



형상 관리 프로그램

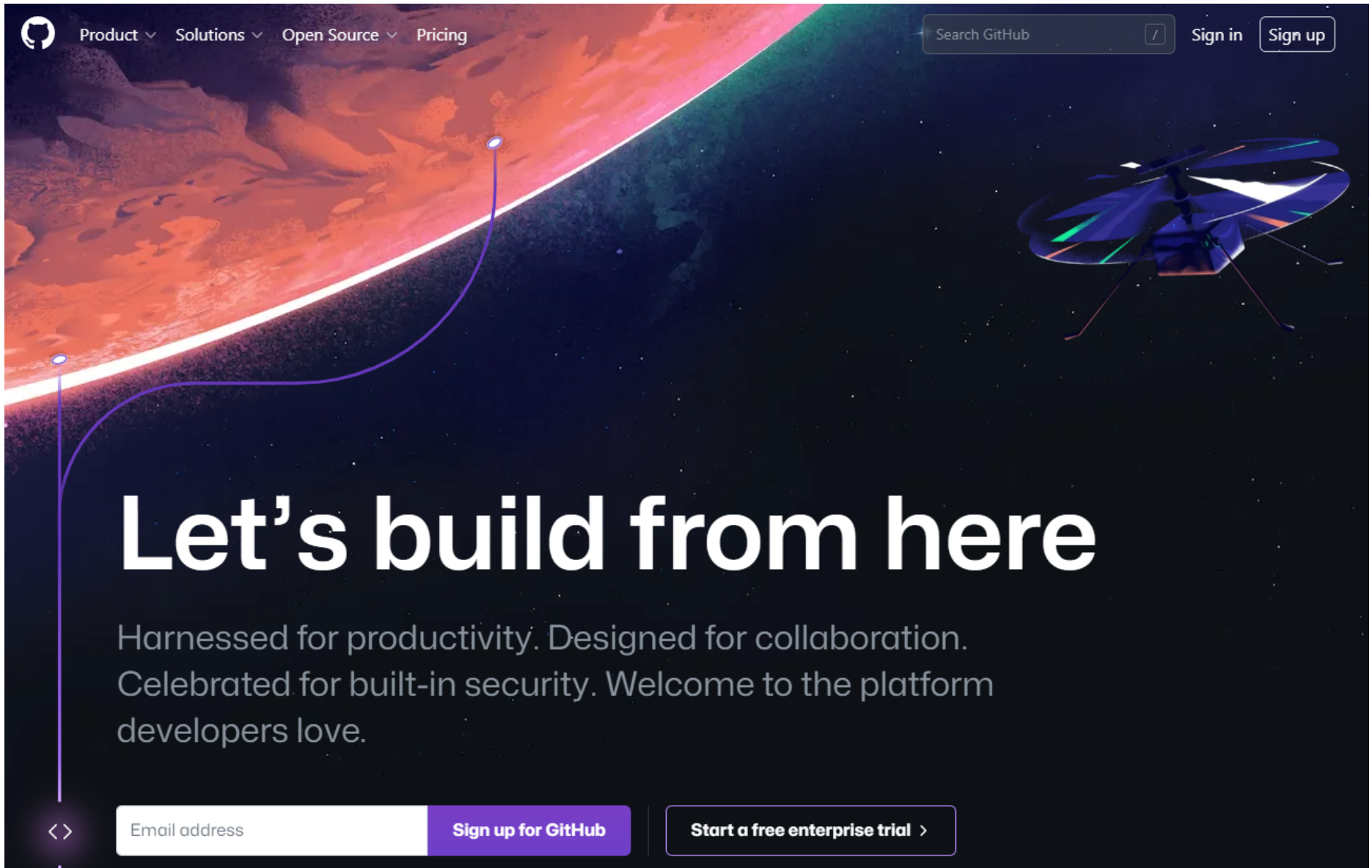
- 필요한 것 🖋️: 깃(git), 깃 허브(Git hub)

1. 깃 설치(os에 맞게) '<https://git-scm.com/>'



