

# JAVA Ch. 2

Prof. Youngchul Jung

2012 – Winter class

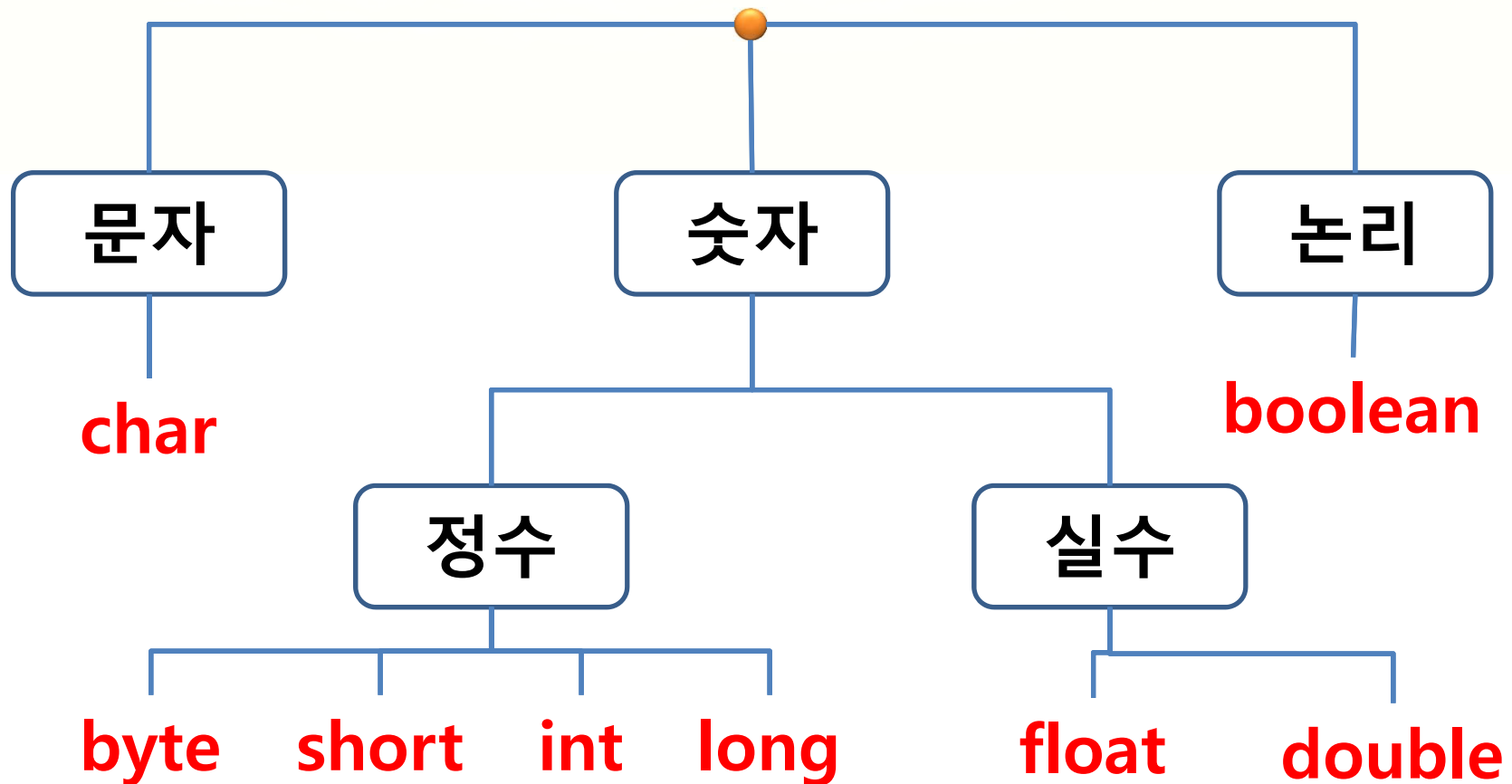


주문식교육의 신실  
영진전문대학

# 변수 (Vairable)

변수(Var i a b l e)란?

