제2장 식과 연산자

식과 연산자

여러가지 계산을 수행하기 위한 “식과 연산자"와,

키보드로부터 문자를 입력 받거나, 화면에 문자를 출력하거나,

난수를 생성하는 등의 “명령문"을 배웁니다.

계산 문

계산 문이란, 변수나 값을 이용하여 여러가지 계산을 컴퓨터에 수행하게 하기 위한 문장이다.

계산 처리라고 한다면, 사칙연산 뿐만이 아니다.

변수에 값을 대입하는 것도 컴퓨터에게는 계산의 일종인 셈이다.

```
int a;
int b;
a = 20;
b = a + 5;
System.out.println(a);
System.out.println(b);
```

식 (expression)

식의 구성요소

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

a, b, 5 : **피연산자** (operand)

+, = : **연산자** (operator)

아무리 복잡한 식이라도, **모든 식은 이 두가지 요소로 구성되어 있다.**

1+5

연산자와 피연산자를 찾아보세요

age = 20

연산자와 피연산자를 찾아보세요

피연산자

피연산자란 “변수나 값" 이라고 생각하면 될까요?

-> 대체로는 맞지만, 좀 더 명확히 하기 위해 특별히 중요한 피연산자인 “**리터럴 (literal)**"을 소개합니다

리터럴 (literal)

피연산자 중에서도 숫자 5 나 문자열 “Hello World” 등, 소스코드에 기술되어 있는 값을 **리터럴** (literal) 이라고 합니다.
그리고 각각의 **리터럴은 데이터 형을 가지고 있습니다**.

리터럴의 종류	표기예	형
소수점이 없는 숫자 (정수)	30	int
소수점이 없는 숫자로 끝이 L 또는 l (큰 정수)	300000L	long
소수점 있는 숫자 (정밀도가 높은 소수)	30.5	double
소수점 있는 숫자로 끝이 F 또는 f (비교적 정밀도가 낮은 소수)	30.5F	float
true (참) 또는 false (거짓)	true	boolean
따옴표로 감싸여진 문자	‘남'	char
이중따옴표로 감싸여진 문자열	“Java”	String

1, '1', "1"

같은 것인지 다른 것인지 생각 해 보세요

정수 리터럴에 대한 응용기법


```
// 0x를 붙이면 16진수
int a = 0x11;    // 17

// 0b를 붙이면 2진수
int b = 0b0011; // 3

// 알기 쉽게 _ 를 붙이는 것도 가능
long price = 2_000_000; // 2,000,000 원
```

이스케이프 시퀀스 (escape sequence)

String 형이나 char 형의 리터럴을 기술할 때, 가끔씩 사용되는 것이 **이스케이프 시퀀스** (escape sequence) 라고 불리우는 특수한 문자이다.
"\ (역슬래쉬) 와 한 문자" 의 총 두 문자로 구성된 기술방법으로, 이 두 문자로 특수한 1문자를 표현한다.

표기	의미
\"	이중따옴표 (")
\'	따옴표 (')
\\	역슬래쉬 또는 원 기호 (\)
\n	개행 (제어문자)

리스트 2-2 이스케이프 시퀀스를 사용하지 않은 예 (에러)


```
System.out.println("내가 좋아하는 기호는 쌍따옴표(")입니다");
```

Java 에서는 **두개의 이중따옴표로 둘러 쌓인 부분** 을 문자열로 취급합니다.

mission: 에러를 수정 해 봅시다.

계산 (evaluation)

Java 가 식에 따라 계산 처리를 하는 것을, 식의 **계산** (evaluation) 이라고 한다.

Java는 3가지 단순한 원칙에 의해 계산을 수행 합니다.

계산 결과로의 치환 법칙

연산자는 주변의 피연산자의 정보를 사용해 계산을 하고, 계산 결과로 치환 한다.

1 + 5 - 3

우선순위의 원칙

식에 연산자가 여러개 있는 경우, **Java 에서는 정해진 우선순위가 높은 연산자 부터 순서대로** 계산 합니다.

