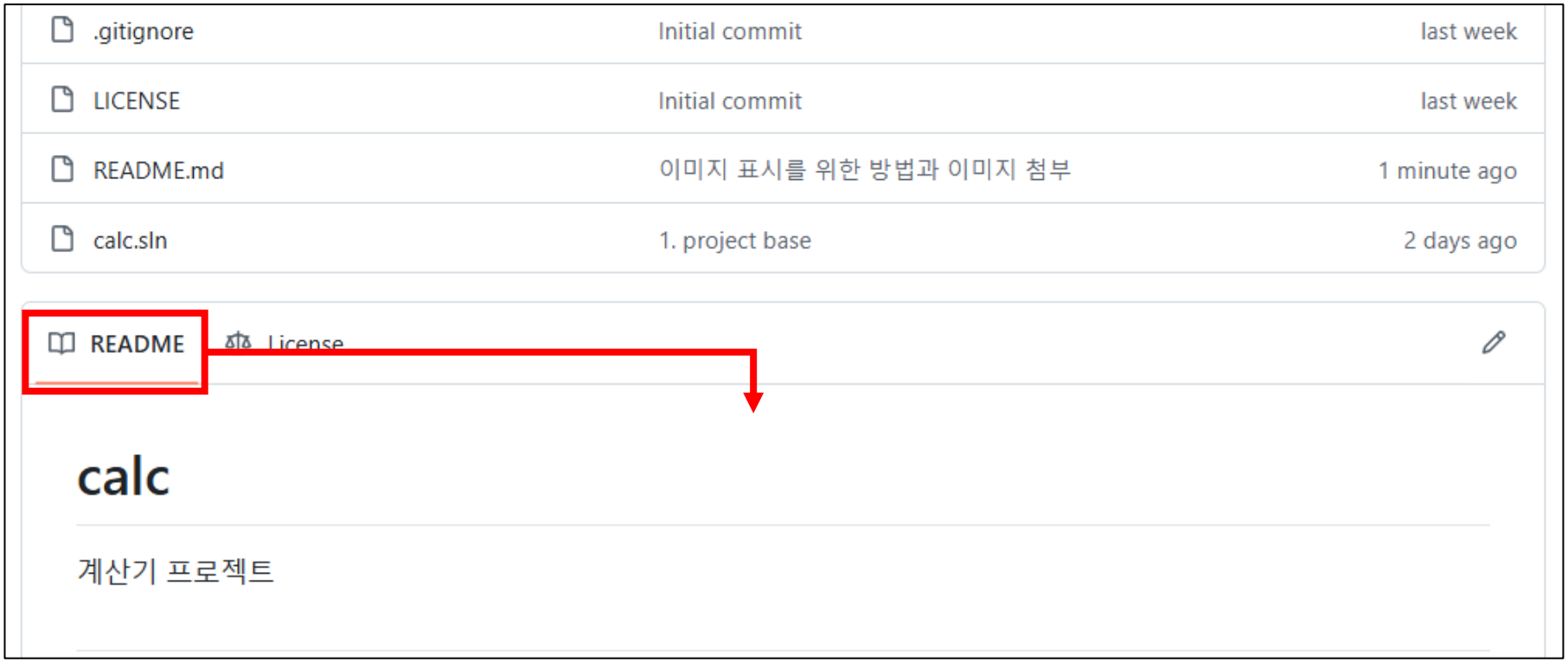
Markdown 언어

About..

컴퓨터소프트웨어공학과
김 원 일

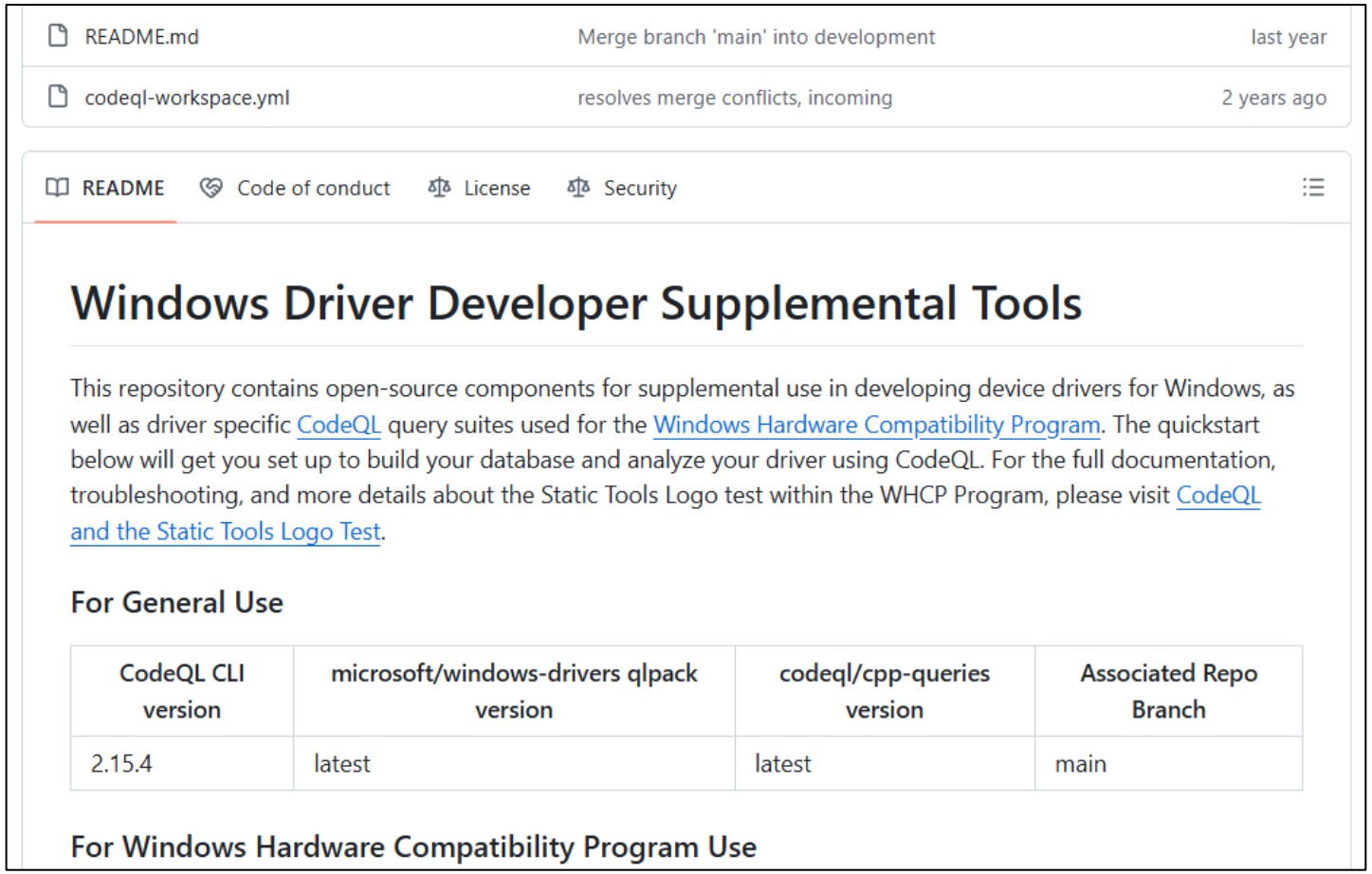
ReadMe.md

- github **메인 화면**
  - **메인 화면을 꾸미는 기본 파일들이 존재**
  - README.md : **메인 화면을 꾸미는 일종의** index.html**과 같은 파일**
  - LICENSE : **프로젝트의 라이선스를 표시하는 파일**
  - **포트폴리오 제출을 위해서는 어느 정도 꾸며야 하는 파일**



