

39주차 발표\_민서연

---

# Static

---

# 01 static

## static 이란

static(=정적) 은 고정된 이라는 의미를 내포

Java에서의 static은 메모리에 한 번 할당되어 프로그램이 종료될 때 해제되는 것을 의미

클래스의 변수, 메서드, 초기화 블록에 사용될 수 있다.

# 01 static

## static 변수 / 메서드

여러 인스턴스에서 공통으로 사용하는 변수/메서드를 선언할 때 사용

클래스가 메모리에 로딩될 때 생성되어 프로그램이 종료 될 때까지 유지 된다.

static 메모리 영역에 존재하기 때문에 객체가 생성 되기 전에 이미 할당 되어 있다.

클래스당 하나만 생성되며 클래스 멤버라고도 불린다.

# 01 static

## static 변수 / 메서드

```
1  class Student{  
2      2 usages  
    static int cnt = 0;  static 변수  
3      3 usages  
    Student(){  
4          cnt++;  
5      }  
6  
7      1 usage  
    public static void printCnt(){  
8        System.out.println("Total student: " + cnt);  
9    }  
10 }  
11  
12 ▶ public class Study {  
13 ▶     public static void main(String[] args) {  
14         Student Min = new Student();  
15         Student Moon = new Student();  
16         Student Lee = new Student();  
17  
18         Student.printCnt();  
19     }  
20 }
```

Total student: 3

# 01 static

## Java 메인 메서드

```
public static void main(String[] args)
```

정적 함수

static으로 함수 또는 클래스를 선언했을 경우 해당 객체는 자바가 컴파일 되는 순간 정의 된다.

## static으로 선언하는 이유

변수나 함수를 메모리에 할당하는 방법 → static 영역에 선언하는 방법 / heap 영역에 선언하는 방법

heap영역에 선언 될 경우 Garbage Collector에 의해 메모리에서 정리가 되어 main 기본 함수가 죽음

static으로 선언해서 메모리에 항상 상주하도록 설정함

# 01 static

## Static 객체와 Bean

### Static과 Bean의 공통점

- 공유 가능  
: static으로 선언된 객체와 빈으로 등록된 인스턴스 모두 공유될 수 있는 인스턴스이고 static객체는 클래스의 모든 인스턴스가 공유하고 빈은 스프링 컨테이너를 통해 공유된다.
- 전역적 접근 가능
- 라이프 사이클이 존재  
: 빈은 스프링 컨테이너에 의해 관리 되고 static 객체는 프로그램이 시작되어 끝날 때 까지를 라이프사이클로 가진다.

# 01 static

## Static 객체와 Bean

### Static과 Bean의 차이점

- static 객체는 다른 객체에 의한 의존성 주입이 불가능하지만 빈은 의존성 주입이 가능하다.
- static 객체는 어떤 상태나 동작을 공유하기 위해 사용하고 빈은 객체의 라이프 사이클 관리나 의존성 주입, 효율적인 리소스 관리 등을 위해 사용한다.
- static 객체는 개발자가 직접 선언하고 초기화 하지만 빈은 스프링 컨테이너에 의해 관리된다.

# 01 static

## Spring Util 클래스

### Static을 사용 할 경우

- 종속성이 없고 객체 생성이 필요 없음  
: 추가적인 설정이나 의존성 주입 필요 없이 직접 클래스 이름으로 호출 할 수 있어 간편함
- 일관된 결과 제공할 때 필요  
: 입력 값이 같으면 항상 동일한 결과 값을 반환하는 함수에 적합
- 메모리 사용에 유리  
: 객체 인스턴스를 생성하지 않기 때문에 메모리 효율성이 높음



# 01 static

## Spring Util 클래스

### Bean을 사용할 경우

- 의존성이 있을 때  
: 의존성 주입을 통해 다른 컴포넌트와의 상호작용이나 상태 관리가 필요한 경우 적합.  
(DB 연결, 외부 서비스와의 통신)
- 테스트 용이성  
: 빈으로 등록된 클래스는 spring의 테스트 프레임워크와 통합되어 mocking 이나 다른 테스트 기법을 적용하기 쉬움
- 결과의 일관성이 보장되지 않는 경우  
: 예측 할 수 없는 외부 요인에 의해 결과가 달라질 수 있는 경우 빈으로 만들어 유연한 처리가 가능함. (네트워크 요청 처리 등)

## 02 출처

<https://coding-factory.tistory.com/524>

<https://skorea6.tistory.com/entry/%EC%9E%90%EB%B0%94-static-%EB%B3%80%EC%88%98-%EC%9D%98%EB%AF%B8-%EC%82%AC%EC%9A%A9%EB%B0%A9%EB%B2%95-%EC%9C%A0%ED%9A%A8%EB%B2%94%EC%9C%84>

<https://velog.io/@yulhee741/java-static-%EB%B3%80%EC%88%98>

<https://hu-coding.tistory.com/61>

<https://devparker.tistory.com/10>

<https://bbogle2.tistory.com/entry/Spring-Static-%EA%B0%9D%EC%B2%B4%EC%99%80-Bean%EC%9D%98-%EC%B0%A8%EC%9D%B4>

[https://curiousjinan.tistory.com/entry/util-class-static-vs-spring-bean#google\\_vignette](https://curiousjinan.tistory.com/entry/util-class-static-vs-spring-bean#google_vignette)