1+5*3 (1+5)*2

결합규칙의 원칙

식 안에 같은 우선순위 그룹에 속한 연산자가 여러개 있을 경우, **연산자별로 정해진 “방향" 부터 순서대로** 계산 한다

10+5+2 a=b=8

연산자

+ 나 * 이외의 연산자 중 대표적인 연산자를 소개합니다

연산자	기능	우선순위	계산의 방향	계산의 예
+	더하기	5	좌->우	3 + 5 -> 8
-	빼기	5	좌->우	10 - 3 -> 7
*	곱하기	4	좌->우	3 * 2 -> 6
/	나누기 (정수 연산에서는 몫)	4	좌->우	3.2 / 2 -> 1.6 9 / 2 -> 4
%	나누기 한 나머지	4	좌->우	9 % 2 -> 1

문자열 결합 연산자

연산자	기능	우선순위	계산의 방향	계산의 예
+	문자열의 결합	5	좌->우	“안녕" + “하세요" -> “안녕하세요"

대입연산자

연산자	기능	우선순위	계산의 방향	계산의 예
=	우변을 좌변에 대입	15	우->좌	a = 10 -> a (10)
+=	좌변과 우변을 더해 좌변에 대입	15	우->좌	a += 2 -> a (a = a + 2 와 동일)
-=	좌변과 우변을 빼서 좌변에 대입	15	우->좌	a -= 2 -> a (a = a - 2 와 동일)
*=	좌변과 우변을 곱해 좌변에 대입	15	우->좌	a *= 2 -> a (a = a * 2 와 동일)
/=	좌변에서 우변을 나누어 좌변에 대입	15	우->좌	a /= 2 -> a (a = a / 2 와 동일)
%=	좌변에서 우변을 나누어 그 나머지를 좌변에 대입	15	우->좌	a %= 2 -> a (a = a % 2 와 동일)
/=	좌변에서 우변을	15	우->좌	a /= 2 -> a

$$a = a + 3$$

증가, 증감 연산자

연산자	기능	우선순위	계산의 방향	계산의 예
++	값을 1 증가 시킴	1	좌->우	a++ -> a (a = a + 1 또는 a += 1 과 동일)
--	값을 1 감소 시킴	1	좌->우	a-- -> a (a = a - 1 또는 a -= 1 과 동일)

```
int a;
a = 100;
a++;
System.out.println(a);
```

먼저 증가 하느냐, 나중에 증가 하느냐


```
int a = 10;
int b = 10;
System.out.println(++a);
System.out.println(b++);
```

형변환 (Cast)

```
double d = 3;
String s = "베스트 " + 3;
```

형이 맞지 않는 대도 에러가 나지 않는 이유는, **java가 식을 계산하는 과정에서 자동적으로 형을 변환하고 있기** 때문입니다.

1. 대입시에 자동형변환
2. 명시적인 형변환
3. 연산시의 자동형변환

대입시의 자동형변환

1장에서 배운대로 **어떤 타입으로 선언 된 변수는 같은 타입의 값을 대입해야 한다**.
int 형 변수에는 int 값을,
String 형 변수에는 String 값을

이것이 원칙임

리스트 2-5 값 보다 큰 타입의 변수에 대입

작은 타입의 값을 큰 타입의 변수에 대입 할 경우에 한해, **자동 형변환이 일어난다**

```
float f = 3;
double d = f;
System.out.println(f);
System.out.println(d);
```

리스트 2-6 데이터 보다 작은 타입 변수에 대입 (에러)


```
public class Main {
    public static void main(String[] args) {
        int i = 3.2;
    }
}
```

리스트 2-7 강제 형변환


```
public class Main {
    public static void main(String[] args) {
        int i = (int) 3.2;
    }
}
```

(int) 를 캐스트 연산자 (cast operator) 라고 한다.

캐스트에 의한 강제 형변환


```
int age;

age = 23;
age = 3.2;
age = (int) 3.2;
```

연산시의 자동형변환

산술연산자에 의한 계산이 수행 될 때에도 “좌우 피연산자는 동일한 타입”
이 원칙.

5 / 2 5.0 / 2.0

5.0 / 2

수치형 끼리의 연산시 형변환 규칙

- 한쪽 피연산자가 double 이면, 다른쪽을 double 로 형변환 됨
- 한쪽 피연산자가 float 이면, 다른쪽을 float 로 형변환 됨
- 한쪽 피연산자가 long 이면, 다른쪽을 long 로 형변환 됨
- 한쪽 피연산자가 int 이면, 다른쪽을 int 로 형변환 됨
- short이나 byte 피연산자는 int 로 형변환 됨

```java
public class Main {
    public static void main(String[] args) {
        double d = 8.5 / 2;
        long l = 5 + 2L;
        System.out.println(d);
        System.out.println(l);
    }
}
```

리스트 2-9 문자열의 연결

한쪽 피연산자가 String 이라면, 다른 쪽도 String 으로 자동 변환되어 연결된다.

```
public class Main {
    public static void main(String[] args) {
        String msg = "나의 나이는 " + 35;
        System.out.println(msg);
    }
}
```

시간 날 때 Java 언어의 spec에 대해 알아 봅시다

위키페디아 또는 Java 공식 문서 https://docs.oracle.com/javase/specs/

명령 실행 문

Java 가 준비해 준 여러가지 명령을 사용하는 것

System.out.println 등.