MarkDown 언어 - 1

- ReadMe.md 파일 적용 언어
  - github 저장소의 메인 페이지를 나타내는 파일을 표시하는 언어로 사용
  - 기본적으로 HTML 형식과 같은 간단한 문법을 사용하여 표시 가능



README.md — Merge branch 'main' into development — last year

codeql-workspace.yml — resolves merge conflicts, incoming — 2 years ago

README | Code of conduct | License | Security

Windows Driver Developer Supplemental Tools

This repository contains open-source components for supplemental use in developing device drivers for Windows, as well as driver specific CodeQL query suites used for the Windows Hardware Compatibility Program. The quickstart below will get you set up to build your database and analyze your driver using CodeQL. For the full documentation, troubleshooting, and more details about the Static Tools Logo test within the WHCP Program, please visit CodeQL and the Static Tools Logo Test.

For General Use

CodeQL CLI version	microsoft/windows-drivers qlpack version	codeql/cpp-queries version	Associated Repo Branch
2.15.4	latest	latest	main

For Windows Hardware Compatibility Program Use

MarkDown 언어 - 2

- **언어의 장/단점**
  - HTML과 같이 간단한 문법으로 가독성과 간결성을 제공
  - 문서 형식을 다른 형식(HTML, PDF 등)으로 변환이 가능
  - 문법이 간단하고 빠르게 문서를 표시할 수 있음
  - 별도의 도구 없이 메모장으로도 작성이 가능
  - 표준이 없기 때문에 변환 도구에 따라 다른 결과물이 출력
  - HTML과 완전하게 호환되지 않음
  - 문법이 간단하여 쉽게 문서를 만들 수 있어 간단하게 사용 가능
  - 간단한 태그는 HTML과 호환되는 경우들도 있어 편하게 사용 가능
  - ReadMe.md 파일에 적용되어 메인 화면 출력하는데 사용
  - 파일에 내용을 기록하면 자동으로 저장소 첫 화면에 표시되도록 구성

MarkDown 언어 – 3

- **언어 학습**
  - **기본적인** HTML **문법을 그대로 사용하면 되므로 복잡하지 않음**
  - **아래 참고 사이트에서 추가적인 문법이나 정보 획득**
  - https://www.markdownguide.org/
  - https://gist.github.com/ihoneymon/652be052a0727ad59601
  - https://www.heropy.dev/p/B74sNE
  - **가장 추천하는 편집기는** Visual Studio Code
  - **무료이면서 회사/기관 관계 없이 어디서나 사용 가능**
  - **화면을 보면서 직접 수정이 가능하므로 가장 추천**

MarkDown 문법 예

- **목록 표시**
  - , <ul>, 와 같은 태그로 변환되는 목록 표시
  - 라인이 "1." 로 시작하면 로 변환되어 화면에 표시
  - 라인이 "-"로 시작하면 로 변환되어 화면에 표시
  - 들여쓰기도 적용되어 HTML과 비슷하게 사용 가능

MARKDOWN

````
`-`로 시작하는 순서가 없는 목록으로 구분합니다.

```markdown
1. 순서가 있는 항목
1. 순서가 있는 항목
    1. 순서가 없는 항목
    1. 순서가 없는 항목
1. 순서가 있는 항목
1. 순서가 있는 항목

- 순서가 없는 항목
- 순서가 없는 항목
    - 순서가 없는 항목
    - 순서가 없는 항목
````

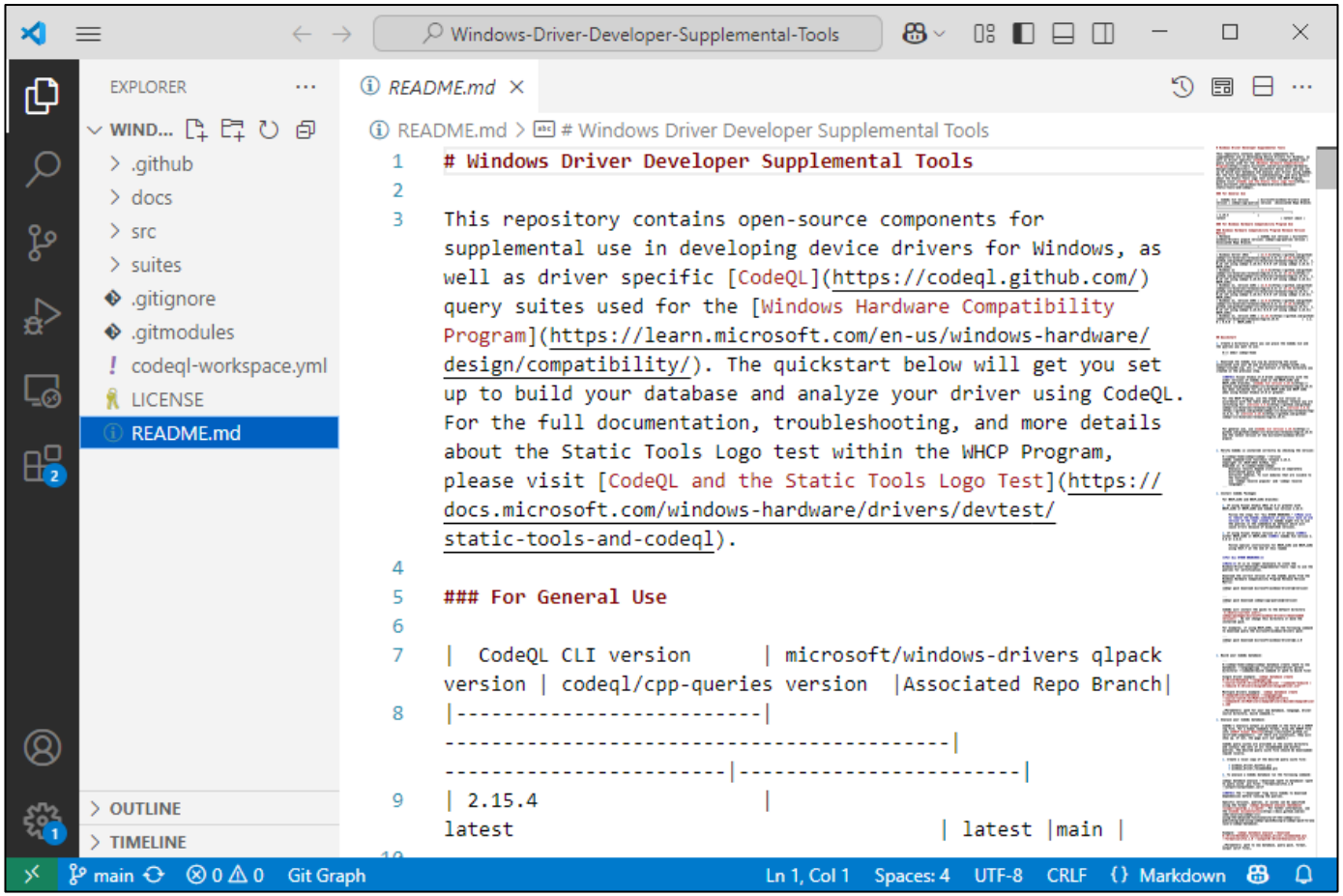
출력 결과:

1. 순서가 있는 항목
2. 순서가 있는 항목
    1. 순서가 없는 항목
    2. 순서가 없는 항목
3. 순서가 있는 항목
4. 순서가 있는 항목

- 순서가 없는 항목
- 순서가 없는 항목
    - 순서가 없는 항목
    - 순서가 없는 항목
````

