

3. 실행 (2)

Object-Oriented Design & Programming
2016년도 봄학기

도입부

- Program = Data + Code

도입부

- Code
 - 어떻게 동작할 지를 계층적으로 기술하는 것
 - Statement
 - Data에게 code란...
 - 유효한 값 생산: `scanner.nextInt()`
 - 값의 유효성 유지: `currentNumber++`
 - 값을 '사용'해 주는 입장: `Say()`
 - 초급 프로그래머에게 중요한 code 관련 키워드
 - 단단함, 깔끔함

도입부

- Program = Data + Code
 - 이 둘은 명백히 서로 구분되는 요소들이지만, 프로그램 안에서 서로 밀접하게 연관되어 있기에 한 쪽의 실수로 인해 다른 쪽이 터질 때가 많습니다.

도입부

- Data를 잘못 사용하면,
code가 정상적으로 실행되지 않습니다.
 - 엉뚱한 값을 넣음
 - 이상한 위치에 값을 넣음(또는 그걸 반대로 당함)
 - 값이 들어가 있던 공간이 증발함

도입부

- 잘못된 code를 사용하면,
원하던 data를 얻을 수 없습니다.
 - 시작 전 준비를 위한 code를 빠뜨림
 - 비교해야 하는데 할당해버림
 - If-else 구조가 허술해서 중간에 새 버림

도입부

- 이미 많이 경험해 본 실수들이지요?
 - 여러분 뿐만 아니라 조교도 종종 틀립니다.
 - 단지 여러분보다 오류 발견/복구 속도가 몇 백 배 빠를 뿐
- 다시 말하면,
여러분이 실수하고 고칠 때마다
조금씩 더 나은 프로그래머가 되어 가고 있다고
생각할 수 있을 것입니다.

도입부

- 네,
우리 모두는, 정도의 차이는 있겠지만,
조금씩 배워 가며 성장하고 있습니다.

도입부

- 이번 학기에...
 - 지금까지 본 내용들을 이해한 경우,
이후 나올 다양한 내용들에 이 논리를 적용해 보세요.
 - 컴퓨터 분야에서 배울 대부분의 개념들은
결국 이 기본 명제에 따라 따라가면 술술 풀리기 마련입니다.
 - 그렇게 생각을 펼쳐 보면서,
자신이 data파인지 code파인지를 가늠해 보는 것도
앞으로의 미래 설계에 작은 도움이 될 수 있을 것입니다.
 - 완벽한 구분법은 아니지만,
변수를 더 투자해서 문제를 푸는 편이면 data파,
메서드를 더 투자해서 문제를 푸는 편이면 code파라고 봅시다.

도입부

- 이번 학기에...
 - 아직 본인이 이 내용들을 이해하지 못 한 경우, 일단 맨 처음에 말한 한 문장만 외워 둔 다음 다양한 초급 문제들을 풀어 보면서 자신만의 관념을 만들기 위해 노력해 보세요.
 - 아직은 늦지 않았습니다.
 - 물론, 하다가 뭔가 막혔거나 더 알고 싶은 게 생기면 지체하지 말고 조교의 도움을 요청해 주세요.

이번 시간에는

- 지난 과제 다시 보기
 - 조교 코드 불러오기 / 내 코드 내보내기
- 조교 코드를 내 코드로 만들기
- Java판 '반복문과 배열'
 - Ctrl + Space 위주로 사용해 볼 예정
- 실습과제#2

지난 과제 다시 보기

- HY-in 학습자료실에서
OODP_HW_1_by_TA.zip을
다운로드해 봅시다.
 - 압축은 풀지 마세요!

