**Struct vs Class**

1. **구조체와 클래스**- 구조체와 클래스 둘 다 연관 있는 데이터를 묶을 수 있는 문법적 장치
2. **접근제어 지시자  
   - public : 어디서든 접근허용  
   - protected : 상속관계에 놓여있을 때, 유도 클래스에서의 접근허용  
   - private : 클래스 내(클래스 내에 정의된 함수)에서만 접근허용**- 구조체의 경우 접근제어 지시자를 따로 선언하지 않는다면 public으로 선언  
   -> 어디서든 접근이 가능해 main함수에서의 초기화가 가능  
   - 클래스의 경우 접근제어 지시자를 따로 선언하지 않는다면 private로 선언
3. **Type 차이**- 구조체 : 값 타입(Value) => Stack 메모리에 생성된다.  
   - 클래스 : 참조 타입(Reference) => Heap 메모리에 생성된다.

텍스트이(가) 표시된 사진

자동 생성된 설명구조체  
함수안에서 1로 변경을 해도 외부 Main에서 값이 변경되지 않는다. **복사된 데이터는 원본 데이터가 아니기 때문에 복사된 값을 수정하면, 스택에 복사된 값을 변경하게 되는것이다.**(복사본에 입력된 값들은 함수가 끝나면 사라짐)

클래스  
**참조타입으로 힙의 주소를 전달**하기 때문에 값이 아닌 같은 주소가 참조된다. 그래서 Main으로 출력했을 때, **값이 변경되지 않고 원본값을 출력한다.(**변경된 데이터들은 함수가 종료되어도 남아있음)

1. **생성시 차이점**a. 구조체는 생성자를 선언할 수 있으나 반드시 파라미터가 있어야한다.  
   b. 구조체는 상속이 불가능하다.  
   c. 구조체는 필드선언 시 const또는 static으로 선언한 경우에만 초기화가 가능하다.  
   d. new연산자를 사용하지 않고 인스턴스화 할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

1. **메모리절약은 클래스, 속도는 구조체  
   - 인스턴스한다면 가비지 컬렉션(속도)**  
   : 구조체는 스택에 바로 할당되기 때문에 가비지컬렉션이 발생하지 않는다.  
   : 클래스는 인스턴스를 생성할 때마다 힙에 메모리 할당한다.  
   : 클래스는 값을 폐기하기 위해서는 가비지 컬렉션이 필요하다.  
   **- 많은 양의 변수를 가지고 있는 구조체는 No(메모리)**  
   : 참조형식인 클래스는 값들을 가리키는 주소만 스택에 저장하지만 구조체는 가지고 있는 변수들의 값들을 모두 스택에 저장되기 때문에 그 크기만큼 스택의 위치 역시 커지게 된다. 하지만 스택은 크기가 제한적이기 때문에 스택오버플로우가 발생할 수 있다.  
   **- 구조체 사용시기**  
   : 변수의 크기가 작거나, 수명이 짧고, 자주 할당되는 객체는 구조체로 만들어주는게 좋다.  
   ex) Unity : position, color, quaternion, rotation, scale 등