



나는 정말 JAVA^를 공부한 적이 있나?

Chapter 02. 변수와 자료형



02-1. 변수의 이해와 활용

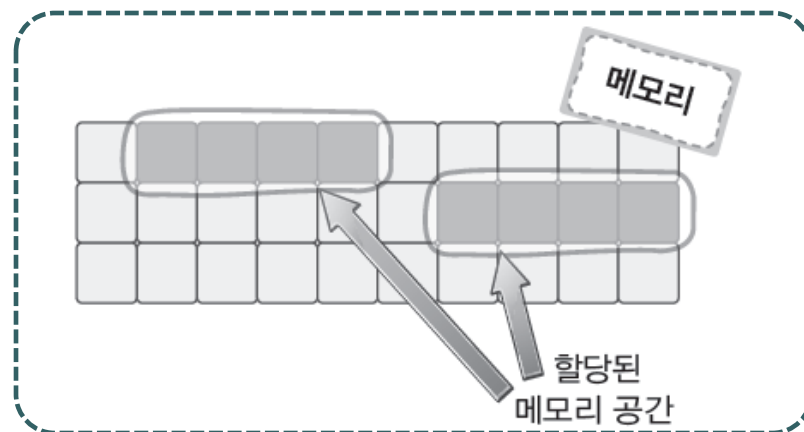
■ 메모리 공간의 활용과 변수와의 관계

✓ 메모리 공간의 활용에 필요한 두 가지 요소

- 데이터 저장을 위한 메모리 공간의 할당 방법
- 할당된 메모리 공간의 접근(저장 및 참조) 방법



'변수(Variable)'라는 것을 통해 이 두가지 방법을 모두 제공



■ 변수(Variable)에 대한 간단한 이해

"난 10진수 정수의 저장을 위한
메모리 공간을 할당하겠다."
"그리고 그 메모리 공간의 이름을
num이라 하겠다."

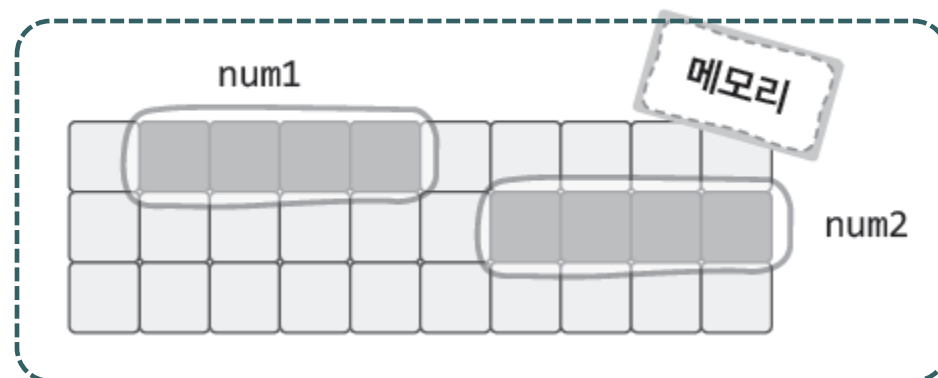


int num;

변수의 선언

int와 같이 변수의 특성을 결정짓는 키워드를 가리켜 자료형이라 한다.

int num1;
int num2;



■ 변수의 선언 및 활용의 예

```
class UseVariable
{
    public static void main(String[] args)
    {
        int num1;
        num1=10;

        int num2=20;
        int num3=num1+num2;
        System.out.println(num1+" "+num2+"="+num3);
    }
}
```

실행결과

10+20=30

■ 자료형의 종류와 구분

자료형	데이터	메모리 크기	표현 가능 범위
boolean	참과 거짓	1 바이트	true, false
char	문자	2 바이트	모든 유니코드 문자
byte	정수	1 바이트	-128 ~ 127
short		2 바이트	-32768 ~ 32767
int		4 바이트	-2147483648 ~ 2147483647
long		8 바이트	-9223372036854775808 ~ 9223372036854775807
float	실수	4 바이트	$\pm(1.40 \times 10^{-45} \sim 3.40 \times 10^{38})$
double		8 바이트	$\pm(4.94 \times 10^{-324} \sim 1.79 \times 10^{308})$

자료형의 분류

- 정수 표현 byte, short, int, long
- 실수 표현 float, double
- 문자 표현 char
- 참과 거짓의 표현 boolean

■ double형 변수의 선언과 활용의 예

```
class VariableDecl
{
    public static void main(String[] args)
    {
        double num1, num2, result;
        num1=1.0000001;
        num2=2.0000001;
        result=num1+num2;

        System.out.println(result);
    }
}
```

