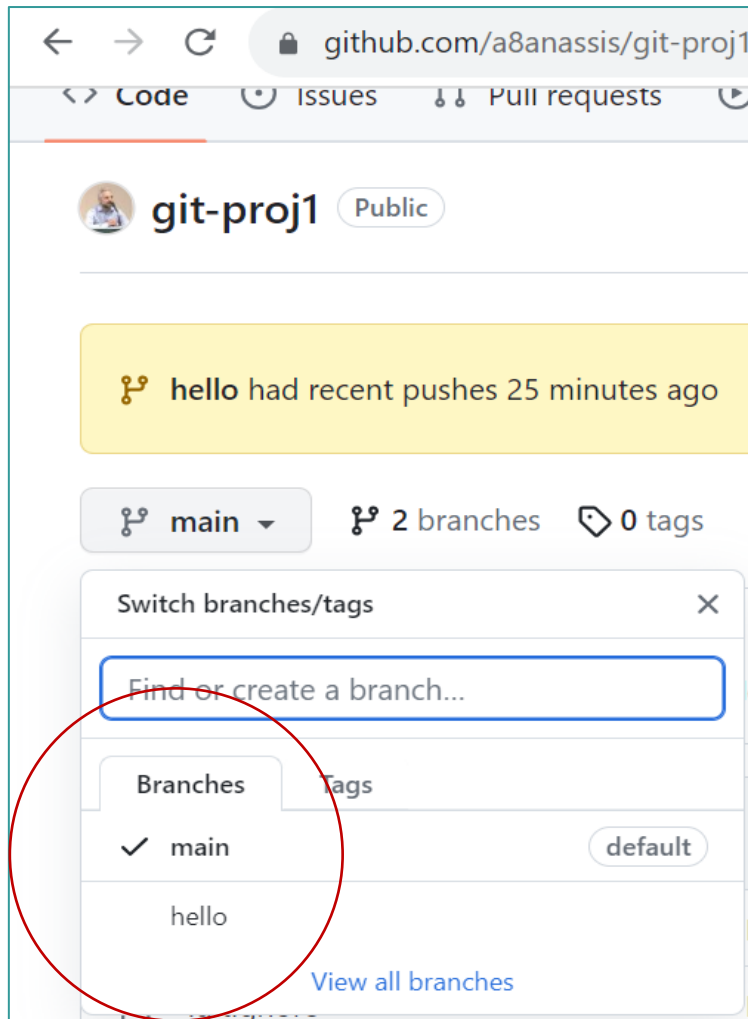
# Push



- Κάνουμε push στο hello branch
- Στο GitHub θα δημιουργηθεί το hello branch και τα περιεχόμενά του