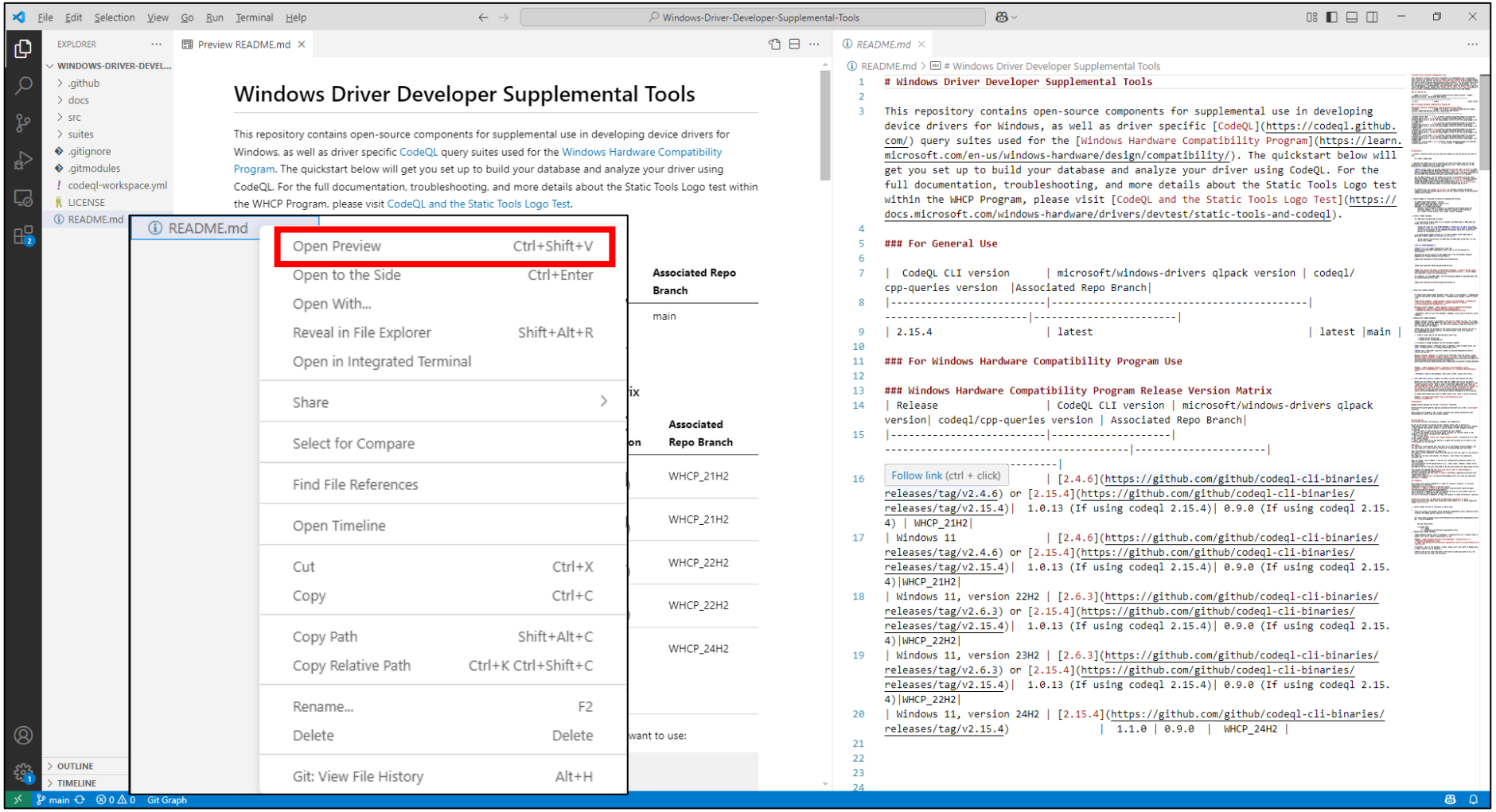
Visual Studio Code 편집 – 1

- ReadMe.md
  - 편집기에서 직접 확인하면 아래와 같이 일반적인 텍스트로 출력
  - 문서 편집기의 텍스트를 보고 화면을 편집하는 것은 쉽지 않음