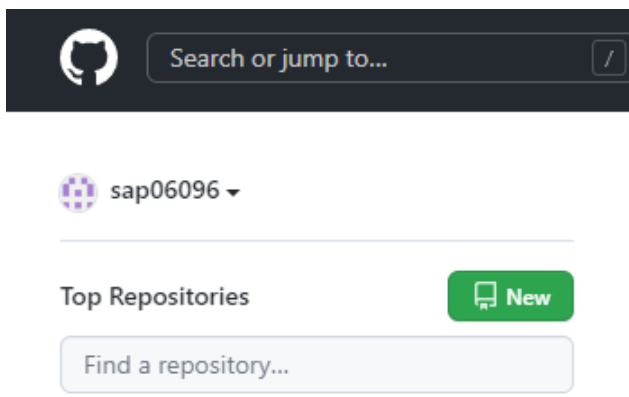
The image is a screenshot of the Git website. At the top left is the Git logo (a red diamond with a white branching diagram) followed by the text "git --everything-is-local". To the right is a search bar with the placeholder text "Search entire site...". Below the header, there is a main content area with a light gray background. On the left, there is text describing Git: "Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency." and "Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**." On the right, there is a 3D illustration of several stacks of papers connected by colored lines (red, blue, yellow) representing a branching model. Below the main content area, there are four circular icons with corresponding text: "About" (gears icon) "The advantages of Git compared to other source control systems.", "Documentation" (book icon) "Command reference pages, Pro Git book content, videos and other material.", "Downloads" (downward arrow icon) "GUI clients and binary releases for all major platforms.", and "Community" (speech bubbles icon) "Get involved! Bug reporting, mailing list, chat, development and more." On the right side of the bottom section, there is a monitor displaying "Latest source Release 2.39.2" and "Release Notes (2023-02-06)" with a button that says "Download for Windows".

2. 깃 허브 회원가입 '<https://github.com/>'



#깃 레파지토리 생성

1. 왼쪽사이드바에 New버튼 클릭



2. 레자피토리 정보 입력

1. Repository name(필수)

- 저장소 명칭을 입력합니다.

2. Description(생략 가능)

- 저장소를 설명하는 내용을 작성합니다.

3. 저장소 접근 권한


- Public: 해당 레포지토리를 공개합니다.
- Private: 비공개로 설정하여 개인 혹은 특정 사용자(혹은 팀원)만 레포지토리를 접근할 수 있습니다.

4. 저장소 초기 설정(생략 가능) —? 나중에 추가 가능함


- Add a README file: README 파일은 마크다운 언어로 작성된 문서로 해당 레포지토리를 사용하는 팀원 또는 사용자에게 프로젝트가 무엇인지 설명합니다.

- Add .gitignore: 특정 파일 또는 폴더를 push 할 때 무시하고 싶은 파일이나 폴더를 지정하는 파일입니다.
- Choose a license: License를 선택하지 않으면 기본 저작권법이 적용됩니다.

Owner *
Repository name *


 sap06096 ▼


/

obesis-T


Great repository names are short and memorable. Need inspiration? How about **expert-octo-broccoli**?

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.

Initialize this repository with:
Skip this step if you're importing an existing repository.


☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

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

.gitignore template: None ▼


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

License: None ▼

 You are creating a private repository in your personal account.

Create repository



3. 레파지토리를 생성한 화면

 sap06096 / obesis-T Private

Unwatch 1
Fork 0
Star 0


Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings

Quick setup — if you've done this kind of thing before

 Set up in Desktop
or
HTTPS
SSH
https://github.com/sap06096/obesis-T.git


Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line




```

echo "# obesis-T" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/sap06096/obesis-T.git
git push -u origin main

```

...or push an existing repository from the command line



```

git remote add origin https://github.com/sap06096/obesis-T.git
git branch -M main
git push -u origin main

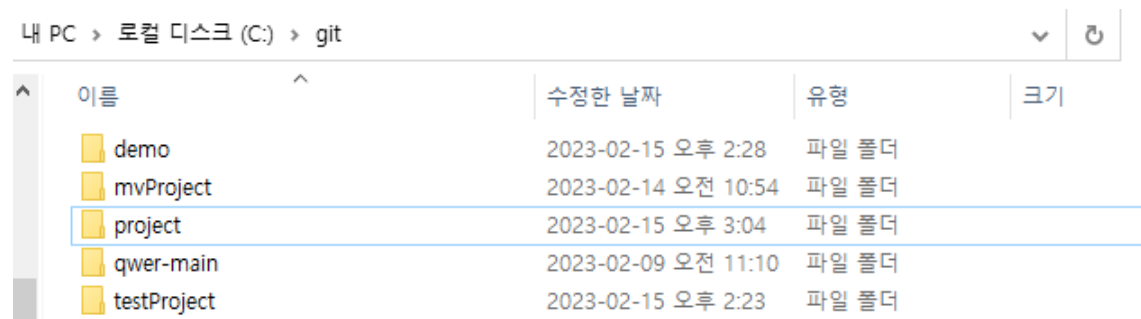
```

