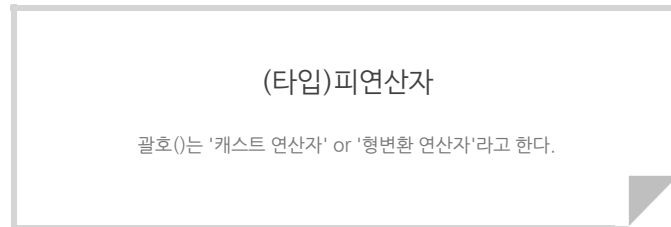


- 형변환(casting): 변수 또는 상수의 타입을 다른 타입으로 변환하는 것.



- ▶ 형변환 연산자는 피연산자의 값을 읽어서 지정하는 타입으로 형변환할 뿐, 피연산자의 변수 값을 바꾸는 것은 아닙니다.
- ▶ 기본형(primitive type)에서 boolean을 제외한 나머지 타입들은 서로 형변환이 가능하다.
- ▶ 기본형과 참조형은 서로 형변환 할 수 없다.
- ▶ 서로 다른 타입의 변수간의 연산은 형변환을 하는 것이 원칙이지만, 값의 범위가 작은 타입에서 큰 타입으로의 형변환은 생략할 수 있다.

● 정수형 간의 형변환

- ▶ 큰 타입에서 작은 타입으로의 형변환 시, 크기의 차이만큼 잘려나가기 때문에 경우에 따라 '값의 손실(loss of data)'가 발생할 수 있다.
- ▶ 반대로 작은 타입에서 큰 타입으로의 형변환 시, 값 손실이 발생하지 않으며 나머지 빈공간은 0 또는 1(변환하려는 값이 음수일 경우)로 채운다. 형변환 후에도 부호를 유지할 수 있도록 하기 위해서이다.
- ▶ Integer.toString(int i): 10진 정수를 2진 정수로 변환한 문자열을 얻을 수 있는 메서드.

● 실수형 간의 형변환

- ▶ 작은 타입에서 큰 타입으로 변환하는 경우, 빈 공간을 0으로 채운다. float에서 double로 형변환 시에, 지수(E)는 float의 기저인 127을 뺀 후 double의 기저인 1023을 더해서 변환하고, 가수(M)는 float의 가수 23자리를 채고 남은 자리를 0으로 채운다.
- ▶ 반대로 double타입에서 float타입으로 변환하는 경우, 지수(E)는 double의 기저인 1023을 뺀 후 float의 기저인 127을 더하고, 가수(M)는 double의 가수 52자리 중 23자리만 저장되고 나머지는 버려진다. 이때, 가수의 24번째 자리의 값이 1이면 반올림이 발생하여 23번째 자리의 값이 1증가하는, 반올림이 발생할 수도 있다.
- ▶ float타입의 범위를 넘는 값을 float으로 형변환하는 경우는 ±무한대(float의 최대값보다 큰 값) 또는 ±0(float의 최소값보다 작은 값)을 결과로 얻는다.
- ▶ float과 double 타입의 변수에 같은 값을 저장해도 정밀도 차이 때문에 서로 다른 값이 저장된다. 저장할 때 이미 값이 달라졌기 때문에, 형변환을 해도 같아지지 않는다.

● 정수형과 실수형 간의 형변환

- ▶ 정수형을 실수형으로 변환: 정수를 2진수로 변환한 다음 정규화를 걸쳐 실수의 저장형식으로 저장된다. 이때, 실수형의 정밀도 제한으로 인해 오차가 발생할 수 있다. 때문에 10진수로 8자리 이상 값을 실수형으로 변환할 때는, float가 아닌 double로 형변환해야 오차가 발생하지 않을 수 있다.
- ▶ 실수형을 정수형으로 변환: 실수형의 소수점 이하 값은 버려진다. 만일 실수의 소수점을 버리고 남은 정수가 정수형의 저장 범위를 넘는 경우에는 정수의 오버플로우가 발생한 결과를 얻는다.

- 자동 형변환: 컴파일러가 생략된 형변환을 자동적으로 추가한다.

- ▶ 산술 변환: 연산과정에서 자동적으로 발생하는 형변환으로 두 타입 중 표현 범위가 더 넓은 타입으로 형변환하여 타입을 일치시킨 다음 연산을 수행한다. 그렇게 하는 것이 값손실의 위험이 더 적어서 올바른 결과를 얻을 확률이 높기 때문이다.
- ▶ 규칙: 기존의 값을 최대한 보존할 수 있는 타입으로 자동 형변환 한다. 표현 범위가 좁은 타입에서 넓은 타입으로 형변환을 하는 경우에는 값 손실이 발생하지 않으므로 두 타입 중 표현 범위가 더 넓은 쪽으로 형변환된다.

byte(1) - short(2), char(2) - int(4) -
long(8) - float(4) - double(8)

- ▶ 표현할 수 있는 값의 범위가 작은 것에서 큰 것으로의 변환은 형변환 연산자를 사용하지 않아도 자동 형변환된다. 예를 들면, float타입의 변수에 int타입의 상수를 저장하는데 아무런 문제가 없다 (int -> float). 하지만 그 반대 방향으로의 변환은 반드시 형변환 연산자를 써줘야 한다.
- ▶ char과 short는 둘 다 2 byte로 크기가 같지만, char의 범위는 양수만 포함되고, short는 반을 쪼개 반은 음수, 반은 양수의 범위를 가지고 있으므로 서로 범위가 달라서 둘 중 어느 쪽으로의 변환도 값 손실이 발생할 수 있으므로 자동 형변환이 수행될 수 없다.
- ▶ 또한 변수가 저장할 수 있는 값의 범위보다 더 큰 값을 저장하려는 경우 형변환을 생략하려면 에러가 발생한다: 'incompatible types: possible lossy conversion from int to byte'. 다만 명시적으로 형변환을 해줬을 경우, 이는 의도적인 것으로 간주하고 컴파일러는 에러를 발생시키지 않는다.

Resource: 자바의 정석