

# JAVA Program 실습

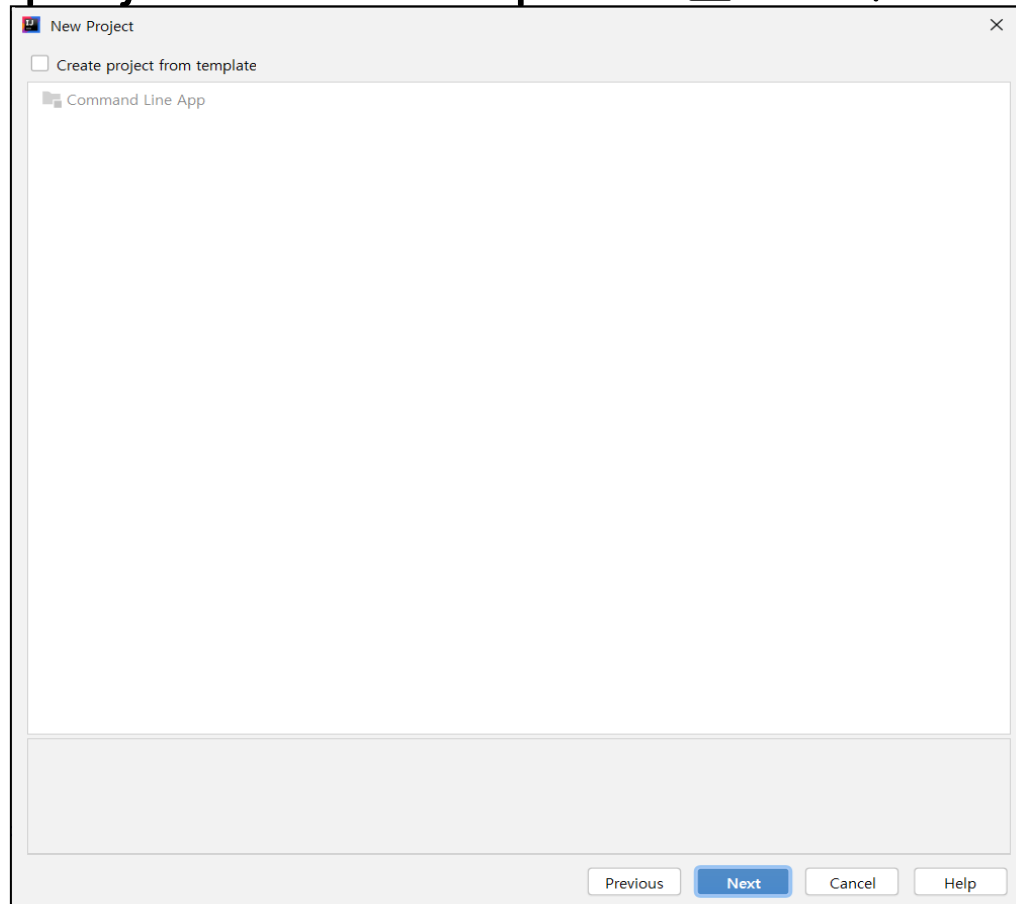
표준 입출력

경북대학교  
소프트웨어융합과  
교수 배희호

# 다양한 클래스 만들기

## ■ New Project

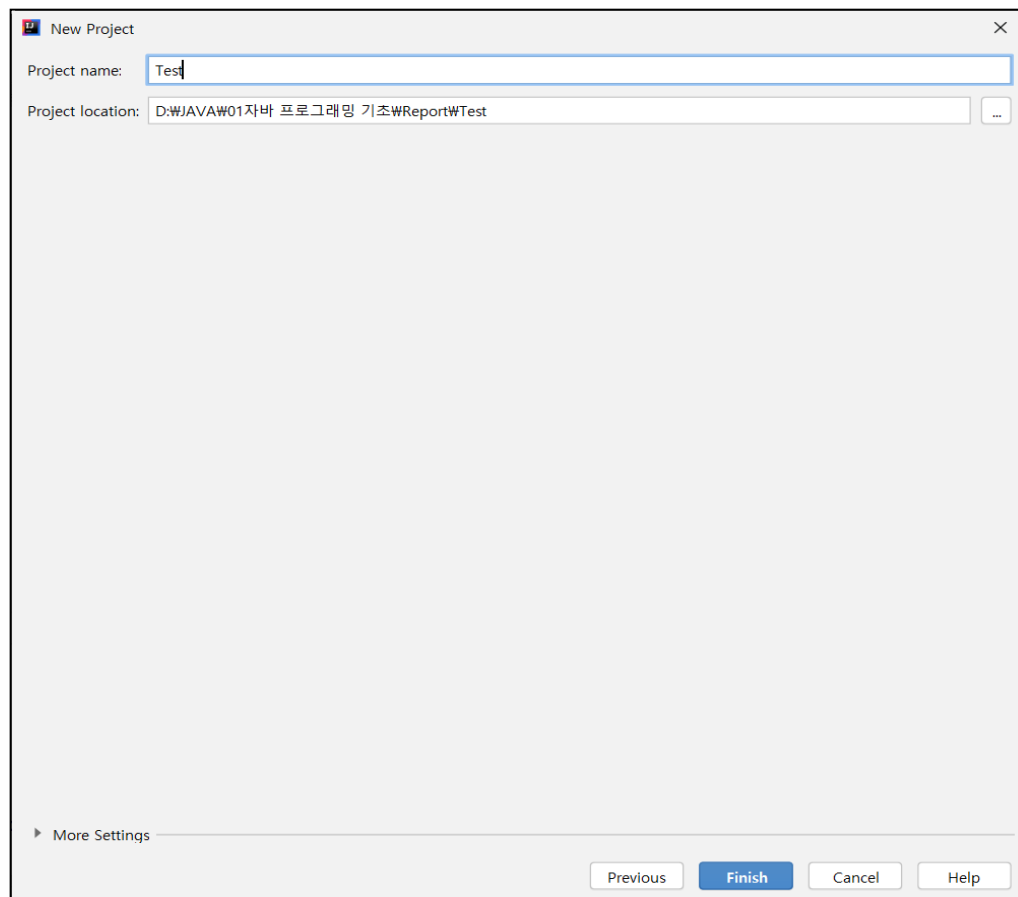
### ■ Create project from template 를 선택 안함



