

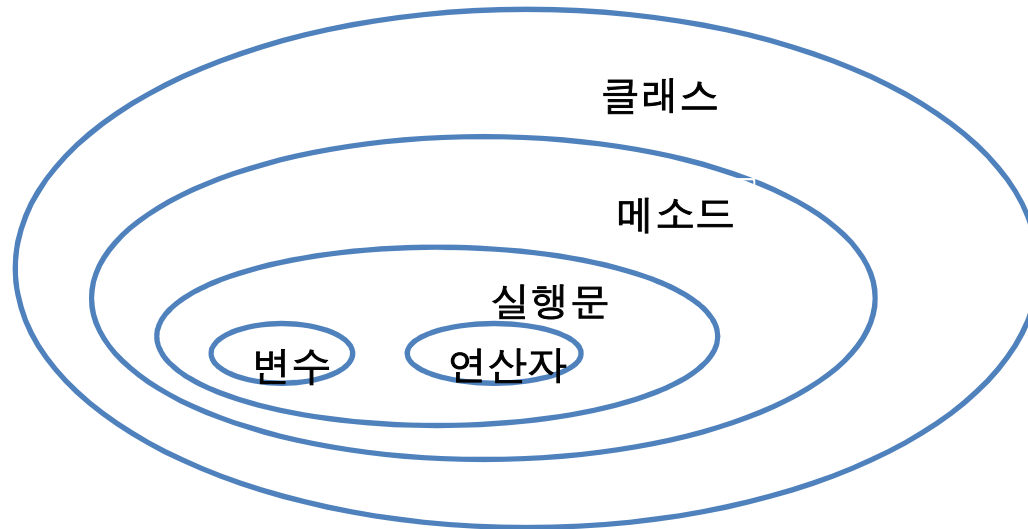


한림대학교 SW중심대학

생 초보를 위한 자바 프로그래밍

2장. 변수와 타입

자바프로그램의 구조



가장 간단한 구조의 자바프로그램

```
public class ClassExec {  
    public static void main(String[] args) {  
        실행문  
    }  
}
```

클래스 시작과 끝

```
public class ClassExec {  
}
```

메소드의 시작과 끝

```
public static void main(String[] args) {  
}
```

주석

- 주석(comment):
 - 프로그램에 대한 설명을 적어 넣은 것
 - 컴파일 안됨
 - // : 1줄 주석처리
 - /* */ : ** 사이는 줄수에 상관없이 주석 처리

```
public class ClassExec {  
  
    public static void main(String[] args) {  
  
        //int a;  
  
        /*  
        int x;  
        int i;  
        int j;  
        */  
  
    }  
}
```



■ 실행문과 세미콜론(;)

■ 실행문

- 변수 선언, 값 저장, 메소드 호출에 해당하는 코드
- 실행문 끝에는 반드시 **세미콜론(;)** 을 붙여 실행문의 끝 표시

```
int x = 1;           //변수 x를 선언하고 1을 저장
int y = 2;           //변수 y를 선언하고 2를 저장
int result = x + y;   //변수 result를 선언하고 변수 x와 y를 더한 값을 저장
System.out.println(result); //콘솔에 출력하는 메소드 호출
```

```
int x = 1; int y = 2;
int result =
    x + y;
```

1절. 변수

■ 변수란?

변수란, 하나의 값을 저장할 수 있는 메모리 공간

■ 변수의 선언 (p.30)

데이터 타입 ← 변수이름 ;

```
int age ;
```

```
double value ;
```

기본(primitive) 타입

논리 타입	boolean	<input type="checkbox"/>
문자 타입	char	<input type="checkbox"/>
정수 타입	byte	<input type="checkbox"/>
	short	<input type="checkbox"/>
	int	<input type="checkbox"/>
	long	<input type="checkbox"/>
실수 타입	float	<input type="checkbox"/>
	double	<input type="checkbox"/>

1절. 변수

■ 변수 이름을 위한 명명 규칙(naming convention)

1. 대소문자 구분
2. 자바 예약어 사용금지
3. 첫 글자는 무조건 알파벳
4. 특수기호 쓰지 않음
5. 알파벳, 숫자만 사용

1.