실행결과

3.0000001999999997

■ 변수의 이름을 짓는 방법

✓ 변수 이름의 제약사항

- 숫자로 시작 불가
- \$와 _ 이외의 다른 특수문자는 사용 불가
- 키워드는 변수의 이름으로 사용 불가

자바의 키워드

boolean	if	interface	class	true
char	else	package	volatile	false
byte	final	switch	while	throws
float	private	case	return	native
void	protected	break	throw	implements
short	public	default	try	import
double	static	for	catch	synchronized
int	new	continue	finally	const
long	this	do	transient	enum
abstract	super	extends	instanceof	null



02-2. 정수 표현방식의 이해

■ 정수를 표현하는 방식

✓ 정수의 표현(양의 정수 기준)

- 가장 왼쪽 비트인 MSB(Most Significant Bit)는 부호를 나타낸다.
- MSB를 제외한 나머지는 크기를 나타낸다.
- 바이트 크기의 차이는 표현범위의 차이로 이어진다.



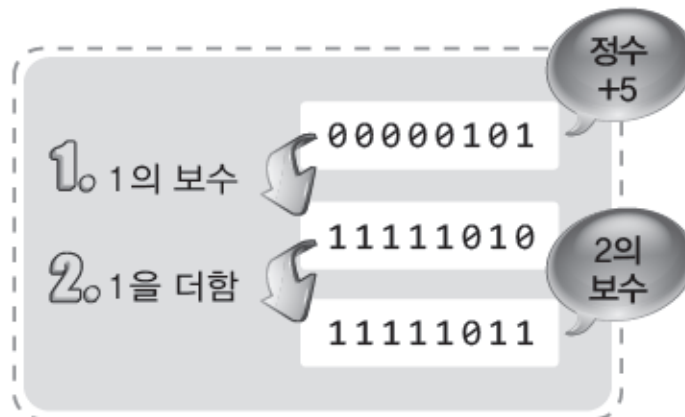
- MSB 0 👉 양 수
- 0011001 👉 25

양의 정수 25

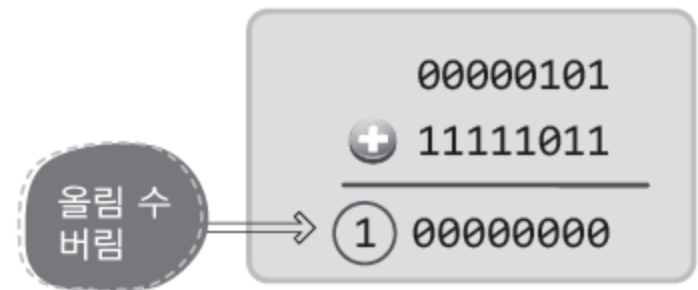
■ 음의 정수 표현방식

✓ 음의 정수 표현

- 양의 정수 표현방식과 다르다.
- 양의 정수와 합의 합이 0이 되는 구조로 정의
- 2의 보수가 음의 정수 표현방식



-5의 2진수 표현



검 증



02-3. 실수 표현방식의 이해

■ 실수표현의 애로사항과 이에 대한 해결책!

✓ 실수표현의 문제점

- 0과 1사이의 실수만 해도 그 수가 무한대
- 단순히 몇 바이트 정도로 모든 실수의 표현은 불가능하다.