# 다양한 클래스 만들기

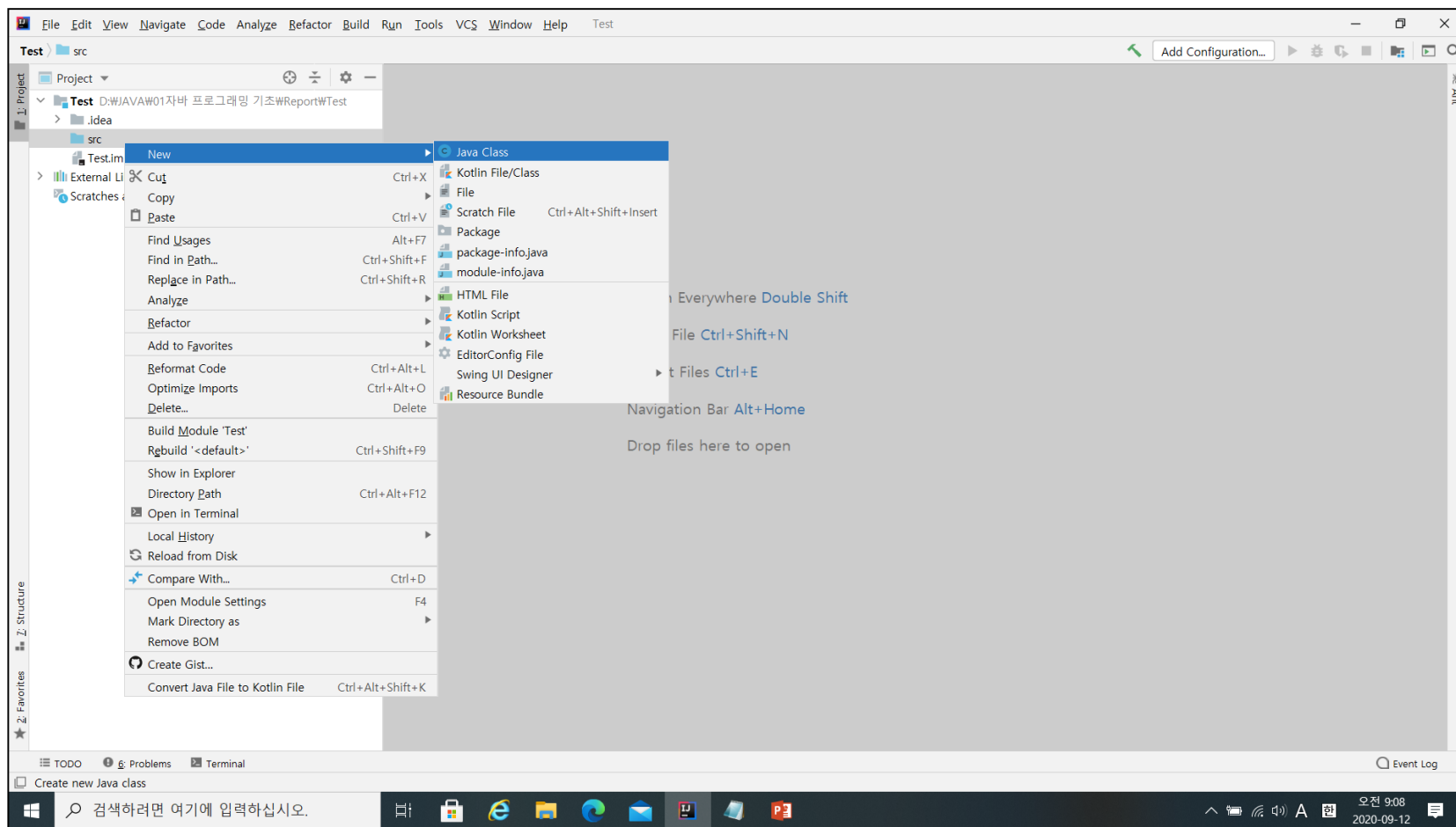
## ■ New Project

### ■ Project name (첫 글자 대문자 사용)



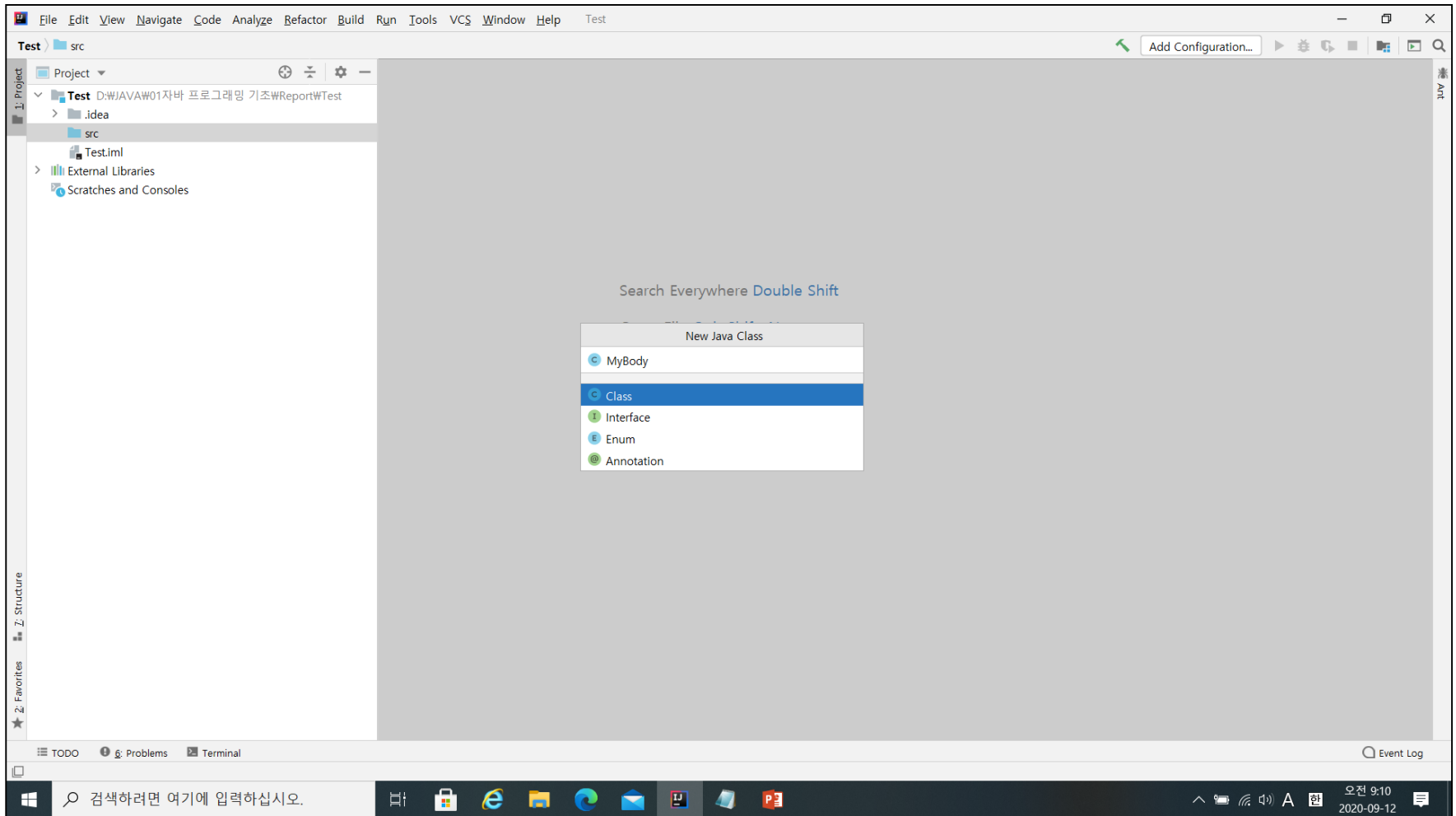
# 다양한 클래스 만들기

## Java Class 만들기



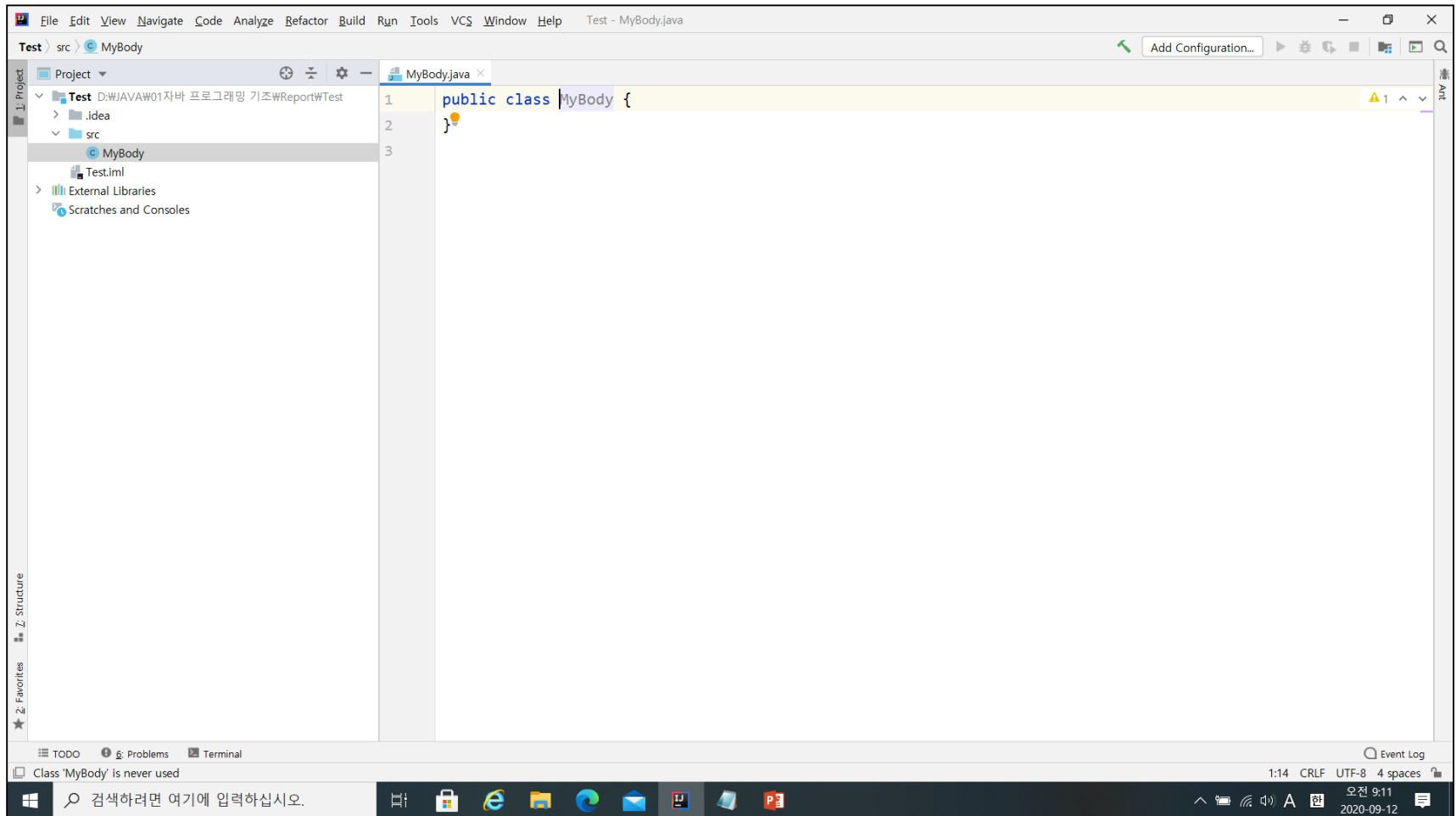
# 다양한 클래스 만들기

## ■ Java Class 만들기



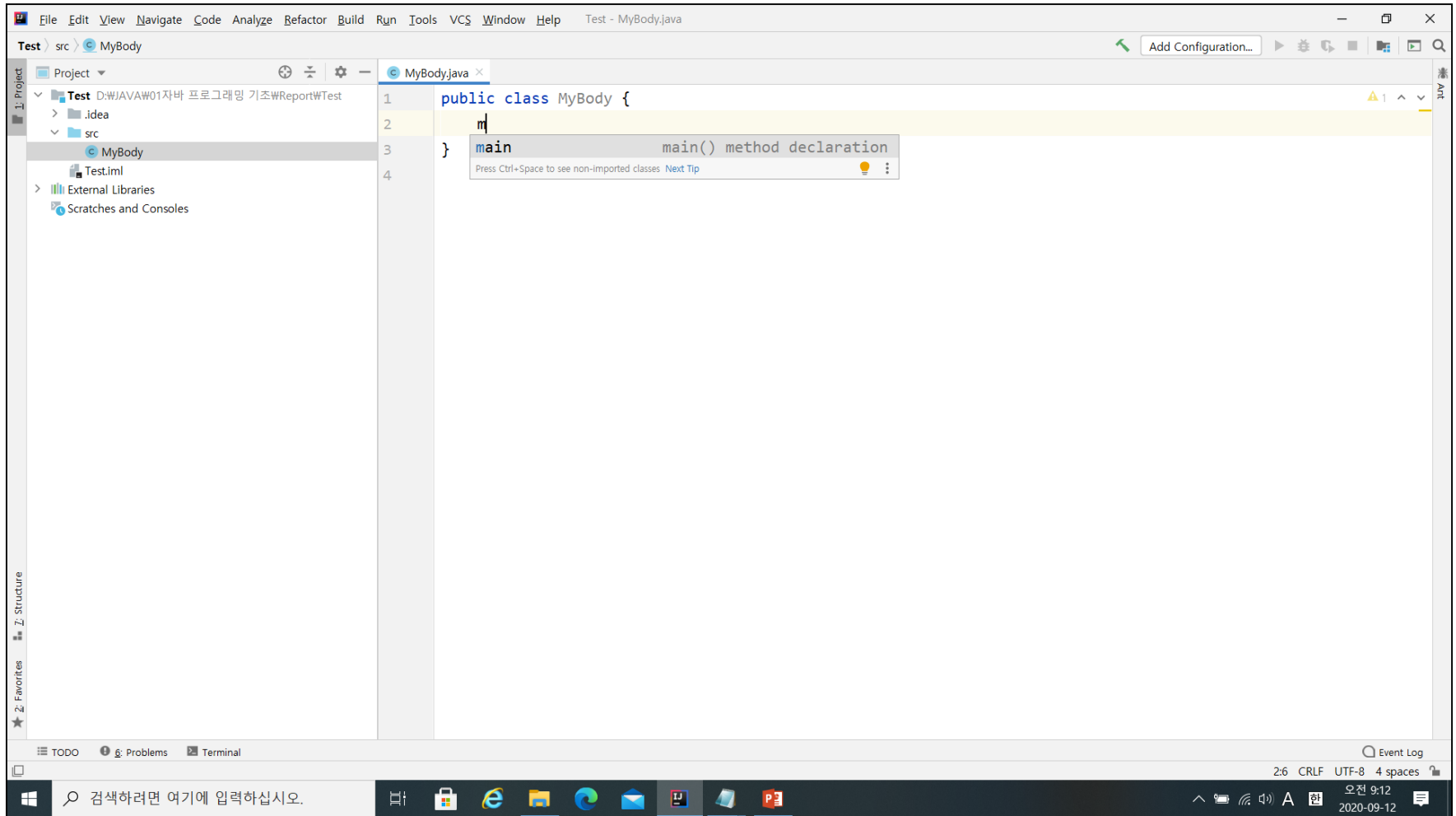
# 다양한 클래스 만들기

## ■ Java Class 만들기



# 다양한 클래스 만들기

## Java Class 만들기



# Character Set

- 대부분의 Programming 언어들은
  - ASCII 또는 ASCII의 변종 또는 EBCDIC Code 사용
- JAVA는 16bit 문자 Code인 Unicode 사용
  - Unicode의 제일 처음 256개의 문자와 Latin-1의 처음 128자는 7-bit ASCII 문자 Code와 동일
  - JAVA는 ASCII 나 ISO Latin-1 File을 읽을 수가 있지만 이들을 읽음과 동시에 즉시 Unicode로 변환
  - 현재 Unicode를 지원하는 편집기가 별로 없기 때문에 JAVA는 Unicode 문자들을 인식 및 변환하기 위해서 제어 문자열(escape sequence) ~~W~~udddd을 사용



# Comments

- // 한 줄 주석

- /\*  
한 줄 또는 여러 줄 주석  
\*/

- // comment 와 /\* \*/ comment 는 C++에서의 comment와 쓰는 방법이 동일

- /\*\*  
document 생성용 주석  
javadoc 명령으로 API 문서를 자동 생성시  
킬 때 사용  
\*/

# Tokens

- Token이란 언어에서 사용하는 기본 단어들을 말함
- JAVA에서 space, tab, newline, form feed와 같은 공백 문자(white space)는 Token들을 구별하기 위해서 또는 문자열의 일부로 사용된 경우 이외에는 별 다른 의미가 없음
- JAVA의 토큰 분석기(greedy tokenizer)는 Token을 얻기 위해서 한번에 가능한 많은 문자들을 취함
  - 예) test = a+++++ test; // INVALID
  - test = a++ ++ + test; // INVALID
  - test = a++ + ++test; // VALID
  - ++++++ 를 ++ + ++이 아니라 ++ ++ +로 해석

# 세미콜론(;),블록(Block),공백

- ;( 세미콜론 ), { }(블록), 그리고 공백(whitespace)들은 Code의 가독성을 높여 줌
  - 논리적인 한 문장이 끝날 때 반드시 세미콜론 (;)을 붙여 줌
- Block은 중괄호(curly brace), 즉 "{"와 "}"로 묶여진 부분을 말하며, 복합문을 구성하는 요소
  - 종속된 문장이 다수 개 일 때 "{" 로 시작하고 "}" 닫아 줌
- 공백(Whitespace)은 빈칸, 탭, 빈 줄 등이 있으며, 이러한 공백은 Source Code를 보기 좋게 하고 알아보기 쉽게 하는데 사용됨