# 변수의 타입 (Data type): 입력 값



# 변수의 타입 (Data type): 메모리 할당

- **기본형(Primitive type)**
  - boolean, char, byte, short, int, long, float, double
  - 실제 값을 저장
- **참조형(Reference type)**
  - 기본형을 제외한 나머지 (String, System 등)
  - 객체의 주소를 저장 (4byte, 0x00000000~0xffffffff)

```
int variable_a;  
  
struct Skel variable_st;
```

C language

```
int variable_a;  
  
struct Skel variable_st;
```

JAVA

# 기본형(Primitive type)

변수 종류에 따른 메모리 크기

종류 \ 크기	byte			
	1	2	4	8
논리형				
문자형		char		
정수형	byte	short	int	long
실수형			float	double

Boolean : 1bit or 1 byte JVM에 의존적



# 변수 범위

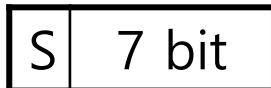
1 bit



8 bit = 1 byte



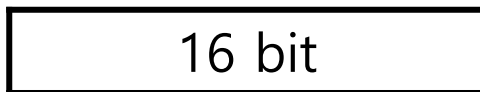
byte  $-2^7 \sim 2^7-1$



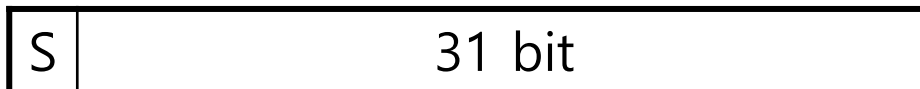
short  $-2^{15} \sim 2^{15}-1$



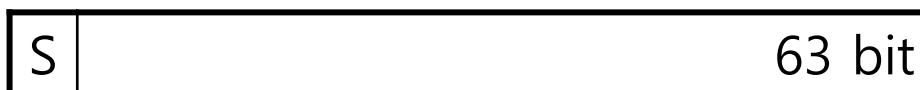
char  $0 \sim 2^{16}-1$



int  $-2^{31} \sim 2^{31}-1$



long  $-2^{63} \sim 2^{63}-1$



float  $1+8+23=32$  bit = 4 byte



double  $1+11+52=64$  bit = 8 byte



# 변수 선언방법

타입    변수명;

int   score;

int   score = 100;

# 변수 명명규칙 (Naming convention)

- 대소문자가 구분되며 길이에 제한이 없다.
- 예약어(Reserved word)를 사용해서는 안 된다.
- 숫자로 시작해서는 안 된다.
- 특수문자는 '\_'와 '\$'만을 허용한다.

`int taeyeon;`

`int _seohyun;`

`int orange2;`

`int king#;`

`int 2pal;`

`int sam$sung;`

`int float;`

`int Float;`



# 변수 명명규칙 권장사항

- 클래스 이름의 첫 글자는 항상 대문자로 한다.
  - 변수와 메서드 이름의 첫 글자는 항상 소문자로 한다.
- 여러 단어 이름은 단어의 첫 글자를 대문자로 한다.
  - lastIndexOf, StringBuffer
- 상수의 이름은 대문자로 한다. 단어는 '\_'로 구분한다.
  - PI, MAX\_NUMBER

```
public class Simpletest {  
  
    public static void main(String[] args) {  
        int generalVariable;  
  
        System.out.println("Hello world");  
    }  
  
    int memberVariableOne;  
}
```

# 변수 사용 예제

```
public class SimpleTest {  
    public static void main(String[] args) {  
        // boolean은 true, false로만 표현된다.  
        boolean varBool    = true;  
  
        byte    varByte    = 1;  
        short   varShort   = 2;  
        int     varInt     = 4;  
        long    varLong    = 8;  
        float   varFloat   = 4.0f;  
        double  varDouble  = 8.0d;  
        char    varChar    = '2';  
  
        System.out.println("varBool:"    + varBool);  
        System.out.println("varByte:"    + varByte);  
        System.out.println("varShort:"    + varShort);  
        System.out.println("varInt:"      + varInt);  
        System.out.println("varLong:"     + varLong);  
        System.out.println("varFloat:"    + varFloat);  
        System.out.println("varDouble:"   + varDouble);  
        System.out.println("varChar:"     + varChar);  
    }  
}
```