지난 과제 다시 보기

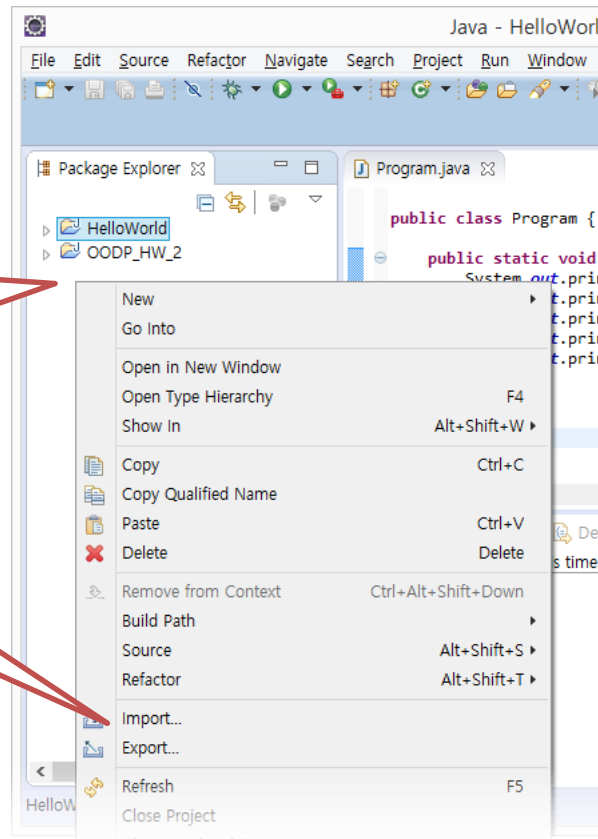
- 방금 받은 압축 파일은...
 - 조교가 만든 실습과제#1-2에 대한 Java project를 Eclipse의 'export' 기능을 사용하여 가져온 것입니다.
 - 이제 이 Java project를 자신의 Eclipse에 'import'해 보도록 합시다.

