

2. 실행 (1)

Object-Oriented Design & Programming
2016년도 봄학기
(조교 결막염+목감기 기념 에디션)

도입부

- OOP는 '복잡한 프로그램'에 유리합니다.
- 복잡한 프로그램이란,
'실행이 복잡한' 프로그램을 말합니다.
- 그래서 우리는, 앞으로 몇 주에 걸쳐,
'복잡한 실행'을 구성하는 방법을
천천히 배워 보려 합니다.

지난 시간 내용 잠시 회상

- 우리가 만든 첫 클래스의 이름
→ Program
 - 말 그대로 '프로그램'입니다.
 - 프로그램 실행의 시작점인 `main()`이 있었습니다.
 - 다른 메서드를 실행하는 `code`가 있었습니다.
 - (해 보진 않았지만) 실행에 쓰일 `data`도 들어갑니다.

지난 시간 내용 잠시 회상

- 잠시 상상해 봅시다.
 - 우리의 `main()`은,
`println()`을 호출함으로써
원 목적이었던 'Hello, World! 출력'을 수행했습니다.
 - `println()`은 `main()`에 의해 호출되었습니다.
 - 그렇다면, `main()`은 누가 호출할까요?
 - Windows같은 **OS**가 호출한다고 보면 적당합니다.

지난 시간 내용 잠시 회상

- 관점을 바꾸어서...
 - 우리의 Program class는,
System.out의 println()을 사용하여
원 목적이었던 'Hello, World! 출력'을 수행했습니다.
 - System.out은 Program class에 의해 **사용**되었습니다.
 - System.out은 System class가 **소유**합니다.
그러나, 권한 설정이 public으로 되어 있으므로
Program class도 **볼** 수 있습니다.

지난 시간 내용 잠시 회상

- 관점을 바꾸어서...
 - 우리의 Program class는,
System.out의 println()을 사용하여
원 목적이었던 'Hello, World! 출력'을 수행했습니다.
 - System.out은 Program class에 의해 **사용**되었습니다.
 - 비슷한 관점에서 보면,
Program.main()은 Program class에 있습니다.
그리고, Program class와 main()은 public입니다.
따라서, OS는 main()을 **볼** 수 있습니다.
 - OS는 Program class를 **사용**할 수 있습니다.

도입부(반복)

- OOP는 '복잡한 프로그램'에 유리합니다.
- 복잡한 프로그램이란,
'실행이 복잡한' 프로그램을 말합니다.
- 그래서 우리는, 앞으로 몇 주에 걸쳐,
'복잡한 실행'을 구성하는 방법을
천천히 배워 보려 합니다.

도입부

- 언젠가 여러분이
Java로 복잡한 실행을 구성할 때는,
방금 본 몇 가지 단어들,
'소유하다', '보이다', '사용하다'와 같은
OOP의 추가 개념들을 이해해야 합니다.
 - 그래서, 아직 갈 길이 멀기에,
조교가 아파도 오늘 실습은 꼭 해야 합니다.

도입부

- 물론, if문 for문을 틀리면,
방금 말한 추가 개념들을 알아도
하등의 소용이 없을 것입니다.
 - 얼른 더 큰 것을 만들어 보고 싶겠지만,
일단 오늘은 작은 과제부터 시작해 봅시다.
- 네, 오늘은 바로 과제로 넘어갑니다.

실습과제#1

- '기본 복습 과제 2종 세트'
 - 우리 실습을 듣기 위해 필요한 기본 요건 점검
- 총 두 가지 프로그램을 만들어야 합니다:
 - 동방예의지국 프로그램
 - 삼육구 게임을 혼자 하다가 정해진 숫자에서 틀리는 프로그램

실습과제#1

- '기본 복습 과제 2종 세트'
 - 나누어 준 과제 설명서를 잘 읽으면서 모르겠으면 즉시 도움을 요청하면서 해 보세요.
 - 다 되면 조교에게 검사를 요청하세요.
 - 검사가 끝나면 HY-in 과제 페이지에 과제를 수행하며 느낀 점을 적어 제출하세요.
 - 기한: 오늘 23:59:59까지

실습과제#1

- '기본 복습 과제 2종 세트'
 - 우리 실습을 듣기 위해 필요한 기본 요건 점검
 - 총 두 가지 프로그램을 만들어야 합니다:
 - 동방예의지국 프로그램
 - 삼육구 게임
- 우선 조교와 함께
첫 번째 프로그램을 만들어 보도록 합시다.

실습과제#1-1

- 일반적인 과제 수행 순서
 - 과제 설명서 잘 읽기
 - 저 조교가 도대체 나한테 뭘 원하는지 파악
 - 과제 요건이 무엇인지, 내가 뭘 하면 되는지 파악
 - 종이와 펜을 꺼내 그림 그려 보기
 - 그림을 보면서 코드 작성
 - 끝!

실습과제#1-1

- 우선 과제 설명서를 읽어 봅시다.

