

Java 프로그래밍 길잡이

✓ 원리를 알면 IT가 맞았다

Java Programming for Beginners



chapter 02.

식별자와 데이터형

- 식별자의 개념 및 예약어에 관하여 학습한다.
- 자바언어에서 사용하는 다양한 데이터 형에 대해 학습한다.
- 변수의 개념 및 사용법에 관하여 학습한다.
- 형변환의 개념과 묵시적 형변환 및 명시적 형변환에 대하여 학습한다.
- 상수의 개념 및 사용법에 관하여 학습한다.

- 식별자는 자바코드 내에서 사용되는 각각의 **단어**를 의미한다.
- 식별자 지정 규칙
 - 첫 문자는 반드시 영문자이어야 한다. 그 다음 문자부터는 숫자와 문자를 혼합해서 사용 가능하다. 단, _ 와 \$는 첫 문자로 사용할 수 있는 특별한 문자이다.
 - 식별자에 대한 길이 제한은 없다.
 - 자바는 대소문자를 구별한다.
- 식별자 종류 2가지
 - a. 시스템 정의 식별자
자바 시스템이 필요에 의해서 먼저 정의한 식별자로서 보통 '예약어', '키워드'라고 부른다.
 - b. 사용자 정의 식별자
자바 개발자가 필요에 의해서 정의한 식별자로서 **클래스명, 변수명, 메소드명**을 지정할 때 사용된다.

○ 시스템 정의 식별자 (예약어, 키워드) 종류

- 자바언어 자체에서 특별한 의미를 가지는 식별자이다.
- 따라서, 시스템이 먼저 지정한 식별자이기 때문에 사용자 지정 식별자로서 사용이 불가능하다.

분류	예약어
기본 데이터 타입	boolean, byte, char, short, int, long, float, double
접근 지정자	private, protected, public
클래스와 관련된 것	class, abstract, interface, extends, implements, enum
객체와 관련된 것	new, instanceof, this, super, null
메소드와 관련된 것	void, return
제어문과 관련된 것	if, else, switch, case, default, for, do, while, break, continue
논리값	true, false
예외 처리와 관련된 것	try, catch, finally, throw, throws
기타	transient, volatile, package, import, synchronized, native, final, static, strictfp, assert

○ 사용자 정의 식별자

사용자가 지정 가능한 식별자로서 클래스명, 변수명, 메소드명, 상수 사용시 이름을 지정할 때 사용된다.

구분	정의 규칙	사용 예
클래스	<ul style="list-style-type: none"> ❑ 첫 문자는 항상 대문자로 표현 ❑ 하나 이상의 단어가 합쳐질 때는 각 단어의 첫 문자들만 대문자로 표현 (Camel 표기법) ❑ 의미있는 명사형으로 지정. 	<pre>class JavaTest{ ...; }</pre>
변수와 메서드	<ul style="list-style-type: none"> ❑ 첫 문자는 항상 소문자로 표현 ❑ 하나 이상의 단어가 합쳐질 때는 두 번째부터 오는 단어의 첫 문자들만 대문자로 표현 (Camel 표기법) ❑ 변수는 의미있는 명사형으로, 메소드는 의미있는 동사형으로 지정. 	<pre>String itLand; public void getTest(){ ...; }</pre>
상수	<ul style="list-style-type: none"> ❑ 모든 문자를 대문자로 표현 ❑ 하나 이상의 단어가 합쳐질 때 공백 필요 시 under score(_)를 사용하여 연결한다. ❑ 의미있는 명사형으로 지정. 	<pre>int javaTest= 10; final int JAVA_TEST = 20;</pre>

○ 데이터형은 자바언어가 처리할 수 있는 데이터 종류를 의미한다.

○ 자바의 데이터형 (Data Type) 2가지

■ 기본 데이터형 (primitive data type : PDT)

- 수치형(정수형): byte ,short , int , long
- 수치형(실수형): float, double
- 논리형: boolean
- 문자형: char

*byte: 이진데이터 처리시 사용됨
*short:C언어와 호환용

*float 보다 double이 정밀도가 높다.