# Identifier

- 대문자와 소문자를 구분하고, 길이 제한이 없음
- 문자와 숫자, 밑줄(\_), 달러기호(\$)를 포함할 수 있음
  - 밑줄(\_)과 달러기호(\$)는 Local System의 규칙이 있거나 부득이한 경우가 아니면 Identifier에 사용하지 않는 것이 좋음
  - Unicode를 지원하기 때문에 한글 사용이 가능하지만, 영문자를 사용하는 것이 좋음
- 문자, 밑줄(\_), 달러기호(\$)로 시작할 수 있음
  - 숫자로는 시작할 수 없음
  - 첫 문자 외에는 문자, 숫자, 밑줄(\_), 달러기호(\$) 등을 임의의 순서로 사용할 수 있음
- 공백을 포함할 수 없음
- Keyword는 Identifier로 사용할 수 없음

# Identifier

## ■ 다음은 잘못된 Identifier의 예

잘못된 예	설명
3test @test this num data	숫자로 시작할 수 없음 첫 문자로 사용할 수 있는 특수문자는 '_', '\$' 뿐 임 키워드는 사용할 수 없음 공백은 사용할 수 없음

# Identifier

- 클래스 이름 / 인터페이스 이름
  - 명사나 형용사를 서술적으로 연결하여 사용
  - 첫 글자는 대문자로 표기
  - 연결된 단어의 첫 글자도 대문자로 표기
  - 나머지 문자는 소문자로 표기
  - "\$"(dollar sign)은 내부 클래스에서 특별한 의미가 있기 때문에 사용을 권장하지 않음
- 변수 이름
  - 명사적 의미를 갖게 사용
  - 첫 글자는 소문자, 연결 단어의 첫 글자는 대문자로 표기
  - 나머지 문자는 소문자로 표기
  - 일반적으로 변수 이름에서는 "\_"(underscore character)를 사용하지 않음

# Identifier

- 메소드 이름
  - 동사적 의미를 갖게 사용
  - 첫 글자는 소문자, 연결 단어의 첫 글자는 대문자로 표기
  - 나머지 문자는 소문자로 표기
  - 메소드 이름 뒤에는 한 쌍의 괄호 "( )"가 뒤따름
  - 일반적으로 메소드 이름에서는 "\_"(underscore character) 를 사용하지 않음

# Program의 일반

- Program 3가지 유형

- IPO model

- 입력(데이터) -> 처리 -> 출력(결과) [정보]

- 출력

- 입력 -> 출력

- 입력 -> 처리 -> 출력



# System 클래스(표준 입출력)

- JAVA에서는 Screen과 Keyboard를 통한 입출력 방법인 표준 입출력을 제공
- 표준 입출력을 제공하는 클래스는 java.lang.System으로 멤버 변수인 in, out, err을 이용해서 표준 입력, 표준 출력, 표준 에러를 제공

# System 클래스(표준 입출력)

## ■ System.in

- System.in은 InputStream 형태로 지정되어 있음
- System 클래스는 JVM(JAVA Virtual Machine)을 구성하고 있는 표준 장치를 뜻하는 클래스
- JVM은 그 자체가 완벽한 하나의 Computer Platform을 가정하고 있기 때문에 독립적으로 동작할 수 있는 구조를 표현하기 위하여 표준 입력과 표준 출력을 스스로의 System 클래스에 등록하여 사용
- 여기에서 주목해야 할 부분은 System.in 변수의 타입이 InputStream이라는 점
- InputStream 클래스는 최상위 클래스이면서 추상 클래스. 따라서 InputStream은 객체를 생성할 수 없는 클래스

# System 클래스(표준 입출력)

## ■ System.err

- System.err 객체는 표준 Error 출력 장치를 의미
- 일반적으로 System.out과 마찬가지로 Monitor로 지정되는 경우가 많음
- 일반적인 정상 출력은 System.out으로 출력되고, Error가 발생할 때 알려주어야 할 내용은 System.err로 출력된다고 볼 수 있음
- 이 변수의 타입도 PrintStream 클래스 타입으로 System.out을 사용하는 방법과 동일

# System 클래스(표준 입출력)

## ■ System.out

- System.out 변수는 표준 출력 장치 객체를 가리키는 대표적인 출력 변수
- JAVA를 처음 배우자마자 사용하는 문장 중 하나가 System.out.println() 메소드
- System.out은 PrintStream 타입으로 선언되어 있는데 PrintStream은 OutputStream 클래스의 후손 클래스로 Exception을 안전하게 처리한 메소드로만 구성
- 이런 이유로 System.out 을 이용하여 출력할 때는 try ~ catch 구문을 작성할 필요가 없음
- 출력 스트림 객체이며 print( ), println( ), printf( ) 등의 문자열 출력을 위한 메소드를 제공

# Data 출력 방법

- System.out의 3가지 메소드
  - print() 메소드
  - println() 메소드 = printf(“~~**W**n”)
  - printf() 메소드

# println() 메소드

- System.out.println()
  - System은 클래스 명
  - out은 System 클래스의 속성 명
  - out은 PrintStream 클래스의 객체
  - Printstream 클래스는 프린트용의 많은 메소드를 제공
  - 예)

```
System.out.println("Hello World")
```

클래스 이름과 메소드 이름은 도트(.)로 구분

System.out.println에 있는 점(.) : ‘~의’를 의미

System의 out의 println 메소드

System 클래스의 PrintStream 클래스의 println 메소드

# println() 메소드

- In은 **line**의 약자로 내용 출력 후 자동 개행
- 개발자가 열심히 Programming을 하더라도 결과를 볼 수 있는 명령문을 실행하지 않으면 절대 직접 눈으로 볼 수 없음
- println()은 JAVA에서 기본 출력문으로 가장 많이 사용되며, 출력하고 싶은 내용 그대로를 입력하여 출력함

```
System.out.println("문자열"); // 불변  
System.out.println(변수명);   // 변화  
System.out.println("문자열" + 변수명);
```

# println() 메소드

## ■ 사용 방법

- 문자열을 출력하기 위해서는 큰 따옴표(“”)로 묶어줌
  - 큰 따옴표 없는 것은 모두 변수 이름으로 인식  
(입력한 변수 이름이 없을 경우 Error 발생)
  - 문자열과 변수 이름을 함께 사용할 수 있으나 반드시 '+'로 연결시켜줘야 함
  - 괄호(( ))안의 내용을 출력한 후 자동으로 줄 바꾸는 기능이 있음
- 
- System.out.print()는 줄 바꿈을 하지 않음
  - 대부분의 경우 println()을 쓰고, 줄 바꿈을 하지 말아야 하는 특수한 경우에만 print()를 사용



# printf() 메소드

- 기본 출력문은 println()은 변수의 값을 그대로 출력하므로, 값을 변환하지 않고는 다른 형식으로 출력할 수 없음
- 반면에 printf()는 형 변환자를 통해 변수의 값을 여러 가지 형식으로 변환하여 출력할 수 있음

System.out.printf("출력 서식", 출력할 내용);

- 출력 후 줄 바꿈을 하지 않는다. 줄 바꿈을 하려면 형 변환자 '\n'을 넣어줘야 함
- 출력하려는 값의 수만큼 형 변환자도 사용해야 함
- 출력될 값과 형 변환자의 순서는 일치해야 함
- 형 변환자를 제외한 문자는 입력한 그대로 출력됨

# printf() 메소드

## ■ 출력 서식

%[-][0][n][.m]지시자

- 출력 서식의 지시자를 제외한 나머지는 생략 가능
  - 예) %d식으로 사용 가능
- n : 출력할 전체 자리수 지정(오른쪽 정렬)
  - 예) %3d, 전체 자리수가 3인 정수
- 0 : 전체 자리수가 지정된 경우 왼쪽의 남은 자리에 0을 출력
  - 예) %03d
- - : 전체 자리수가 지정된 경우 왼쪽 정렬하고 빈칸에 공백 출력
- .m : 소수점 아래 자리수 지정. 잘리는 소수점 자리수는 반올림 시켜서 표시
  - 예) 3.2f

# printf() 메소드

## ■ 변환 사양자의 형식

%

- ✓ format 의 종류 (형문자)
- ✓ 정밀도 지정자 (.digit)
- ✓ 폭 지정자 (digit)
  - ✓ 숫자인 경우는 고정된 자릿수
- ✓ 플래그(flag)
  - ✓ +와 무 부호는 오른쪽 기준
  - ✓ -는 왼쪽 기준
- ✓ 변환 사양 기호 (%)

# printf() 메소드

## ■ 형 변환자

형 변환자	설명
%b	args가 null인 경우에는 false를 출력하고, args가 boolean 혹은 Boolean 타입인 경우에는 String.valueOf ( )의 값이 출력. 그렇지 않은 경우에는 true를 출력
%d	10진 정수 형식으로 출력
%o	8진수 정수의 형식으로 출력
%x 또는 %X	16진수 정수의 형식으로 출력
%f	소수점 형식으로 출력
%c	문자형식으로 출력
%s	문자열 형식으로 출력
%n	줄 바꿈 기능
%e 또는 %E	지수 표현식의 형식으로 출력

# printf() 메소드

## ■ Escape Sequence 문자

\n	new line (줄 바꿈 기능)
\r	return (커서를 맨 앞으로)
\t	tab
\b	backspace
\f	프린트 출력 시 줄 변경
\\	\ 사용시
\"	"(큰따옴표) 사용시
'	'(작은따옴표) 사용시

# printf() 메소드

## ■ printf( )의 정밀도 지정자

지정자	출력 변환에 미치는 영향
.10진수	<ul style="list-style-type: none"><li>✓ 숫자인 경우는 소수점 이하의 자리수를 나타냄</li><li>✓ 문자열인 경우는 최대 출력 문자 수를 지정</li><li>✓ 0(ZERO)이면, 숫자인 경우는 소수점 이하를 출력하지 않고, 문자열인 경우는 아무것도 출력하지 못함</li></ul>

# printf() 메소드

## ■ printf( )의 정밀도 지정자

형 문자	정밀도 지정이 변환에 미치는 영향
e, E, f형	<ul style="list-style-type: none"><li>소수점 아래의 유효자리 개수를 나타냄</li><li>마지막 자리 수에서 반올림</li></ul> 예) <code>System.out.printf("%.1f", 367.89);</code> 367.9
g, G형	<ul style="list-style-type: none"><li>소수점 아래의 최대의 유효 자리수를 나타냄</li></ul> 예) <code>System.out.printf("%.1g", 367.89);</code> 4e+02
s형	<ul style="list-style-type: none"><li>출력되는 최대문자의 개수를 의미</li><li>최대문자 이후의 문자는 출력되지 않음</li></ul> 예) <code>System.out.printf("%.3s.", "ABCDEFGH");</code> ABC <code>System.out.printf("%.0s", "ABCDEFGH");</code> 출력 없음
c형	<ul style="list-style-type: none"><li>출력에 아무런 영향을 주지 못함</li></ul>

# printf() 메소드

## ■ printf( )의 폭 지정자

폭 지정자	출력 폭에 대한 영향
10진수	<ul style="list-style-type: none"><li>✓ 인수의 최소 출력 폭을 지정</li><li>✓ 데이터의 자리수가 지정된 것보다 더 작은 경우는 남은 좌측은 공백(blank)으로 채워진다</li><li>✓ 반대로 지정된 자리 수보다 큰 경우는 폭 지정을 무시하고 갖고 있는 자리수로 출력</li><li>✓ 부동 소수점형 데이터의 소수점도 한 자리로 여김</li></ul> <p>예) <code>System.out.printf("/%5d/", 12);</code>    <code>/ 12/</code> <code>System.out.printf("%1d", 123);</code>    <code>123</code></p>



