

02장 변수와 자료형



변수와 자료형

◆ 상수 (Constant)

항상 일정한 값을 유지하는 데이터를 메모리에 저장해서 사용

◆ 변수 (Variable)

특정 상황에 따라 변화할 수 있는 데이터를 메모리에 저장해서 사용

◆ 자료형(Data Type)

데이터의 크기(Size)와 형태(Type)을 규정지어 만들어 둔 약속



변수와 자료형

◆ 변수의 선언 방법

```
public class Human {
    static int age;    // 변수의 선언

    public static void main(String[] args) {
        age = 20;      // 변수에 값 할당
        String name = "정찬웅";    // 변수를 선언함과 동시에 값을 할당
    }
}
```

◆ 변수의 명명 규칙

공백문자 및 특수문자 사용 불가

영어 대소문자를 구분한다 (NextIT ≠ nextit)

숫자는 첫 글자가 아닐 때 사용 가능

글자 수의 제한은 없다

자바 예약어 사용 불가

영어 대소문자 및 맨 앞에 \$, _ 기호를 쓰는 것은 허용

변수와 자료형

◆ 예약어 (Keyword)

자바에서 미리 특정한 기능을 정의해 둔 단어

◆ 식별자 (Identifier)

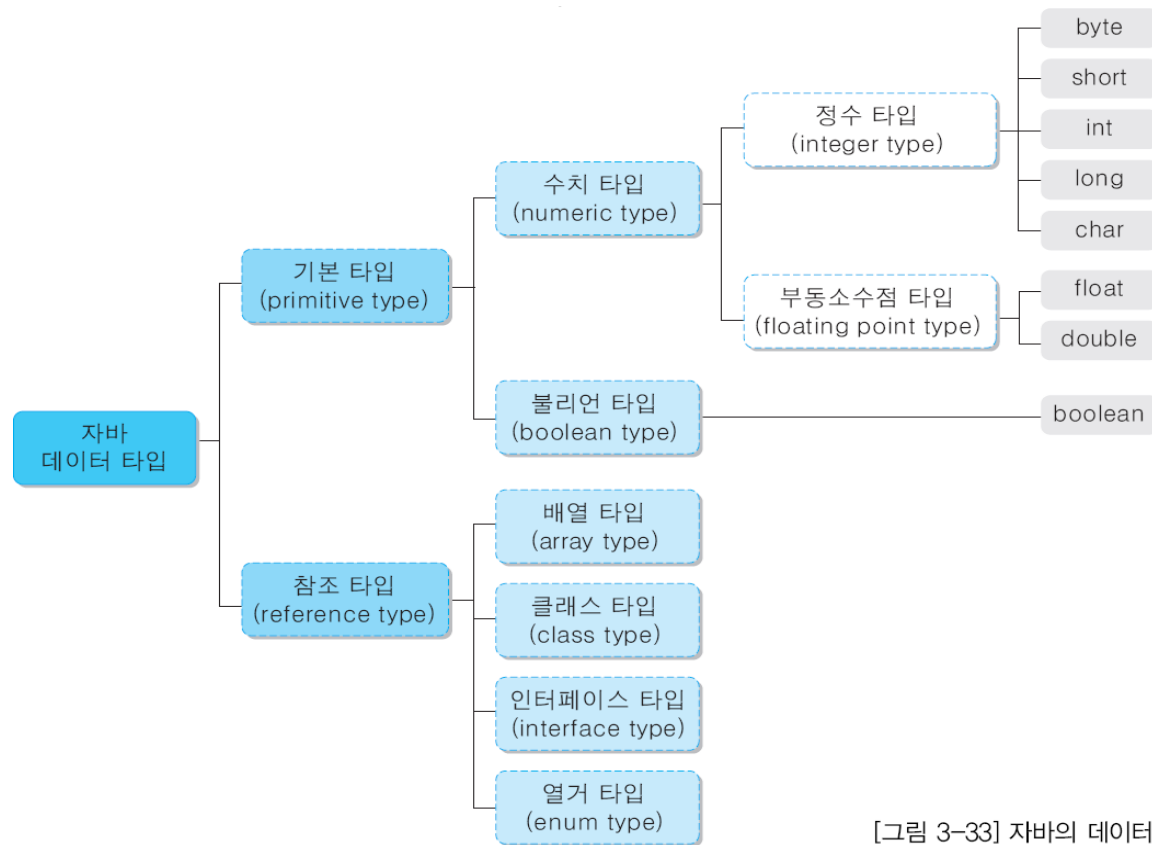
예약어를 제외한, 사용자에게 의해 정의된 단어



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

변수와 자료형

◆ 자바의 데이터 타입



[그림 3-33] 자바의 데이터 타입



변수와 자료형

◆ 기본 타입(Primary Type)

