묵시적(암묵적) 형변환

크기가 작은 데이터 타입의 값을 크기가 큰 데이터 타입의 값에 넣을 때 암묵적으로 크기가 큰 데이터 타입으로 변경되는 경우

Ex)

```
int a = 5;
long b = a;
System.out.println(b);
```

```
Console : 5
```

Int의 데이터 타입을 가진 변수 a의 값 5가 **long**의 데이터 타입을 가진 변수 b에 들어가면서 크기가 더 큰 **long**으로 변하게 되었다.

데이터 타입의 형변환 순서

byte -> short -> int -> long -> float -> double
char ↗

명시적 형변환

크기가 큰 데이터 타입의 값을 크기가 작은 데이터 타입의 값에 넣거나 숫자형의 데이터 값과 문자형의 데이 터 값을 변환할 때 사용된다.

Ex)

```
int a = 5;
byte b = (byte)a;
System.out.println(b);
```

```
Console : 5
```

Int의 데이터 타입을 가진 변수 a의 값 5를 더 작은 데이터 타입인 **byte**의 변수 b에 넣기 위해 a를 **byte**로 변환 시켰다.

명시적 형변환 구조

(변환할 데이터 타입)변환할 변수

Ex)

```
(int)a
(float)b
(double)c
(short)d
```

숫자형 데이터 타입과 문자형 데이터 타입의 변환

문자형 -> 숫자형

Ex)

```
String a = "123";
int b = Integer.parseInt(a);
System.out.println(b);
```

Console : 123

String의 데이터 타입을 가진 변수 a가 int로 변경되어 변수 b에 저장되었다.

숫자형 -> 문자형

Ex)

```
int a = 123;
String b = Integer.toString(a);
System.out.println(b);
```

Console : 123

int의 데이터 타입을 가진 변수 a가 String로 변경되어 변수 b에 저장되었다.

과제 1 - 묵시적 형변환, 명시적 형변환 예제 4개 작성

묵시적 형변환 – 1

두 값을 더했을 때 데이터 타입의 값 이상(오버 플로우)가 나올 때 예제

```
short a = 31264;
short b = 30045;

int c = a+b;

System.out.println(c);
```

Consloe : 61309

short의 최대 양수 값은 32,767이다 즉 31,264와 30,045를 더하면 **short**으로 나타낼 수 없다. 따라서 **int**로 형변환 시켜 올바른 값이 나오도록 사용하였다.

묵시적 형변환 – 2

두 값을 뺄 때 데이터 타입의 값 이하(언더 플로우)가 나올 때 예제

```
short a = 32767;
short b = 32767;
short c = 0;

int d = c-a-b;

System.out.println(d);
```

Console : -65534

short의 최소 음수 값은 -32,768이다 즉 0에 32,767와 32,767를 빼면 **short**으로 나타낼 수 없다. 따라서 **int**로 형변환 시켜 올바른 값이 나오도록 사용하였다.

명시적 형변환 – 1

변수를 저장할 때 일정 값 이하일 때 저장 공간의 이익을 위해 변수를 변경하는 경우의 예제

```
long a = 120;
byte b = 0;

if(a < 127) {
	b = (byte)a;
}

System.out.println(b);
```

Console : 120

long의 크기는 64bit이고 **byte**의 크기는 8bit이다. 즉 120이라는 값을 저장하기에는 굳이 long을 쓸 필요가 없다. 따라서 **byte**로 형 변환시켰다.

명시적 형변환 – 2

크기가 큰 데이터 타입을 크기가 작은 데이터 타입으로 이동시킬 경우의 예제

```
int a = 50;
byte b = (byte)a;

System.out.println(b);
```

Console : 50

int안에 있는 값을 **byte**으로 옮기기 위해 형변환 시켰다.

과제 2 - 문자형 -> 숫자형으로 형변환 하는 예제

두 정수를 더해야 하는데 한 변수의 값이 문자형 일 때 예제

```
int a = 5;
String b = "5";

System.out.println(a + Integer.parseInt(b));
```

Console : 10

String의 값을 int으로 바꿔서 두 int형의 합으로 바꾸었다.

과제 3 – 숫자형 -> 문자형으로 형변환 하는 예제

정수형의 값을 문자형으로 바꿔야 할 때의 예제

```
int a = 5;
int b = 5;

System.out.println(Integer.toString(a) + b);
```

Console : 55

int의 값을 String으로 바꿔서 문자열의 연속으로 바꾸었다.

비교 연산자

두 값을 비교하는 연산자

종류

a <= b
a가 b보다 작거나 같다.
a >= b
a가 b보다 크거나 같다.
a != b
a는 b와 같지 않다.
a == b
a와 b는 같다.
a > b
a는 b보다 크다.
a < b

a는 b보다 작다.