# printf() 메소드

## ■ printf( )의 플래그

플래그	지정 내용
-	<ul style="list-style-type: none"><li>• 왼쪽 끝에 맞추어 결과를 출력. 남은 우측은 공백으로 채움</li><li>• “-” 플래그가 지정되지 않은 경우는 오른쪽을 기준으로 출력</li></ul> 예) <code>System.out.printf("/%-5d/", 12);</code> <code>/12_ _ _/</code> <code>System.out.printf("/%5d/", 12);</code> <code>/_ _ _12/</code> <code>System.out.printf("/%-5.2s/", "ABCDEF");</code> <code>/AB_ _ _/</code>
+	<ul style="list-style-type: none"><li>• 숫자를 출력하는 경우 양수이면 (+)를 음수이면 (-)를 출력</li></ul> 예) <code>System.out.printf("/%+5d/", -123);</code> <code>/_ -123/</code>
아무것도 없음	<ul style="list-style-type: none"><li>• 숫자를 출력하는 경우 양수이면 (+)를 붙이지 않고 공백(blank) 음수이면 (-)를 붙여서 출력</li></ul> 예) <code>System.out.printf("/%5d/", 123);</code> <code>/_ _123/</code> <code>System.out.printf("/%5d/", -123);</code> <code>/_ -123/</code>
0(zero)	<ul style="list-style-type: none"><li>• 숫자 값에 대하여 출력 폭의 남은 왼쪽 공간을 0으로 채워서 출력</li></ul> 예) <code>System.out.printf("/%05d/", 123);</code> <code>/00123/</code>
#	<ul style="list-style-type: none"><li>• 형 문자에 따라서 형의 교체를 지정</li></ul>

# printf() 메소드

## ■ printf( )의 #플래그

형 문자	# 플래그가 데이터 형에 미치는 영향
o	<ul style="list-style-type: none"><li>8진수를 의미하는 O을 인수 앞에 붙여줌 예) <code>System.out.printf("%#o", 8);</code>      010</li></ul>
x, X	<ul style="list-style-type: none"><li>16진수를 의미하는 0x, 0X가 인수 앞에 붙여줌 예) <code>System.out.printf("%#x", 16);</code>      0x10</li></ul>
e, E, f	<ul style="list-style-type: none"><li>소수점 이하의 자리수가 없어도 결과는 항상 소수점을 포함 예) <code>System.out.printf("%#e", 123.0);</code> 1.23000e+02</li></ul>

# String.format( )

- 숫자를 표기할 때 자리점(,)를 찍어 써서 달러 표시하는 법

```
int val = 10000000;
```

```
String str = String.format("$%,d", val);
```

```
System.out.println(str);
```

\$10,000,000

- 포매팅 대상의 인자가 여러 개이고 문자열을 포함하는 법
- argument\_index 옵션 : \$

```
System.out.println(
```

```
String.format("스마트폰의 가격은 %2$,d 원 이며,  
재고가 %1$d대 남았습니다.", 3, 890000));
```

스마트폰의 가격은 890,000 원 이며, 재고가 3대 남았습니다.

# String.format( )

- '<'를 사용하여 하나의 인자로 모두 사용 가능

```
double val = 10000.432342342;  
String str = String.format("%f    %<,f    %<,.2f", val);  
System.out.println(str);
```

10000.432342    10,000.432342    10,000.43

# 표준 입출력 예제 1

- 다음과 같이 출력되는 Program을 작성하세요.



- 출력 방법만 알고 있으면 된다

# 표준 입출력 예제 1

```
public class Sample1 {  
    public static void main(String[] args) {  
  
        이곳에 Program 작성  
  
    }  
}
```

# 표준 입출력 예제 2

- 다음과 같은 출력을 생성하는 Program을 작성해봅시다



# 표준 입출력 예제 2

```
public class Sample2 {  
    public static void main(String[ ] args) {  
  
    }  
}
```

이곳에 Program 작성



# 표준 입출력 예제 3

- 다양한 printf() 메소드를 사용하여 다음을 Program 하여라
- 빨강색은 데이터로 만들어 사용



# 표준 입출력 예제 3

```
public class Sample3 {  
    public static void main(String[ ] args) {
```

이곳에 Program 작성

```
}  
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 표준 입출력 예제 4

- 다양한 printf() 메소드를 사용하여 다음을 Program 하여라
- 빨강색은 데이터로 만들어 출력



# 표준 입출력 예제 4

```
public class Sample4 {  
    public static void main(String[ ] args) {
```

이곳에 Program 작성하세요

```
    }  
}
```

# 표준 입출력 예제 5

■ 다음 Program에서 이름은 ?

```
public static void main(String args[]) {  
    String 이름 = "홍길동";  
    System.out.println("안녕하세요. " + 이름 + " 님!");  
}
```

변수를 한글로도 작성 가능

## 표준 입출력 예제 6

■ 다음 문장들의 출력 결과를 적으세요. Error가 있는 문장의 경우, 괄호 안에 ‘오류’라고 적으시오.

- ① `System.out.println("1" + "2")` → ( )
- ② `System.out.println(true + "")` → ( )
- ③ `System.out.println('A' + 'B')` → ( )     'A' = 65  
`System.out.println('1' + 2)` → ( )     '1' = 49
- ④ `System.out.println('1' + '2')` → ( )
- ⑤ `System.out.println('J' + "ava")` → ( )
- ⑥ `System.out.println(true + null)` → ( )

# 표준 입출력 예제 6

- ① : 문자열("1") + 문자열("2") = 문자열 ("12")
- ② : `true + "" → "true" + "" → "true"`
- ③ : `'A'+'B'`의 경우, `char + char → int + int → int`의 과정을 통해 최종결과는 `int`형 정수값이 된다.  
문자와 문자의 덧셈연산의 결과는 `int`형 정수값이 된다.  
왜냐하면 `int`형보다 작은 타입(`byte`, `char`, `short`)은 `int`형으로 변환된 후에 덧셈연산이 진행되기 때문이다  
`'A'+'B' → 65 + 66 → 131` 'A'와 'B'의 문자코드의 값은 각각 65와 66이다.
- ④ : `'1' + 2 → 49 + 2 → 51` '1'의 문자코드의 값은 49이다
- ⑤ : `'1'+'2' → 49 + 50 → 99`
- ⑥ : `'J' + "ava" → "J" + "ava" → "Java"`
- ⑦ : `true + null → 오류`

# 표준 입출력 예제 7

■ 다음 Program의 출력 결과를 설명하여라 ?

```
public static void main(String[] args) {  
    final int PAGES = 365;
```

```
    System.out.printf("/%d/\n", PAGES);  
    System.out.printf("/%2d/\n", PAGES);  
    System.out.printf("/%10d/\n", PAGES);  
    System.out.printf("/%-10d/\n", PAGES);  
    System.out.printf("/%+10d/\n", PAGES);  
    System.out.printf("/%3o/\n", PAGES);  
    System.out.printf("/%3x/\n", PAGES);  
    System.out.printf("\n%c\n", PAGES);  
}
```

```
/365/  
/365/  
/          365/  
/365      /  
/          +365/  
/555/  
/16d/  
ü
```



# 표준 입출력 예제 7

- ① : 변환 사양자 '%d'는 변경자 없는 기본형으로 출력 데이터의 정수와 같은 폭의 10진 정수(decimal)로 출력
- ② : 변환사양에서 최소 폭을 2자리로 지정하였지만 출력 데이터의 자릿수 보다 작게 지정되어 있으므로 지정된 출력 폭은 무시되고 자동적으로 출력 데이터의 자릿수만큼 확장되어 출력
- ③ : 대응된 인수를 최소 10 자리에 표현하면서 일반적인 오른쪽 맞춤을 함
- ④ : 최소 10 자리에 표현하면서 플래그 '-'에 의해서 왼쪽 맞춤을 함
- ⑤ : 최소 10 자리에 표현하면서 플래그 '+' 때문에 부호를 표시 함
- ⑥⑦ : 변환 사양자 '%o'는 대응되는 인수 365를 8진 정수(octal)로 변환하여 출력 하고, '%x'는 데이터를 16진 정수(hexadecimal)로 변환하여 출력

# 표준 입출력 예제 8

■ 다음 Program의 출력 결과를 설명하여라 ?

```
public static void main(String[] args) {  
    final double RENT = 1234.56;
```

```
    System.out.printf("/%f/\Wn", RENT);  
    System.out.printf("/%e/\Wn", RENT);  
    System.out.printf("/%4.2f/\Wn", RENT);  
    System.out.printf("/%3.1f/\Wn", RENT);  
    System.out.printf("/%10.3f/\Wn", RENT);  
    System.out.printf("/%10.5e/\Wn", RENT);  
    System.out.printf("/%+4.2f/\Wn", RENT);  
    System.out.printf("/%010.2f/\Wn", RENT);
```

```
}
```

```
/1234.560000/  
/1.234560e+03/  
/1234.56/  
/1234.6/  
/ 1234.560/  
/1.23456e+03/  
/+1234.56/  
/0001234.56/
```

# 표준 입출력 예제 8

- ① : 변환 사양자 '%f'는 지수를 포함하지 않은 부동 소수점 (float, double)형으로 출력. 일반적으로 소수점 이하 6 자리까지 출력
- ② : 변환 사양자 '%e'나 '%E'는 지수를 포함한 부동 소수점 형으로 출력
- ③ : 변환 사양자 앞의 4.2는 총 4 자리에서 소수점 아래 자릿수를 2 자리로 변환하라는 지시자 이지만 전체 자리수가 원래 값보다 작으므로 무시되고 소수점 이하 2 자리만 유지. 즉, 주어진 데이터의 자리수가 꼭 지정자보다 큰 경우 꼭 지정자는 무시되고 데이터의 본래 자릿수를 확보하여 출력
- ④ : 총 3 자리에 소수점 아래 자릿수를 1 자리로 변환하라는 변환 지시자 이지만 전체 자리수가 인수 값보다 작으므로 무시되고 소수점 이하 1 자리만 유지하려고 소수점 2 번째 자리에서 반올림

# 표준 입출력 예제 8

- ⑤ : 변환 사양자 '%e'는 지수를 포함한 부동 소수점형으로 출력하는 데 소수점을 최대 5자리로 표현
- ⑥ : 총 4 자리에 소수점 아래 자릿수를 2 자리로 변환하고 부호를 삽입하라는 변환 지시자. 전체 자리수가 대응된 인수 값보다 작으므로 무시 됨
- ⑦ : 총 10 자리에 소수점 아래 자릿수를 2 자리로 변환하고 남은 앞자리에는 0(zero)을 채우라는 변환 지시자. 그리고 소수점 이하 3 번째 자리에서 반올림

# 표준 입출력 예제 9

■ 다음 Program의 출력 결과를 설명하여라 ?

```
public static void main(String[] args) {  
    final String MSG = "KyungBok University";  
  
    System.out.printf("/%2s/\n", MSG);  
    System.out.printf("/%22s/\n", MSG);  
    System.out.printf("/%22.5s/\n", MSG);  
    System.out.printf("/%-22.5s/\n", MSG);  
}
```

```
/KyungBok University/  
/  KyungBok University/  
/                               Kyung/  
/Kyung                          /
```

# 표준 입출력 예제 9

- ① : 변환 사양자 앞의 폭 지정자 4가 있다. 이것은 최소 4자리 문자열로 출력하라는 의미인데 주어진 문자열의 길이가 4 자리를 넘으므로 폭 지정자는 무시되고 원래 값의 크기로 출력
- ② : 문자열을 위해서 22 자리가 마련되고 오른쪽에 맞추고, 왼쪽의 남은 자리는 공백으로 채움
- ③ : 문자열을 위해서 22 자리가 마련되고 문자열은 최대 4자리 문자열을 오른쪽 맞춤으로 출력하고, 남은 왼쪽의 자리는 공백으로 채움
- ④ : 주어진 문자열을 위해서 최소 22 자리가 마련되고 최대 4 자리 문자열을 왼쪽 맞춤으로 출력하고, 남은 오른쪽의 자리는 공백으로 채움

# 표준 입출력 예제 10

■ 다음 Program의 출력 결과를 설명하여라 ?

```
public static void main(String[] args) {  
    System.out.printf("\n\t This is \n sample\n"  
                       "Escape sequence.\n");  
    System.out.printf("\t\n\t\t = tab\n");  
    System.out.printf("\t'0x13' = %d\n", 0x13);  
    System.out.printf("\t\t 0x13 = %d\n", 0x13);  
    System.out.printf(" This is %d%% of list.\n", 80);  
}
```

This is “sample” Escape sequence.

“\t” = tab

‘0x13’ = 19

0x13 = 19

This is 80% of list.

# 표준 입출력 예제 11

■ 다음 Program의 출력 결과를 설명하여라 ?