# 변수, 상수, 리터럴

- **변수**(variable) – 하나의 값을 저장하기 위한 공간
- **상수**(constant) – 한 번만 값을 저장할 수 있는 공간
- **리터럴**(literal) – 그 자체로 값을 의미하는 것

```
int score = 100;
```

```
score = 200;
```

```
final int MAX = 100;
```

```
MAX = 200;
```

```
char ch = 'A';
```

```
String str = "abc";
```

# 리터럴과 접미사

**boolean** power = true;

**char** ch = 'A';

**char** ch = '₩u0041';

**char** tab = '₩t';

**byte** b = 127;

**short** s = 32767;

**int** i = 100;

**int** oct = 0100;

**int** hex = 0x100;

**long** l = 1000000000000**L**;

**float** f = 3.14**f**

**double** d = 3.14**d**

**float** f = 100**f**;

10.

.10

10f

3.14e3f

1e1

# 변수의 기본값과 초기화

- 변수의 초기화 : 변수에 처음으로 값을 저장하는 것
  - 지역변수는 사용되기 전에 반드시 초기화해주어야 한다

자료형	기본값
boolean	false
char	'\u0000'
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d 또는 0.0
참조형 변수	null

```
boolean isGood = false;
char grade = ' '; // 공백
byte b = 0;
short s = 0;
int i = 0;
long l = 0; // 0L로 자동변환
float f = 0; // 0.0f로 자동변환
double d = 0; // 0.0로 자동변환
String s1 = null;
String s2 = ""; // 빈 문자열
```

# 변수 초기화 관련 예제

```
public class SimpleTest {  
  
    public static void main(String[] args) {  
  
        short    varShort;  
  
        varShort++;  
  
        System.out.println("varShort:" + varShort);  
    }  
}
```

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:  
    The local variable varShort may not have been initialized  
  
    at ycj.SimpleTest.main(SimpleTest.java:11)
```



# 문자와 문자열

**char** ch = 'A';

**char** ch = 'AB'; // 에러

**String** s1 = "AB";

**char** ch = "; // 에러

**String** s1 = "";

String s1 = "A" + "B"; // "AB"

"" + 7 → "" + "7" → "7"

""+7+7 → "7"+7 → "7"+"7" → "77"

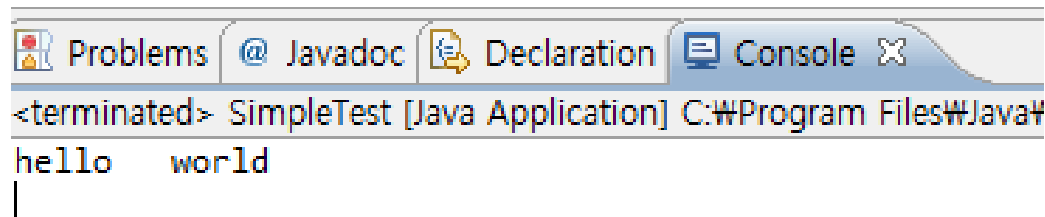
7+7+" " → 14+" " → "14"+" " → "14"

문자열 + any type → 문자열

any type + 문자열 → 문자열

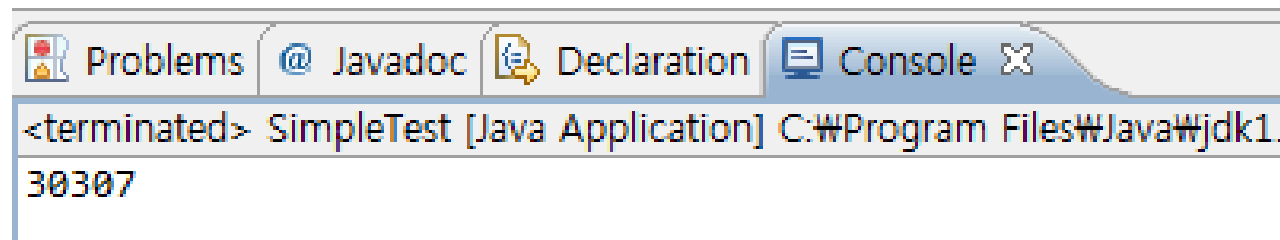
# 문자열 처리 예제 (1)

```
public class SimpleTest {  
  
    public static void main(String[] args) {  
  
        String leftString, rightString, printString;  
  
        leftString = "hello";  
  
        rightString = "world";  
  
        printString = leftString + "\t" + rightString;  
  
