



JAVA 프로그래밍

Chapter 02 변수와 자료형

목차

- ❖ 상수와 변수 그리고 자료형
- ❖ 변수의 선언 방법
- ❖ 데이터 타입
- ❖ 타입 변환

상수와 변수 그리고 자료형

❖ 상수(Constant)

- 항상 일정한 값을 유지하는 데이터

❖ 변수(Variable)

- 데이터를 저장할 수 있는 메모리 공간이다.
- 특정 상황에 따라 변화하는 데이터.

❖ 자료형(Data Type)

- 변수의 크기를 규정지어 미리 만들어 둔 크기의 약속.
- 자료형은 데이터의 크기(Size)와 데이터의 형태(Type)를 지정하기 위한 도구이다.

❖ 변수의 선언 방법

- 자료형 변수명;
- `int age;` // 변수의 선언
- `age = 20;` // 변수에 값 할당
- `String str = "Hello";`

식별자 명명 규칙(naming convention)

- ❖ 첫 번째 글자는 문자이거나 '\$', '_', 영어 대, 소문자
- ❖ 문자 수의 제한은 없다
- ❖ 공백문자 및 특수 문자 사용불가
- ❖ 숫자는 첫 글자가 아닐 때 사용 가능
- ❖ 영어 대소문자를 구분한다.
- ❖ 자바 예약어(Keyword) 사용 불가

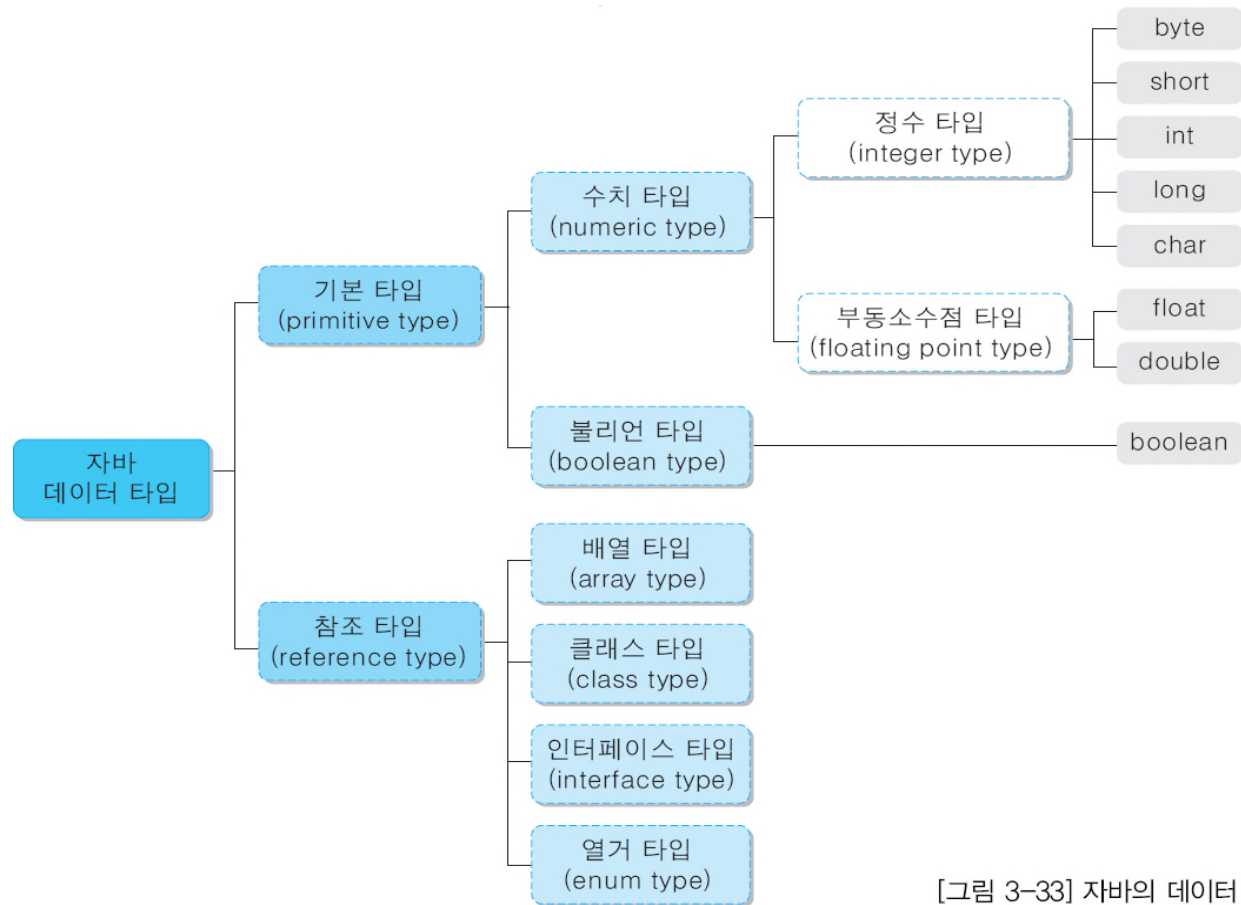
자바 예약어(Keyword)

- ❖ 예약어(Keyword)
 - 자바에서 미리 특정한 기능을 정의해둔 단어.
- ❖ 식별자(Identifier)
 - 키워드를 제외한 사용자에 의해 정의된 단어.