1절. 변수

■ 변수의 사용

■ 변수값 저장 : 할당연산자(=) 이용

```
int score;    //변수 선언  
score = 90;   //값저장
```

recommand !!

초기값은 변수를 선언함과 동시에 줄 수도 있다.

```
int score = 90;
```

■ 할당 연산자 기준

- 좌측변수 : 값을 대입 받는 쪽
- 우측변수, 상수 : 값을 대입해 주는 쪽

■ 값의 할당

- 좌측변수 = 우측변수, 상수

1절. 변수

■ 변수의 사용

■ 변수값 읽기

- 변수는 초기화가 되어야 읽기 가능
- 잘못된 코딩의 예

```
int value;           //변수 value 선언 (초기화 안됨) → 컴파일에러 발생  
int result = value + 10; //변수 value 값을 읽고 10을 더한 결과값을 변수 result에 저장
```

- 맞게 고친 후의 코드

```
int value = 30;       //변수 value가 30으로 초기화 됨  
int result = value + 10; //변수 value 값을 읽고 10을 더한 결과값(40)을 변수 result에 저장
```


- int 형 변수 i,j,x,y 생성 : 초기값 0
- 변수 i,j,x,y 출력
- i : 4, j : 6, x : 9, y : 10 값 대입
- 변수 i,j,x,y 출력

- i에 x의 값을 대입후 i의 값 출력
- j에 y의 값을 대입후 j의 값 출력

1절. 변수

- 문자 : 문자 하나를 “ ” 로 묶어서 표현

```
char c = ' c' ;
```

- 문자열 : 하나 이상의 문자를 “ ” 로 묶어서 표현

```
String s=“ orange” ;
```

예제

```
public class ByteExample {  
    public static void main(String[] args) {  
        char c = 'c';  
        String s="orange";  
  
        System.out.println("c : "+c);  
        System.out.println("s : "+s);  
  
    }  
}
```

- **System.out.println()에서 ()에 명시**
- **문자열을 출력할 경우 : ()안에 “ ” 를 이용**
 - System.out.println(“test”);
- **변수를 출력할 경우 : ()안에 변수명을 명시**
 - System.out.println(sum);
- **문자열과 변수를 출력할 경우 : ()안에 문자열이나 변수 상수를 +로 연결**
 - System.out.println(“test” +sum);
- **다수의 변수들을 출력할 경우 : ()안에서 각 변수들 사이를 +” ” + 구분**
 - System.out.println(sum+” ” +div+” ” +mul);
- **정수형 변수 a 에는 10을 저장, 정수형 변수 b 에는 20을 저장**
 - a, b를 하나의 System.out.println() 출력
- **문자형변수 c 에는 z를 저장, 문자열변수 str에는 orange를 저장**
 - ❖ c, str를 하나의 System.out.println() 출력