```
public static void main(String[] args) {  
    System.out.printf("This is a sample program. \n");  
    System.out.printf("This\t is\t a\t sample\t program. \n");  
    System.out.printf("This\r is\r a\r sample\r program. \n");  
    System.out.printf("This \t\b is \t\b a \t\b sample \t\b program.\n");  
    System.out.printf("This\t\t is\t\t a\t\t" sample program. \n");  
    System.out.printf("This\t\b is\t\t a\t\t sample\t\r program. \n");  
}
```

This is a sample program.

This      is          a          sample          program.  
program.

Thisisasampleprogram.

This\t is' a" sample program.

Thi is      a  
program.



# 표준 입출력 예제 12

■ 다음 Program의 출력 결과를 설명하여라 ?

```
public static void main(String[] args) {  
    System.out.printf("%d + %d 은 %d 입니다.\n", 10, 20, 30);  
    System.out.printf("%2$d + %1$d 은 %3$d 입니다.\n", 10, 20, 30);  
}
```

10 + 20 은 30 입니다.  
20 + 10 은 30 입니다.

인자 순서는 argument의 치환 순서를 지정하는 것  
1\$는 첫 번째, 2\$는 두 번째, 3\$는 세 번째 argument를 의미

# 표준 입출력 예제 13

■ 다음 Program의 출력 결과를 설명하여라 ?

```
public static void main(String[] args) {  
    float x = 21.567f;  
  
    System.out.printf("/%6.2f/\n", x);  
    System.out.printf("/%1.3f/\n", x);  
    System.out.printf("/%1.1f/\n", x);  
    System.out.printf("/%.0f/", x);  
}
```

```
/ 21.57/  
/21.567/  
/21.6/  
/22/
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 표준 입출력 예제 14

■ 다음 Program의 출력 결과를 설명하여라 ?

```
public static void main(String[] args) {  
    System.out.printf("/%5d/Wn", 12);  
    System.out.printf("/%1d/Wn", 123);  
  
    System.out.printf("/%-5d/Wn", 12);  
    System.out.printf("/%5d/Wn", 12);  
    System.out.printf("/%+5d/Wn", 123);  
  
    System.out.printf("/%5d/Wn", 123);  
    System.out.printf("/%5d/Wn", -123);  
    System.out.printf("/%05d/Wn", 123);  
  
    System.out.printf("/%#o/Wn", 8);  
    System.out.printf("/%#x/Wn", 16);  
    System.out.printf("/%#e/Wn", 123.0);  
}
```

```
/ 12/  
/123/  
  
/12  /  
/ 12/  
/ +123/  
/ 123/  
/ -123/  
/00123/  
/010/  
/0x10/  
/1.230000e+02/
```

# 표준 입출력 예제 15

■ 다음 Program의 출력 결과를 설명하여라 ?

```
public static void main(String[] args) {  
    int test = 56;  
  
    System.out.printf("/%4d/\n", test);  
    System.out.printf("/%-4d/\n", test);  
    System.out.printf("/%05d/\n", test);  
    System.out.printf("/%-5d/\n", test);  
    System.out.printf("/%+5d/\n", test);  
}
```

```
/ 56/  
/56 /  
/00056/  
/56 /  
/ +56/
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 표준 입출력 예제 16

■ 다음 Program의 출력 결과를 설명하여라 ?

```
public static void main(String[] args) {  
    System.out.printf("/%.1f/Wn", 367.89);  
    System.out.printf("/%.1g/Wn", 367.89);  
    System.out.printf("/%.3s/Wn", "ABCDEFGH");  
    System.out.printf("/%.0s/Wn", "ABCDEFGH");  
    System.out.printf("/% -5.2s/Wn", "ABCDEFGH");  
}
```

```
/367.9/  
/4e+02/  
/ABC/  
//  
/AB  /
```

# 표준 입출력 예제 17

■ 다음 Program의 출력 결과를 설명하여라 ?

```
public static void main(String[] args) {  
    System.out.println("" + 7 + 7);  
    System.out.println(7 + "");  
    System.out.println(7 + " ");  
    System.out.println(" " + 7);  
    System.out.println(7 + "7");  
    System.out.println(7 + 7 + "");  
    System.out.println(" " + 7.5);  
    System.out.println(true + "" + 7);  
    System.out.println(null + "" + 7);  
}
```

```
77  
7  
7  
7  
77  
14  
7.5  
true7  
null7
```

# Data 입력 방법

- Scanner 클래스
  - Scanner 객체 생성
  - Scanner 클래스 메소드 사용
  - Scanner 객체 닫기
- System.in.read() 메소드 사용하기
  - 문자 단위 입력
- BufferedReader 클래스
  - File 처리 방법
- JOptionPane 클래스
  - GUI 방식

# Scanner 클래스

- System.in
  - Keyboard로부터 숫자, 문자 또는 문자열 등의 Data를 입력하기 위해서 사용하는 입력 Stream 객체
- System.in 객체와 Scanner 클래스
  - java.util 패키지에 속해 있는 Scanner 클래스는 여러 종류의 Data Type의 값을 입력할 수 있는 방법을 제공

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

```
String name = scan.read( );    // 문자열을 입력하고 name에 저장  
int age = scan.readInt( );     // 정수를 입력하고 age에 저장
```

홍길동 20 ↵

⇒ name 변수에 “홍길동”이 age 변수에 “20”이 저장



# Scanner 클래스

- Scanner는 사용자가 입력하는 값을 공백('Wt', 'Wf', 'Wr', ' ', 'Wn')으로 구분하여 읽음
- 탭(tab), 띄어쓰기 등으로 구분하여 읽음
- Program을 실행시키고 콘솔 창에 문자열을 입력하고 'Enter'를 누르면 됨

# Scanner 클래스

## ■ 메소드

메소드	설명
next()	입력 토큰을 String(띄어쓰기로 구분)으로 반환
nextLine()	입력 토큰을 String(한 문장 전체, 엔터키로 구분)으로 반환
nextFloat()	입력 토큰을 float(실수형)으로 반환
nextDouble	입력 토큰을 double(실수형)으로 반환
nextBoolean()	입력 토큰을 boolean(논리형)으로 반환
nextInt()	입력 토큰을 int(정수형)으로 반환
nextLong()	입력 토큰을 long(정수형)으로 반환
nextByte()	입력 토큰을 byte(정수형)으로 반환
nextShort()	입력 토큰을 short(정수형)으로 반환

# Scanner 클래스

- 문자열을 입력 받을 때 사용하는 함수 2개

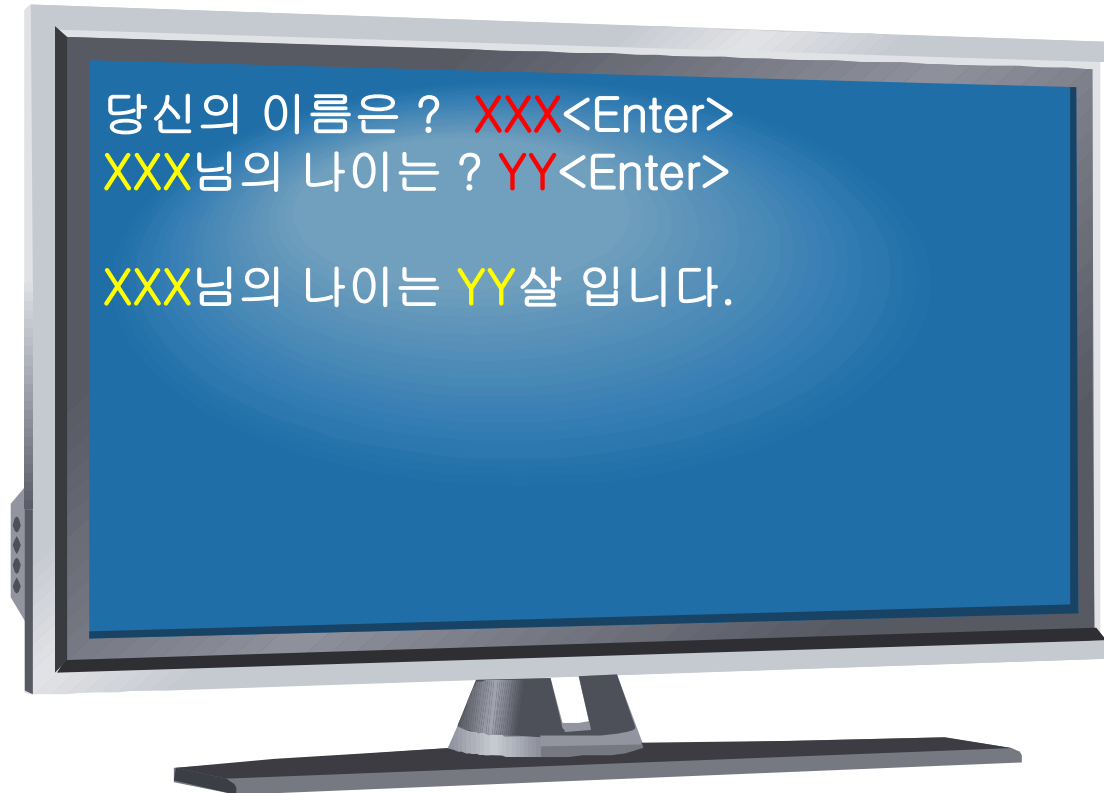
```
String str = scn.next();  
String strLine = scn.nextLine();
```

- 2개의 차이점은 무엇인가?
  - next()는 공백 단위로 문자열을 읽음
  - nextLine()는 Enter 단위로 문자열을 읽음  
(Enter 입력은 `\r\n`으로 구성된 두개의 문자이다)
  - 그러므로 next()와 nextLine()을 섞어서 사용할 때 주의  
해야 함

# Scanner 클래스

- Scanner 클래스 주의 사항
  - Scanner 클래스의 객체를 만들려면 표준 입력 Stream을 나타내는 System.in이라는 이미 정의 되어있는 객체를 전달해야 함 (File로부터의 입력은 File 클래스의 객체를 전달할 수도 있음)
  - 특정 Data Type의 값을 읽기 위해 사용하는 메소드는 **next~()** 메소드
    - 예) nextInt(), nextDouble() 등과 같은 메소드를 사용
  - 문자열, 한 줄(Enter Key 기준으로)을 모두 읽기 위해서는 **nextLine()** 메소드 사용
  - 단일 문자(char)를 읽기 위해서 **next()**와 **charAt()** 메소드를 함께 사용

# Scanner Data 입력 예제 1



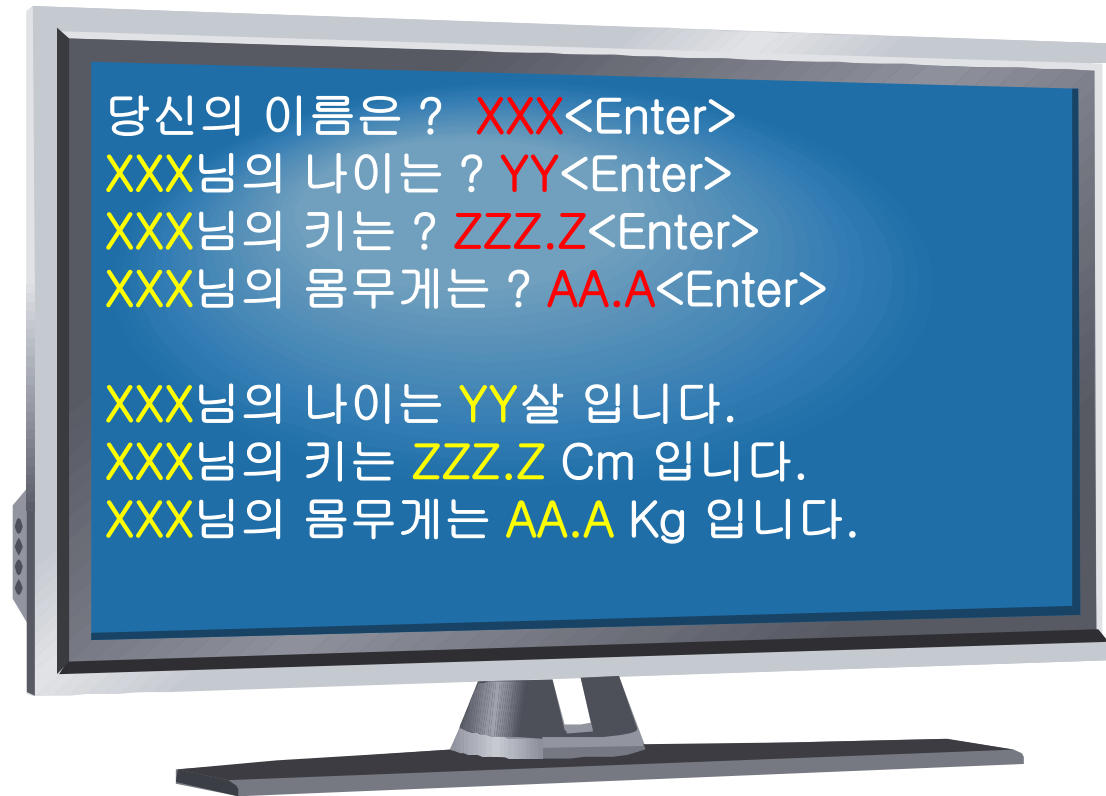
# Scanner Data 입력 예제 1