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

데이터 타입

❖ 자바의 데이터 타입



[그림 3-33] 자바의 데이터 타입

데이터 타입

❖ 기본 타입(primitive type)

- 키워드로 정의된 8개의 기본 데이터 타입
- 정의된 형태의 값을 변수에 저장하기 위한 데이터 타입
- 문자 데이터를 저장할 경우 유니코드 값을 사용
- 데이터 타입을 사용하여 변수를 선언하면, 해당 타입에 따라 정해진 메모리 크기가 변수에 할당

기본형		
구분	기본 타입	메모리 크기
정수 타입	byte	1바이트
	short	2바이트
	int	4바이트
	long	8바이트
	char	2바이트
부동 소수점 타입	float	4바이트
	double	8바이트
불리언 타입	boolean	정해져있지 않음

데이터 타입

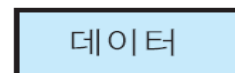
❖ 참조 타입(reference type)

- 참조값(reference value)을 다루기 위한 데이터 타입
- 참조값 : 메모리 주소를 계산할 수 있는 값
- 기본 타입을 제외한 나머지 타입이 이에 해당됨.
- 배열, 클래스, 인터페이스, 열거 타입 등

❖ 기본 타입과 참조 타입의 비교

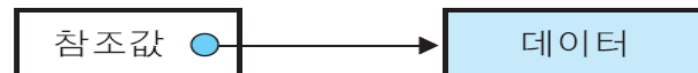
(a) 기본 타입의 변수

기본 타입 변수



(b) 참조 타입의 변수

참조 타입 변수



자바 기본 자료형

❖ 논리형 타입(boolean)

- boolean 타입은 1byte(8bit)로 표현되는 논리값(true/false)을 저장할 수 있는 데이터 타입.
- 상태값에 따라 조건문과 제어문의 실행 흐름을 변경하는데 주로 이용.

❖ 정수형 타입

자료형	크기	값의 범위	
byte	1byte	-128 ~ 127	S 7bit
short	2byte	-32768 ~ 32767	S 15bit
char	2byte	0 ~ 65535	16bit
int	4byte	-2147483648 ~ 2147483647	S 31bit
long	8byte	-922경 ~ 922경	S 63bit

정수 타입

❖ byte 타입

- byte 타입은 색상 정보 및 파일 또는 이미지 등의 이진(바이너리) 데이터를 처리할 때 주로 사용.
- 정수 타입 중 가장 작은 범위의 수를 저장.
- 128 ~ 127 ($-2^7 \sim 2^7-1$) 까지의 값의 변위를 가지며, 양수가 2^7-1 인 이유는 0을 포함하기 때문.

0	1	1	1	1	1	1	1	→	127
0	1	1	1	1	1	1	0	→	126
....									
0	0	0	0	0	0	0	1	→	1
0	0	0	0	0	0	0	0	→	0
1	1	1	1	1	1	1	1	→	-1
1	1	1	1	1	1	1	0	→	-2
....									
1	0	0	0	0	0	0	1	→	-127
1	0	0	0	0	0	0	0	→	-128

정수 타입

❖ char 타입

- 자바는 모든 문자를 유니코드(Unicode)로 처리한다.
- 유니코드는 세계 각국의 문자들을 코드값으로 매핑한 국제 표준 규약.
- 유니코드는 하나의 문자에 하나의 코드값을 부여, 'A' 및 '가'도 하나의 코드값을 갖는다.
- 유니코드는 0 ~ 65535 범위의 2byte 크기를 가진 정수값
- 0 ~ 127까지는 아스키(ASCII) 문자(특수기호 및 영어 알파벳)가 할당.
- 44032 ~ 55203까지는 한글 11172자가 할당되어 있다.

