

4주차 cpp

202004123 최진형

202004151 백승원

1. 클래스 선언부 작성은 했지만 구현부 작성에서 형식을 몰라 막힘
2. 갑자기 cout에서 << 인지 >> 인지 헷갈려서 안됨.
3. 출력문에서 축약을 하려고 했다, -> cout << donut.getArea(); 이렇게 코드축약을 하려 했는데 어떤 이유에서인지 실행이 되지않아서 double donutArea 변수를 선언 후 대입해서 cout << donutArea 로 출력하니 잘 나왔음. << 다시 해보니 축약이 됨...!! 무언가 오류가 있었던것 같음
4. Rectangle 예제를 실습해보니 확실히 circle 예제를 처음 할 때보다 쉽게 코드가 짜여짐.
예제에는 rectangle 클래스의 width, height 매개변수들이 Int 값으로 쓰여있길래, 본인은 double 의 실수형태로 선언하여 rect 객체에서 width, height 값을 실수로 넘겨주어도 소수점까지 계산이 되도록 구현함.

클래스

- 객체를 만들어내기 위해 정의된 설계도, 틀
- 클래스는 객체가 아님. 실체도 아님
- 멤버 변수와 멤버 함수 선언

객체

- 캡슐화(encapsulation), < 객체의 특성
- 일부분만 외부에 보이며, 상태와 행동으로 구성되어있음
- 객체는 생성될 때 클래스의 모양을 그대로 가지고 탄생
- 멤버 변수와 멤버 함수로 구성
- 메모리에 생성, 실체(instance)라고도 부름
- 하나의 클래스 틀에서 찍어낸 여러 개의 객체 생성 가능
- 객체들은 상호 별도의 공간에 생성

생성자

- 객체가 생성되는 시점에서 객체에게 필요한 초기화를 위해 자동으로 호출되는 멤버 함수
- 반드시 클래스 이름과 동일한 멤버 함수여야함!
- 딱 한번만 호출된다!
- 리턴 타입 x

소멸자

- 객체가 소멸되는 시점에서 자동 으로 호출되는 함수.
- 딱 한번만 자동 호출,!! 임의로 호출할 수 없다
- ~ 를 붙여서 코딩.

생성자, 소멸자의 원리를 수업시간에는 잘 이해하지 못해서 추후 검색을 통해 추가로 이해하였음.

-> 생성자가 하나도 작성되어 있지 않은 클래스에는 기본적으로 컴파일러가 기본생성자를 만들어주지만 클래스에 생성자가 하나라도 선언된 경우에는 컴파일러가 기본 생성자를 자동으로 생성해주지 않는다고 생각 하였음.

제일 나중에 생성된게 가장 먼저 소멸된다는 개념을 이해함.

지역객체 = 함수내에 선언된 객체, 함수가 종료되면 소멸된다

전역객체 = 함수의 바깥에 선언된 객체, 종료할때 소멸된다

실행순서

프로그램이 로딩될때 가장먼저 전역객체부터 실행 -> main 함수의 main 객체들 실행 -> f() 지역 객체 실행
소멸순서 -> 실행순서의 역순

접근 지정자

- 기본적으로는 private ()
- public 은 모든 다른 클래스들에 허용
- protect -> 본인과 상속받은 자식 클래스만 허용

인라인 함수

- 함수 호출에 따른 오버헤드가 존재하지 않는다 |
- 프로그램의 실행 속도 개선
- 자주 호출되는 짧은 코드의 함수 호출, 시간 대폭 감소

과제

- 156p 12번을 풀어서 과제로 제출.
- 코드를 분리해서 선언부, 구현부, 메인함수, 다 나눠서 코딩.