정의된 형태의 값을 변수에 저장하기 위한 데이터 타입

데이터 타입을 사용하여 변수를 선언하면 해당 타입에 따라 메모리 크기가 변수에 할당

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

↓ $2^8 = 256$ 배

변수와 자료형

◆ 정수 타입

char를 제외한 나머지는 숫자를 저장하는데 주로 사용

char는 모든 문자를 **유니코드(Unicode)**로 처리

자료형	크기	값의 범위
byte	1byte	-128 ~ 127
short	2byte	-32768 ~ 32767
char	2byte	0 ~ 65535
int	4byte	-2147483648 ~ 2147483647
long	8byte	-922경 ~ 922경

↓ $2^8 = 256$ 배

유니코드

세계 각국의 문자들을 코드값으로 매핑한 국제 표준 규약

0 ~ 65535 범위의
2 byte의 크기를 가진 정수값

0 ~ 127까지는 아스키(ASCII)
문자 (특수기호 및 영어 알파벳)
이 해당

44032 ~ 55203까지는 한글
11172 자가 해당

변수와 자료형

◆ 실수 타입

소수점이 있는 실수 데이터를 저장할 수 있는 타입

`double` 은 실수를 그대로 써도 저장이 가능하지만
`float` 은 실수 뒤에 `f` 나 `F`를 붙여주어야 저장이 가능하다.

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

◆ boolean 타입

`true/false` 의 논리값(참/거짓)을 저장할 수 있는 데이터 타입

값에 따라 조건문과 제어문의 실행 흐름을 변경하는데 주로 이용



변수와 자료형

◆ 참조 타입(Reference Type)

참조값(Reference Value)을 다루기 위한 데이터 타입

기본 타입을 제외한 나머지 타입들이 여기에 해당

배열, 클래스, 인터페이스, 열거타입 등

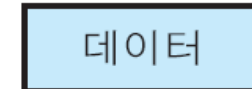
참조 타입을 가지는 변수는 메모리 주소를 가지지만 데이터는 없는 null 값을 가질 수 있다.

참조값

메모리 주소를 계산할 수 있는 값

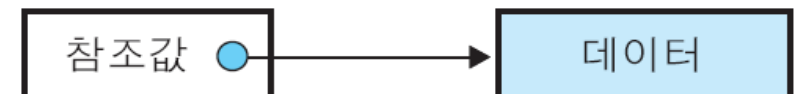
(a) 기본 타입의 변수

기본 타입 변수



(b) 참조 타입의 변수

참조 타입 변수



변수와 자료형

◆ 문자열(String)

자바의 문자열은 String 클래스의 인스턴스로 관리된다.

메서드		기능
int	length()	문자열의 길이를 리턴
char	charAt(int index)	주어진 index위치의 문자를 리턴
boolean	equals(Object anObject)	문자열의 내용 비교, 같으면 true, 다를 때 false
String	substring(int beginIndex, int endIndex)	beginIndex부터 endIndex-1까지의 문자열을 리턴
String	concat(String str)	문자열 뒤에 str을 연결
String	trim()	문자열의 앞뒤 공백을 제거
String	toUpperCase()	문자열의 모든 소문자를 대문자로 교체
String	toLowerCase()	문자열의 모든 대문자를 소문자로 교체
String	replace(char oldChar, char newChar)	문자열에 포함된 oldChar를 모두 newChar로 교체

변수와 자료형

◆ 형 변환(Casting)

데이터 타입을 다른 데이터 타입으로 변환하는 것

자동 형 변환과 강제 형 변환이 있다.

