**1. 클래스**

1. **구조체와 클래스 차이**

구조체는 값타입(value), stack메모리에 생성된다.

- 구조체는 생성자를 선언할 수 있으나 반드시 파라미터가 있어야 한다.

- 구조체는 상속이 불가능하다.

- 구조체는 필드선언시 const 또는 static으로 선언한 경우에만 초기화가 가능하다.

\* 구조체는 원본을 복사하여 복사본을 만들어 사용하기에 복사본의 값을 변환하더라도 원본은 수정이 안된다. 즉 독립적인 구조체로서 남게된다.

- new연산자를 사용하지 않고 인스턴스화 할 수 있다.

클래스는 참조타입(reference), 힙메모리에 생성된다.

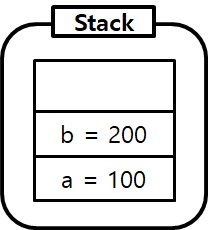
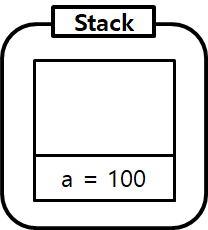
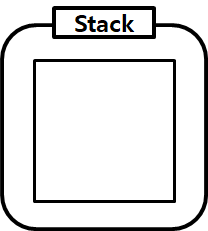
- 클라스는 반드시 생성자를 선언한다.

- 클라스는 상속이 가능하다.

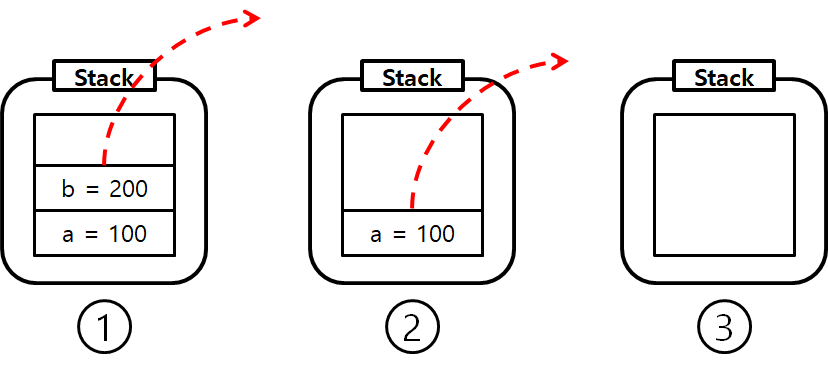
\* 클라스는 참조타입으로 heap의 주소를 전달하기에 값이 아닌 같은 주소를 참조하게 되기에 이에 대한 내용을 변경할 경우 원본을 수정하게 되어 같이 참조하고있는 곳에서 읽어올 시 같은 변경내용을 보게 된다.

1. **값형식과 참조형식**

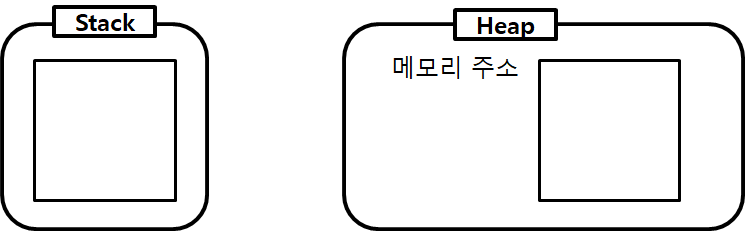
- 값형식의 Stack:

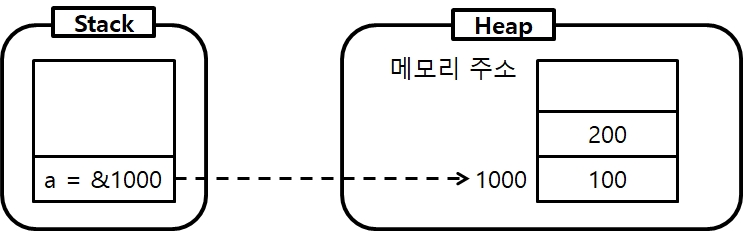


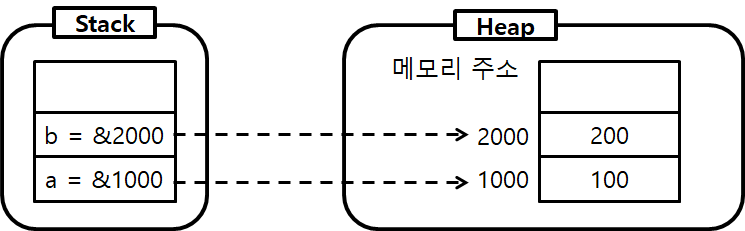
위의 그림과같이 메모리의 stack영역에 쌓이며 블록, 즉 ‘{‘ 와 ‘}’사이에서 데이터가 순차적으로 생성되고 마지막으로 넣은 순서대로 소멸이 되는 방식을 사용하다.