■ 참조 데이터형 (reference data type : RDT)

- 기본 데이터형을 제외한 나머지 데이터형 이다.
- 대표적으로 클래스 , 배열 , 인터페이스가 있다.

○ 기본 자료형의 종류

[표 2-5] 기본 자료형의 종류

자료형	키워드	크기	기본값	표현 범위
논리형	boolean	1byte	false	true 또는 false(0과 1이 아니다)
문자형	char	2byte	\u0000	0 ~ 65,535
정수형	byte	1byte	0	-128 ~ 127
	short	2byte	0	-32,768 ~ 32,767
	int	4byte	0	-2,147,483,648 ~ 2,147,483,647
	long	8byte	0	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807
실수형	float	4byte	0.0	-3.4E38 ~ +3.4E38
	double	8byte	0.0	-1.7E308 ~ +1.7E308

○ 정수형의 기본은 int 이고 실수형의 기본은 double 이다.

○ 자바 언어가 처리하는 실제 데이터를 의미한다.

a. 문자 리터럴

하나의 문자를 의미한다. 반드시 “(single quotes)으로 표현한다.

예> ‘A’ , ‘남’

다음은 특별한 형태의 escape 문자이다.

이스케이프 문자	용도	유니코드
\t	수평 탭	0x0009
\n	줄 바꿈	0x000a
\r	리턴	0x000d
\"	“ (큰따옴표)	0x0022
'	' (작은따옴표)	0x0027
\\	\	0x005c

○ 문자열은 반드시 “”(double quotes)으로 표현한다. (문자와 문자열은 구분됨)

예> “홍길동” , “서울시”

b. 정수 리터럴

일반적인 정수 데이터를 의미한다. 10진수, 8진수, 16진수로 표현이 가능하다. 8진수는 숫자 0부터 7까지의 숫자데이터 조합으로 표현 가능하고, 16진수는 숫자 0과 x의 뒤에 숫자 0부터 9까지, 문자 A부터 F까지의 조합으로 표현한다.

예> 234 //10진수
030 //8진수
0xA4 //16진수

int 보다 큰 값은 long타입으로 표현하기 위해서 값 마지막에 L 문자를 추가한다.

예> long ssn = 8012101234567L;

c. 실수 리터럴

일반적인 소수점을 가진 실수 데이터를 의미한다.

예> 3.14 // 일반적인 표현방식

6.02E23 // 지수표현방식으로 큰 실수데이터 표현시 사용

2.71F // 간단한 float 형을 표현

실수형의 기본은 double이다. 따라서 float을 표현하기 위해서는 3.14F 또는 3.14f를 사용하며 명시적 double 표현법은 3.14D 또는 3.14d 형식을 사용한다.

d. 논리형 리터럴

참/거짓을 표현할 때 사용하는 논리 데이터이다. 자바에서는 소문자 true 또는 false 로 논리값을 표현한다.

예> true
false

- 프로그램에서 사용하는 데이터(리터럴)를 저장하기 위한 용도로 사용된다.
- 복수개의 값이 아닌, 단 하나의 값만 저장이 가능하다. 복수개의 값을 저장하기 위해서 배열 또는 컬렉션을 사용한다.
- 변수에는 다양한 타입의 값을 저장하지 못하고, 한가지 타입만 저장 가능하다.
- 저장된 데이터는 언제든지 변경이 가능하기 때문에 ‘변경이 가능한 수’ 즉, 변수라고 부른다.

변경이 불가능한 수는 ‘상수’라고 부른다.

- 기본형 데이터를 저장하면 ‘기본형 변수’ 라고 하고, 참조형 데이터를 저장하면 ‘참조형 변수’라고 한다.
- 다음과 같은 3가지 단계로 변수를 사용할 수 있다.
 - a. 변수 선언
 - b. 값 할당(초기화)
 - c. 값 변경

a. 변수선언

변수선언은 자바 프로그램에게 저장 시킬 데이터형과 코드에서 식별해서 사용하기 위한 용도인 변수명을 사용하여 표현한다.