◆ 자동 형 변환

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
```

변수와 자료형

◆ 강제 형 변환

큰 크기의 타입이 작은 크기의 타입으로 변환되는 경우에는

자동으로 변환할 수 없으며, 강제 형 변환은 캐스팅 연산자()를 사용하여 형 변환한다.

데이터의 손실이 발생할 수 있다.

작은 크기 타입 = (작은 크기 타입) 큰 크기 타입

```
int intVal = 130;
```

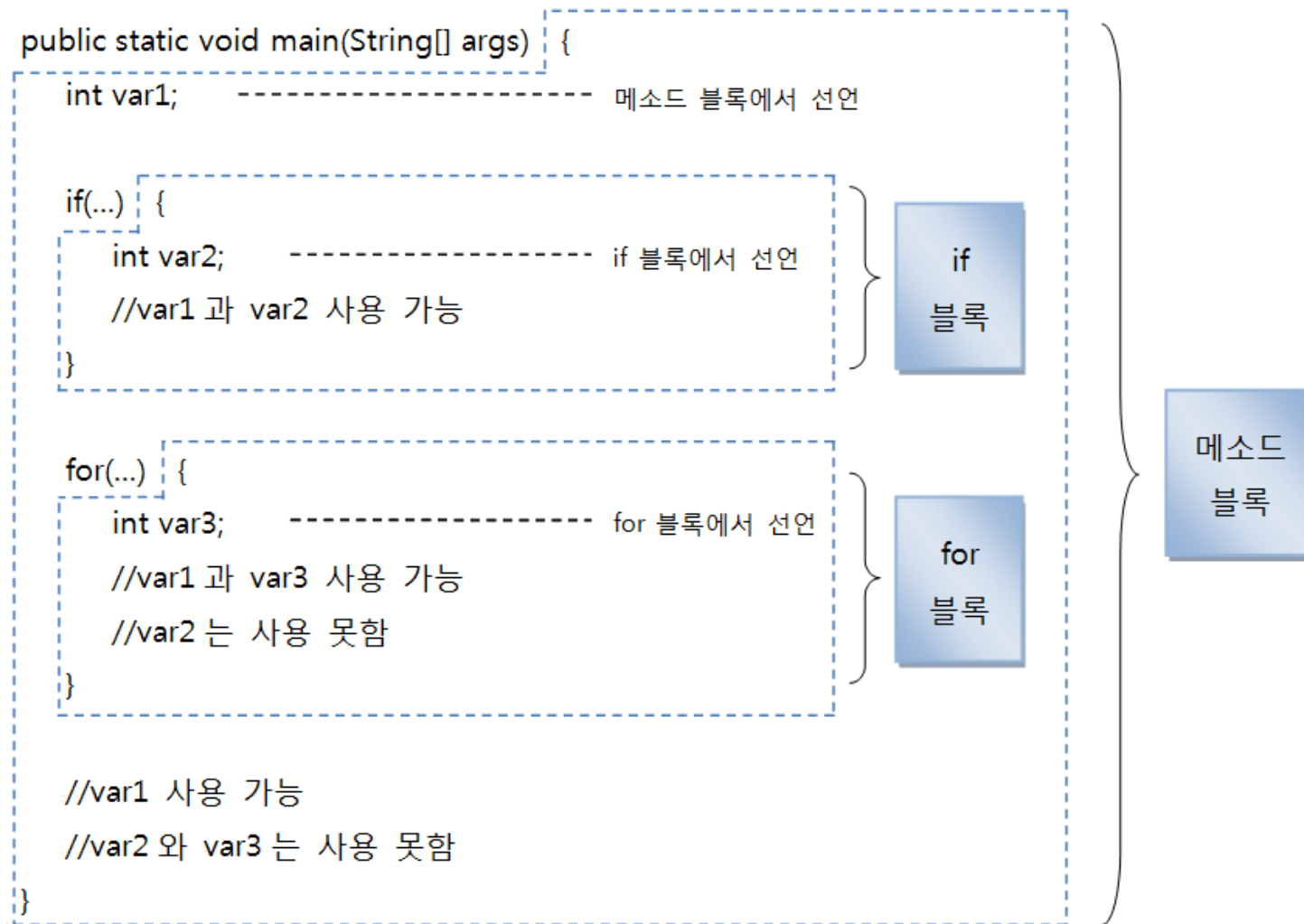
```
byte byteVal = (byte)intVal;           // 강제 형 변환(캐스팅)
```

byteVal = ?

변수와 자료형

◆ 변수의 사용 범위(Scope)

변수는 중괄호 블록{} 내에서 선언되고, 사용된다.



변수와 자료형

◆ 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