리스트 2-10 명령(화면출력) 실행문

```
String name = "오준석";
String message;
message = name + "씨, 안녕하세요";
System.out.println(message);
```

```
String name = "오준석";
System.out.println(name + "씨, 안녕하세요");
```

인자 (argument, 매개변수)


```
String name = "오준석";
System.out.println(name + "씨, 안녕하세요");
```

리스트 2-11 개행 없이 화면에 출력하는 명령


```
String name = "오준석";
System.out.print("내 이름은 ");
System.out.print(name);
System.out.print("입니다");
```

더 큰 숫자를 대입하는 명령


```
int a = 5;
int b = 3;
int m = Math.max(a, b);
System.out.println("비교실험 : " +
    a + "와 " + b + " 중 큰 쪽은.. " + m);
```

리스트 2-13 String 형을 int 형으로 변환하는 명령


```
String age = "31";
int n = Integer.parseInt(age);
System.out.println("당신은 내년에 " + (n + 1) +
"살이 됩니다");
```

리스트 2-14 랜덤한 수를 생성하는 명령


```
int r = new java.util.Random().nextInt(90);
System.out.println("랜덤한 수 " + r);
```

리스트 2-15 키보드로부터 입력을 받는 명령


```
System.out.println("당신의 이름을 입력 해 주세요");
String name = new java.util.Scanner(System.in).nextLine();

System.out.println("당신의 나이를 입력 해 주세요");
int age = new java.util.Scanner(System.in).nextInt();

System.out.println("반갑습니다. " + age + "살의 " + name + "씨");
```

수식

- 수식은 연산자와 피연산자로 구성되어 있다
- 리터럴에도 형(타입)이 있어 기술방법에 의해 결정 된다
- 연산자가 계산 되면, 그 연산자와 피연산자는 결과로 변한다
- 연산자는 우선순위와 결합규칙에 따라 계산된다.

형변환

- 큰 변수에 작은 데이터를 대입하면, 자동적으로 형이 변환되어 대입 된다
- 작은 변수에 큰 데이터를 대입하면, 캐스트를 하여 대입할 수 있다
- 수식의 계산시, 큰 데이터에 맞게 자동적으로 형이 변환 된다

명령의 실행

- Java에 준비되어 있는, 여러가지 명령을 실행할 수 있다

연습문제 2-1

```java
public class Main {
    public static void main(String[] args) {
        int x = 5;
        int y = 10;
        String answer = "x+y는" + x + y;
        System.out.println(answer);
    }
}
```

"x+y는 15" 가 표시되도록 수정하시오

연습문제 2-2

다음 중 문법이 올바른 것을 모두 고르시오.

1. int x = 3 + 5.0;
2. double d = 2.0F;
3. int i = “5”;
4. String s = 2 + “ 명째";
5. byte b = 1;
6. double d = true;
7. short s = (byte)2;

연습문제 2-3

다음 내용의 프로그램을 작성 하시오.

1. 화면에 “점을 보세요" 라고 표시합니다.
2. 화면에 “이름을 입력해 주세요" 라고 표시합니다.
3. 키보드로부터 문자열을 입력 받아, String 형 변수 name 에 넣습니다.
4. 화면에 “나이를 입력 해 주세요" 라고 표시합니다
5. 키보드로부터 문자열 입력을 받아, String 형 변수 ageString 에 넣습니다
6. 변수 ageString 의 내용을 int 형으로 변환하고, int 형 변수 age 에 대입합니다
7. 0 부터 3 까지의 난수를 생성해, int 형 변수 fortune 에 대입합니다
8. fortune 의 수치를 증가연산자로 1 증가시켜, 1 부터 4 까지의 난수로 합니다.
9. 화면에 “점꾀가 나왔습니다!” 라고 표시합니다.
10. 화면에 “(나이)살의 (이름)씨, 당신의 운세번호는 (난수) 입니다" 라고 표시합니다.
    (나이) 에는 변수 age를, (이름) 에는 변수 name 을, 그리고 (난수) 에는 8. 에서 만든 난수의 숫자를 표시합니다.
11. 화면에 “1:대박 2:중박 3:보통 4:망" 이라고 표시합니다