

4주차 노트정리

13조 <소재현, 전대원>

○ 예제 3-1

<전대원> Circle 클래스와 donut 객체가 나뉘지는 것을 보고 클래스와 객체는 공존하는 것인가에 질문

<소재현> 클래스를 정의한 후에도 객체를 생성하지 않을 수 있다 클래스는 객체를 생성하기 위한 설계도 역할을 한다

○ 예제 3-2 <전대원> int main(){ 다음 부분에 객체에도 데이터형을 결정해줘야 한다고 판단 후 Double rect 으로 처리했다가 계속된 오류 발생

<소재현> 객체는 클래스의 데이터형식을 그대로 따라가지 않을까 하여 double을 지운 후 실행해서 정상적으로 결과가 나옴

○ 대화 주제 : 위임 생성자를 사용하는 이유

<소재현> 코드를 간략하게 하고

데이터 초기화를 편리하게 시키기 위해서 사용함

초기화를 하지 않을 시 코드 쓰레기 값이 나온다

<전대원> 많이 만들어진 생성자의 중복을 피하기

위해서 위임 생성자를 사용한다

단순히 생성자를 여러 타입으로 사용하는 것이

낫지 않을까에 대한 의문점

○ 예제 3-6 <Rectangle 클래스 만들기>

<소재현> 클래스 선언 까지는 잘 처리했지만

클래스 구현을 할 때 생성자의 Rectangle이 아닌

클래스 선언에 없는 새로운 멤버 함수를 사용하여서

인식이 안되는 오류 발생하였다

<전대원> main함수 전에 클래스를 구현하는 파트에서는 반드시 생성자가 나오거나, 클래스 선언했던 멤버함수가 나와야 한다, Rectangle() 의 괄호안에 들어가는 함수 개수만 달라져도 새로운 생성자로 인식하는 것을 확실하게 익혔음

○ 예제 3-7 & 소멸자 <전대원, 소재현>

소멸자 특징 : 매개변수 없이 사용 가능하고, 클래스에서 딱 한 개만 사용 할 수 있다
~를 붙여 사용하고 기본 출력의 '역순'으로 나오게 된다

○ 예제 3-3의 소스를 헤더 파일과 cpp 파일로 분리하여 작성한 사례

<소재현> 분리하지 않고 하나의 cpp로 만든 후
.h circle.cpp main.cpp 로 분리 후 실행 오류

<전대원> #include "Circle.h" 의 헤더 파일 입력을 통해
오류 해결

○ 클래스 선언부, 클래스 구현부, 실행 파트의 구분

헤더파일 (.h) 구현파트의 (.cpp) main 실행파트(.cpp)

알게된 내용 : 초반부의 내용에서 클래스를 선언하는 곳과
멤버함수, 멤버함수를 연결해주는 클래스 구현부를 굳이 선
을 그어 구분하는 이유가 의문이었으나, 후반부 내용에서 다
른 파일로서 구분해서 활용하게 된다는 것으로 고민 해결