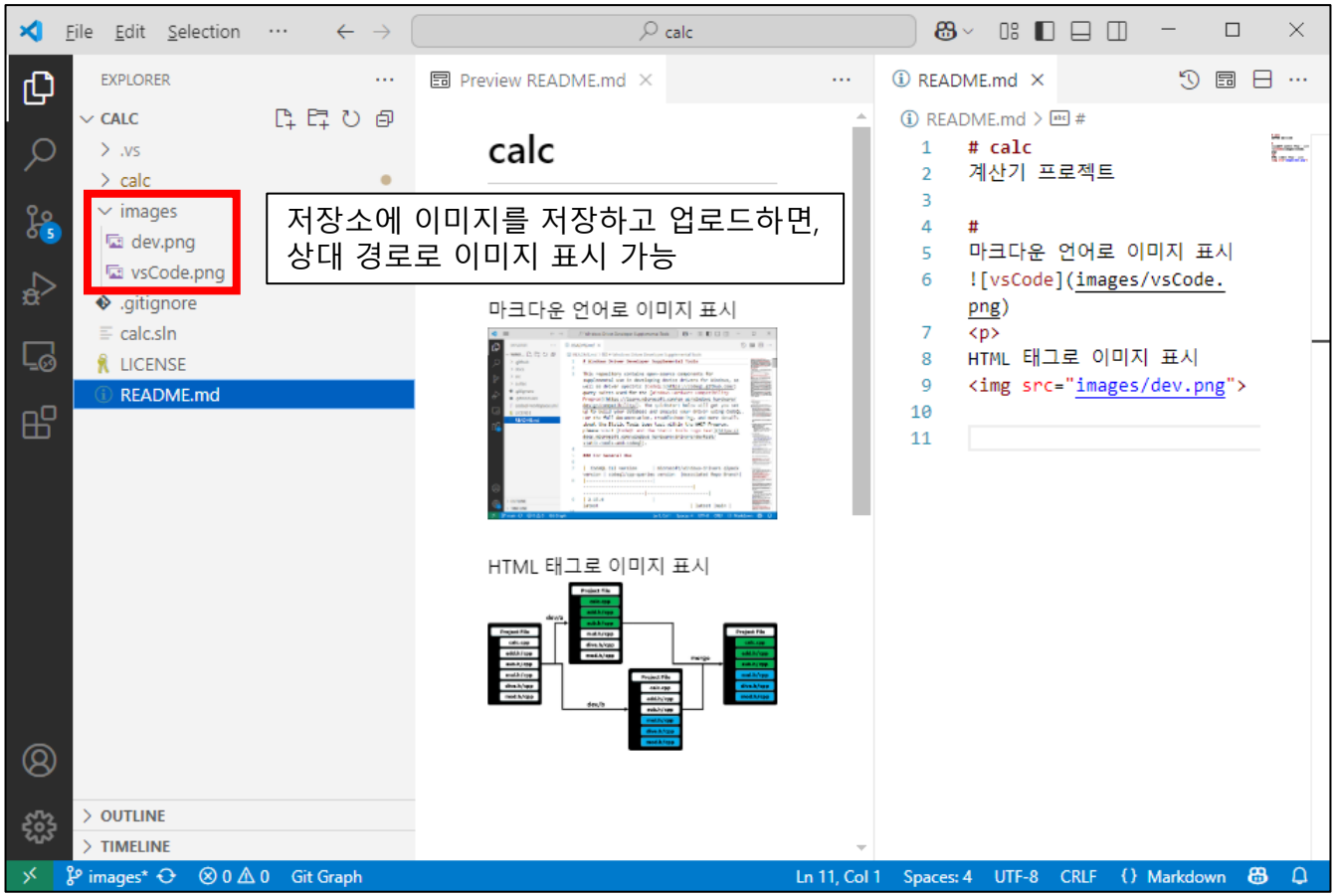
Visual Studio Code 편집 – 2

- 보면서 즉시 편집과 확인이 가능하도록 구성



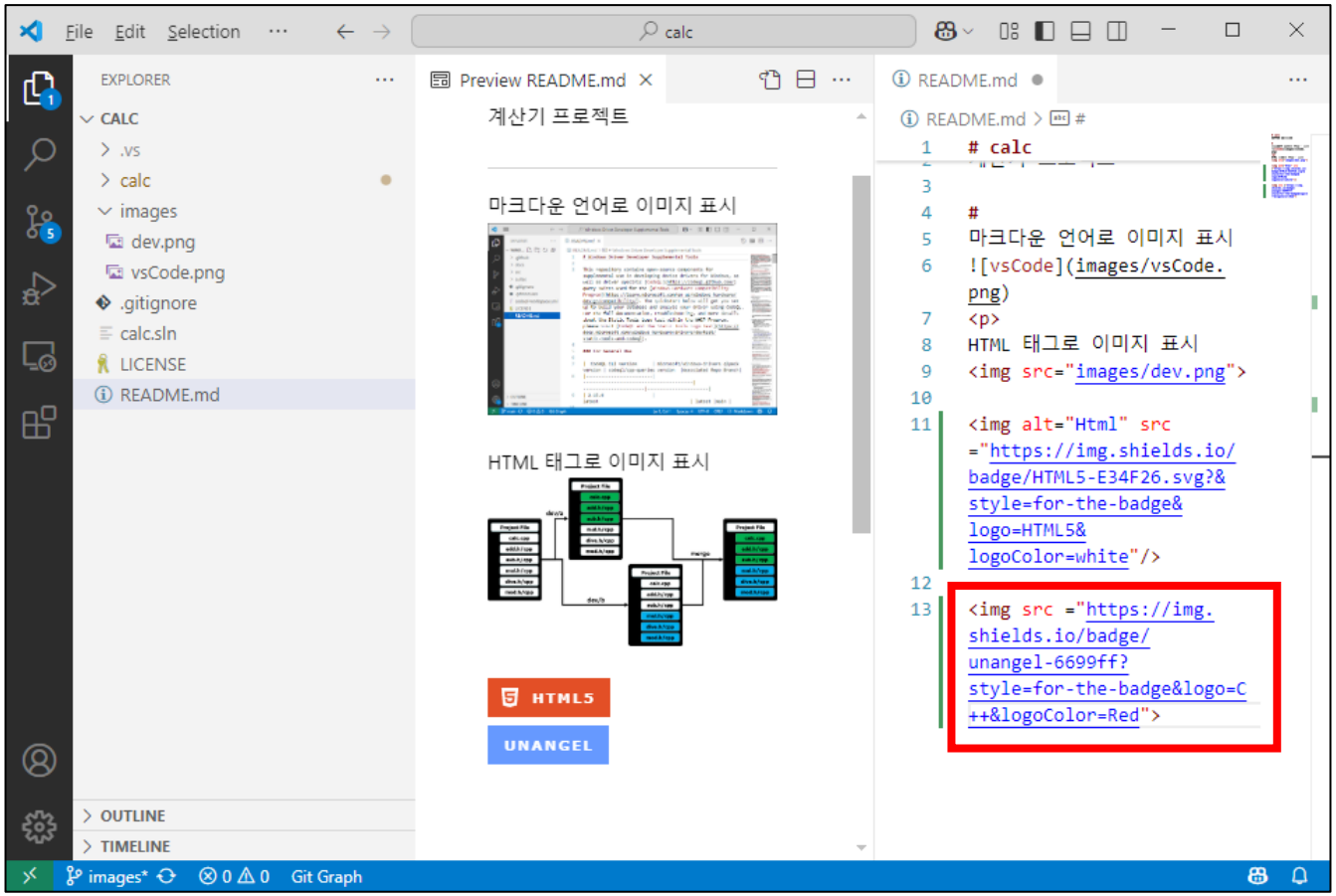
주 화면에 이미지 표시

- **이미지 표시**
  - MarkDown : ![**텍스트**](**이미지 경로**)로 **이미지를 표시**
  - HTML : **이미지 표시와 동일한** <img src="**이미지 경로**">로 **표시**



