

- 1, **가독성, 재사용성**, 무 결성, 순차지향보다 더 짧고 읽기 쉬운 프로그래밍이 가능
- 2, **객체** = instance - new ObjectName()
속성 = 필드/멤버, 클래스 내에 멤버변수/필드로 선언할 수 있다.
기능 = 메서드
- 3, 추상클래스는 **객체생성**이 불가능하다.
- 4, 인터페이스는 오로지 **추상메서드, 상수필드**, 디폴트 메서드, 정적 상수 필드, 정적 메서드만을 가진다.
- 5, 인터페이스는 자식클래스에서 상속받을 때 자식클래스의 선언부 뒤쪽에 **implements<Interface Name>**을 선언한다.
- 6, 자식클래스는 인터페이스에서 선언한 추상메서드들을 전부 강제로 **Implements** 해야한다.
- 7, 인터페이스는 일반클래스에서 **다중상속**이 가능하다.
- 8, Class 영역 : (**클래스, interface, static변수, static메서드**)정보 등이 저장된다.
- 9, Call Stack 영역 : **일반메서드**가 사용하는 영역
 - FIFO(First In First Out) : 큐
 - LIFO(Last In First Out) : 스택
- 10, Heap 영역 : 동적으로 할당하여 사용할 수 있는 메모리 **인스턴스**가 가리키는 스택의 주소를 저장한다.
- 11, 클래스에 사용시 : **상속**
필드에 사용시 : **값 변경**
메서드에 사용시 : **오버라이드**
- 12, 인터페이스는 크게 **리스트, set, map**으로 나뉜다.
 - 12-1 : **리스트**
 - 12-2 : **Set**
 - 12-3 : **Map**
- 13, 모든 Exception관련 클래스의 최상위 부모 클래스는 **Exception** 클래스이다.
- 14-1 : **Thread Class**
14-2 : **Runnable Interface**
- 15-1 : **바이트**
15-2 : **문자**
- 16-1 : **Connection**
16-2 : **Statement/PreparedStatement**
16-3 : **ResultSet**