변수선언은 사용자가 지정한 식별자로서 식별자 규칙에 맞게 지정해야 된다. **동일한 변수이름으로 중복 선언이 불가능하다.**

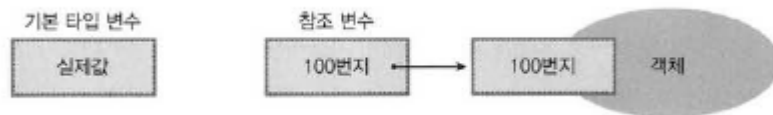
문법:

데이터형 변수명;

예> int num; //기본형 변수
 String name; //참조형 변수

int age=10,height,weight; //권장 안함.

*기본형 변수와 참조형 변수 차이점



b. 변수 초기화 (initialization)

선언된 변수에 가장 처음 값을 입력시키는 작업을 ‘변수 초기화’라고 한다.

문법:

변수명=값;

```
예> int num;      String name;  
    num = 10;    name="홍길동";
```

*다음과 같이 변수선언과 초기화 작업을 한꺼번에 처리 가능하다.

```
예> int num = 10;  
    String name = "홍길동";
```

c. 변수값 변경

변수에 저장된 데이터는 항상 똑같은 데이터를 가지기 보다는 프로그램이 실행되면서 변경되는 것이 일반적이다.

이런 이유로 저장된 값이 변경될수 있기 때문에 ‘변수’라고 한다.

예>

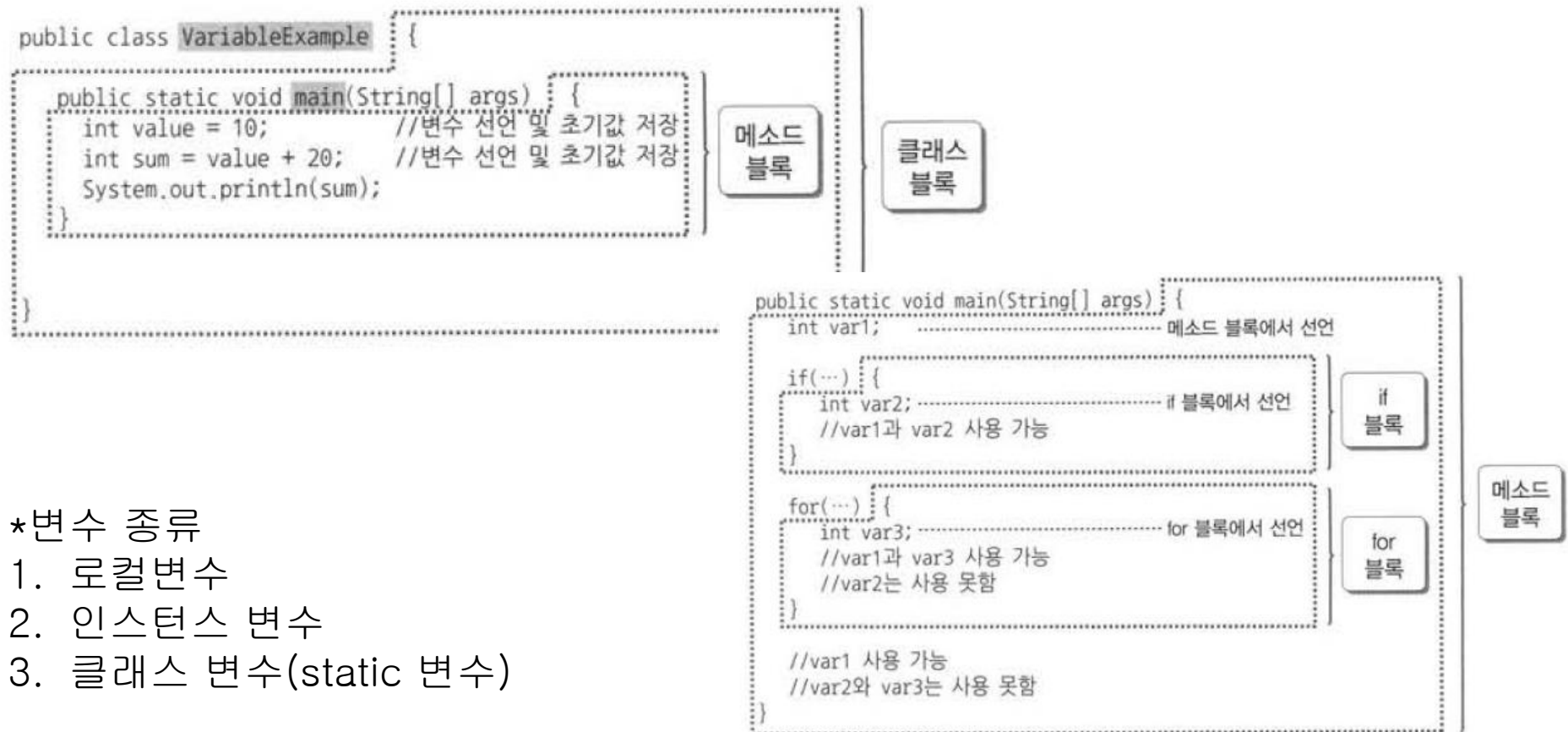
```
int age =10;
```

```
..
```

```
age = 20; // 프로그램 실행중 에 변수값 데이터 변경가능.
```

