

⚠ 본 사이트를 이용함으로써 **쿠키 정책**과 **개인정보처리방침**을 읽고 이해하였다는 의사표시를 하게 됩니다.

동의함

COMP2500 실습 1

목차

[I. 윈도우 운영체제에 IntelliJ 설치하기](#)

[II. 깃 클라이언트 설치하기](#)

[III. 웹 기반 비공개 깃 저장소 만들기](#)

[IV. 빌드봇을 내 깃 저장소에 초대하기](#)

[V. 깃 저장소 설정 마무리](#)

[저장소 클론하기](#)

[클론한 폴더에 템플릿 집어넣기](#)

[VI. '나의 POCU'에서 내 정보 업데이트하기](#)

[슬랙 가입하기](#)

[VII. 간단한 프로그램 작성하기](#)

[VIII. 코드 제출 및 채점받기](#)

POCU 아카데미에서 실습을 진행하기 위해서는 먼저 환경을 구축해야 합니다. 이번 실습뿐만 아니라 앞으로 모든 실습과 과제를 할 때 사용하는 환경이니 이번 한 번만 잘 따라 해 주세요. 이번 실습을 마치고 나면 자신의 깃(Git) 저장소에 코드를 저장(제출)하는 법과 슬랙에서 빌드봇에게 채점을 받는 법을 알게 될 겁니다.

I. 윈도우 운영체제에 IntelliJ 설치하기

본 과목에서 기본적으로 사용하는 운영체제와 통합개발환경(IDE)은 윈도우와 IntelliJ이며 자바는 Adoptium 그룹이 제공하는 Temurin OpenJDK 버전 11.0.14를 사용합니다.

- 다음의 가이드([링크](#))를 참고하여 Temurin OpenJDK를 설치해주세요.
- 다음의 가이드([링크](#))를 참고하여 IntelliJ를 설치해 주세요.
 - IntelliJ에서 Temurin OpenJDK를 사용하도록 설정하시고 JDK의 이름(Name)을 'Java-11'으로 바꿔주세요.

공식적으로 지원하는 운영체제는 아니지만 맥 및 리눅스를 사용하셔도 IntelliJ와 Temurin OpenJDK를 설치하실 수 있습니다. 단, POCU 아카데미의 공식 문서나 지원은 윈도우를 기초로 하니 본인 스스로 웹 검색을 하며 문제를 해결해야 할 수도 있습니다.

II. 깃 클라이언트 설치하기

깃(Git)을 써 본 적이 없으신가요? 그러면 소스트리(Sourcetree)라는 프로그램을 추천합니다.

소스트리를 설치하는 방법은 다음의 가이드([링크](#))를 참고해주세요.

혹시 이미 깃 사용법을 알고 계신 분들은 본인이 원하는 다른 클라이언트 프로그램을 사용하셔도 됩니다. 하지만 POCU의 공식 문서나 지원은 소스트리에 한하니 그 점을 유의해주세요.

III. 웹 기반 비공개 깃 저장소 만들기

이 수업에서 과제 및 실습을 제출할 때는 깃 저장소(repo)를 사용합니다. POCU 아카데미에서 지원하는 무료 깃 저장소 서비스는 아래와 같습니다.

- 깃허브(GitHub)
- 깃랩(GitLab)

본인의 입맛에 알맞은 깃 저장소를 고르세요. 잘 모르시겠으면 가장 흔히 사용하는 깃 저장소는 깃허브이니 그냥 그걸 고르세요 ^_^

웹 기반 비공개 깃 저장소를 만드는 방법은 이 가이드([링크](#))를 참고해주세요. 하지만 반드시 명심할 점이 하나 있습니다! 반드시 **비공개(private)** 저장소를 만드셔야 합니다. **공개(public) 저장소를 사용하실 경우 부정행위로 간주하여 0점 처리합니다.**

IV. 빌드봇을 내 깃 저장소에 초대하기

밤낮 가리지 않고 여러분이 제출한 과제를 채점해 주는 근면 성실한 로봇이 바로 빌드봇입니다. 하지만 빌드봇이 채점을 하려면 여러분의 코드를 볼 수 있어야겠죠? 따라서 여러분의 깃 저장소에 빌드봇을 초대해주세요.