**Pushed hello to new branch origin/hello**

**1 file committed: Main Updated to Hello Coding Factory**

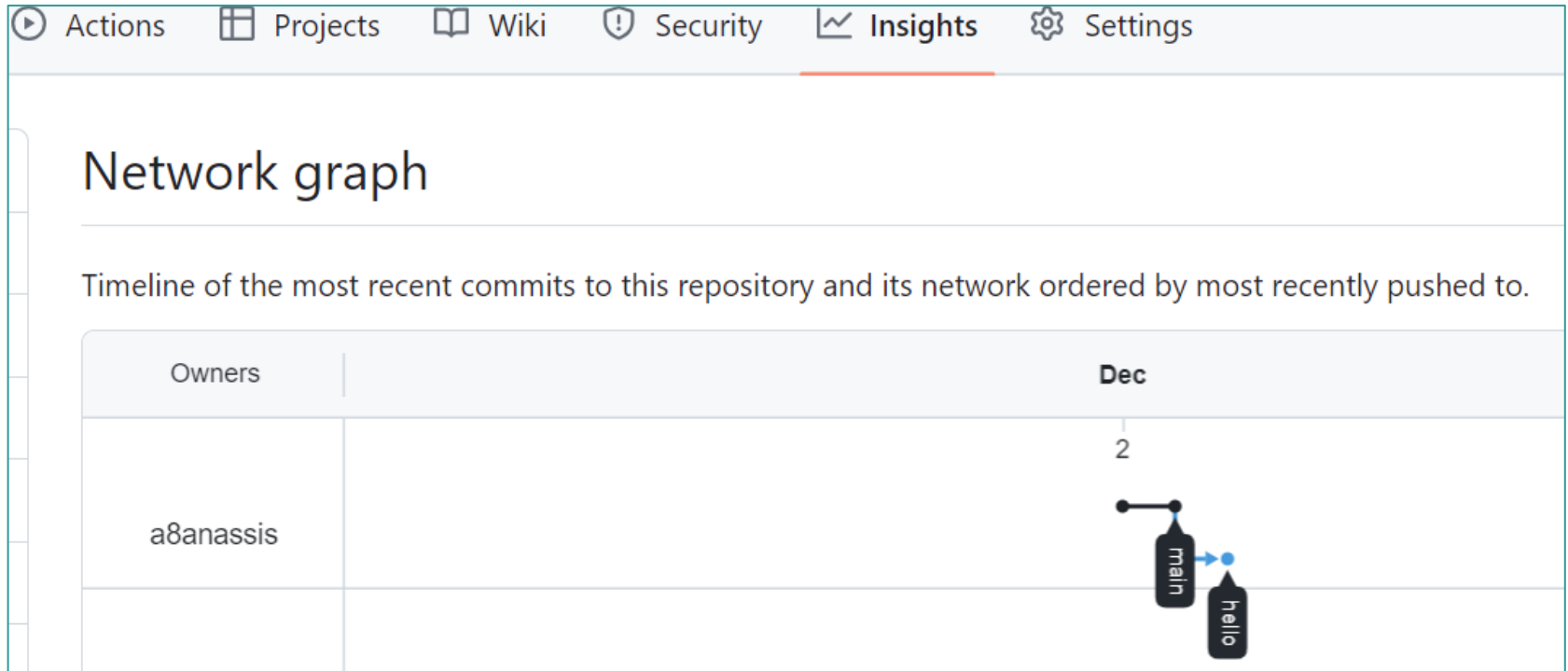


- Μετά το push, στο GitHub έχουμε το branch hello
- Και έχουμε και τις αλλαγές που έχουν γίνει σε αυτό το branch



# Network Graph

Version Control με Git / GitHub



- Στο Network Graph βλέπουμε τα commits. Το hello είναι ένα commit μπροστά από το main branch



# Collaborators pull & checkout

Version Control με Git / GitHub

```
a8ana@thanassis-pc MINGW64 ~/OneDrive/GitProjects/git-proj1/src
$ git pull
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 7 (delta 2), reused 7 (delta 2), pack-reused 0
Unpacking objects: 100% (7/7), 463 bytes | 15.00 KiB/s, done.
From github.com:a8anassis/git-proj1
 * [new branch]      hello      -> origin/hello
Already up to date.
```

```
a8ana@thanassis-pc MINGW64 ~/OneDrive/GitProjects/git-proj1/src/gr/aueb
/cf (main)
$ git checkout hello
Switched to a new branch 'hello'
branch 'hello' set up to track 'origin/hello'.
```

- Στο Git, όταν κάνουμε **git pull** χωρίς να προσδιορίσουμε branch, τότε φέρνει τις αλλαγές που έχουν γίνει στο current checked-out branch και, ως μέρος της διαδικασίας, φέρνει και όλα τα νέα branches
- Αυτά τα νέα branches αποθηκεύονται locally ως remote tracking branches. Όταν κάνουμε checkout στο νέο branch δημιουργείται το νέο branch και τοπικά



# And .. The Magic happens

Version Control με Git / GitHub

```
a8ana@thanassis-pc MINGW64 ~/OneDrive/GitProjects/git-proj1/src/gr/aueb/cf (hello)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

a8ana@thanassis-pc MINGW64 ~/OneDrive/GitProjects/git-proj1/src/gr/aueb/cf (main)
$ java Main.java
Hello git
```

```
a8ana@thanassis-pc MINGW64 ~/OneDrive/GitProjects/git-proj1/src/gr/aueb/cf (main)
$ git checkout hello
Switched to branch 'hello'
Your branch is up to date with 'origin/hello'.

a8ana@thanassis-pc MINGW64 ~/OneDrive/GitProjects/git-proj1/src/gr/aueb/cf (hello)
$ java Main.java
Hello Coding Factory
```