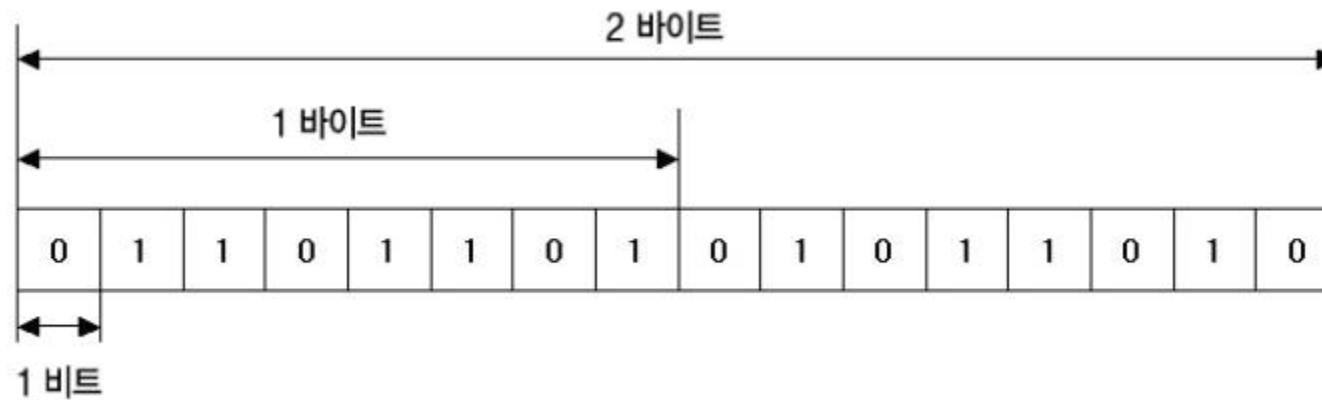
2절. 데이터 타입

- 기본(primitive) 타입
 - 정수, 실수, 문자, 논리 리터럴을 직접 저장하는 타입
 - 메모리의 최소 기억단위인 bit가 모여 byte 형성

값의 종류	기본 타입	메모리 사용 크기		저장되는 값의 범위
정수	byte	1 byte	8 bit	$2^7 \sim 2^7 - 1$ (-128~127)
	char	2 byte	16 bit	$0 \sim 2^{16} - 1$ (유니코드: \u0000~\uFFFF, 0~65535)
	short	2 byte	16 bit	$-2^{15} \sim 2^{15} - 1$ (-32,768~32,767)
	int	4 byte	32 bit	$-2^{31} \sim 2^{31} - 1$ (-2,147,483,648~2,147,483,647)
	long	8 byte	64 bit	$-2^{63} \sim 2^{63} - 1$
실수	float	4 byte	32 bit	(+/-)1.4E-45 ~ (+/-)3.4E38
	double	8 byte	64 bit	(+/-)4.9E-324 ~ (+/-)1.7E308
논리	boolean	1 byte	8 bit	true, false

데이터 크기

- 비트(Bit)
 - 컴퓨터의 최소 데이터단위
 - 1비트는 2진수(0,1) 1개의 값을 저장할수 있는 메모리 공간의 크기
- 바이트(Byte)
 - 비트가 8개 모인것을 1바이트라 함



```
public class ByteExample {  
    public static void main(String[] args) {  
        byte var1 = -128;  
        byte var2 = -30;  
        byte var3 = 0;  
        byte var4 = 30;  
        byte var5 = 127;  
        //byte var6 = 128; //컴파일 에러  
  
        System.out.println(var1);  
        System.out.println(var2);  
        System.out.println(var3);  
        System.out.println(var4);  
        System.out.println(var5);  
    }  
}
```