- 깃 저장소에 다른 유저(빌드봇)를 초대하는 방법은 이 가이드([링크](#))를 참고해주세요.

- 본인이 사용하는 저장소에 맞게 빌드봇을 초대합니다.
 - 깃허브
 - 'full name': **pocubuildbot**
 - 깃랩
 - 'Select members to invite': **pocubuildbot**
 - 'Choose a role permission': Reporter

V. 깃 저장소 설정 마무리

저장소 클론하기

- 소스트리를 실행합니다.
- 새 탭에서 상단에 위치한 Clone 버튼을 누릅니다.
- '소스 경로 / URL(Source Path / URL)' 입력란에 자신의 저장소의 HTTPS URL을 붙여 넣습니다. HTTPS URL은 자신의 저장소에서 확인할 수 있습니다. ('Code' 또는 'Clone' 버튼을 눌렀을 때 'Clone with HTTPS' 아래 나오는 주소)
- 사용자 인증을 진행합니다.
 - 깃허브
 - 깃허브 사용자는 깃허브 계정 비밀번호 대신 Personal Access Token을 사용해야 합니다.
 - 토큰 생성 방법은 이 가이드([링크](#))를 참고해주세요.
 - 자신의 깃허브 계정의 Username과 토큰으로 로그인합니다. 토큰은 Password 란에 입력하시면 됩니다.
 - 다른 깃 저장소
 - 자신의 저장소 계정의 Username과 Password로 로그인합니다.
- 목적지 경로(Destination Path)를 바꾸고 싶으면 바꿔도 됩니다.
- 클론(Clone) 버튼을 누르면 잠깐의 시간이 흐른 후 완료됩니다.
- 소스트리를 잠시 내버려 두고, 파일 탐색기를 열어 방금 클론한 폴더로 갑니다. 비어 있는 디렉토리(또는 .git이라는 숨겨진 폴더)를 볼 수 있을 겁니다.
 - (중요) 운영체제가 맥이신 분의 경우 클론한 폴더를 열고 shift +command + . 키를 입력해줘야 합니다.

클론한 폴더에 템플릿 집어넣기

- 웹 브라우저를 열고 <https://github.com/POCU/COMP2500StarterPack>으로 이동합니다.
- 'Code' 버튼 클릭 > 'Download ZIP' 선택
- 다운로드한 파일의 압축을 앞서 클론한 폴더 안에 풉니다. 클론한 폴더 내부 **최상위에** 다음과 같은 폴더 및 파일들이 보여야 합니다. 예: 깃 리포를 클론한 폴더의 경로가 'C:\ProgrammingStudy'라면 'C:\ProgrammingStudy\Assignment1'과 같이 압축이 풀려야 합니다.

```
.git (이 폴더는 숨겨진 폴더라 안 보일 수 있습니다.)
Assignment1
Assignment2
Assignment3
Assignment4
Lab1
...
Lab11
.gitignore
LICENSE
README.md
```

- 다시 소스트리로 돌아갑니다. '스테이지에 올라가지 않은 파일'에 보면 수많은 파일들이 보일 겁니다. '모두 스테이지에 올리기(Stage All)' 버튼을 눌러 주세요.
- 커밋 메시지를 적절히 입력하세요. '-에 바뀐 내용 즉시 푸시' 옆 체크박스를 체크하고 '커밋(Commit)' 버튼을 누릅니다. (처음에 이름과 이메일을 입력하라는 창이 뜰 수 있습니다. 입력을 완료하고 다시 '커밋(Commit)' 버튼을 눌러서 완료해 주세요)
- 위 두 단계(4와 5)가 바로 실습 및 과제를 제출하는 방법입니다. 잘 모르겠다고요? 걱정 마세요! 우린 이 과정을 VIII. 코드 제출 및 채점 받기에서 다시 할 거예요.