뱃지 아이콘 사용하기

- 기술 스택 또는 각종 정보 표시 아이콘
  - https://shields.io/ 에서 직접 생성하여 적용할 수 있음
  - 2가지 방식에서 원하는 형태로 뱃지를 표시하여 보다 많은 정보 제공 가능



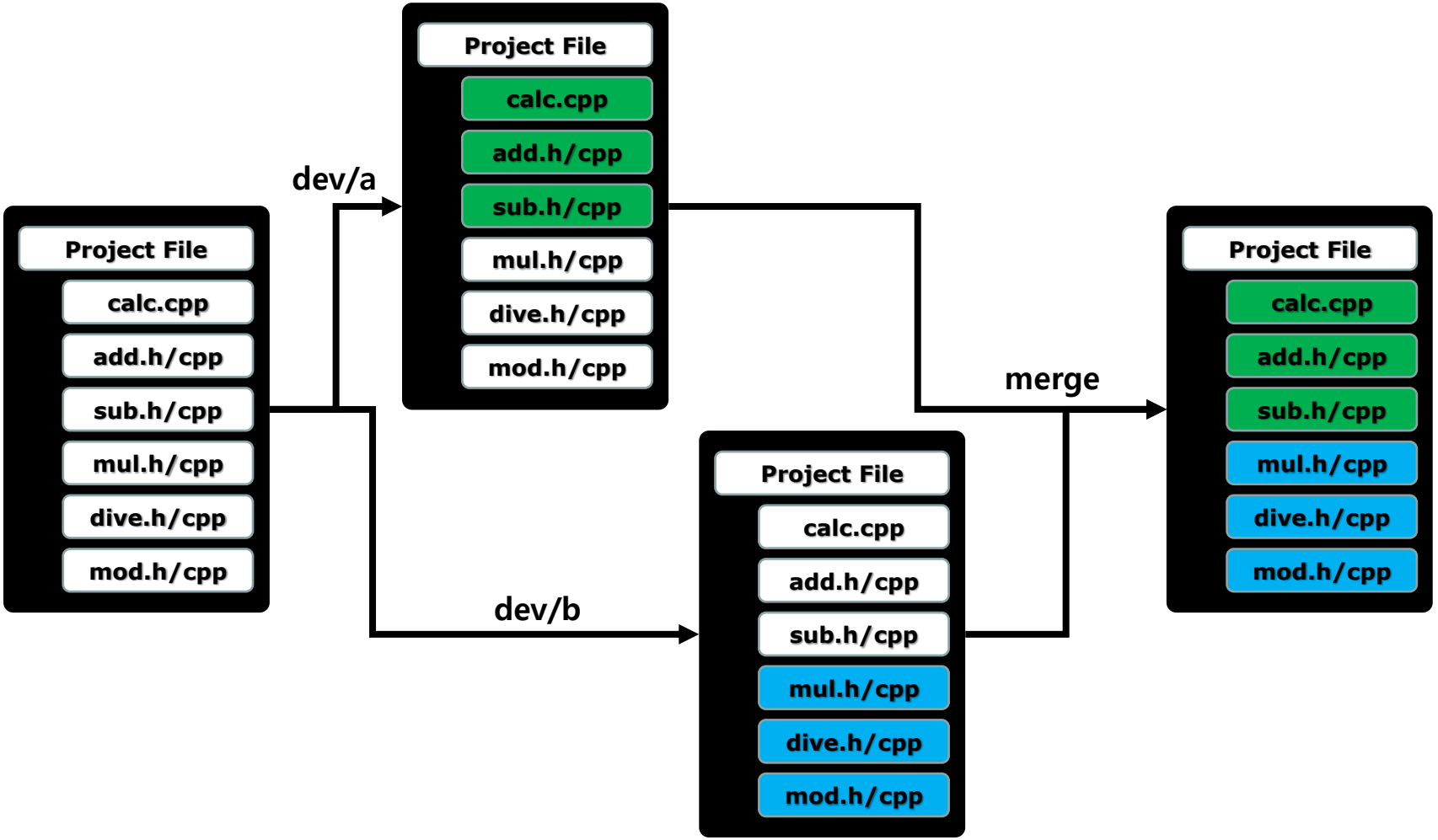
비 선형 개발과 충돌 해결

About..