```
import java.util.Scanner;
public class InputName {
    public static void main(String[] args) {
        Scanner test = new Scanner(System.in);
        String name;
        int 나이;

        System.out.print(" 당신의 이름은 ? "); // Prompt
        name = test.next( );
        System.out.printf(" %s님의 나이는 ? ", name);
        나이 = test.nextInt( );
        System.out.printf(" %s님의 나이는 %d살 입니다.\n", name,
                                                                    나이);
    }
}
```



# Scanner Data 입력 예제 2



# Scanner Data 입력 예제 2

```
import java.util.Scanner;
public class InputName1 {
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        String name;
        int age;
        float height, weight;

        System.out.print("당신의 이름은 ? ");
        name = keyboard.nextLine();
        System.out.printf("%s님의 나이는 ? ", name);
        age = keyboard.nextInt();
        System.out.printf("%s님의 키는 ? ", name);
        height = keyboard.nextFloat();
        System.out.printf("%s님의 몸무게는 ? ", name);
        weight = keyboard.nextFloat();
    }
}
```





# Scanner Data 입력 예제 2

```
System.out.printf("%s님의 나이는 %d살 입니다.\n", name, age);
System.out.printf("%s님의 키는 %.1f Cm 입니다.\n",
                  name, height);
System.out.printf("%s님의 몸무게는 %.1f Kg 입니다.\n",
                  name, weight);
}
}
```

# Scanner Data 입력 예제 3

- 다음과 같이 입 출력되는 프로그램을 작성 하여라.

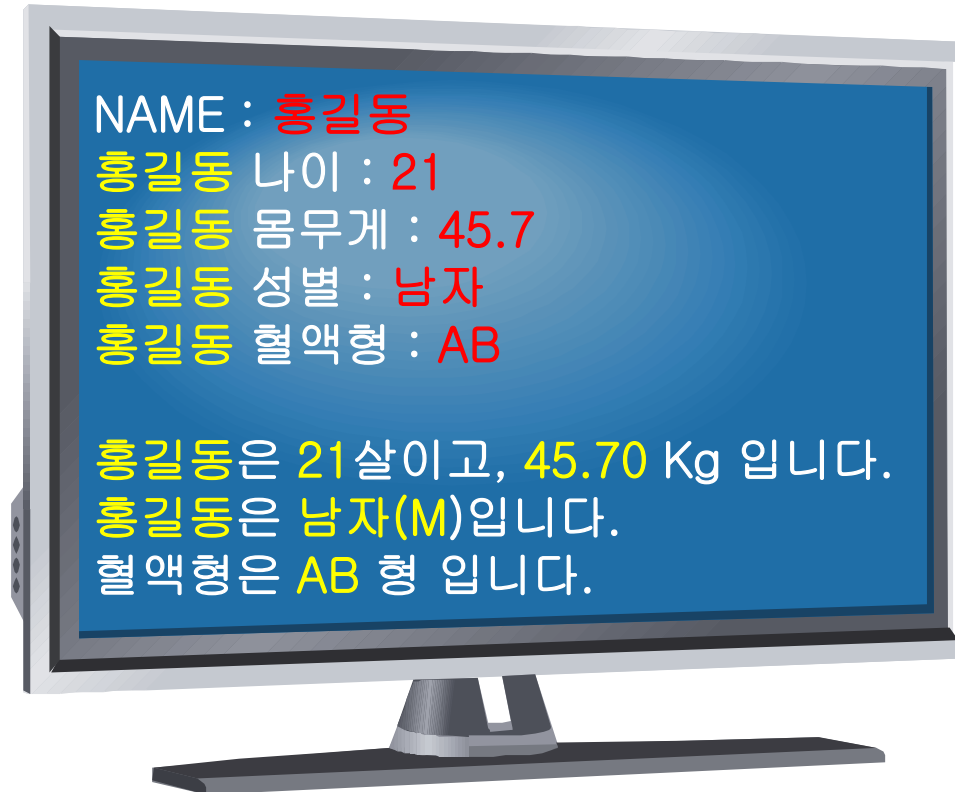


# Scanner Data 입력 예제 3

[illegible]

# Scanner DATA 입력 예제 4

- 다음 프로그램을 Scanner를 사용하여 프로그램 하여라



# Scanner사용 데이터 입력 예제 4

```
public class InputName4 {  
    public static void main(String[] args) throws Exception {  
        Scanner keyboard = new Scanner(System.in);  
        String name, blood;  
        int 나이;  
        float weight;  
        char gender, temp;  
  
        System.out.print(" 당신의 이름은 ? ");  
        name =   
        System.out.printf(" %s 나이 : ", name);  
        나이 =   
        System.out.printf(" %s 몸무게 : ", name);  
        weight =   
        System.out.printf(" %s 성별 : ", name);  
        gender = 
```



# Scanner Data 입력 예제 4

```
System.out.printf(" %s 혈액형 : ", name);  
blood =   
temp = (gender == '남') ? 'M' : 'F';  
System.out.printf(" %s은 %d살이고, %.2f Kg 입니다.\n",  
                                                             name, 나이, weight);  
System.out.printf(" %s은 %c자(%c)입니다.\n", name, gender,  
                                                             temp);  
System.out.printf(" 혈액형은 %s 형 입니다.\n", blood);  
}  
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 문자열에서 Scanner 예제 5

- 다음과 같이 입 출력되는 프로그램을 작성 하여라.



키보드가 아닌,  
문자열을 대상

"12 5 7"

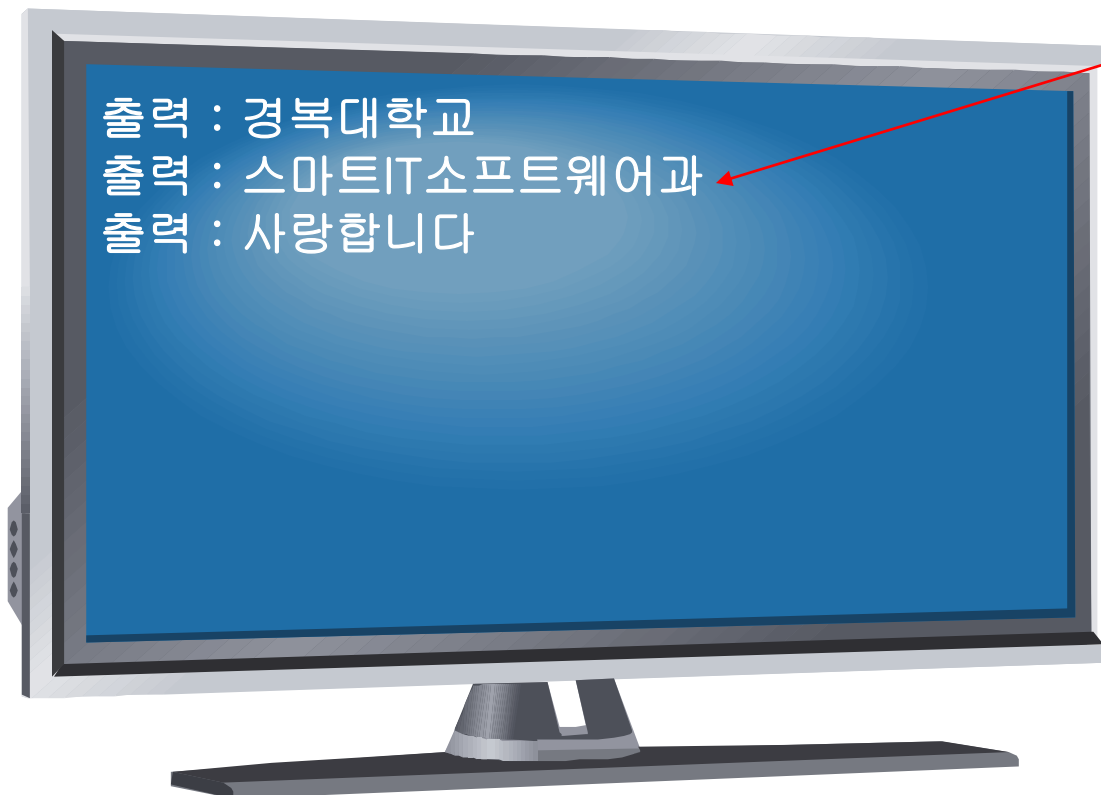
# 문자열에서 Scanner 예제 5

```
public static void main(String[] args) {  
    String source = "12 5 7";  
    Scanner input = new Scanner(source);  
  
    int num1 = input.nextInt();  
    int num2 = input.nextInt();  
    int num3 = input.nextInt();  
    int sum = num1 + num2 + num3;  
  
    System.out.printf("문자열에 저장된 %d, %d, %d의 합은 %d\n",  
        num1, num2, num3, sum);  
}
```



# 문자열에서 Scanner사용 예제 6

- 다음과 같이 입 출력되는 프로그램을 작성 하여라.



키보드가 아닌,  
문자열을 대상

```
"WnWn경북대학교Wn스마트IT소프트웨어과Wn사랑합니다Wn";
```

# 문자열에서 Scanner 예제 6

```
public static void main(String[] args) {  
    String data = "\n\n경북대학교\n스마트IT소프트웨어과\n사랑합니다\n";  
  
    Scanner input = new Scanner(data);  
  
    String str;  
    while (input.hasNext()) {  
        str = input.next();  
        System.out.println("출력 : " + str);  
    }  
}
```

# Scanner Data 입력 예제 7



주소 입력 잘 되나요?