논리 연산자

논리 기호를 표현해 주는 연산자

종류

&& - and

|| - or

! – not

부호 증감 연산자

부호만을 적어 변수의 값에 일정한 값을 더하거나 빼는 연산자

++a

a++

=> a = a + 1

--a

a—

=>a = a – 1

부호가 앞에 있으면 뒤에 붙은 변수에 부호에 맞는 연산을 한 뒤 작업한다
부호가 뒤에 있으면 변수 앞의 작업을 한 뒤 부호에 맞는 연산을 실행한다

Ex)

```
int a = 5;
int b = ++a;

System.out.println(a);
System.out.println(b);
```

Console : 6

6

```
int a = 5;
int b = a++;

System.out.println(a);
System.out.println(b);
```

Console : 6

5

대입 연산자

a += b

a 에 b 를 더하여 a 에 대입

a -= b

a 에 b 를 빼서 a 에 대입

a *= b

a 에서 b 를 곱해서 a 에 대입

a /= b

a 에서 b 를 나누어서 몫을 a 에 대입

a %= b

a 에서 b 를 나누어서 나머지를 a 에 대입

삼항 연산자(조건 연산자)

조건에 따라 값이 변화하는 연산자

구조

변수 = (조건) ? true 일 때 값 : false 일 때 값

Ex)

```
int bool = (5 > 4) ? 10 : 20;

System.out.println(bool);
```

Console : 40

과제 4 – 비교 연산자 예제 3 개 작성

1 – 계산이 맞는지 확인하는 예제

```
System.out.println((7 + 5) == 5);
```

Console : False

계산 결과와 값이 다르므로 False 를 반환 한다.

```
System.out.println((7 + 5) == 12);
```

Console : True

계산 결과와 값이 같으므로 True 를 반환한다.

2 – 변수의 크기를 비교하는 예제

```
int a = 5;
int b = 8;

System.out.println(a > b);
```

Console : False

A 는 b 보다 크다 따라서 False 값이 반환된다.

3 – 값이 크거나 같음을 나타내는 예제

```
System.out.println("사촌동생의 나이는 3 살로 동생의 나이인 5 살 이하다" + (3 <= 5));
```

Console : 사촌동생의 나이는 3 살로 동생의 나이인 5 살 이하다 true

간단하게 크거나 같음을 표현할수도 있다.

과제 5 – 논리 연산자에 따른 결과 값 정리

&& - AND

모두 True 일 때만 True 값을 반환한다.

True && True -> True

False && True -> False

True && False -> False

False && False -> False

|| - OR

한쪽만 True 라도 True 값을 반환한다.

True || True -> True

False || True -> True

True || False -> True

False || False -> False

! – NOT

뒤에 붙는 조건을 반전 시킨다.

!True -> False

!False -> True

과제 6 – 삼항 연산자(조건 연산자) 예제 2 개 작성

1 – 조건을 판별하여 그룹을 나눌 때

```
int age = 20;
String a = (age >= 20) ? "성인" : "미성년자";

System.out.println(a);
```

Console : 성인

```
int age = 19;
String a = (age >= 20) ? "성인" : "미성년자";

System.out.println(a);
```

Console : 미성년자

`int age` 의 값에 따라 성인인지 미성년자 인지 구별 할 수 있다.

2 – 조건을 판별하여 숫자를 구별할 때

```
int a = 5;
String b = (a > 0) ? "0이하의 정수" : "양수인 정수";

System.out.println(b);
```

Console : 양수인 정수

조건식을 이용하여 숫자를 판별하는데 이용 할 수 있다.

과제 7 – 부호 증감 연산자 예제 4 개 작성

++a 활용

```
int a = 5;
int b = ++a;
```

a = 6, b = 6

a++활용

```
int a = 5;
int b = a++;
```

a = 6, b = 5

--a 활용

```
int a = 5;
int b = --a;
```

a = 4, b = 4

a--활용

```
int a = 5;
int b = a--;
```

a = 4, b = 5

과제 8 – 연산자 우선 순위

우선순위	연산자	내용
1	( ) , [ ]	괄호 / 대괄호
2	!, ~, ++, --	부정 / 증감 연산자
3	*, /, %	곱셈 / 나눗셈 연산자
4	+, -	덧셈 / 뺄셈 연산자
5	<<, >>, >>>	비트단위의 쉬프트 연산자
6	<, <=, >, >=	관계 연산자
7	==, !=	
8	&	비트단위의 논리연산자
9	^	
10	|	
11	&&	논리곱 연산자
12	||	논리합 연산자
13	? :	조건 연산자
14	=, +=, -=, *=, /=, %=, <<=, >>=, &=, ^=, ~=	대입 / 할당 연산자

과제 9 – 입력문 정리

1. Scanner 사용

```
import java.util.Scanner;
```

해당 **import** 문을 사용해야 한다.

```
Scanner sc = new Scanner(System.in);
```

Scanner 의 명칭을 선언해준 뒤(명칭 : sc)

입력할 데이터 타입에 맞춰 입력문을 선언 해준다

byte 입력

```
byte scByt = sc.nextByte();.
```

short 입력

```
short scShort = sc.nextShort();
```

int입력

```
int scInt = sc.nextInt();
```

long 입력

```
long scLong = sc.nextLong();
```

float 입력

```
float scFloat = sc.nextFloat();
```

double 입력

```
double scDouble = sc.nextDouble();
```

문자형 입력

띄워 쓰기 불가능

```
String scString = sc.next();
```

띄워 쓰기 가능

```
String scString1 = sc.nextLine();
```

2. in.read 사용

```
import java.io.IOException;
```

해당 **import** 문을 사용해야 한다.

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

throws 을 사용하여 예외 처리해준다.

```
int inInt = System.in.read();
```

원하는 변수를 선언 하고 이러한 형태로 선언해주면 된다.