...or import code from another repository
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

#생성한 레파지토리에 프로젝트 올리는 법

1. 문서에서 자신이 올릴 프로젝트 폴더를 선택한다.



2. 마우스 우클릭 > Git Bash Here 클릭(git을 설치해야 쓸 수 있음)

3. 파일 준비(명령어 입력)

```
git init      #.git 파일 생성

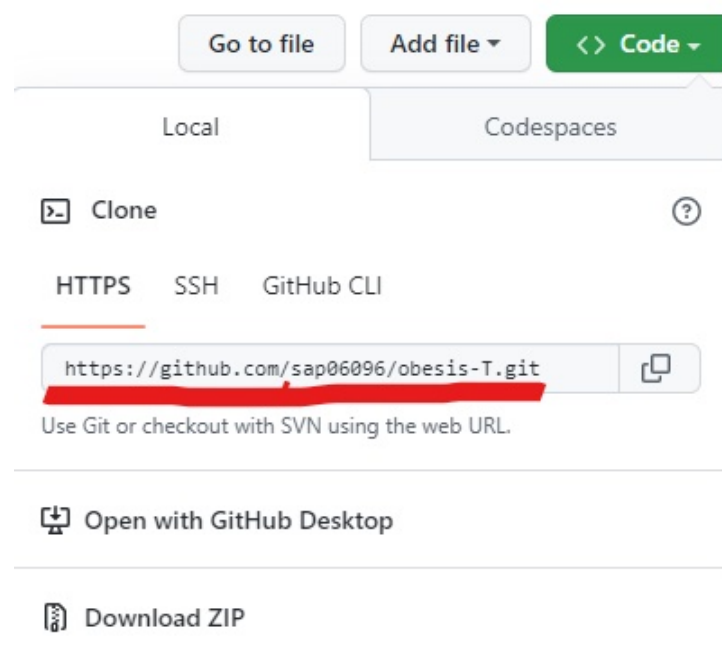
git add .     #선택한 프로젝트 폴더 내의 모든 파일 관리
              -> 특정파일만 하고 싶다면 git add 파일이름.파일형식 ex) git add a.txt

git commit -m "커밋하는 메세지쓰기(ex.first commit)"    #커밋
```

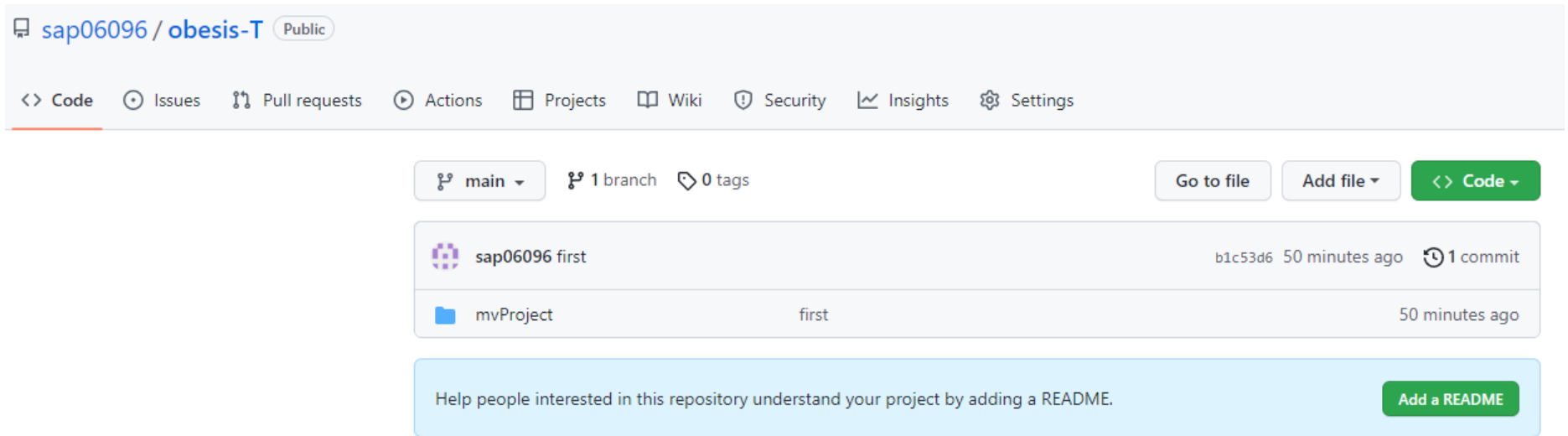
4. 업로드 하기(명령어 입력)

```
git remote add origin {아래 사진의 생성한 레파지토리의 코드에 깃허브 저장소 주소입력}

git push -u origin master #push
```