Eclipse에 Java project를 담는 방법

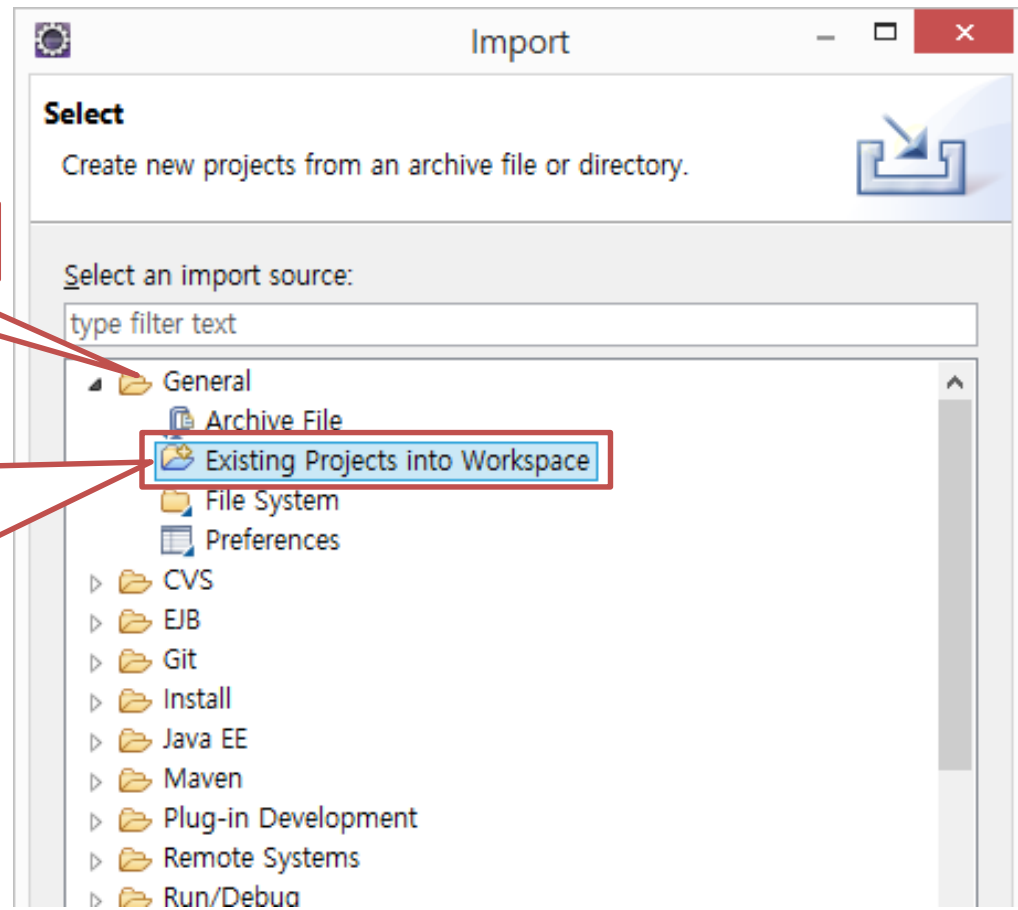
1) Eclipse 켜기

2) Package Explorer
빈 영역을
오른쪽 클릭

3) 뜨는 메뉴에서
'Import...' 클릭



Eclipse에 Java project를 담는 방법



4) 'General' 탭의...

5) 요 놈 선택

주의: 위에꺼 아님!