컴퓨터소프트웨어공학과
김 원 일

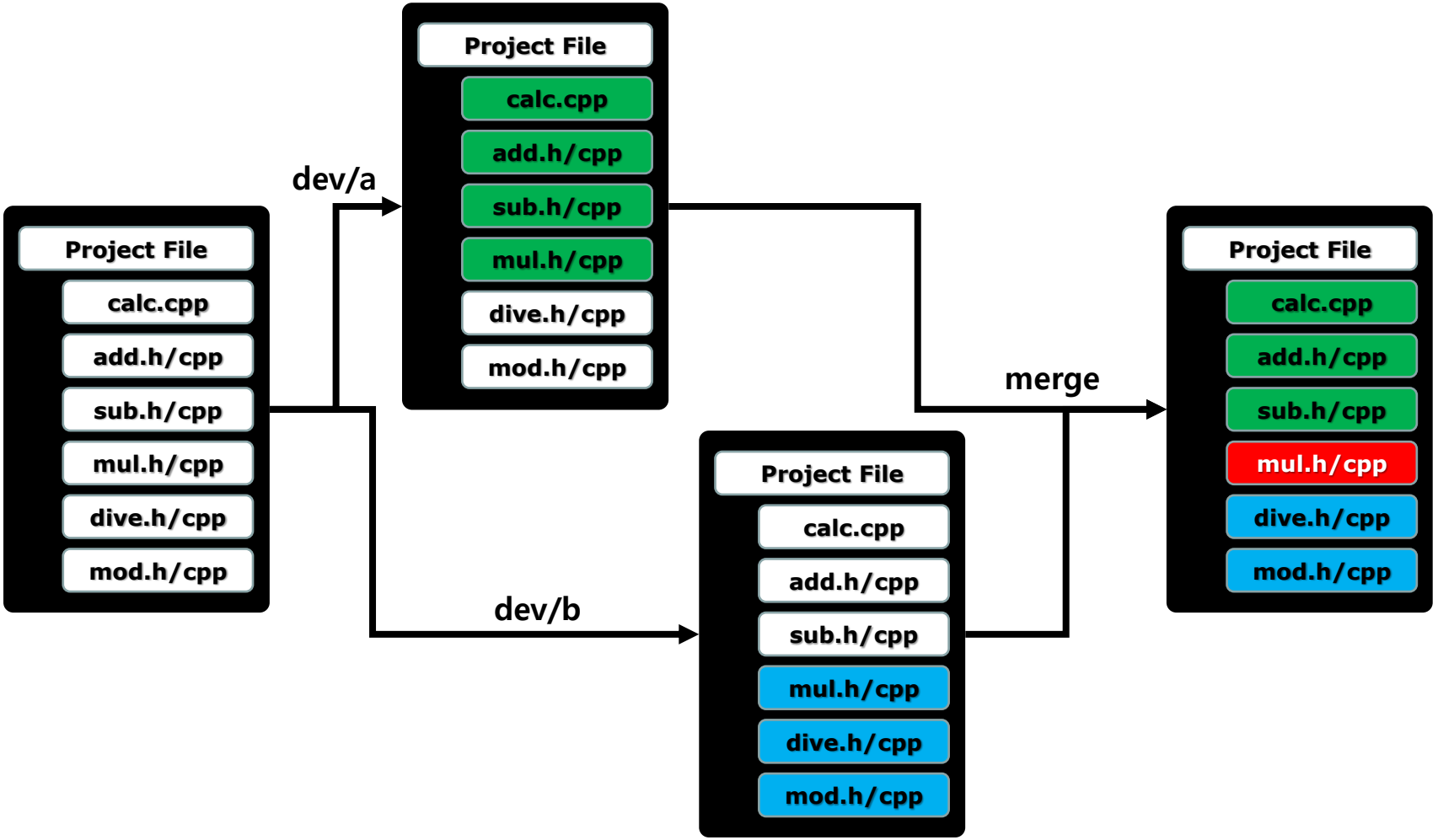
비 선형 개발 - 1

- 개발 실습
  - 브렌치에서 작업한 내용이 겹치지 않도록 각자 개발하고 병합



비 선형 개발 - 2

- **브렌치 병합**
  - mul.h/cpp를 동시에 수정한 경우 병합에 문제가 발생



코드 충돌 예제 - 1

- **계산기 프로젝트 복제**
  - git clone https://github.com/ycs-wikim/calc.git
  - **복제한 저장소에서** dev/a, dev/b **2개의 브렌치를 생성**
  - **생성한 브렌치에 코드 충돌이 발생할 수 있도록 코드를 수정 후 병합**

```
MINGW64:/f/202507001/calc

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main)
$ git branch
* main

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main)
$ git branch dev/a

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main)
$ git branch dev/b

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main)
$ git branch
  dev/a
  dev/b
* main

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main)
$
```

코드 충돌 예제 - 2

- **코드 변경** - 1
  - "dev/a" **브렌치로 브렌치를 변경하고,** mul.h **파일을 수정**
  - **수정 후** Commit **하여** "dev/a" **브렌치의 작업을 종료**

```
MINGW64:/f/202507001/calc
unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main)
$ git checkout dev/a
Switched to branch 'dev/a'

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (dev/a)
$ vi calc/
add.cpp                calc.vcxproj.user     mul.cpp
add.h                  dive.cpp              mul.h
calc.cpp               dive.h                sub.cpp
calc.vcxproj           mod.cpp               sub.h
calc.vcxproj.filters   mod.h

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (dev/a)
$ vi calc/mul.h

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (dev/a)
$ git add calc/mul.h

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (dev/a)
$ git commit
```

코드 충돌 예제 - 3

- **코드 변경** - 2
  - "dev/b" **브렌치로 브렌치를 변경하고**, mul.h **파일 수정**
  - **수정 후** Commit **하여** "dev/b" **브렌치의 작업을 종료**
  - **즉, 동일한 파일을** 2**개의 브렌치에서 수정한 경우를 생성**

```
MINGW64:/f/202507001/calc
unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main)
$ git checkout dev/b
Switched to branch 'dev/b'

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (dev/b)
$ vi calc/mul.h

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (dev/b)
$ git add calc/mul.h

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (dev/b)
$ git status
On branch dev/b
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   calc/mul.h


unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (dev/b)
$ git commit
```

코드 충돌 예제 - 4

- **동일 파일 수정 - 1**
  - 각 브렌치에서 동일한 파일을 각기 다른 형태로 수정

MINGW64:/f/202507001/calc

```
1 #pragma once
2
3 /// edit in dev/a
4
5 int mul(int, int);
6
~
~
~
~
calc/mul.h[+] [dos] (20:40 04/06/2025)                    6,0-1 All
```

MINGW64:/f/202507001/calc

```
1 #pragma once
2
3 // edit dev/b branch
4 int mul(int, int);
~
~
~
~
calc/mul.h[+] [dos] (17:05 09/06/2025)                    3,20 All
```

코드 충돌 예제 - 4

- **동일 파일 수정 - 1**
  - 각 브렌치에서 동일한 파일을 각기 다른 형태로 수정

MINGW64:/f/202507001/calc

```
1 #pragma once
2
3 /// edit in dev/a
4
5 int mul(int, int);
6
~
~
~
~
calc/mul.h[+] [dos] (20:40 04/06/2025)                    6,0-1 All
```

MINGW64:/f/202507001/calc

```
1 #pragma once
2
3 // edit dev/b branch
4 int mul(int, int);
~
~
~
~
calc/mul.h[+] [dos] (17:05 09/06/2025)                    3,20 All
```

코드 충돌 예제 - 5

- **동일 파일 수정** - 2
  - **각 브렌치에서 동일한 파일의 내용을 확인**
  - **파일 내용이 다른 형태를 확인하고** main **브렌치로 이동**

```
MINGW64:/f/202507001/calc
unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (dev/b)
$ cat calc/mul.h
#pragma once

// edit dev/b branch
int mul(int, int);

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (dev/b)
$ git checkout dev/a
Switched to branch 'dev/a'

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (dev/a)
$ cat calc/mul.h
#pragma once

/// edit in dev/a

int mul(int, int);


unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (dev/a)
$
```

코드 충돌 예제 - 6

- **브렌치 병합**
  - main **브렌치에서** dev/a, dev/b **브렌치 순으로 병합 시도**
  - **먼저** dev/a **브렌치를 병합할 때** Fast-forward**로 병합되는 것을 확인**
  - **이어** dev/b **브렌치를 병합하려고 할때** calc/mul.h **파일로 인해 병합 불가**
  - **동일 파일에 수정이 발생하여 병합되지 않고 병합 중인 상태로 변경**