❖ char 변수의 선언

```
char var1 = 'A';           // 유니코드: 0x0041, 2진수: 00000000 01000001
char var2 = 'B';           // 유니코드: 0x0042, 2진수: 00000000 01000010
char var1 = '가';          // 유니코드: 0xAC00, 2진수: 10101100 00000000
char var1 = '나';          // 유니코드: 0xAC01, 2진수: 10101100 00000001
```

```
char c = 'A'; // 문자로 선언
char c = 65; // 10진수로 선언
char c = 'Wu0041'; // 16진수로 선언, 유니코드라는 의미에서 'Wu + 16진수값'
```

❖ 문자열(String)

- String name = "홍길동";

정수 타입

❖ short 타입

- short 타입은 2byte(16bit)로 표현되는 정수값을 저장할 수 있는 데이터 타입.
- 값의 범위 : -32,768 ~ 32,767($-2^{15} \sim 2^{15}-1$)

❖ int 타입

- int 타입은 4byte(32bit)로 표현되는 정수값을 저장할 수 있는 데이터 타입.
- 값의 범위 : -2,147,483,648 ~ 2,147,483,647($-2^{31} \sim 2^{31}-1$)
- int 타입은 자바에서 정수 연산을 하기 위한 기본 타입.
- byte 또는 short 타입의 변수를 + 연산하면 int 타입으로 변환된 후 연산되고 연산의 결과 역시 int 타입이 된다.

```
int number = 10;  
int octNumber = 012;  
int hexNumber = 0xA;
```

❖ long 타입

- long 타입은 8byte(64bit)로 표현되는 정수값을 저장할 수 있는 데이터 타입.
- 값의 범위 : -9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807($-2^{63} \sim 2^{63}-1$)
- long 타입은 정수값 뒤에 소문자 'l' 이나 대문자 'L'을 붙인다.

실수 타입(float, double)

❖ 실수 타입

- 실수 타입은 소수점이 있는 실수 데이터를 저장할 수 있는 타입으로 float와 double이 있다.
- 실수는 정수와 달리 부동 소수점(floating-point) 방식으로 저장되기 때문에 정수 타입보다 큰 범위의 값을 저장할 수 있다.

$$\begin{array}{ccc} + & m & \times 10^n \\ \text{부호} & \text{가수(mantissa)} & \text{지수(exponent)} \end{array}$$

float(4byte) : 부호(1bit) + 지수(8bit) + 가수(23bit) = 32bit = 4byte

1	지수 (8bit)	가수 (23bit)
---	-----------	------------

double(8byte) : 부호(1bit) + 지수(11bit) + 가수(52bit) = 64bit = 8byte

1	지수 (11bit)	가수 (52bit)
---	------------	------------

실수 리터럴의 기본 타입은 double이다. float 타입은 리터럴 뒤에 'f' 나 'F' 를 붙인다.

```
double var1 = 3.14;  
float var2 = 3.14; // 컴파일 에러(Type mismatch : cannot convert form double to float)  
float var3 = 3.14F;
```

데이터 형 변환

❖ 형 변환

- 형 변환이란 데이터 타입을 다른 데이터 타입으로 변환하는 것.
- 자동(묵시적) 형 변환, 강제(명시적) 형 변환이 있다.

❖ 자동 형 변환(Promotion)

- 작은 크기를 가지는 타입이 큰 크기를 가지는 타입에 저장될 때 발생하며, 자동으로 변환이 일어난다.

```
큰 크기 타입 = 작은 크기 타입  
  
byte byteVal = 10;  
  
int intVal = byteVal;    // 자동 형 변환
```

데이터 형 변환

❖ 강제 형 변환(Casting, Demotion)

- 큰 크기의 타입이 작은 크기의 타입으로 변환되는 것.
- 자동으로 변환 할 수 없으며, 강제 형 변환은 캐스팅 연산자()를 사용하여 형 변환한다.
- 데이터 손실이 발생할 수 있다.

작은 크기 타입 = (작은 크기 타입) 큰 크기 타입

```
int intVal = 12010000;
```

```
byte byteVal = (byte)intVal;           // 강제 형 변환(캐스팅)
```

제어문자

❖ 역슬래쉬(₩)가 붙은 이스케이프(escape) 문자는 특수한 용도로 사용된다.

Escape Sequence	의미	Unicode
₩b	백스페이스(backspace BS)	0x0008
₩t	수평 탭(horizontal tab HT)	0x0009
₩n	줄 바꿈 문자(line feed LF)	0x000a
₩f	새 페이지 문자 (form feed FF)	0x000c
₩r	리턴 문자(carriage return CR)	0x000d
₩"	큰따옴표(double quote ")	0x0022
₩'	작은 따옴표(single quote ')	0x0027
₩₩	백슬래쉬(backslash ₩)	0x005c
₩8진수	8진수에 해당 하는 unicode 문자 예)₩8, ₩42, ₩377	0x0000 ~ 0x00ff
₩u	Unicode의 16진수 표현 예)'₩uC790'	

Wrapper 클래스

❖ 기본 데이터 타입의 값을 담은 객체를 생성하는 클래스

클래스 이름	해당 기본데이터 타입
Byte	byte
Short	short
Integer	int
Long	long
Character	char
Float	float
Double	double
Boolean	boolean