        System.out.println(printString);  
    }  
}
```



## 문자열 처리 예제 (2)

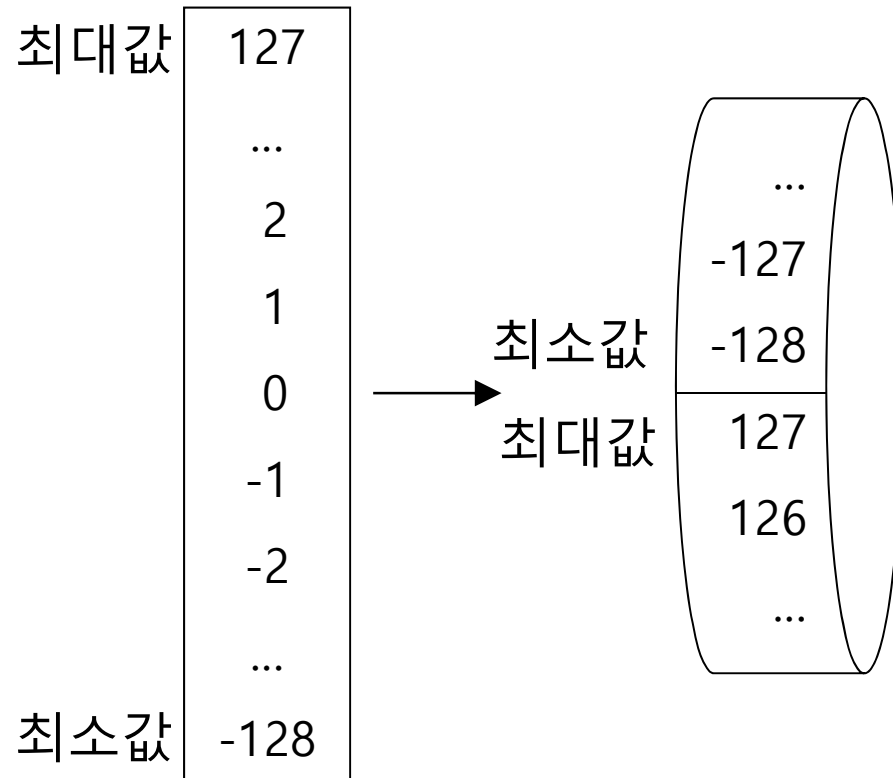
```
public class SimpleTest {  
  
    public static void main(String[] args) {  
  
        String printString;  
  
        printString = "7";  
  
        printString = 10 + 20 + "30" + printString;  
  
        System.out.println(printString);  
    }  
}
```



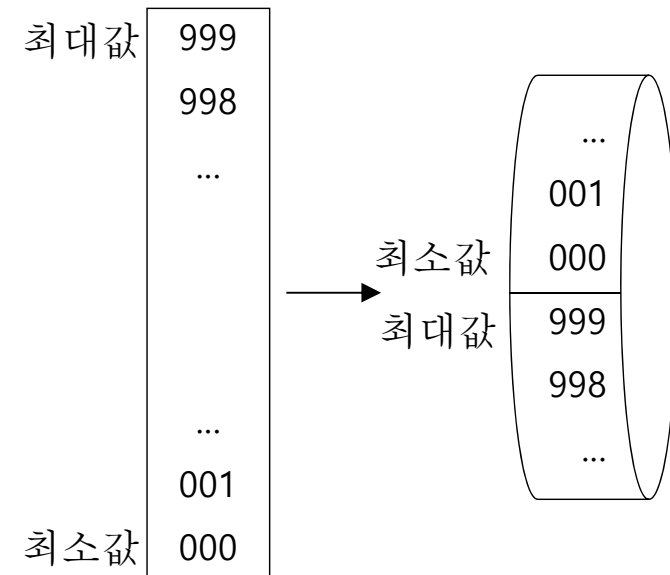
# 정수의 오버플로우(Overflow)

**byte** b = 127;

b = b + 1; // b에 저장된 값을 1증가



**byte** b = 128; //에러



**자동차 주행 미터기**

# 정수 오버플로우 예제

```
public class SimpleTest {
```

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