- Με **git checkout** αλλάζουν τα περιεχόμενα του dir/folder (αλλάζει το working dir)
- Παρατηρούμε ότι με **git checkout main**, στον ίδιο folder αλλάζουν τα περιεχόμενα του folder και το java Main.java δίνει το αντίστοιχο και διαφορετικό output





## 2ο Σενάριο (1)

Version Control με Git / GitHub

- Στο 2<sup>ο</sup> σενάριο θεωρούμε ότι η ομάδα του backend έχει δύο ή περισσότερα μέλη που συνεργάζονται
- Κάθε μέλος αναλαμβάνει ένα task, ενώ ο manager κάνει merge τις αλλαγές στο main branch και μετά τις κάνει commit και push



## 2ο Σενάριο (2)

Version Control με Git / GitHub

- Υποθέτουμε ότι δουλεύουμε στο `hello branch` και κάνουμε `merge` σε αυτό. Στο τέλος αν όλα είναι καλά, θα κάνουμε `merge` στο `main branch`
- Θα δημιουργήσουμε ένα branch ***feature-hello-method***, που θα προσθέτει μία μέθοδο, που θα εμφανίζει το `hello` και θα το αναλάβει ένα μέλος της ομάδας και ένα branch ***hotfix-doc-comments*** που θα εισάγει `doc comments` στην κλάση
- Το *feature-hello-branch* και το *hotfix-doc-comments* θα γίνουν branch μέσα από το `hello branch`



# Νέο feature branch

Version Control με Git / GitHub

```
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (hello)
$ git branch feature-hello-method
```

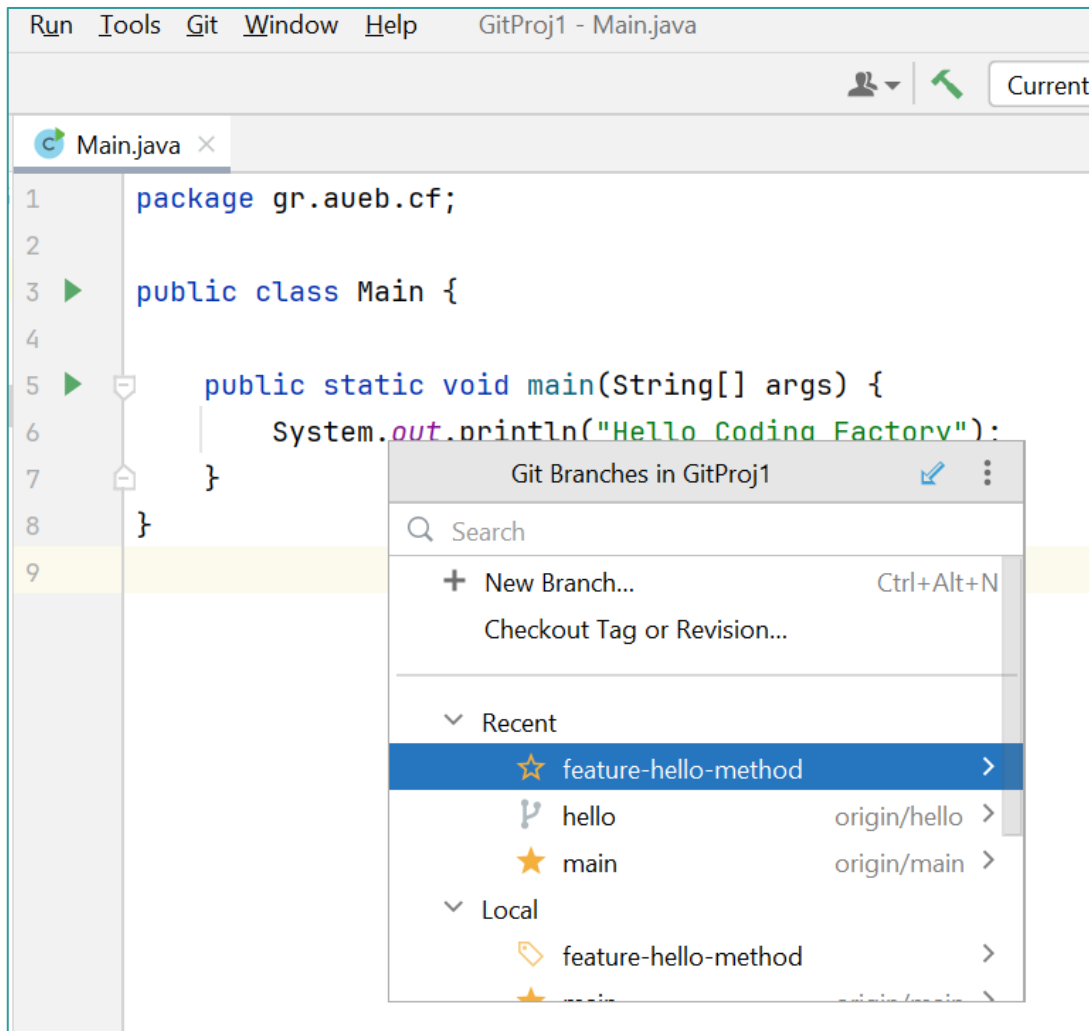
```
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (hello)
$ git checkout feature-hello-method
switched to branch 'feature-hello-method'
```

