

## 1. static

---

1. static은 해석하면 정적인 이라는 뜻을 가지고 있다.
2. Java에서 정적이다라는 것은 메모리에서 삭제되지 않는 것을 의미한다.
3. static 키워드를 이용하면 프로그램 시작부터 종료까지 삭제되지 않는 변수나 메소드를 만들 수 있다.

## 2. 인스턴스 변수와 메소드

---

1. 지역변수는 선언된 {}블록 안에서만 사용이 가능하다.
2. 인스턴스 변수와 인스턴스 메소드는 객체를 생성했을 때 사용가능한 변수와 메소드. 일반적으로 클래스에 선언되어 있는 변수와 메소드를 의미함.
3. 객체가 메모리에 저장되어 삭제되기 전까지 인스턴스 변수와 인스턴스 메소드를 사용할 수 있다.

## 3. 클래스 변수와 메소드

---

1. 클래스 변수와 클래스 메소드는 어플리케이션 시작부터 종료까지 메모리에서 삭제되지 않고 계속해서 사용가능한 변수와 메소드를 의미한다.
2. 클래스 변수와 클래스 메소드는 static 영역에 저장되고 객체를 생성하지 않고도 사용할 수 있다.
3. 클래스 변수와 클래스 메소드는 static 키워드를 이용해서 선언할 수 있다.
4. 접근제어자 static 타입 변수명;  
접근제어자 static 리턴타입 메소드명() {}
5. 클래스원형(클래스명).변수명 이나 클래스원형(클래스명).메소드명으로 클래스 변수와 클래스 메소드를 사용할 수 있다.
6. 프로그램이 구동될 때 계속적으로 필요하거나 많은 클래스에서 공통적으로 사용되는 변수나 메소드를 static 키워드를 사용해서 클래스 변수와 클래스 메소드로 만들어 준다.

## 4. 제어자의 조합

---

1. 클래스 : public, default 접근제어자, abstract(추상화해주는 키워드), final
2. 메소드 : 모든 접근 제어자, final, abstract, static
3. 멤버변수 : 모든 접근 제어자, final, static
4. 지역변수 : final
5. 제어자의 조합에서 주의할 점

- 메소드에 static과 abstract를 동시에 선언할 수 없다.
- 클래스에 abstract와 final을 동시에 사용할 수 없다.
- abstract 메소드는 private을 사용할 수 없다.
- 메소드에서 private과 final은 같은 의미(확장불가)이므로 둘 중 하나만 선언한다.