

01-15 [Java]

 소유자	 종수 김
 태그	

지역변수와 스코프

지역변수 : 특정 지역 내에서만 사용 가능한 변수

{코드블록} 내에서만 사용이 가능, 이를 벗어나면 사용이 불가능

스코프 : 변수가 접근 가능한 범위

스코프의 존재 이유

코드블록내에서만 필요한 변수를 더 넓은 범위의 코드블록에서 선언한다면,

- 비효율적인 메모리 사용
 - 특정 코드블록 내에서만 사용가능한 코드는 코드 블록내에서 선언 및 사용하면 코드 블록이 끝남과 동시에 메모리가 해제되지만, 넓은 범위의 코드 블록에서 선언한다면 해당 코드 블록이 끝날 때 까지 메모리가 유지된다.
 - 물론 GC에 의해 해제되긴함.
- 코드 복잡성이 증가한다
 - 해당 코드가 넓은 범위의 코드블록 내에서 선언되고 좁은 범위의 코드블록에서만 사용된다면, 넓은 범위의 코드블록이 끝날 때 까지 개발자는 해당 변수의 값을 신경 쓸 수 밖에 없으므로 이처럼 비효율적인 스코프로 선언된 코드가 늘어난다면 코드 복잡성이 증가한다.
- 변수는 꼭 필요한 범위로 한정해서 사용하는 것이 좋음.

형 변환

- 작은 범위에서 큰 범위로는 당연히 값을 넣을 수 있다.
 - int → long → double
- 큰 범위에서 작은 범위는 문제를 일으킬 수 있다.
 - 소수점 버림
 - 오버 플로우

- Java에서 작은 범위에서 큰 범위로의 형 변환은 인정해줌.
 - 하지만 내부적으로는 타입을 맞추어 줘야 함.
 - 자동 형변환 / 묵시적 형변환
 - 작은 범위에서 큰 범위로의 형 변환은 캐스팅을 자동으로 진행

▼ 예시

```
doubleValue = intValue;
=
doubleValue = (double) intValue;
```

명시적 형변환

반대로 큰 범위에서 작은 범위로의 대입은 ?

```
double doubleValue = 1.5;
int intValue;
intValue = doubleValue; // 컴파일 오류
intValue = (int) doubleValue; // 명시적 형변환
```

위 경우 double이 표현할 수 있는 범위가 int가 표현할 수 있는 범위보다 크기에 컴파일 오류가 발생하며, 이를 위처럼 명시적으로 형 변환할 경우 값이 누락될 수 있다.

형변환은 변경하고 싶은 데이터 타입을 (int)와 같이 명시적으로 입력

개발자가 직접 형변환 코드를 입력한다고 해서 '명시적 형변환' 이라고 한다.

- 형변환을 한다고 해서 doubleValue값 자체를 바꾸는 것은 아님.
즉 doubleValue의 1.5를 1로 바꾸는 것이 아님. doubleValue는 여전히 1.5

형변환과 오버플로우

만약 형변환시 작은 데이터가 표현할 수 있는 값을 넘어간다면?

- int로 변경되어 들어가는 값이므로, 오버플로우 (-)가 발생

계산과 형 변환

- 같은 타입끼리의 계산은 같은 타입의 결과를 낸다
 - int + int = int

- `double + double = double`
- 서로 다른 타입의 계산은 큰 범위로 자동 형변환이 일어난다.
 - `int + long = long + long`
 - `int + double = double + double;`
-