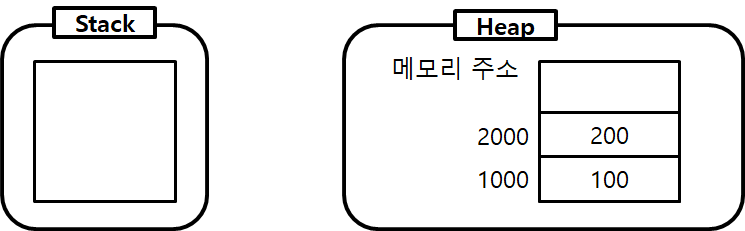


- 참조형식 Heap









참조형식은 실제 데이터를 Heap영역에 저장을 하고 스텍영역에서 이를 불렀을 때 stack안에 있는 값이 아닌 힙 내 실제 원본으로 접속하여 이를 사용하는 방식이다.

1. **얕은복사 깊은복사**

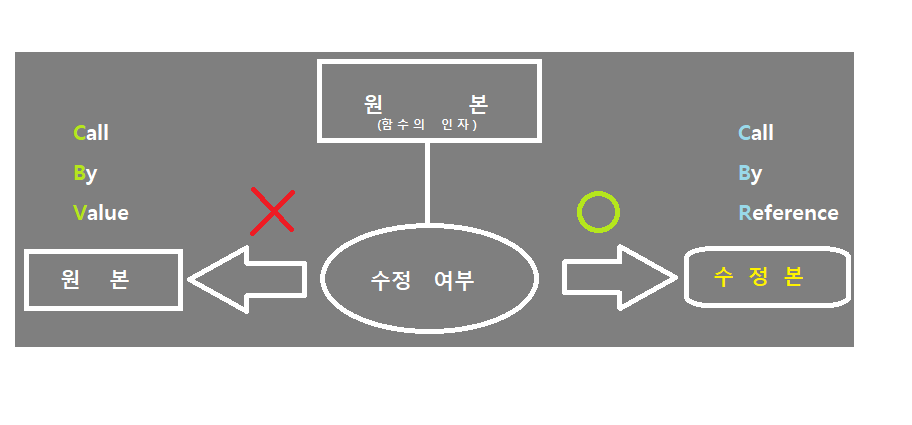
얕은복사

객체를 복사할 때 주소만 복사 즉 참조만 복사하는 방식. (클라스)

깊은복사

객체를 복사할 때 원본을 복사해 새로운 원본을 만들어 이를 참조 또는 가져오는 방식을 말한다. 기본적으로 structure의 경우 깊은 복사를 하며, Class의 경우 따로 DeepCopy함수를 선언하여 복사한다고 볼 수 있다.

1. **Call by Value, Call by Reference**

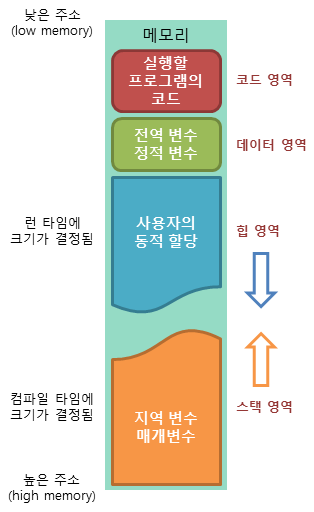


Call by Value와 Cal by Reference는 함수가 인수를 전달할 때 사용되는 방식을 나타낸다.

Call by Value는 함수가 인수로 전달받은 값을 복사하여 처리하는 방식으로 전달된 값은 변경을 해도 원본 값은 변경되지 않는다. 즉 값만 복사해 온다고 볼 수 있다.

**2. 메모리**

1. **메모리 구조**

코드 영역: 실행할 프로그램의 코드가 저장되는 영역. 텍스트 영역이라고도 부른다. CPU는 코드 영역에 저장된 명령어를 하나씩 가져가서 처리하게 된다.

데이터 영역: 프로그램의 전역 변수와 정적 변수가 저장되는 영역이다. 프로그램의 시작과 함께 할당되며, 프로그램이 종료되면 소멸한다.

스택 영역: 함수의 호출과 관계되는 지역 변수와 매개변수가 저장되는 영역이다. 스택영역은 함수의 호출과 함깨 할당되며, 함수의 호출이 완료되면 소멸한다.

힙 영역: 사용자가 직접 관리할 수 있는 메모리 영역이다. 힙 영역은 사용자에 의해 메모리 공간이 동적으로 할당되고 해제된다. 힙영역은 메모리의 낮은 주소에서 높은 주소의 방향으로 할당된다. C또는 C++에서는 사용자가 이 메모리를 직접 관리할 수 있으며, 잘못 관리하면 언더플로우를 일으킬 수 있다. C#에서는 GC 즉 가비지 콜렉터가 자동으로 돌아가기에 이를 관리하는 수고를 덜어준다.

1. **각 영역의 저장되는 데이터와 접근범위, 생존범위**
2. **스택영역**
3. **힙영역**
4. **데이터영역**

**3. 가비지 콜랙터**

[**https://velog.io/@cedongne/C-%EA%B0%80%EB%B9%84%EC%A7%80-%EC%BB%AC%EB%A0%89%ED%84%B0**](https://velog.io/@cedongne/C-%EA%B0%80%EB%B9%84%EC%A7%80-%EC%BB%AC%EB%A0%89%ED%84%B0)

1. **가비지 콜랙터에 대상**
2. **가비지 콜랙터에 메모리를 수거하는 알고리즘 : 마크/스위프/컴백션**
3. **가비지 콜랙터가 메모리를 관리하는 기법 : 세대관리**