5. 레파지토리 화면으로 돌아와서 F5하면

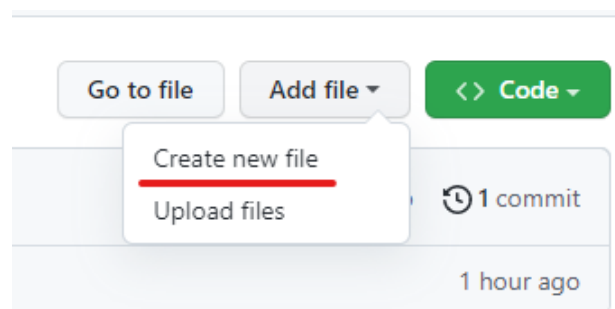


mvProject올리기 성공..!

#깃 명령어를 사용 안하고 레파지토리에 파일 업로드하기

내용에 앞서 깃 허브에 파일을 올릴 때 보통 폴더안에 정리해서 넣는다.
근데 깃 허브는 문서처럼 빈 폴더를 생성못함. ==> txt파일이나 README.md파일을 넣어줘야 됨.

1. Add file → Create new file선택



2. 폴더생성

2-1) 폴더 구조 작성

obesis-T / in main

☀폴더 구조

맨 앞 - Repository 이름

중간 - Folder 이름

마지막 - File 이름

*입력할 때는 폴더 이름을 쓰고 '/'를 넣어주면 다음 폴더 또는 파일 이름을 작성 할 수 있음.

obesis-T / sources / java / class / in main

*예를 들어서 만든 위에 사진 폴더 구조

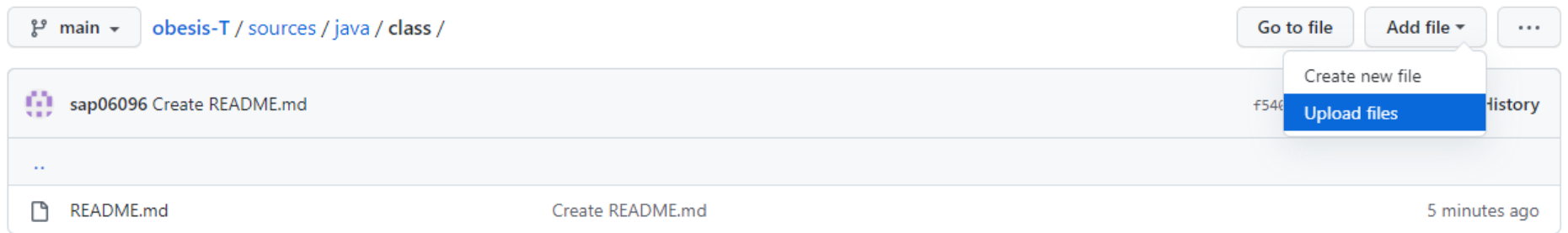
sources폴더 → java폴더 → class폴더 → README.md파일