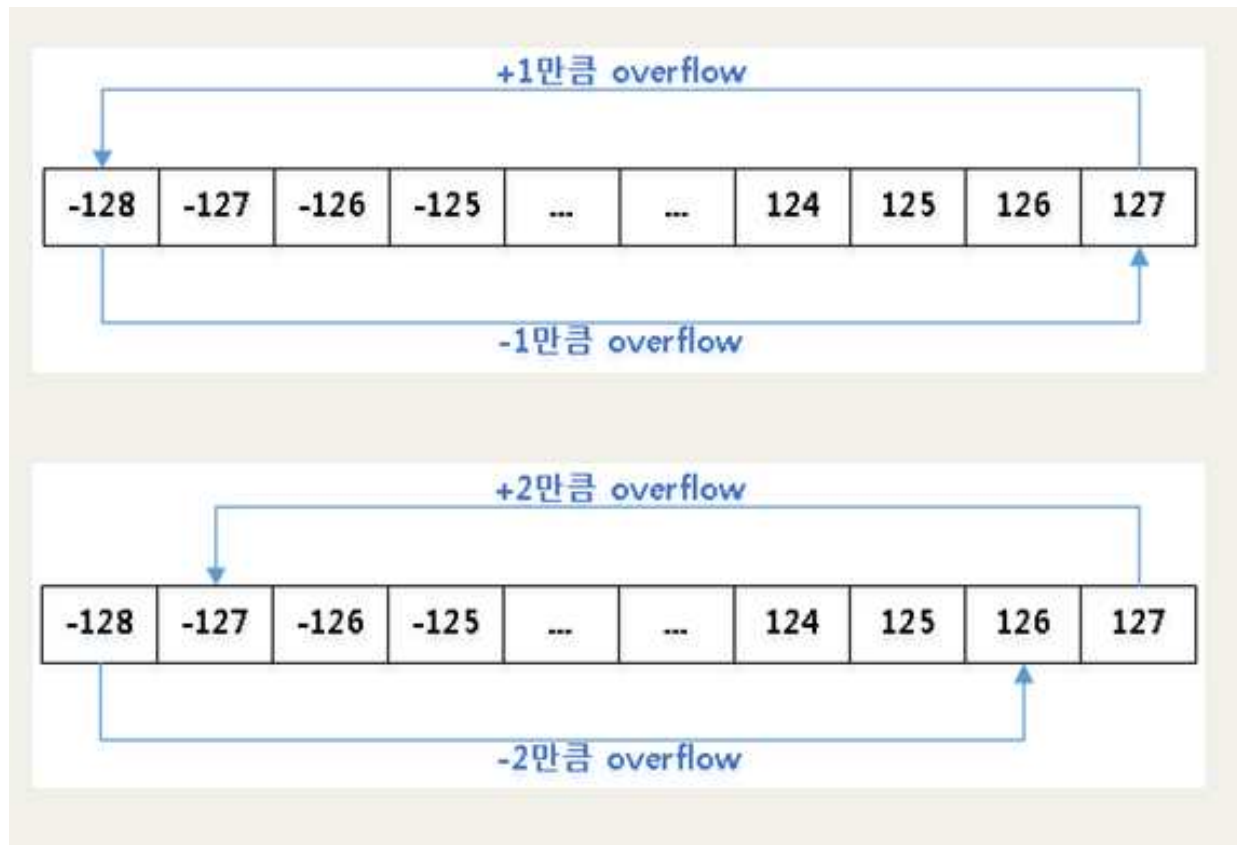
오버플로우, 언더플로우

■ 오버플로우

- 산술연산의 결과가 취급할 수 있는 수의 범위 보다 커지는 상태

■ 언더플로우

- 산술연산의 결과가 취급할 수 있는 수의 범위 보다 작아지는 상태



3절. 타입 변환

- 타입크기순서

byte(1) < short(2) < int(4) < long(8) < float(4) < double(8)