코드 충돌 예제 - 7

- **병합 상태 확인**
  - **병합을 진행 중인 상태에서 충돌이 발생한 내용을 수정해야 병합 가능**
  - "git merge --abort"**로 병합 진행 상태를 완전히 취소할 수 있음**
  - **병합에서 문제가 발생한 파일에 대한 정보가 출력됨**
  - **2개의 브렌치에서 서로 변경이 발생하여 병합되지 않는 것을 확인 가능**

```
MINGW64:/f/202507001/calc
unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main|MERGING)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   calc/mul.h

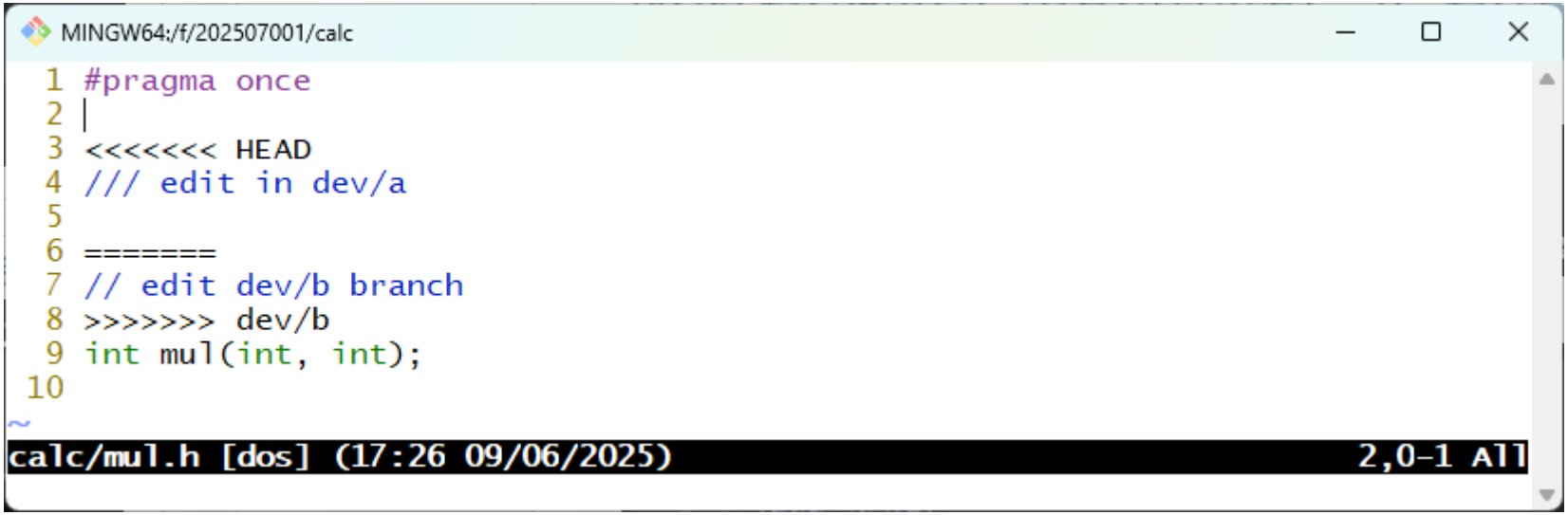
no changes added to commit (use "git add" and/or "git commit -a")

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main|MERGING)
$
```

브렌치 상태 : 병합 진행 상태

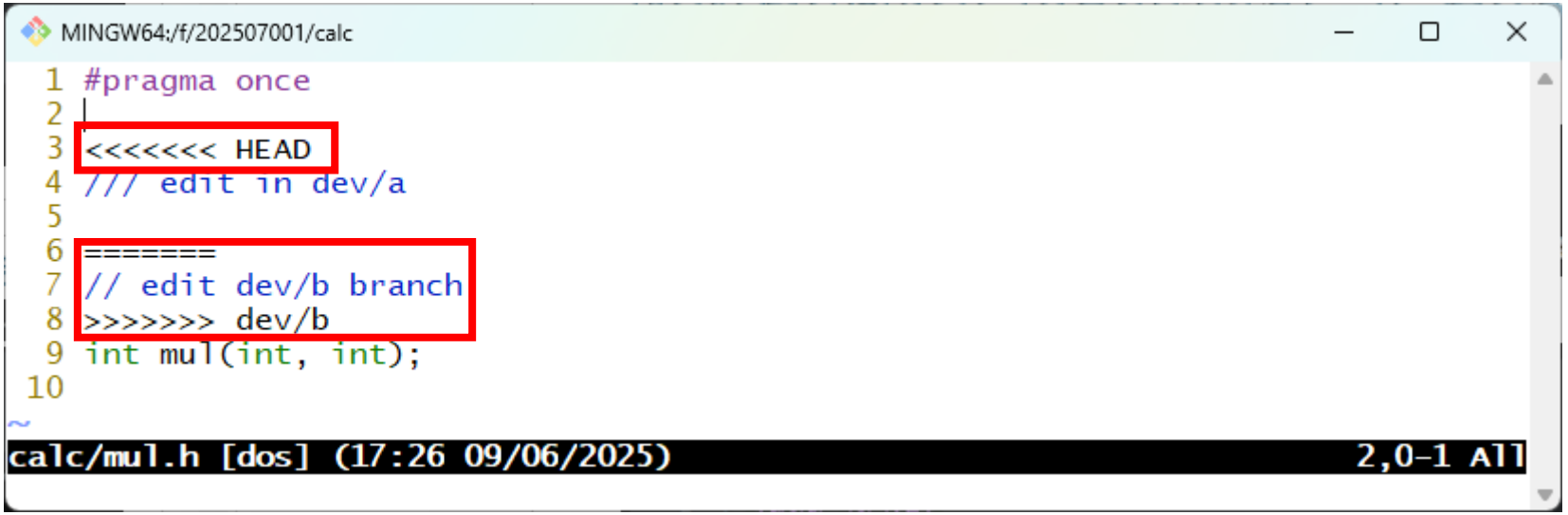
코드 충돌 예제 - 8

- **코드 상태**
  - 양쪽에서 수정된 파일을 열어보면 아래와 같이 나타남
  - 먼저 병합한 dev/a 브렌치에서의 수정 내용과 병합하려는 dev/b의 차이점
  - 현재 상태에서는 간단한 수정이기 때문에 출력이 비교적 적음
  - 복잡한 코드가 중첩되면 내용 수정이 매우 어려워지기 때문에 주의
  - 현재 브렌치의 내용은 "<<<<<<< HEAD" 형태로 출력
  - "=======" 이후가 병합하려는 브렌치의 서로 다른 내용을 나타냄
  - ">>>>>>> dev/b"가 병합 브렌치의 코드 내용 마지막을 나타냄



코드 충돌 예제 - 9

- **코드 수정**
  - 2개 브렌치 중에 현재 합쳐야 할 코드와 그렇지 않는 코드를 구분해야 함
  - 현재 HEAD가 가리키고 있는 main 브렌치의 내용이 맞는 경우
    - ➔ dev/b의 코드와 정보 제공 라인들을 삭제하고 병합을 수행
    - ➔ 아래 그림의 붉은 박스 라인들을 삭제하면 HEAD와 동일한 상태
  - 병합하려는 dev/b 브렌치의 내용이 맞는 경우
    - ➔ 병합 취소 후 코드를 수정하거나 병합 방법 변경 등 복잡해짐
    - ➔ 정해진 방법은 없으나 git에서 충돌 해결 방법을 제시하기도 함