VI. '나의 POCU'에서 내 정보 업데이트하기

- [POCU 웹사이트](#)에 로그인 합니다.
- '나의 POCU'로 이동합니다.
- 현재 진행 중인 수업으로 이동합니다.
- 조금 아래로 스크롤하여 '내 정보'를 찾습니다.
- '깃 저장소 주소': 'V. 깃 저장소 설정 마무리'에서 클론할 때 사용했던 HTTPS URL을 입력합니다.
- 저장을 누르고, 제대로 저장되었는지 다시 한번 확인합니다.

슬랙 가입하기

혹시 아직 슬랙(Slack)에 가입하지 않으셨나요?

'나의 POCU' 코스 페이지에서 '슬랙 방 가입하기' 버튼을 눌러서 가입해 주세요.

이제부터 이 페이지에서 '토론하기' 버튼을 눌러서 슬랙 방으로 이동할 수 있습니다.

VII. 간단한 프로그램 작성하기

이 부분이 오늘 실습에서 테스트 및 채점을 받을 있는 부분입니다. 아주 간단한 두 정수를 더하는 코드를 작성해 보겠습니다.

우선 IntelliJ를 열어 Lab1 폴더를 엽니다. 이제 `academy.pocu.comp2500.lab1` 패키지 안에 `Lab1.java` 파일을 만듭니다 (`Program.java`가 들어 있는 패키지의 상위 패키지). 만드는 방법을 모르겠다면 이 가이드([링크](#))를 참고해 주세요. 참고로 앞으로 모든 실습/과제에서 학생들이 작성해야 하는 Java 파일들은 `Program.java`의 상위 패키지 안에 만들어야 합니다. (예: `academy.pocu.comp2500.lab1`, `academy.pocu.comp2500.lab3`, `academy.pocu.comp2500.assignment1` 등등) 그렇지 않은 파일들은 빌드봇이 무시하니 꼭 기억하세요.

`Lab1.java` 파일도 만들었다면 이제 그 파일 안에 아래 코드를 추가한 뒤 저장해 주세요.

```
package academy.pocu.comp2500.lab1;

public class Lab1 {
    public static int add(int a, int b) {
        return a + b;
    }
}
```

자, 이제 코드를 제출할 준비가 되었습니다.

VIII. 코드 제출 및 채점받기

이제 코드를 제출하고 채점을 받을 차례입니다.

- 1. 소스트리를 사용하여 내 코드를 커밋(commit) 및 푸시(push) 합니다. ([가이드 링크](#))
- 2. 본 코스의 슬랙 방에 접속합니다. (슬랙 방의 링크는 '나의 POCU' 페이지에서 찾으실 수 있습니다)
- 3. 아무 대화방이나 가서 다음의 빌드 명령어를 **복붙 하지 않고 직접 타이핑**하여 입력합니다. (다음의 명령어는 명령어를 입력한 사람에게만 보입니다)

```
/build lab 1
```

- 4. 빌드를 진행하겠다는 메시지가 나올 것입니다.
 - 혹시 'timeout' 메세지가 나왔나요? 그럼 빌드 명령어(단계 3)를 다시 시도해 보세요. 빌드가 이미 진행되고 있다고 알려줄 것입니다.
- 5. 몇 분쯤 기다리면 빌드봇(@buildbot)이 채점결과 보고서를 보여줍니다.
- 6. 점수를 잃은 부분이 있다면 채점결과 보고서에 있는 링크(예: 테스트 - 1 / 4 통과)를 눌러 왜 점수를 잃었는지 확인한 뒤, 코드를 수정하고 단계 1~5를 다시 반복합니다.
- 7. 100%를 맞았다면 이것으로 실습 1은 끝입니다! 수고하셨습니다!

