

필드 (Field)

▶ 필드 (Field)

✓ 필드 표현식

```
[접근제한자] [예약어] class 클래스명 {
```

```
    [접근제한자] [예약어] 자료형 변수명 [= 초기값];
```

```
}
```

✓ 필드 예시

```
public class Academy {  
    public int temp1;  
    protected int temp2;  
    int temp3;           //접근제한자 생략 시 (default)  
    private int temp4;   //캡슐화 원칙으로 private 사용  
}
```

▶ 필드 (Field)

✓ 필드 접근제한자

구분		해당 클래스 내부	같은 패키지 내	후손 클래스 내	전체
+	public	○	○	○	○
#	protected	○	○	○	
~	(default)	○	○		
-	private	○			

▶ 필드 (Field)

✓ 필드 예약어 - static

같은 타입의 여러 객체가 공유할 목적의 필드에 사용하며,
프로그램 start시에 **정적 메모리(static) 영역**에 자동 할당되는 멤버에 적용

✓ static 표현식

```
public class Academy {  
    private static int temp1;  
}
```

▶ 필드 (Field)

✓ 필드 예약어 - final

하나의 값만 계속 저장해야 하는 변수에 사용하는 예약어

✓ final 표현식

```
public class Academy {  
    private final int TEMP1 = 100; // final 키워드가 붙은 필드명은  
                                   // 모두 대문자로 표기  
    private int temp4;  
}
```

▶ 필드 (Field) - 클래스 초기화 블록

✓ 클래스 초기화 블록

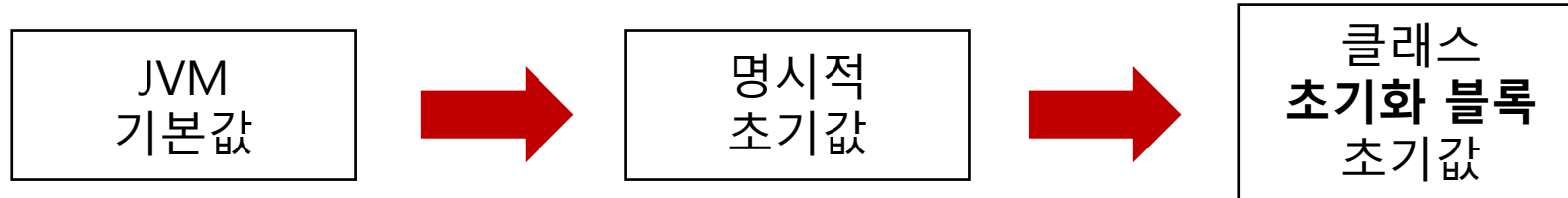
- 인스턴스 블록 ({ }) - 인스턴스 변수를 초기화 시키는 블록으로 객체 생성시 마다 초기화
- static(클래스) 블록 (static{ }) - static 필드를 초기화 시키는 블록으로 프로그램 시작 시 한 번만 초기화

✓ 클래스 초기화 블록 표현식

```
[접근제한자] [예약어] class 클래스명 {  
    [접근제한자] static 자료형 필드1 = 10;  
    [접근제한자] 자료형 필드2 = 20 // 명시적 초기화;  
  
    static{ 필드1 = 30; }  
    { 필드2 = 40; }  
}
```

▶ 필드 (Field) - 초기화 순서

✓ 클래스 변수



✓ 인스턴스 변수