코드 충돌 예제 - 10

- **충돌 해결 : HEAD가 맞는 경우 - 1**
  - **병합 상태에서 HEAD의 코드만 남기도록 파일 수정과 준비 상태로 변경**

```
MINGW64:/f/202507001/calc

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main|MERGING)
$ cat calc/mul.h
#pragma once

/// edit in dev/a

int mul(int, int);


unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main|MERGING)
$ git add calc/mul.h

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main|MERGING)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)


unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main|MERGING)
$
```

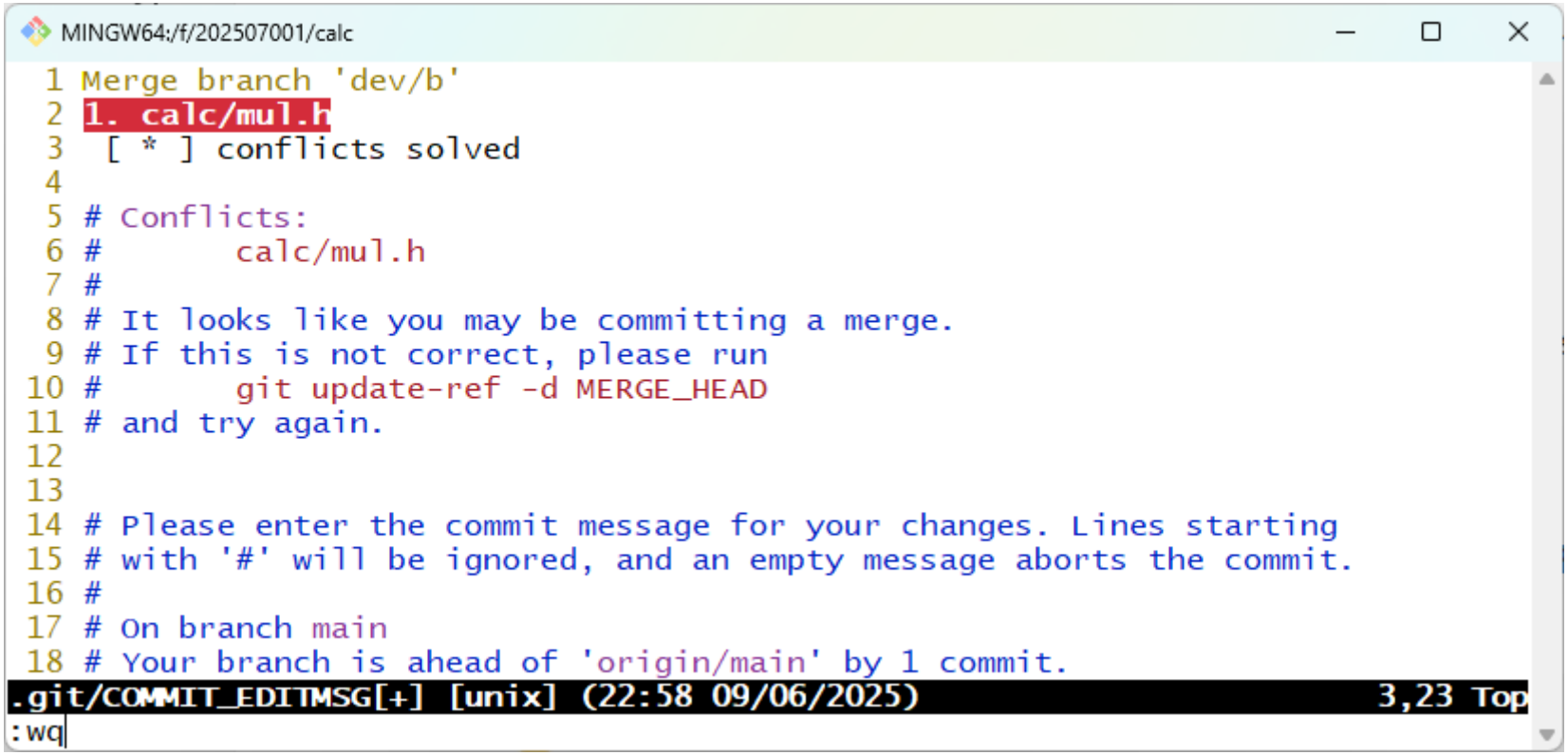
HEAD와 동일하게 파일 내용을 수정한 상태

수정한 다음 충돌이 발생한 파일을 준비 상태로 변경

상태 정보 출력 내용이 변경
충돌이 고쳐졌으나 아직 병합 중임을 알림

코드 충돌 예제 - 11

- **충돌 해결 : HEAD가 맞는 경우 - 2**
  - **준비 상태에서 그대로 Commit을 수행하면 아래 그림과 같이 정보 출력**
  - **별도 정보를 남기고 싶다면 추가적인 메시지를 입력하는 것도 좋음**
  - **충돌 파일에 대한 정보와 병합 브렌치를 rebase 하는 옵션도 출력**



코드 충돌 예제 - 12

- **충돌 해결 : HEAD가 맞는 경우 - 3**
  - Commit **완료 후** main **브렌치에 정상적으로 병합된 것을 확인**
  - Commit **메시지도 그대로 적용되어 있음**
  - **추가 코드 작업이 가능하나** dev/b **브렌치 작업은 여전히 충돌 발생 가능**
    - dev/b **브렌치의 코드를 수정하거나 브렌치를 삭제하는 것도 방법**

```
MINGW64:/f/202507001/calc

[main 61c976f] Merge branch 'dev/b' 1. calc/mul.h  [ * ] conflicts solved

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

unangel@DESKTOP-I8OQG85 MINGW64 /f/202507001/calc (main)
$ git log --oneline
61c976f (HEAD -> main) Merge branch 'dev/b' 1. calc/mul.h  [ * ] conflicts solve
d
a59fd21 (dev/b) 1. calc/mul.h  [ * ] modify file mul.h
2256807 (dev/a) 1. calc/mul.h  [ * ] edit for comments
f1daa88 (origin/main, origin/HEAD) 1. project base  [ + ] for calc project
7f62ade Initial commit
```

코드 충돌 해결 과제

- **충돌 해결 : dev/b가 맞는 경우**
  - **현재 dev/a가 main 브렌치에 이미 병합되어 있는 상태임**
  - **이미 main에 적용된 코드가 잘못 병합되어 있음**
  - **새로 병합해야 하는 브렌치가 올바른 코드이므로 이를 적용해야 함**
  - **할 수 있는 모든 방법을 총 동원하여 팀원끼리 해결할 것**
  - **해결 과정을 순서대로 정리하고, 화면을 캡처하여 과제로 제출 후 확인**
    - **과제 : 해결 과정과 해결된 상태에 대한 git-flow 화면 캡처를 PPT로 제출**