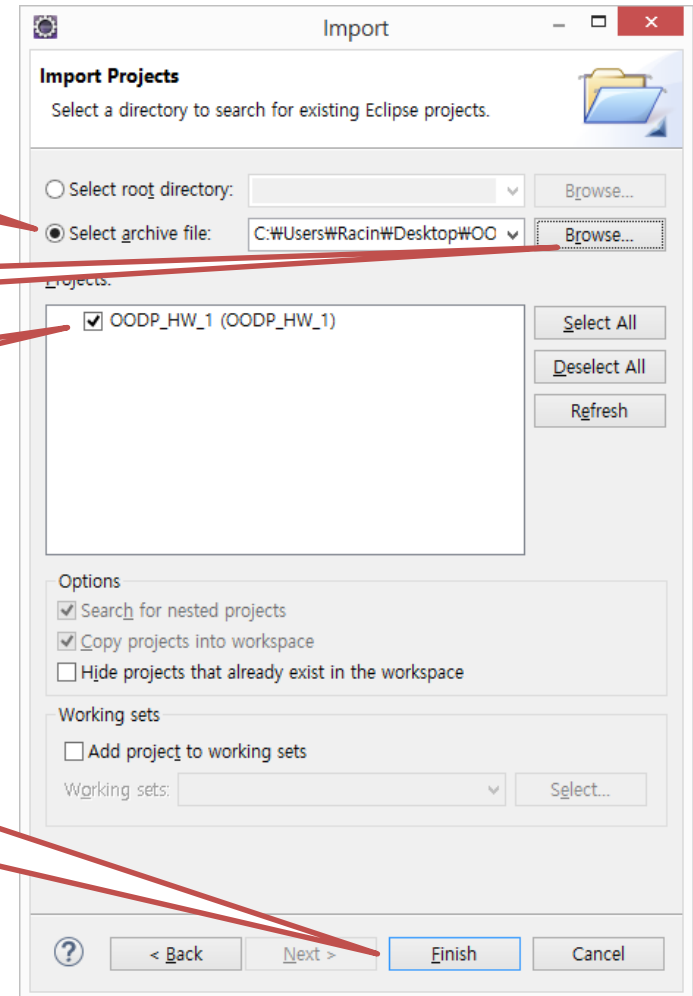
Eclipse에 Java project를 담는 방법

6) 'Select archive file' 선택

7) 'Browse...' 클릭 후 zip 파일 선택

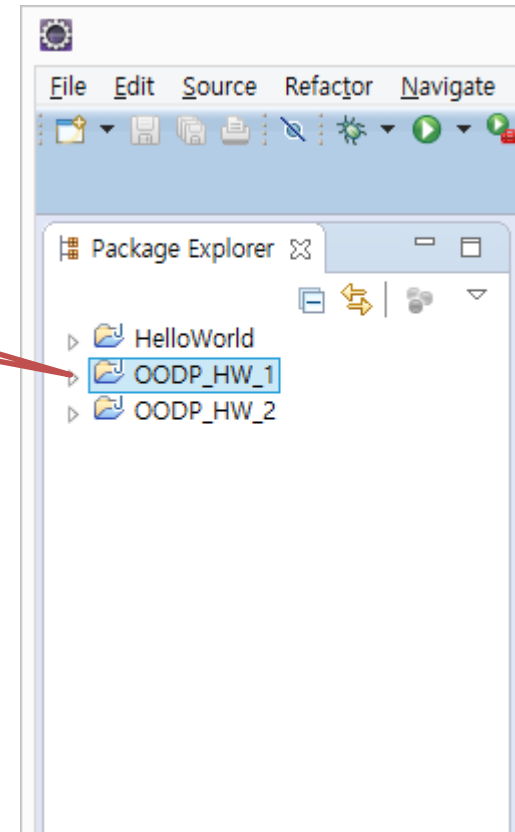
8) Import하려는 Java project가
잘 체크되어 있나 확인
(체크가 안 되는 경우 다다음 슬라이드 확인 ^^)

9) 엔터 키를 강타하여 Finish



Eclipse에 Java project를 담는 방법

A) 원하는 Java project가
Package Explorer에 잘 들어 있으면
성공!



Eclipse에 Java project를 담는 방법

- Import할 때 주의할 점

- 이름 중복 안 됨

- 이전 Java project를 클릭 → F2 눌러서 이름 바꾸기 가능
 - 만약 기존 것을 지우고 새로 import하려는 경우
기존 project 우클릭 → Delete... 선택한 다음 뜨는 창에서

반드시 체크 박스 체크한 다음 OK를 눌러야 함.

- 조교가 부적절한 강조 표현을 쓸 정도로 중요함.
이거 체크 안 하고 지우면 귀찮게 C:\w 탐방을 해야만 함.
➢ 귀찮은 일이니 그냥 손 들고 조교 도움 요청 ㄱㄱ

Eclipse에 Java project를 담는 방법

- Import할 때 주의할 점
 - Import는 기존 Java project 설정을 그대로 가져옴.
따라서, 실습 환경과 다른 곳에서 export해 온 것을
실습 환경에 import하는 경우 문제가 생길 수 있음
 - 이건 현재 우리 수준에서 해결하기 어려우니
일단 이 문제가 뜨면 조교 도움을 요청하시고
집에 가서 JDK, Eclipse 다시 설치하는 것을 추천
 - 이건 매 년 한 두 명씩 겪는 일이니
아마 우리 중에도 곧 범인이 나오게 될 것임

Java project를 내보내는 방법

- Export는 import의 정 반대입니다.
 - 프로젝트 폴더 우클릭 → Export... 선택
 - 맨 위의 General 폴더에서 Archive File 선택
 - Export할 Java project 체크
 - Browse... 버튼을 눌러 내보낼 위치와 이름 선택
 - Finish!
- Import를 잘 한다면 이건 어렵지 않으니 설명은 생략하도록 할게요.

Eclipse에 Java project를 담는 방법

- Import가 다 끝났으면
'조교의 코드'를 잠시 살펴보도록 합시다.

농친 학생들을 위한 슬라이드

- `/** 주석 */`

- 일반 주석과 달리 파란 색입니다.
- 메서드 바로 위에 달려 있습니다.
- 해당 메서드 이름에 마우스를 갖다 대면 파란 주석의 내용이 뜹니다.
- 왠지 멋집니다. 나도 이런 거 하고 싶어집니다.

→ 하면 됩니다. 자주 할 수록 재미가 붙게 될 것입니다.