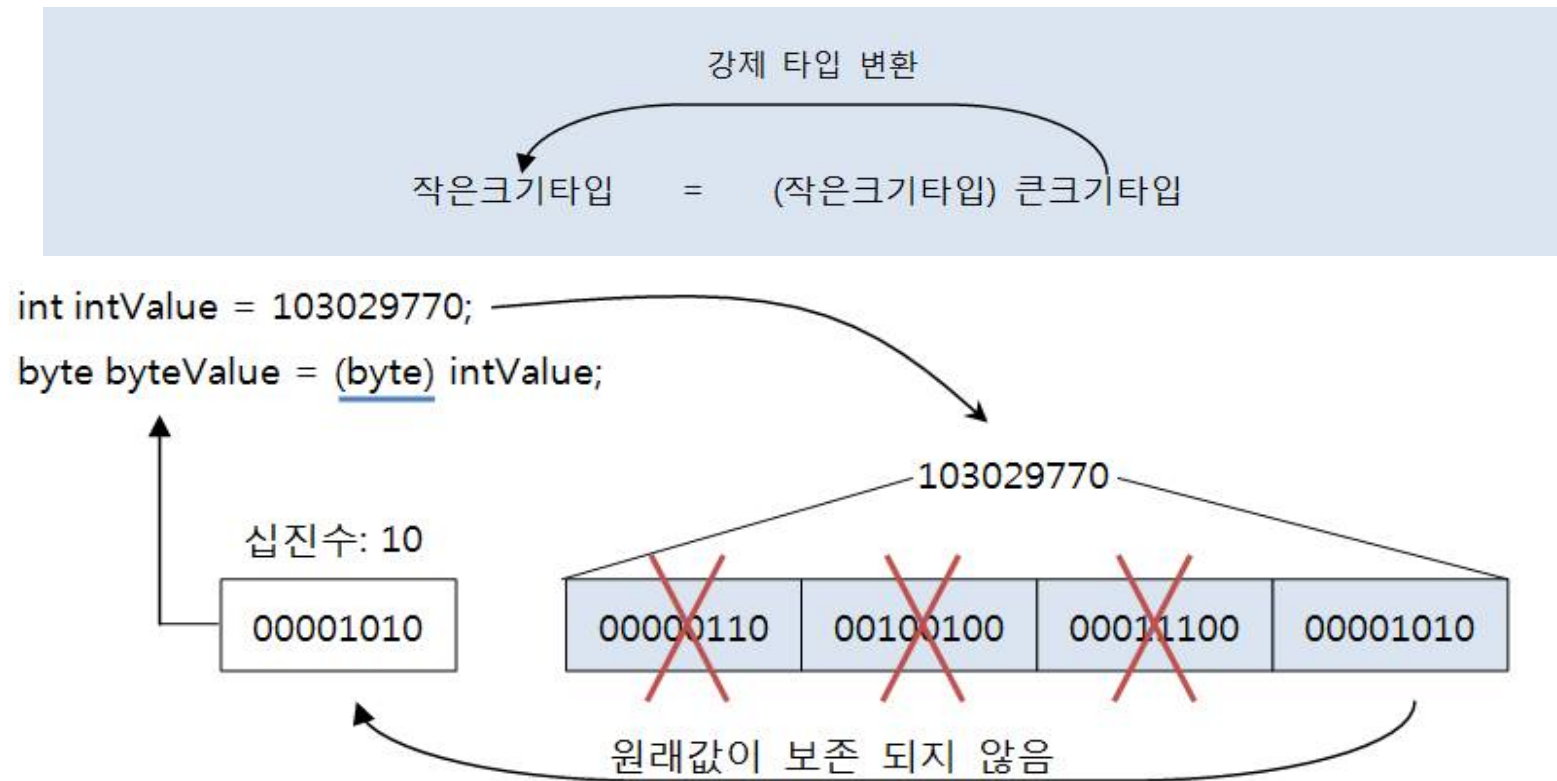
- 타입 변환

- 데이터 타입을 다른 타입으로 변환하는 것
 - byte ↔ int, int ↔ double

- 타입변환이 일어나는 시기

1. 값의 대입
2. 연산식

강제 타입 변환



■ 형식

- 변수a = (변수a의 타입) 변수b or 연산식

```
int intValue = 44032;  
long longValue = 500;  
intValue = (int) longValue;
```

```
int intValue = 44032;  
double doubleValue = 3.14;  
intValue = (int) doubleValue;
```

```
public class CastingExample {  
    public static void main(String[] args) {  
        int intValue = 44032;  
        char charValue = (char) intValue;  
        System.out.println(charValue);  
  
        long longValue = 500;  
        intValue = (int) longValue;  
        System.out.println(intValue);  
  
        double doubleValue = 3.14;  
        intValue = (int) doubleValue;  
        System.out.println(intValue);  
    }  
}
```