



Static의 특징은?

≡ 분류

Java

[Static정의]

Static 키워드를 사용한다는 것은 메모리에 한번 할당되어 프로그램이 종료될 때 해제되는 것을 의미한다.

[Static 메모리]



일반적으로 우리가 만든 Class는 Static 영역에 생성되고, new 연산을 통해 생성한 객체는 Heap영역에 생성된다. 객체의 생성 시에 할당된 Heap영역의 메모리는 Garbage Contror

을 통해 수시로 관리 를 받는다.

하지만 Static키워드를 통해 Static영역에 할당된 메모리는 모든 객체가 공유하는 메모리라는 장점을 지니지만, Garbage Collector의 관리 영역 밖에 존재하므로 Static을 자주 사용하면 프로그램의 종료 시까지 메모리가 할당된 채로 존재하므로 시스템의 퍼포먼스에 악영향을 주게된다.

[Static 변수]

- Static 변수는 클래스 변수이다.
- 객체를 생성하지 않고도 Static 자원에 접근이 가능하다.
- Static 정적 변수 : 메모리에 고정적으로 할당되어, 프로그램이 종료될 때 해제되는 변수

```
public class MyCalculator {  
    public static appName = "MyCalculator";  
  
    public static int add(int x, int y) {  
        return x + y;  
    }  
  
    public int min(int x, int y) {  
        return x - y;  
    }  
}
```

```
MyCalculator.add(1, 2);    // static 메소드 이므로 객체 생성 없이  
MyCalculator.min(1, 2);   // static 메소드가 아니므로 객체 생성후0
```

```
MyCalculator cal = new MyCalculator();
```

```
cal.add(1, 2);    // o 가능은 하지만 권장하지 않는 방법
cal.min(1, 2);    // o
```

예시) "호범" 이라는 이름을 가진 사람이 세상에 존재할 때

```
public class Person {
    private String name = "호범";

    public void printName() {
        System.out.println(this.name);
    }
}
```

위와 같은 클래스를 통해 100명의 Person 객체를 생성하면, "호범" 이라는 같은 값을 갖는 메모리가

100개나 중복해서 생성되게 된다.

이러한 경우 static을 사용하여 여러 객체가 하나의 메모리를 참조하도록 하면 메모리 효율이 더 높아 질 것이다. 또한 "호범" 이라는 이름은 결코 변하지 않는 값이므로 final을 붙여주고, 일반적으로 Static은 상수의 값을 갖는 경우가 많으므로 public으로 선언을 하여 사용한다.

일반적으로 public static final로 활용된다.

```
public class Person {
    public static final String name = "호범";

    public static void printName() {
        System.out.println(this.name);
    }
}
```

위의 코드를 보면, printName이라는 함수 역시 static 키워드가 붙어서 static메소드가 되었는데, static 변수는 static 메소드를 통해 접근하도록 권장되기 때문이다.

[Static 메소드]

Static Method는 객체의 생성 없이 호출이 가능하며, 객체에서는 호출이 불가능하다. 일반적으로는 유틸리티 관련 함수들은 여러 번 사용되므로 static 메소드로 구현을 하는 것이 적합한데, static

메소드를 사용하는 대표적인 Util Class로는 java.util.Math가 있다.

```
//사용 예시 Math
public class Test {
    private String name1 = "천";
    private static String name2 = "호범";

    public static void printMax(int x, int y) {
        System.out.println(Math.max(x, y));
    }

    public static void printName(){
        // System.out.println(name1); 불가능한 호출
        System.out.println(name2);
    }
}
```

두 수의 최대값을 구하는 경우에 Math클래스를 사용하는데, static 메소드로 선언된 max 함수를 초기화 없이 사용한다. 하지만 static 메소드에서는 static이 선언되지 않은 변수에 접근이 불가능한데, 메모리 할당과 연관지어 생각해보면 당연하다.

우리가 Test.printName() 을 사용하려고 하는데, name1은 new 연산을 통해 객체가 생성된 후에 메모리가 할당된다. 하지만 static 메소드는 객체의 생성 없이 접근하는 함수이므로,

할당되지 않은 메모리 영역에 접근을 하므로 문제가 발생하게 된다. 그러므로 static 변수에 접근하기 위한 메소드는 반드시 static 메소드가 되어야 한다.

```
//Singleton Pattern
public class Singleton {
    private static Singleton instance;
    private LazySingleton() {}
    public static synchronized Singleton getInstance() {
        if(instance == null) {
            instance = new Singleton();
        }
        return instance;
    }
}
```

프로그램에서 클래스당 하나의 객체만 사용하도록 유도하는 Singleton Pattern은 static을 적절히 사용한 대표적인 예시이다.

getInstance 함수를 별도의 객체 생성 없이 호출할 수 있도록 static 멤버로 선언했으며, 클래스 변수인 Instance도 static으로 선언해 여러개의 객체가 생성되더라도 동일한 객체만 리터할 수 있도록 했다.

/

[주의사항]

- Out of Memory(OoM)

객체가 제거되어도 static 멤버는 메모리에 남아있기 때문에 코드 상에서 별도로 처리해 두지 않으면 모든 메모리 공간을 static 인스턴스가 차지하는 * 메모리 Leak 이 발생할 수 있다.

```

public class OomClass {
    static Vector oomVector;
    static void addOomVector(int a) {
        oomVector.add(a);
    }
    public OomClass() {
        for (int i = 0; i < 100000; i++)
            addOomVector(i);
    }
}

```

위의 OomClass는 생성 할 때마다 static 변수는 oomVector 벡터 클래스에 0~99999까지 엔트리를 추가하는데 OomClass 한 두 개 정도 생성할 때는 별 문제가 되지 않지만 그 수가 많아지면 메모리에서 차지하는 비율이 높아져 OOM이 발생할 수 있다.




메모리 Leak (메모리 누수 현상)

컴퓨터 프로그램이 필요하지 않은 메모리를 계속 점유하고 있는 현상. 사용한 다음 반환하지 않는 것이 누적되면 메모리가 낭비된다.

Reference

JAVA의 static

절차 지향인 C언어에서 static 변수는 함수 또는 변수를 작성 중인 파일 내에서만 사용하고 싶을 때만 사용하기 때문에 사용법이 비교적 간단했는데 JAVA에서는 객체지향 개념과 연관되어 있어 사용 시 다소

 <https://selfish-developer.com/entry/JAVA%EC%9D%98-static-1>



[Java] static변수와 static 메소드


Java에서 Static 키워드를 사용한다는 것은 메모리에 한번 할당되어 프로그램이 종료될 때 해제되는 것을 의미합니다. 이를 정확히 이해하기 위해서는 메모리 영역에 대한 이해가 필요합니다. [Static의 메모리

 <https://mangkyu.tistory.com/47>



OOM이란 무엇인가요?

OOM이란? 사용자에게 보이는 앱의 비정상 종료는 대부분 예외 발생에 따른 Crash가 그 원인이지만, 기타 주요 원인으로는 Out Of Memory(OOM)를 들 수 있습니다. 애플리케이션이 대량의 메모리를

 <https://support.smrtbeat.com/hc/ko/articles/360000049202-OOM%EC%9D%B4%EB%9E%80-%EB%AC%B4%EC%97%87%EC%9D%B8%EA%B0%80%EC%9A%94->