```
        byte varByte = 125;
```

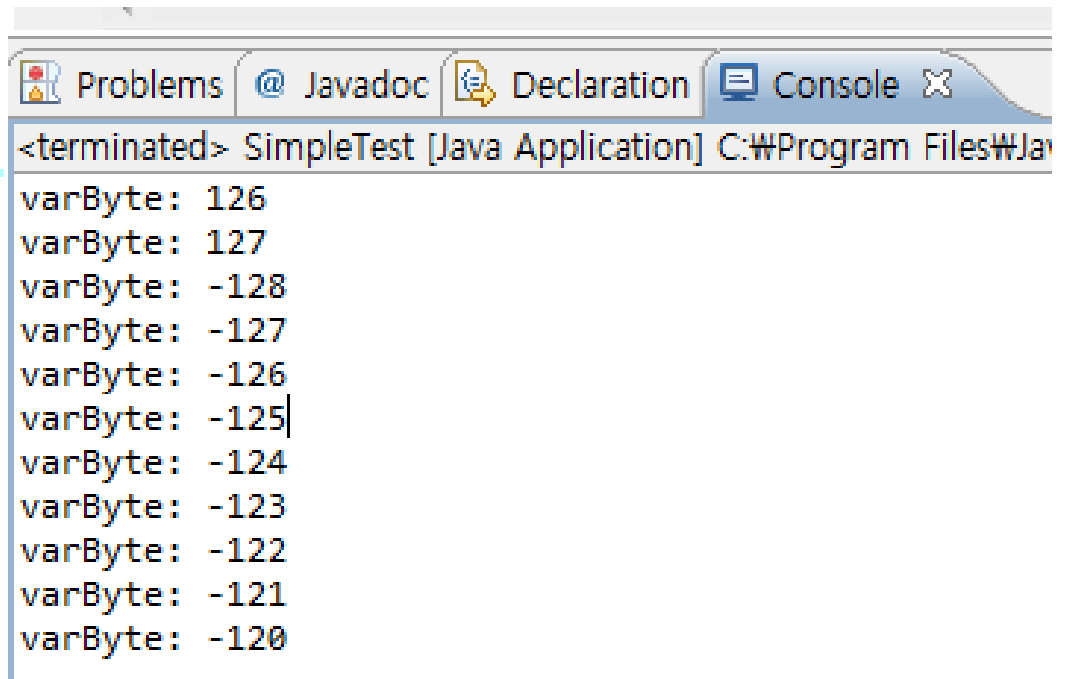
```
        for(int count = 0 ; count <= 10 ; count++)
```

```
        {
            varByte++;
            System.out.
```

```
        }
```

```
    }
```

```
}
```



```
Problems @ Javadoc Declaration Console X
<terminated> SimpleTest [Java Application] C:\Program Files\Ja
varByte: 126
varByte: 127
varByte: -128
varByte: -127
varByte: -126
varByte: -125
varByte: -124
varByte: -123
varByte: -122
varByte: -121
varByte: -120
```

# 형변환 (Type casting) (1)

- 형변환이란?

- 값의 타입을 다른 타입으로 변환하는 것이다.
- boolean을 제외한 7개의 기본형은 서로 형 변환이 가능하다.

**float f = 1.6f;**

**int i = (int) f;**

변 환	수 식	결 과
int → char	(char) 65	'A'
char → int	(int) 'A'	65
float → int	(int) 1.6f	1
int → float	(float) 10	10.0f



# 형변환 예제 (1)

```
public class SimpleTest {  
  
    public static void main(String[] args) {  
  
        float    varFloat    = 2.4f;  
        int      varInt      = 0;  
  
        byte     varByte     = 65;  
        char     varChar     = 'A';  
  
        varInt = (int) varFloat;  
        System.out.println("varInt: " + varInt);  
  
        varChar = (char) varByte;  
        System.out.println("varChar: " + varChar);  
    }  
}
```

---