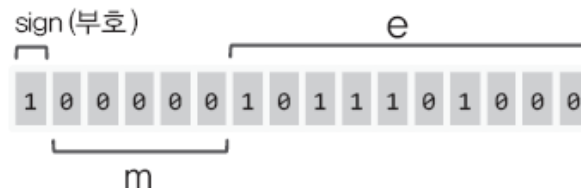
✓ 문제점에 대한 해결책

- 정밀도를 포기하고, 대신에 표현할 수 있는 값의 범위를 넓히자

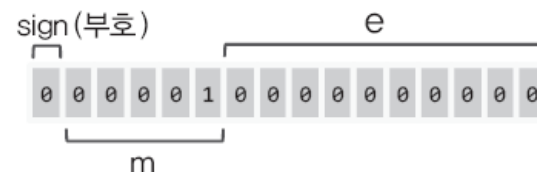
실수의
표현을 위한 수식

$$\pm (1.m) \times 2^{e-127}$$

수식에
반영하는 비트 구성



실수표현을 위한 수식의 도입



$$+ (1.1) \times 2^{-127}$$

표현의 예

■ 실수의 저장 및 참조 원리

참조는

역순의 과정을 거침

```
double num1=2.00000001
```



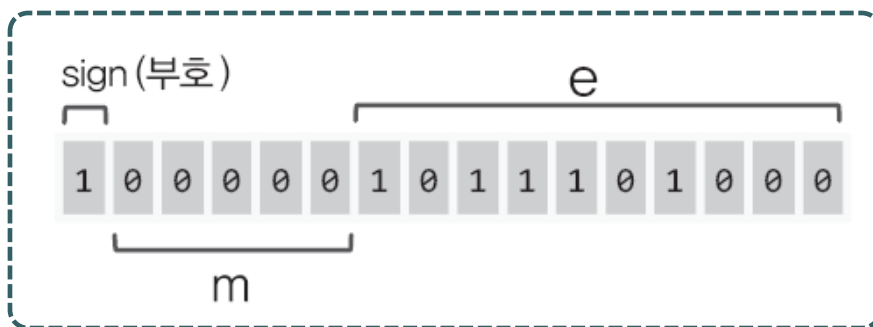
이 수식을 근거로

2.00000001에 대한 근사치를 조합해 냄

$$\pm (1.m) \times 2^{e-127}$$



부호, m 그리고 e를 다음과 같이 배치하여
변수 num1의 메모리 공간에 저장





02-4. 자료형의 이해

■ 정수 자료형에 대한 논의

✓ 자바의 4가지 정수 자료형

- byte, short, int, long
- 정수를 표현하는데 사용되는 바이트 크기에 따라서 구분이 됨

✓ 작은 크기의 정수 저장에는 short? 아니면 int?

- CPU는 int형 데이터의 크기만 연산 가능
- 때문에 연산직전에 short형 데이터는 int형 데이터로 자동변환
- 변환의 과정을 생략할 수 있도록 int를 선택한다!

✓ 그럼 short와 byte는 왜 필요한가?

- 연산보다 데이터의 양이 중요시 되는 상황도 존재!
- MP3 파일, 동영상 파일
- 데이터의 성격이 강하다면 short와 byte를 활용!

■ 실수 자료형에 대한 논의

✓ 자바의 2가지 실수 자료형

- float, double
- float는 소수점 이하 6자리 double은 12자리 정밀도

✓ 실수 자료형의 선택기준

- float와 double 모두 매우 충분한 표현의 범위를 자랑한다!
- 이 둘의 가장 큰 차이점은 정밀도!
- 따라서 필요한 정밀도를 바탕으로 자료형을 결정한다.
- 일반적으로 double의 선택이 선호됨.

■ 실수의 e 표기법과 16진수 8진수 표현의 예

```
class ENotation
{
    public static void main(String[] args)
    {
        double e1=1.2e-3;
        double e2=1.2e+3;
        int num1=0xA0E;
        int num2=0752;

        System.out.println(e1);
        System.out.println(e2);
        System.out.println(num1);
        System.out.println(num2);
    }
}
```

1.2×10^{-3}

$1.2 \times 10^{+3}$

16진수 A0E

8진수 752

■ 문자 자료형 char

✓ 자바의 문자 표현

- 문자 하나를 2바이트로 표현하는 유니코드 기반으로 표현
- 유니코드는 전 세계의 문자를 표현할 수 있는 코드 집합
- 문자는 작은 따옴표로 표현한다.
- 문자는 char형 변수에 저장한다. 저장 시 실제로는 유니코드 값 저장

오른쪽의 두 코드는
사실상 동일한 코드

```
char ch1='A';  
char ch2='한';
```



변환 발생

```
char ch1=65;           // 65는 16진수로 0x41  
char ch2=54620;        // 54620은 16진수로 0xD55C
```

■ '참'과 '거짓'을 표현하기 위한 자료형: boolean

나는 정말
JAVA를
공부한 적이 없어요

✓ 논리적인 표현을 위한 두 가지 약속

- true '참'을 의미하는 키워드
- false '거짓'을 의미하는 키워드

✓ 키워드 true와 false에 대한 좋은 이해

- 숫자의 관점에서 이해하려 들지 말자.
- 자바에서의 true와 false는 그 자체로 저장 가능한 데이터이다.
- true와 false의 저장을 위한 자료형 boolean!

```
class Boolean
{
    public static void main(String[] args)
    {
        boolean b1=true;
        boolean b2=false;

        System.out.println(b1);
        System.out.println(b2);
        System.out.println(3<4);
        System.out.println(3>4);
    }
}
```

실행결과

```
true
false
true
false
```