# Scanner Data 입력 예제 7

```
public static void main(String args[]) throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    String name, address;  
    int age;  
  
    System.out.println("당신의 이름을 입력하세요.");  
    name = keyboard.next();  
    System.out.println("당신의 나이를 입력하세요.");  
    age = keyboard.nextInt();  
    System.out.println("당신의 주소를 입력하세요.");  
    address = keyboard.nextLine(); //address = keyboard.next();  
  
    System.out.println("이름 : " + name);  
    System.out.println("나이 : " + age);  
    System.out.println("주소 : " + address);  
}
```



# Scanner Data 입력 예제 7

- 주소를 받아들이지 못하는 원인과 대책

- 원인

- nextInt() 메소드를 이용하여 나이를 입력 받으면서 [Enter] Key를 입력하면 값은 나이 변수에 전달되지만 [Enter("WnWr")] Key 값은 입력 Buffer에 남아 있다가 다음의 nextLine() 메소드에서 받아 들음

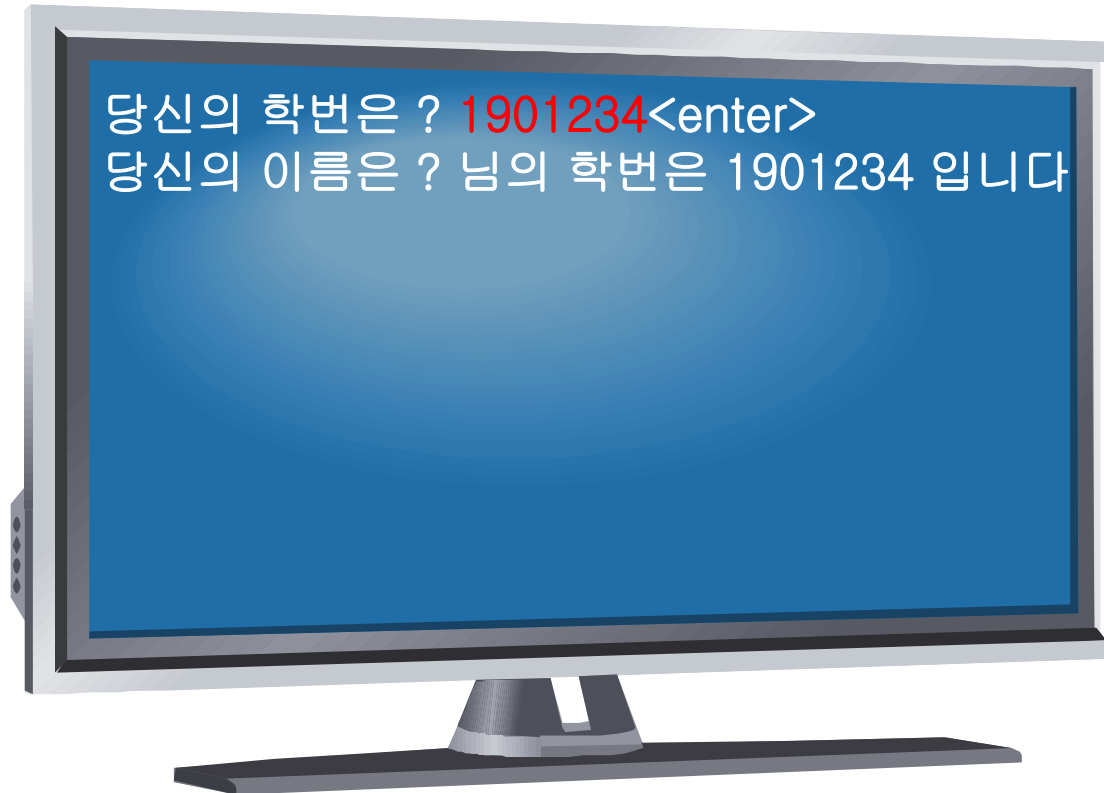
- 대책

- 입력 Buffer에 남아 있는 [Enter] Key를 읽어 버리면 되므로 nextLine( ) 메소드를 한번 더 읽음

# Scanner Data 입력 예제 7-1

```
public static void main(String args[]) throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    String name, address;  
    int age;  
  
    System.out.println("당신의 이름을 입력하세요.");  
    name = keyboard.next();  
    System.out.println("당신의 나이를 입력하세요.");  
    age = keyboard.nextInt();  
    * keyboard.nextLine();  
    System.out.println("당신의 주소를 입력하세요.");  
    address = keyboard.nextLine();    //address = keyboard.next();  
  
    System.out.println("이름 : " + name);  
    System.out.println("나이 : " + age);  
    System.out.println("주소 : " + address);  
}
```

# Scanner Data 입력 예제 8



이름을 입력 받을 수가 없네요?

# Scanner Data 입력 예제 8

```
public static void main(String args[]) {  
    Scanner keyboard = new Scanner(System.in);  
    String hakbun;  
    String name;  
  
    System.out.printf(" 당신의 학번은 ? ");  
    hakbun = keyboard.next();  
    System.out.printf(" 당신의 이름은 ? ");  
    name = keyboard.nextLine();  
  
    System.out.printf("%s님의 학번은 %s 입니다\n", name, hakbun);  
}
```

이름을 입력할 수 없을 텐데



# 문자열에서 Scanner 예제 9

```
public static void main(String[] args) throws IOException {  
    String data = "1701008Wn홍길동";  
    Scanner string = new Scanner(data);  
  
    int num = string.nextInt();  
    String name = string.nextLine();  
  
    System.out.println("학번 : " + num);  
    System.out.println("이름 : " + name);  
}
```

학번 : 1701008  
이름:

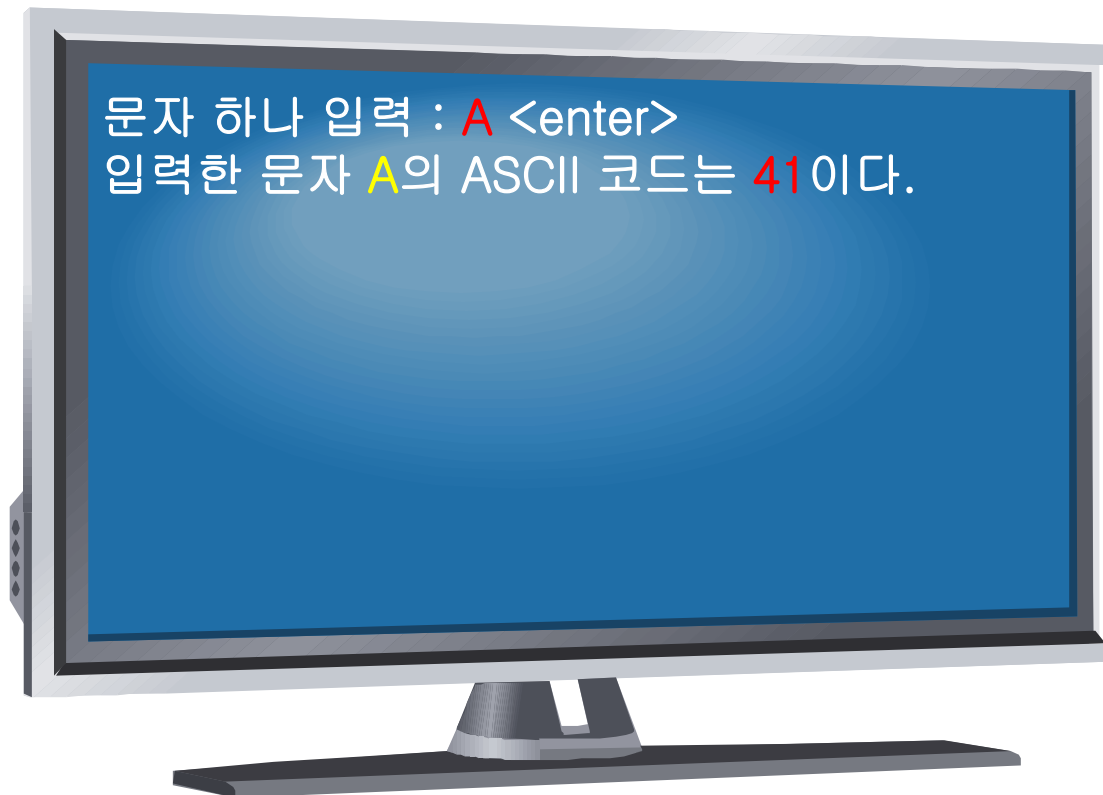
이 Program은 올바른 것 같지만, nextInt()와 nextLine()을 섞어 사용했을 때 범하기 쉬운 Error

# 문자열에서 Scanner 예제 10

- 보통 Keyboard를 통한 입력은 한 줄 단위 입력을 많이 하기 때문에 `nextLine()`으로 입력 받아, 필요한 타입으로 변환하여 입력하는 방법이 좋음

```
public static void main(String[] args) {  
    String data = "21\n홍길동";  
    Scanner string = new Scanner(data);  
  
    int age = Integer.parseInt(string.nextLine());  
    String name = string.nextLine();  
  
    System.out.println("나이 : " + age);  
    System.out.println("이름 : " + name);  
}
```

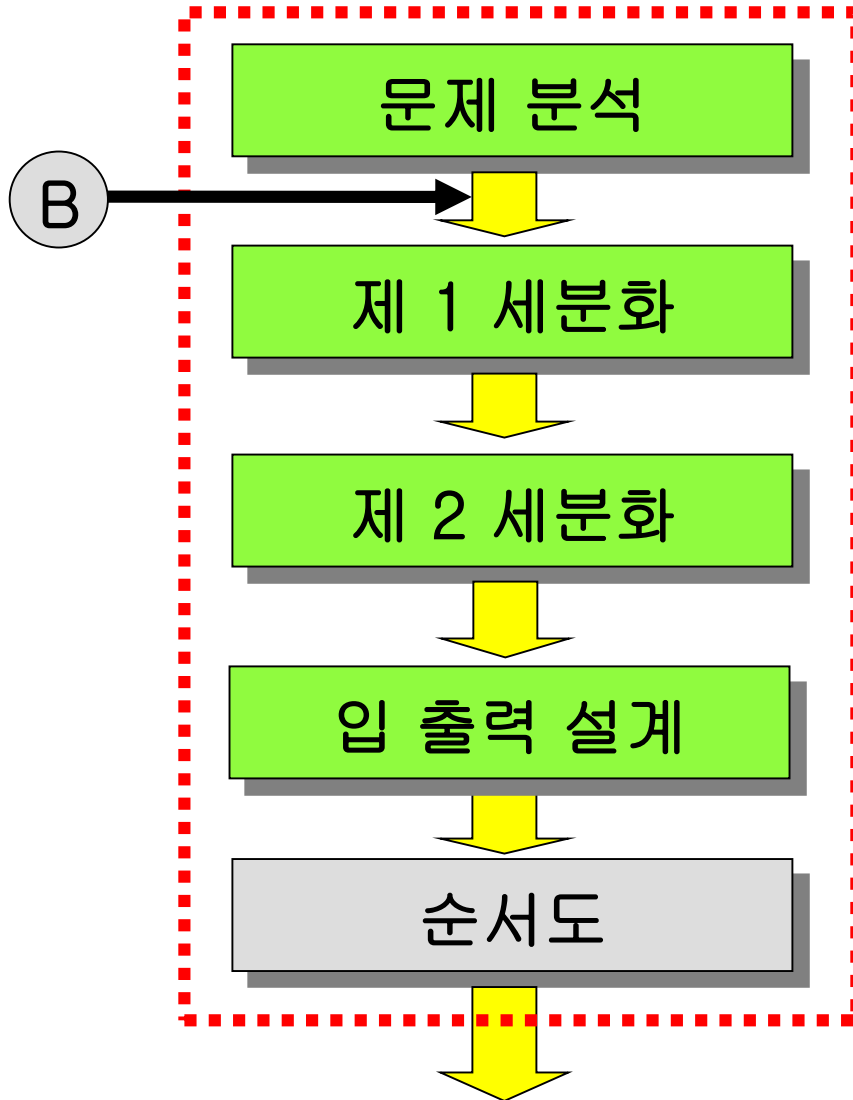
# Scanner 문자 입력 예제 11



# Scanner 문자 입력 예제 11

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
    char ch;  
  
    System.out.printf("문자 하나 입력 : ");  
    ch = scanner.next().charAt(0);  
  
    System.out.printf("입력한 문자 %c의 ASCII 코드는 %h 이다\n",  
                                                                ch, ch);  
}
```

# Program 개발 흐름



열거식으로 일 처리를 나열함  
상세한 과정을 표현하기보다  
주요내용만을 먼저 생각

제1세분화 내용을 좀더 상세하게 기술

필요한 기억장소 설계

제2세분화 내용을 순서도 작성

# Program 개발 흐름

프로그램 코드 작성

순서도를 기반으로 코드를 작성