2-2) commit

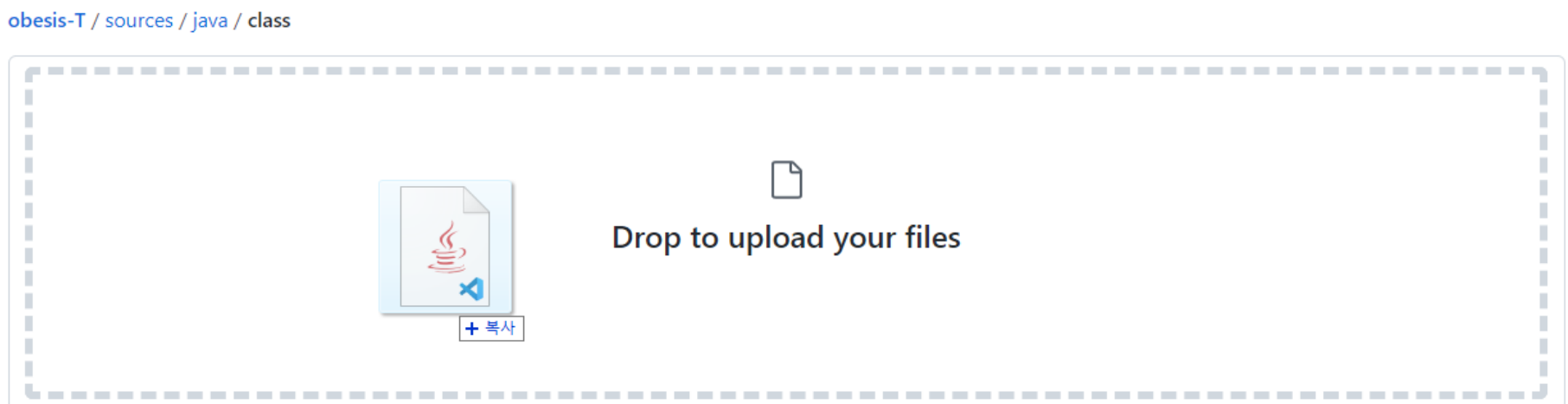
다 작성하면 화면 맨 아래에 Commit new file버튼을 클릭 해줘야 함.

방금 만든 class폴더 안에 java코드 파일 업로드하기

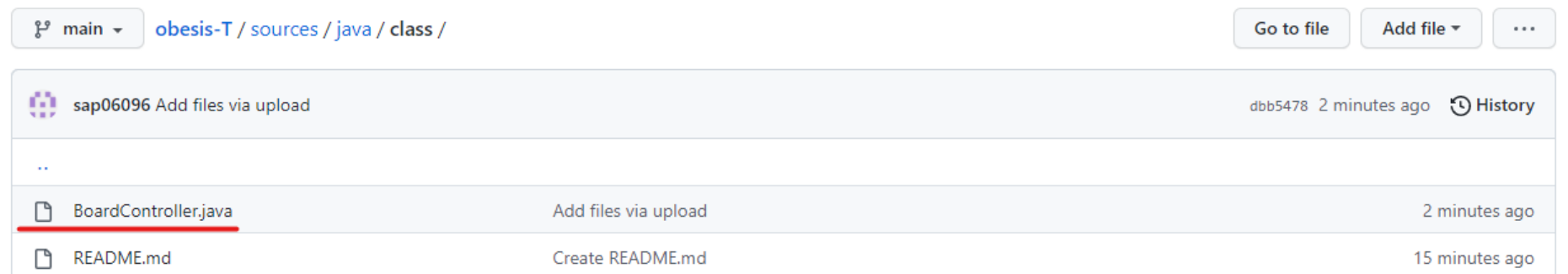
3. Add file → Upload Files (반드시 넣고자 하는 폴더 안에서 업로드 파일 버튼 누르기!!!)



4. 파일을 끌어다가 Drag → Uploading 완료되면 Commit changes



파일을 넣은 폴더 안으로 가보면



파일 업로드 완료....!!

#정리

Upload file과 Git Bash Here 둘 중 어떤 게 파일 올리기에 좋나?

Upload file

장점

- 폴더 구조를 보기 편하기 때문에 원하는 폴더에 빠르게 파일을 넣기에 좋음

단점

- 파일 업로드 갯 수 100개로 제한
- 프로젝트 파일같이 폴더 안에 구조가 복잡하고, 파일이 많은 내용을 업로드할 때는 폴더 구조까지 만들어주면서 파일을 업로드해야 되기 때문에 반복 수행을 해야 하는 단점

*파일이 많은 프로젝트 or 용량이 큰 파일을 담은 폴더들을 올릴 때

Git Bash Here로 올리는 것 추천!!!

*파일이 많지 않고 수정한 파일들을 넣어줄 때

Upload file로 드래그 하는 방식 추천!!!