1. Circle 클래스 객체 생성 및 활용

우선 클래스의 활용을 오랜만에 하였기에 클래스 선언 부분을 눈에 먼저 익혔다. circle 클래스를 선언할 때 객체를 여러개 사용해야하나 싶었지만 객체지향 프로그래밍의 목적과 맞게 객체를 최대한 적게 쓰는 쪽으로 작성하였다. 저번에 사용한 getArea 함수를 사용해 선언을 하고싶었다. 교수님이 주신 pdf에는 '::' 연산자를 사용하여 getArea 함수를 선언하였는데 '::' 연산자에 대해서 잘 모르겠어서 조원과 함께 토론하였다. 범위 지정 연산자라는 사실을 알게 되었고, Circle이라는 class 안에 getArea 함수를 사용한다는 범위를 지정해 주었고, 각각 인수를 설정해서 도넛과 피자의 면적을 구하였다. 구하는 과정 중 area 변수의 재정의가 가능한지 궁금해서 토론 후 해봤더니 되었다. 마무리하는 과정중 class에 public 선언을 하지 않아서 결과 값을 도출해 낼 수 없었다는 사실을 발견하였다. class에 public 선언이 되어야 class 안에 있는 내용을 끌어다가 쓸 수 있다는 사실을 알게 되었다.

2. Rectangle 클래스 객체 예제

앞서 했던 Circle 클래스와 하는 방법이 크게 다르지 않아서 둘 다 막힘 없이 잘했다. Rectangle 클래스에 getArea의 범위를 지정하는 생성자 선언 부분에 객체의 타입을 지정하지 않아서 오류가 생겼었지만 토론 후 double이라는 타입을 선언하니 문제없이 실행되었다.

3. 위임 생성자 예제

위임 생성자 예제를 푸는데에는 문제가 없었다. 위임 생성자를 생성함으로써 어떤 이득을 얻을 수 있는지 토론해본 결과 위임 생성자를 통해 생성자에 미리 인수 값을 넣어줌으로써 나중에 class에 곁들여 사용할 객체들의 인수를 선정할 필요가 없다는 결론을 내렸다. 객체지향 프로그래밍의 주요 목적인 편리성과 간단성을 생각해봤을 때 위임생성자는 나중의 인수 선언을 생략하면서 코딩의 줄이 많이 줄어들 것이 편리한 점이라고 생각했다.

4. 객체 생성자의 유무

생성자를 생성할 때 인수의 값을 ()안에 포함한 생성자만을 선언을 한 상태에서, 생성자에 값을 넣은 pizza라는 객체는 오류가 없지만, 생성자에 값을 넣지않은 donut이라는 객체는 컴파일 오류가 발생 될 것이라는 결론을 내렸다.

5. 3-6 rectangle 클래스 예제

isSquare 함수마저 bool 타입의 생성자를 선언해서 만들어야 하는 것이 조금 어려웠다. 그 외에도 Rectangle 생성자를 세 번이나 만들어야한다는 점에서 이 경우에 더 편리하게 할 수 있는 방법이 없나 생각을 해보게 되었다.

6. 지역객체와 전역객체의 생성과 소멸 과정

지역객체가 제일 먼저 실행이 되고, 생성자 순서를 따라 반지름이 1인 원 먼저 생성이 된 후에 main 함수 내의 원이 실행이 되고, f함수의 원이 실행이 된 후에 그 뒤 소멸자는 거꾸로 f함수, main 함수 반지름이 1인 원 지역객체 순으로 소멸 될 것이라고 생각했다.