• 변수의 사용 범위 (scope)

변수는 블록({})내에서 선언되고 사용된다. 즉 블록 scope를 따른다.



*변수 종류

1. 로컬변수
2. 인스턴스 변수
3. 클래스 변수(static 변수)

- 서로 다른 종류의 데이터가 프로그램 실행 중에 원래의 데이터형을 버리고 새로운 데이터 형으로 변경되는 것을 의미한다. 단, 같은 종류의 데이터끼리만 변경이 가능하다.

- 데이터 형 변환 2가지 형태

- a. 묵시적 형변환

- 자동형 변환 또는 promotion이라고 한다.
 - 기본 데이터형 또는 참조 데이터형(상속관계) 모두 가능하다.
 - 기본적으로 수치형은 상호간에 데이터 형변환이 다음과 같이 데이터형이 큰 쪽으로 자동으로 변환 가능하다. ('up casting')

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

char -> int

- 큰 타입 데이터와 작은 타입 데이터가 연산될 때 결과는 큰 타입으로 자동으로 변경된다.

```
예> int n = 10;  
    float f = 20.3f;  
    double = n + f;    //연산결과는 실수
```

- int 보다 작은 데이터형간의 연산결과는 int로 자동으로 변경된다.

```
예> short s = 10;  
    short s2 = 20;  
    short s3 = s + s2; //에러
```

- 문자열과 다른 데이터간에 + 연산자 사용시 문자열로 자동으로 변경된다.

```
예> String str = "hello" + 1 + 2 + 3; // hello123
```

b. 명시적 형 변환

- 강제형변환 또는 'type casting'이라고 한다.
- 작은 데이터 형으로 변환되기 때문에 'down casting'이라고도 부른다. 따라서 데이터 손실이 발생 될 수도 있다.
- 다음과 같은 형 변환 연산자를 사용한다.

문법:

(데이터형)값;

예> int num = 10;
short s = (short)num;

값이 변경될 수 있는 것을 ‘변수’ 라고 하며,
값을 변경하지 못하는 것을 ‘상수’라고 한다.
final 키워드를 사용하여 ‘상수’로 처리 가능하다.
일반적으로 상수명은 대문자로 설정한다.

문법:

final 데이터형 상수명=값;

예>

public static final int NUM = 30;

기본 타입	최대값 상수	최소값 상수
byte	Byte.MAX_VALUE	Byte.MIN_VALUE
short	Short.MAX_VALUE	Short.MIN_VALUE
int	Integer.MAX_VALUE	Integer.MIN_VALUE
long	Long.MAX_VALUE	Long.MIN_VALUE
float	Float.MAX_VALUE	Float.MIN_VALUE
double	Double.MAX_VALUE	Double.MIN_VALUE



Thank you
