

[Java] Primitive Type, Wrapper Class

Java의 변수 타입에는 기본형(Primitive Type)과 참조형(Reference Type) 이 있습니다.

기본형은 short, int, long, float, double, byte, char, boolean 으로 총 8개가 지원 됩니다.

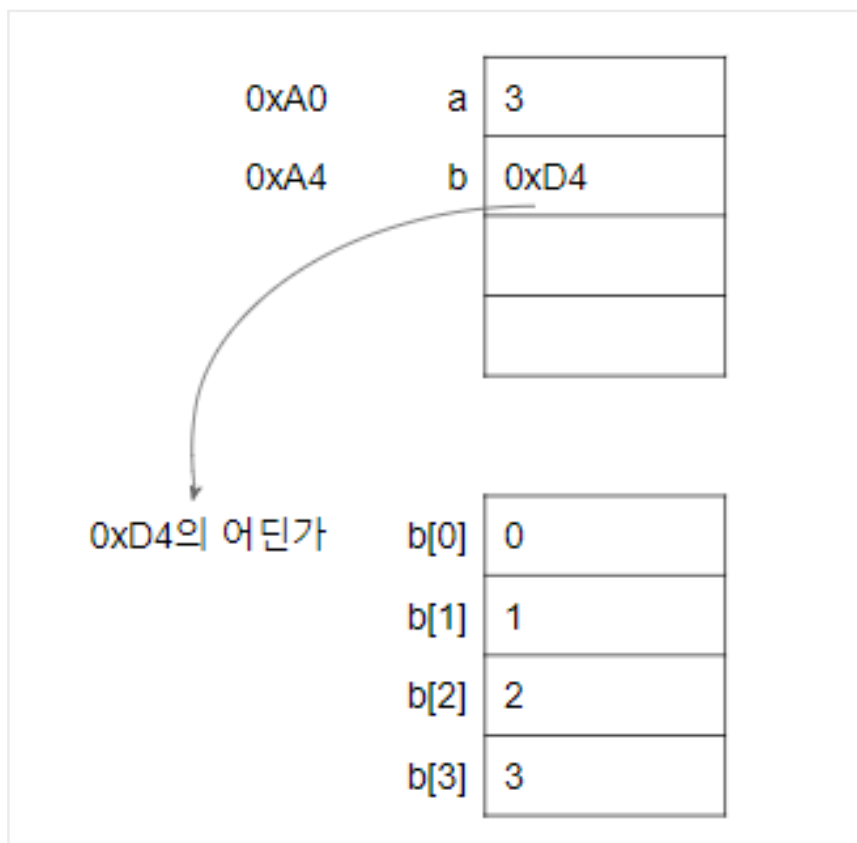
기본형과 참조형의 차이

기본형 변수는 값을 그대로 저장합니다.

반면 참조형 변수는 객체의 레퍼런스를 저장합니다.

```
int a = 3; // 기본형
```

```
int b[] = new int[]{1,2,3} //참조형
```



기본형은 사이즈가 고정되어 있기 때문에 해당 변수의 메모리 공간에 그대로 값을 할당하면 됩니다.

그런데 참조형에는 객체가 할당되어야 하는데, 그 사이즈가 유동적입니다.

그래서 바로 값을 할당 할 수 없고 힙(Heap)영역에서 별도의 메모리 공간을 차지하여 그곳에 할당한 뒤 변수에는 해당 힙 영역의 주소값을 가리키게 합니다.

? 그러면 왜 기본형과 참조형 두가지의 개념이 있는걸까?

기본형도 참조형처럼 레퍼런스를 참조하게 되면 개발자는 기본형과 참조형의 차이를 두지 않고 개발할 수 있을거 같은데 왜 두 개념을 구분했을 까 라는 생각이 드실 수 있습니다.

우선 기본형을 참조형처럼 레퍼런스로 참조하는 방식을 사용하면 기본형으로 사용했을 때에 비해 참조 비용이 증가하게 됩니다.

변수가 몇개 안되면 성능 차이가 안느껴질 수 있지만 `int[]` 의 사이즈가 매우 크다고 가정하면 모든 요소를 참조할 때 마다 힙 영역의 메모리 주소를 또 다시 참조해야 한다고 하면 비용 차이가 커질 수 있습니다.

이렇기 때문에 기본형으로 처리할 수 있는 것은 기본형으로 쓰는 것이 나을 것입니다.

래퍼클래스(Wrapper Class)

Java는 객체지향 언어이다. 모든 것을 객체로 다룹니다.

문제는 기본형은 객체가 아니라는 것입니다. `List`에 `int`를 넣고 싶다고 해서 `List<int> a;` 와 같이 선언이 불가능 합니다.

"저게 왜 안돼?" 라는 생각이 들 수있는데, 모든 것을 객체로 다루는 Java에서 기본형까지 절너식으로 사용할 수 있게 하려면 그 구현이 매우 복잡했을 것이여서 그렇지 않을까 싶습니다.

이럴 때 사용하는 것이 래퍼클래스 입니다. 래퍼 클래스는 기본형을 객체로 한번 감싼 "클래스" 입니다.

이것 때문에 기본형을 객체로 다루는 것이 가능합니다.

ex) `public class Integer{ private final int value;}`

위의 래퍼 클래스를 사용하게 되면 아까 안됐었던

`List<int> a;` 를 `List<Integer> a;` 로 구현 가능합니다.

여기서 질문이 생길 수 있습니다.

"그럼 기본형 대신 항상 래퍼클래스를 쓰면 되는거 아닌가?"

하지만 비용문제가 크기 때문에 필요할 때만 사용하는것이 좋습니다.

