



7-2. Object의 생성과 파괴

창의적소프트웨어프로그래밍
2022년도 여름학기
Racin

이번 시간에는

- 유리수 **object**의 칸권을 조금 더 단단하게 보장해 봅니다.
 - 새 **object**를 도모하는 선언/수식을 적을 때
그 프로그래머가 '반드시' 처음에 담아 둘 Data를 지정해야 하도록 만들어 봅니다
 - '생성자'를 정의해 볼 거예요
 - 사용이 끝난 **object**를 처분하는 방법을 정해 봅니다
 - '파괴자'를 정의해 볼 거예요
 - 사실 유리수 **object**는 워낙 단순해서 특별히 케어할 것이 별로 없긴 해요
 - 컴파일러의 '국선' **생성자** / **파괴자** 제공 기능을 확인하고,
그 기능을 직접 요청하거나 아예 끄는 방법을 확인합니다
- '새 **object**를 도모하는 시점에 칸권 보장이 기대되지 않는 경우'를 짚어봅니다

이번 시간에는

- 유리수 **object**의 칸권을 조금 더 단단하게 보장해 봅니다.
 - 새 **object**를 도모하는 선언/수식을 적을 때
그 프로그래머가 '반드시' 처음에 담아 둘 Data를 지정해야 하도록 만들어 봅니다
 - '생성자'를 정의해 볼 거예요

- 사용이 끝난 **object**를 처분하는
 - '파괴자'를 정의해 볼 거예요
 - 사실 유리수 **object**는 워낙 단순해

- 컴파일러의 '국선' **생성자** / **파괴자**
그 기능을 직접 요청하거나 아예 고

뭔가 거창하게 적어 놓았는데 실제로는 매우 간단해요.

이전 내용(칸권 이야기) 복습같은 느낌만 딱 가져가고
중간고사 다음 시간에 정식으로 정리해 볼게요.

오늘은 추가 슬라이드 설명 없이 현장실습만 하고 갑시다.

- '새 **object**를 도모하는 시점에 칸권 보장이 기대되지 않는 경우'를 짚어봅니다

이번 시간에는

- 아무튼 시작해 봅시다.
 - 방금 전까지 같이 구경해 본 .cpp 파일을 다시 열어주세요
- 강사와 함께 **함수 선언/정의**를 몇 개 적어보면서,
'유리수가 뭔지 미리 정하는' 우리가 고를 수 있는 선택지들을 한 번 확인해 봅시다
 - 직접 따라 적어봐도 좋고, 가볍게 견학해 봐도 좋아요.
단, 그 특성상 일부만 따라 적었다면 칸권이 망할 수 있으니
조금 불안하다면 미리 .cpp 파일 내용을 백업해 둔 다음 참여해 봐요

마무리

- 다양한 선택지가 있었고, 마치 유리수 개념의 창시자가 된 것처럼 `main()` 정의 적는 프로그래머의 자유도를 높이거나 낮출 수 있었습니다.
- 지금은 여러 선택지들을 그냥 주욱 열람해 본 셈이고, 일단 중간고사 본 다음에 각각의 의의를 다시 한 번 정리해 볼게요.
 - 잠시 쉬었다가 오늘 실습으로 넘어가 봅시다