- Το μέλος της ομάδας George αναλαμβάνει αυτό το feature. Δημιουργεί ένα νέο branch feature-hello-method και θα δουλέψει σε αυτό το branch



# IntelliJ νέο feature branch

Version Control με Git / GitHub

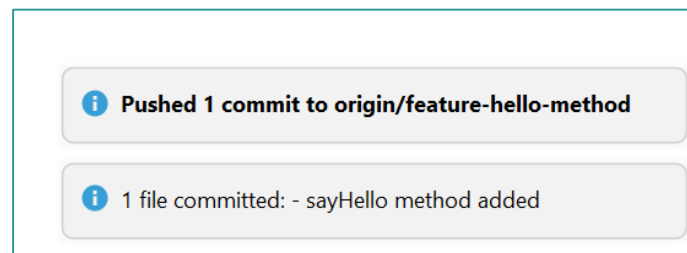
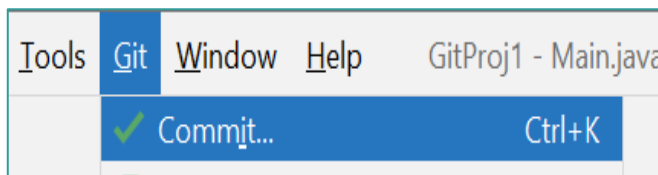
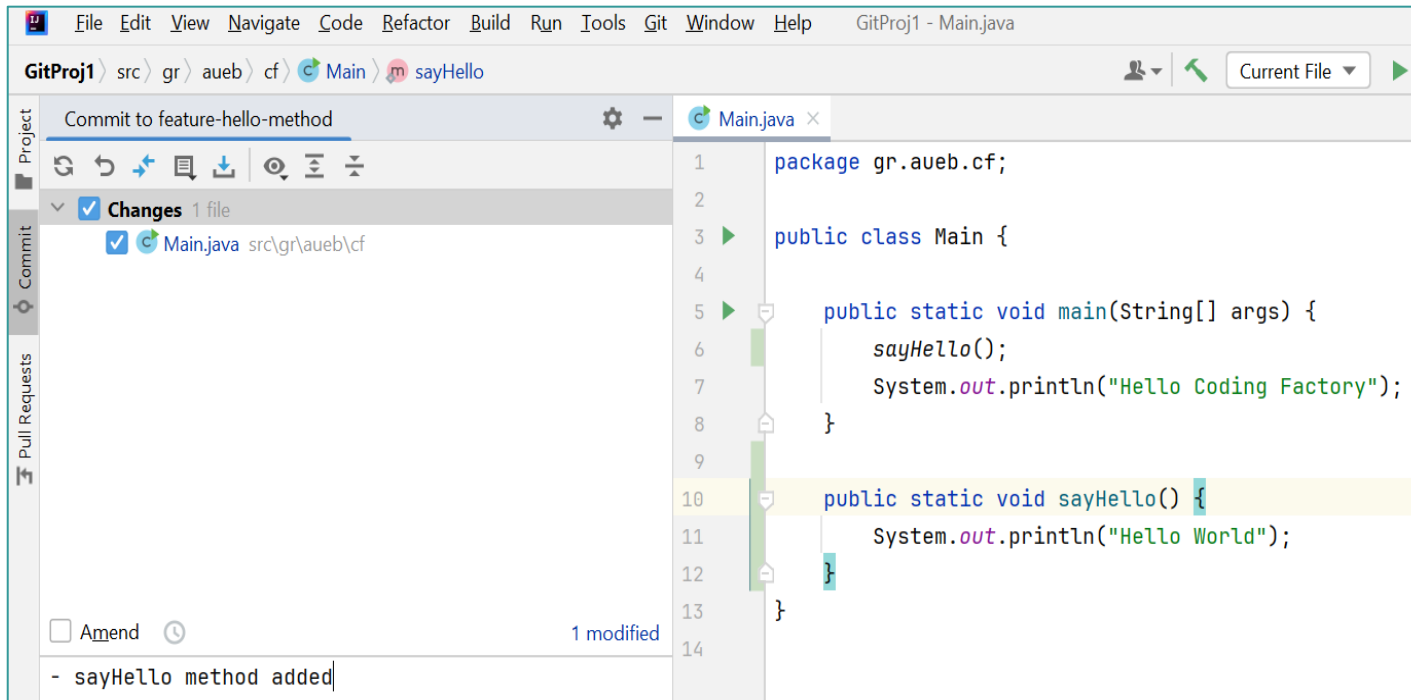




# Commit changes

Version Control με Git / GitHub

- Το μέλος της ομάδας κάνει αλλαγές / προσθήκες στο δικό του branch και μετά τις κάνει **commit** και **push**





# Manager (1)

Version Control με Git / GitHub

- Ο **manager** κάνει **pull** το **νέο branch** με τις **αλλαγές** και αφού ελέγξει θα τα κάνει **merge**
- Ο Manager μετά κάνει **checkout** στο **hello**. Το **hello branch** λειτουργεί ως **dev branch**

```
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (feature-hello-method)
$ git checkout hello
Switched to branch 'hello'
Your branch is up to date with 'origin/hello'.
```

- Κάνει **merger** το **feature branch** στο **hello**. Όλα τα **merge** τα κάνουμε πρώτα εδώ (στο **hello**) κι αν όλα είναι καλά κάνουμε **merge** στο **main branch**



# Manager (2)

Version Control με Git / GitHub

```
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (hello)
$ git merge --no-ff feature-hello-method
Merge made by the 'ort' strategy.
 src/gr/aueb/cf/Main.java | 5 +++++
1 file changed, 5 insertions(+)
```

- Το merge εφόσον έχουμε `--no-ff` αυτόματα δημιουργεί ένα merge commit ακόμα και αν το feature branch είναι ένα commit μπροστά, οπότε τότε θα μπορούσε να γίνει το merge με fast-forward, αλλά δεν θα φαινόταν στο history το merge, ενώ τώρα φαίνεται



# Manager (3)

Version Control με Git / GitHub

```
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (hello)
$ git push origin hello
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 233 bytes | 233.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:a8anassis/git-proj1.git
 4391c87..72548b7  hello -> hello
```

- Ο manager κάνει push στο hello branch. Σε αυτό το branch συλλέγονται (γίνονται merge) όλα τα επιμέρους branches