실습과제#1-1

- 만들 프로그램에 대한 그림을 그려 봅시다.
 - 이번 과제는 순서도를 그려 보면 적절할 듯
- 순서도를 그린다는 것은...
 - 내가 모르는 부분이 정확히 어디인지 찾기
 - 그 부분을 완성하기 위해 필요한 것이 무엇인지 파악하기
 - 이 때 필요한 것은 보통 실습자료 및 과제 설명서에 있음
 - 코드 작성하다 멍해졌을 때
핫식스 한 캔 비우고 돌아오기 위한 세이브 포인트

실습과제#1-1

- 만들 프로그램에 대한 그림을 그려 봅시다.
 - 이번 과제는 순서도를 그려 보면 적절할 듯

-

실습과제#1-1은 워낙 간단해서 금방 끝나니
그림은 조교만 그려 보도록 할게요

기

하트 한 칸 비우고 돌아오기 위한 세이프 포인트

실습과제#1-1

- 자, 이제 그림과 과제 설명서를 보면서
조교와 함께 실제 코드를 작성해 봅시다!

농친 학생들을 위한 슬라이드

- Java에서 키보드 입력을 받는 방법 (1/2)
 - java.util 패키지에 있는 Scanner를 씁니다.
 - GetAge() 안에서 S 치고 Ctrl + Space를 누른 다음 목록에서 Scanner를 골라 엔터를 칩시다.
 - 엔터 치면 파일 맨 위에 `import java.util.Scanner;` 라고 추가됩니다.
 - 이 부분은 오늘은 일단 넘어갑시다.
 - 이제 `Scanner scanner = new Scanner(System.in);`을 적으면 scanner 변수를 통해 키보드 입력을 받을 수 있습니다.

농친 학생들을 위한 슬라이드

- Java에서 키보드 입력을 받는 방법 (2/2)
 - scanner 변수에 .을 찍은 다음 next를 치면...
 - next(): 다음 키보드 입력에서 한 단어를 가져 옵니다 → %s
 - nextInt(): 다음 키보드 입력에서 int값을 가져 옵니다 → %d
 - nextDouble(): %lf
 - ...등등등등 중에서 필요한 것을 골라 엔터 치면 됩니다.
next어쩌구()의 반환값이 바로 우리가 원하는 키 입력이므로
int age = scanner.nextInt();와 같이 코드를 구성해 씁니다.
 - 영어가 된다면, scanner 내용물에 달린 설명들을 찬찬히 읽어 보는 것도 좋겠군요.

농친 학생들을 위한 슬라이드

- static 메서드 (1/2)
 - 아직은, 왜 static을 안 붙이면 오류가 나는지에 대해 명확히 이해할 수 없습니다.
 - 오늘은 그냥 그러려니 하고 넘어갑시다.

농친 학생들을 위한 슬라이드

- static 메서드 (2/2)
 - 다행히도 Eclipse는 Ctrl + Space를 눌렀을 때 여기서 지금 '보이는' 것들만 목록에 띄워 줍니다.
 - GetAge() 또는 Insa()가 main()에서 보이지 않는 경우,
 - 무식하게 강 직접 치면 빨간 줄이 그어집니다.
 - 이 때 Ctrl + 1을 누르면 현명한 Eclipse가 해결책을 보여주며 해결책들 중 가장 위에 있는 것을 고르면 대부분의 상황에선 문제가 아름답게 해결됩니다.
 - 여기선 GetAge() 또는 Insa()를 static 메서드로 바꿔 줍니다.

농친 학생들을 위한 슬라이드

- If, elseif, else 자동 완성 기능
 - sysout 치고 Ctrl + Space 하면 변환되듯, if나 elseif, else를 치고 Ctrl + Space 하면 우리가 원하는 바로 그 코드가 자동으로 나옵니다.
- 탭 키를 통해 코드 내 요충지를 탐색할 수 있습니다.
- 엔터 키를 치면 중괄호 안으로 커서가 이동합니다.
 - 중괄호 안에 녹색 바가 그려져 있는 게 보일 것입니다.

농친 학생들을 위한 슬라이드

- '이름 바꾸기' 기능
 - 이 기능은 현재 컴파일 오류가 없는 경우에만 쓰세요.
 - 어떤 이름이든(변수명, 메서드명, 클래스명 등), 커서를 갖다 대고 Alt + Shift + R을 누르면 그 이름에 네모가 쳐 집니다.
 - 그 때 현란한 Del, BS 연타 후 이름을 바꾼 다음 엔터 키를 딱 치면...
 - 일단 네모가 사라집니다.
 - 그 이름이 사용되던 모든 곳에 바뀐 이름이 적용됩니다. Ctrl + Z를 눌러 취소할 때도 모든 변경 사항이 복구됩니다.

오늘 내용 정리

- 순서도를 그릴 수 있다면,
코드로 옮길 수 있다.
 - 키보드 잡기 전에 펜부터 잡자.
- Java 구문 문법은 C와 비슷하다.
 - Ctrl + Space를 통해 더 쉽게 쓸 수 있다.
- 새 단축키: Ctrl + 1과 Alt + Shift + R

실습과제#1-2

- 이제 여러분 차례입니다.
과제 설명서를 잘 읽어 보고,
궁금한 것이 생기면 조교를 찾으면서
자유롭게 과제를 진행해 봅시다.
 - '과제 수행 조건'은 꼭 지켜야 합니다.
 - 반대로, 조건이 적혀 있지 않은 요소들은
얼마든지 자유롭게 해도 좋습니다.
 - 다 되었다 싶으면 조교에게 검사를 요청하세요!