농친 학생들을 위한 슬라이드

- 추가 변수 쓰기 / 추가 메서드 쓰기
 - 여러분이 변수를 덜 쓸 수록 코드는 지저분해집니다.
여러분이 메서드를 덜 쓸 수록 코드는 길어집니다.
 - 결국 코드 완성에 걸리는 시간이 그만큼 길어집니다.
 - 과제를 시작하기 전에 먼저 그림을 그려 보는 것은,
어떤 변수가 더 필요할지, 어떤 메서드가 더 필요할지
미리 짐작할 수 있다는 점에서
전체 과제를 더 빨리 끝내는 데 큰 도움을 줍니다.
 - 물론 그 퀄리티 및 '코드가 내뿜는 간지' 또한 현저히 증가

농친 학생들을 위한 슬라이드

- 알아먹기 쉽게 적당히 긴 이름 쓰기
 - a, b, c, i, j, k는 그리 좋은 이름은 아닙니다.
 - 여러분이 Dijkstra 아저씨를 정말 좋아하는 경우가 아닌 이상, 각 변수의 목적에 맞는 이름을 붙여 주는 것이 좋습니다.
 - '낙타등 타법'을 쓰면 이름이 길어도 빠르게 쓸 수 있습니다.
 - 어휘가 딸리면 강 한국말 써도 됩니다.
 - 여러분이 나중에 외국 개발자와 협업을 하게 된다면, Alt + Shift + R로 이름을 바꿀 수 있다는 사실만 기억하세요.

농친 학생들을 위한 슬라이드

- 같은 과제라도 다양한 해결책이 있었음
 - 배열을 씀: 당신은 data파일 가능성이 있습니다.
 - switch를 씀: 당신은 code파일 가능성이 있습니다.
 - 친구꺼 베낌: 당신은 CEO 적성일 가능성이 있습니다.
 - (아무도 조교한테 적발되지 않았으니 성공)

조교 코드를 내 코드로 만들기

- 이제 우리 앞에 있는 조교 코드를 '내 코드'로 만들어 보도록 합시다.

조교 코드를 내 코드로 만들기

- 이제 우리 앞에 있는 조교 코드를 '내 코드'로 만들어 보도록 합시다.
 - 코드 창 아무 데나 클릭한 다음, Ctrl + Shift + F를 눌러 보세요.

조교 코드를 내 코드로 만들기

- 바뀐 점
 - 중괄호, 띄어쓰기 등이 일괄적으로 바뀌었다.
 - 원래 조교 스타일로 작성되어 있던 코드가 'Eclipse 스타일'로 바뀌었다.
- 여러분이 이전에 작성한 코드도, Ctrl + Shift + F를 누르면 좀 더 보기 좋게 잘 정렬됩니다.

조교 코드를 내 코드로 만들기

- 하지만 이건 '제 코드'가 아니예요...
 - 그래서 오늘 실습에서는,
Eclipse를 커스터마이징해서
아까 그 단축키를 눌렀을 때
여러분이 원하는 스타일의 코드가 나오도록
설정하는 방법을 배워 보려 합니다.

조교 코드를 내 코드로 만들기

- 하지만 이건 '제 코드'가 아니예요...
 - 그래서 오늘 실습에서는,
Eclipse를 커스터마이징해서
아까 그 단축키를 눌렀을 때
여러분이 원하는 스타일의 코드가 나오도록
설정하는 방법을 배워 보려 합니다.
- 그리고, 그 과정 자체가 오늘 과제의 일부입니다.

실습과제#2

- 아마도 우리 실습 마지막 기본 복습 과제
 - 총 세 개의 부분과제로 구성되어 있습니다.
- 이번 과제 설명서는 중요하므로,
조교와 함께 자세히 읽어 볼 예정입니다.
- 각자의 선택에 따라 과제를 수행한 다음,
과제 페이지에 느낀 점을 적어 제출하세요.
 - 제출기한: 오늘 23:59:59까지

오늘 내용 정리

- Eclipse엔 좋은 기능이 많다.
 - Import, export 기능
 - 커스터마이징 기능
- Java의 몇몇 문법은 C와 약간씩 다르다.
 - 그래도 아직은 감내하고 쓸 만한 수준
- 새 단축키: Ctrl + Shift + F, 낙타등 타법