```
varInt: 2  
varChar: A
```

## 형변환 (Type casting) (2)

## 1. byte → int

```
byte b = 10;
```

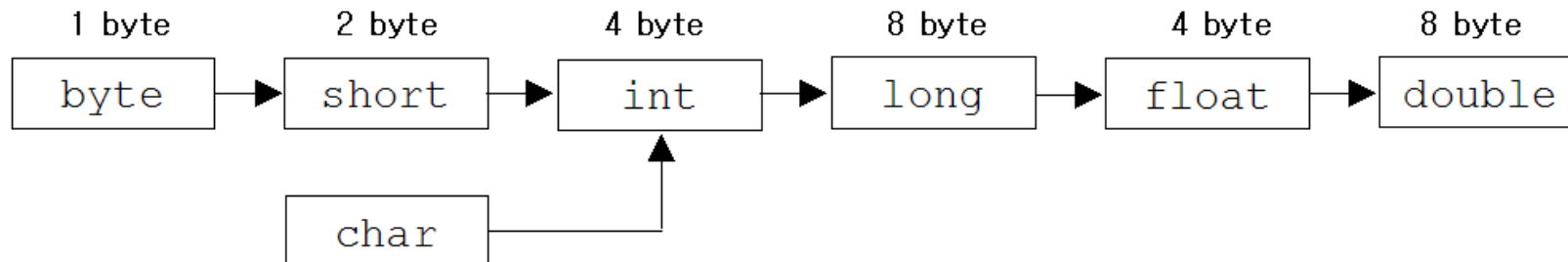
```
int i = (int) b; // 생략가능
```

## 2. int $\rightarrow$ byte

```
int i2 = 300;
```

```
byte b2 = (byte)i2; // 생략불가
```

변환	2진수	10진수	값손실
byte ↓ int	<div style="text-align: right;">0 0 0 0 1 0 1 0</div> <div>0 1 0 1 0</div>	10  10	없음
int ↓ byte	<div>0 1 0 0 1 0 1 1 0 0</div> <div style="text-align: right;">0 0 1 0 1 1 0 0</div>	300  44	있음



## 형변환 예제 (2)

```
public class SimpleTest {  
  
    public static void main(String[] args) {  
  
        byte    varByte = 100;  
        short   varShort = 0;  
  
        varShort = (short)varByte;  
        System.out.println("varShort: " + varShort);  
  
        varShort = 400;  
        varByte = (byte) varShort;  
        System.out.println("varByte: " + varByte);  
    }  
}
```

# Extra slides

# JAVA에서 콘솔로 부터 입력 받기

```
import java.util.*;
```

```
public class SimpleTest {
```

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

```
        Scanner scan = new Scanner(System.in);
```

```
        int varInt = 0;
```

```
        float varFloat = 0;
```

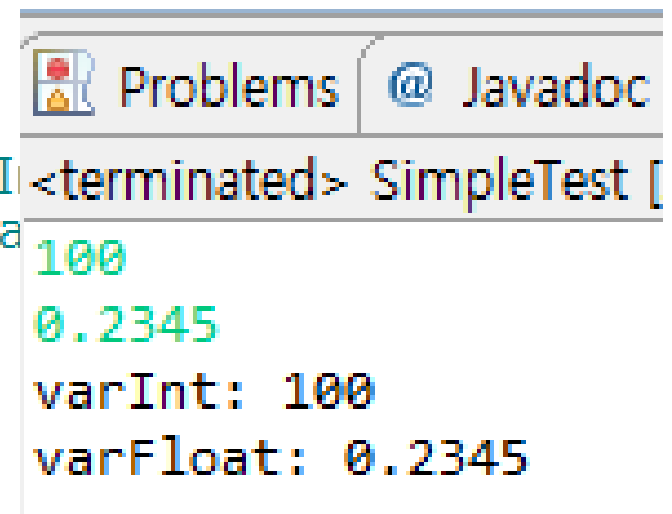
```
        varInt = scan.nextInt();
```

```
        varFloat = scan.nextFloat();
```

```
        System.out.println("varInt: " + varInt);
```

```
        System.out.println("varFloat: " + varFloat);
```

```
    }  
}
```



```
<terminated> SimpleTest [  
100  
0.2345  
varInt: 100  
varFloat: 0.2345
```

# 예제 프로그램

- 정수 두 개를 키보드로부터 입력 받아 더한 값을 출력하시오

```
import java.util.*;

public class SimpleTest {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int fistInput    = 0;
        int secondInput  = 0;

        fistInput    = scan.nextInt();
        secondInput  = scan.nextInt();

        System.out.println(fistInput + "+" + secondInput + "=" + (fistInput+secondInput));

    }
}
```