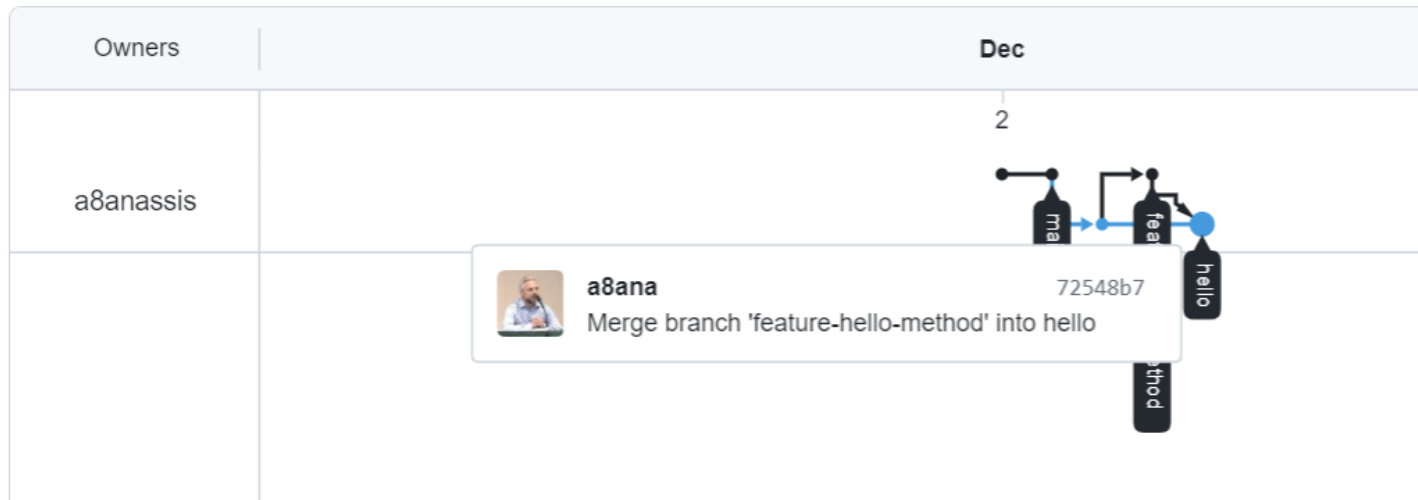




# Graph

## Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



- Παρατηρούμε ότι έχει καταγραφεί το merge commit και το history είναι πιο καθαρό (clear)



# Final merge

```
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (main)
$ git merge --no-ff hello
Merge made by the 'ort' strategy.
 src/gr/aueb/cf/Main.java | 7 ++++++-
 1 file changed, 6 insertions(+), 1 deletion(-)

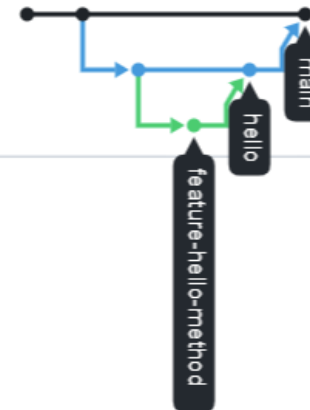
a8ana@thanassis-pc MINGW64 ~/IdeaProjects/idea1023/GitProj1 (main)
$ git push origin main
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 216 bytes | 216.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:a8anassis/git-proj1.git
 ca3515e..f092fc1  main -> main
```

- Τελικά ο manager κάνει merge το hello branch στο main branch. Πάλι με `--no-ff`. Μετά κάνει push
- Μετά μπορούν να κάνουν pull οι collaborators



## Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.





# Εργασία

Version Control με Git / GitHub

- Δημιουργήστε ένα repository στο GitHub και ανεβάστε όλα τα projects που έχουμε αναπτύξει στο πλαίσιο του Structured Programming
- Σε ένα 2<sup>ο</sup> repository ανεβάστε όλες τις ασκήσεις που έχετε κάνει ανά chapter
- Ανεβάστε στην Υποβολή Εργασίας, ένα .txt αρχείο με τα δύο GitHub links