```
class FourRulesExample1 {  
    public static void main(String args[]) {  
        System.out.println(1 / 2 - 3);  
        System.out.println(2.0 + 1.5 * 2.0);  
        System.out.println(10 / 3 / 2);  
        System.out.println(2 * 5 / 2);  
    }  
}
```

소스코드 생성

Library

컴파일

테스트 & 디버깅

B

최종프로그램

# 일상생활에서의 일 처리 과정 도식화

## ■ 작성방법

- 주어진 문제를 크게 몇 가지 처리과정으로 나눔 (What : 개념적 단계)

- Information과 Data를 파악
- 일을 시간 순서로 나열 (일 처리 순서로 나열)
- 특별히 처리해야 할 일이 있는지 검토
- 제1 세분화

- 제 1 세분화의 내용을 세분화하여 순서도를 작성 (How : 실체적 단계)

- 일 처리 순서를 재점검 (구체적 표현)
- 제2 세분화

# 제 1 세분화

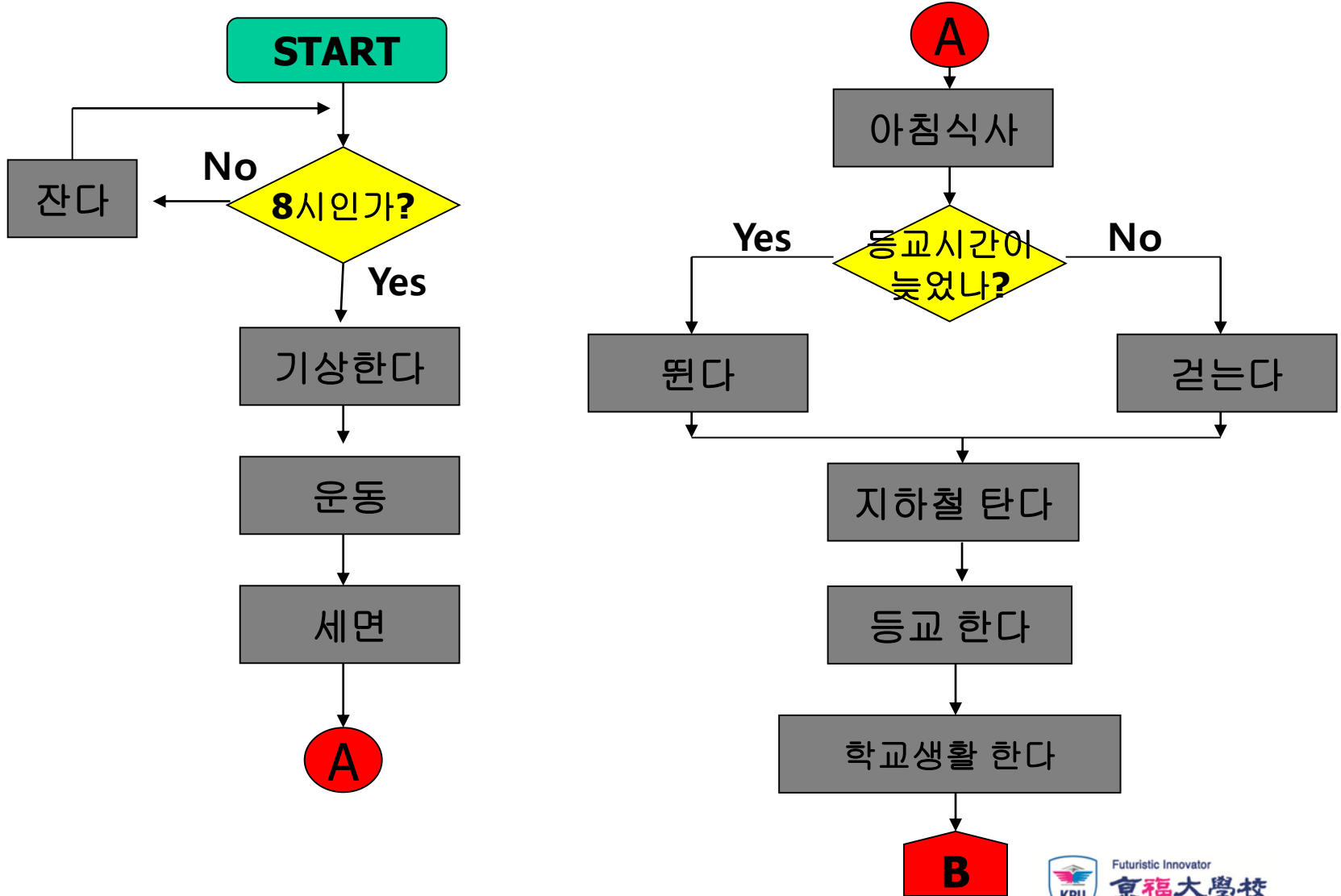
- 어느 학생의 일과를 소개
  - 상세한 과정을 표현하기에 앞서 **주요 일과만 먼저 생각해 보자**
- 제 1 세분화
  1. 기상한다
  2. 등교한다
  3. 학교생활을 한다
  4. 자유시간을 갖는다
  5. 종료(취침)



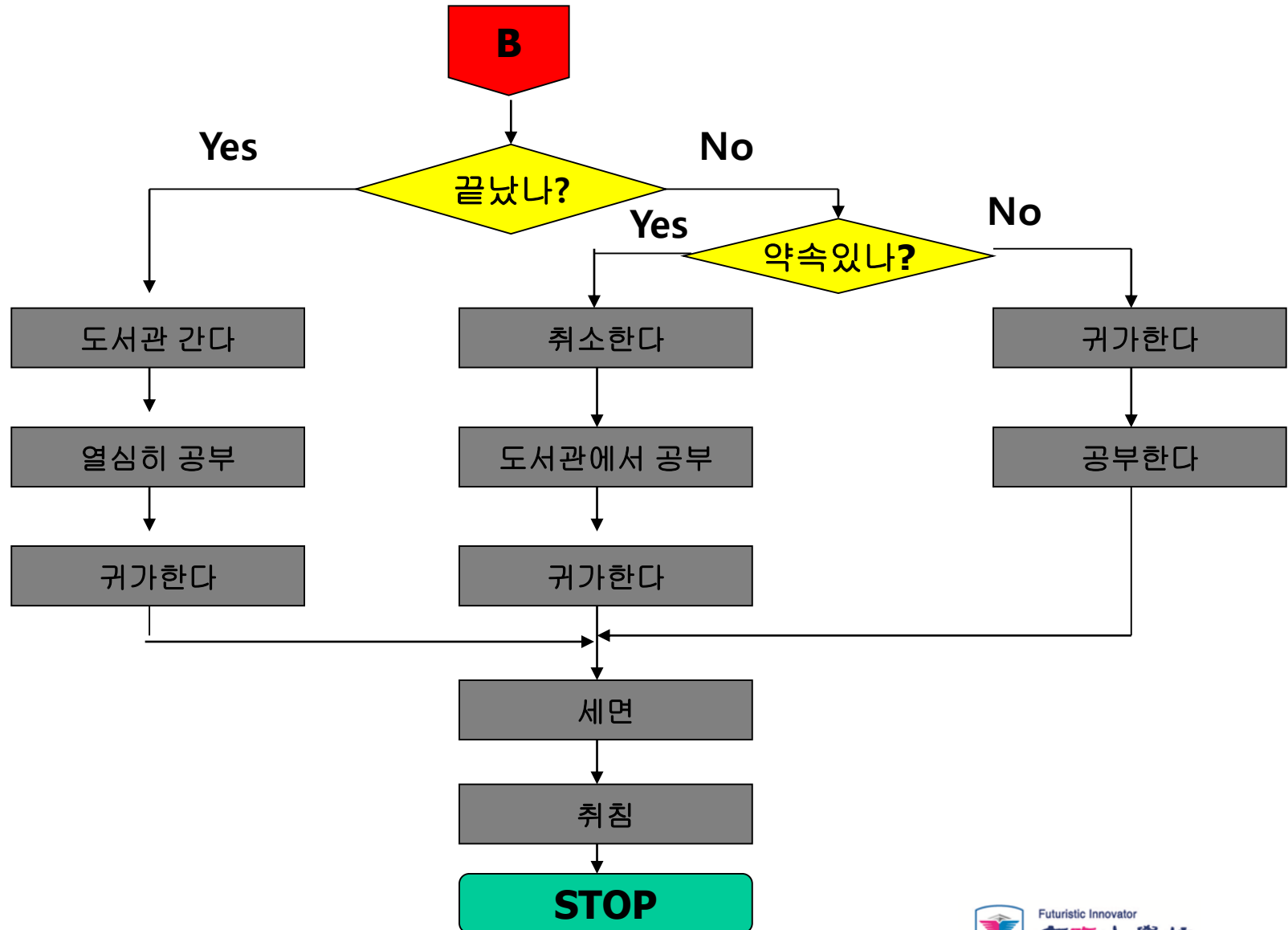
# 제 2 세분화

- 1. 기상한다
  - 1.1 운동
  - 1.2 세면
  - 1.3 아침식사
- 2. 등교한다
  - 2.1 버스를 타고 지하철로 이동
  - 2.2 지하철 탄다.
  - 2.3 버스를 탄다
- 3. 학교생활을 한다
  - 3.1 오전 수업
  - 3.2 점심 식사
  - 3.3 오후 수업
- 4. 자유시간을 갖는다
- 5. 종료(취침)

# 순서도 작성(1/2)

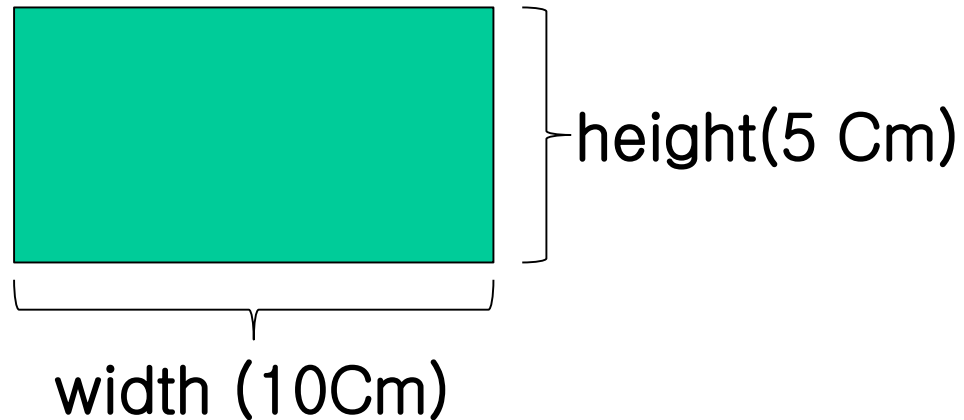


# 순서도 작성(2/2)



# 도입 실습

- 직사각형의 둘레(perimeter)와 면적(area)을 구하는 프로그램을 작성하여 보자



# 도입 실습

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 가로(width) – int (10 Cm)
- 세로(height) – int (5 Cm)

## ■ 출력

- 면적(area) – int
- 둘레(perimeter) – int

## ■ 계산 방법

- $\text{area} = \text{width} * \text{height};$
- $\text{perimeter} = 2 * (\text{width} + \text{height});$

# 도입 실습

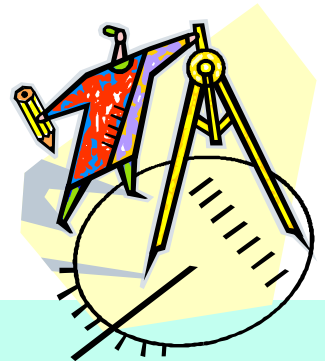
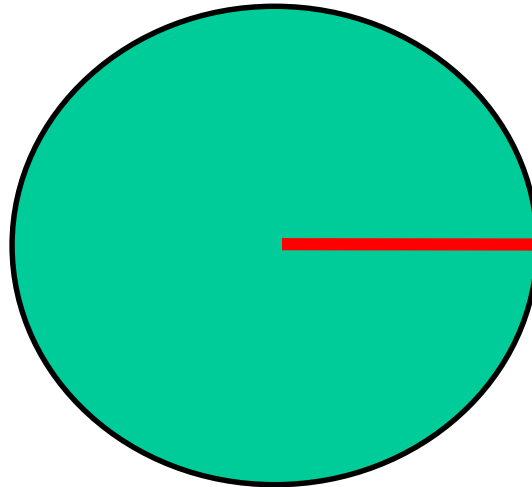
```
public static void main(String[] args) {  
    int width = 10;  
    int height;  
    int area;  
    int 1  
  
    height = 5;  
    area = width * height;  
    perimeter = 2  
    System.out.println(" 사각형 넓이 : " + area);  
    System.out.println(3);  
}
```



정보만 표시하지 말고 데이터도 표시하자

# 원의 면적

- 반지름(radius)이 100Cm인 원의 면적(area)을 구하는 프로그램 작성하여라.



◆ 원 면적(area) 구하는 공식  
= 3.141592 x 반지름(radius) x 반지름(radius)

# 원의 면적

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력 (데이터)

- 반지름(radius) – int (100 Cm)

## ■ 출력 (정보)

- 원의 면적(area) – double

## ■ 계산 방법

- 원 면적(area) 구하는 공식

$$= 3.141592 \times \text{반지름(radius)} \times \text{반지름(radius)}$$

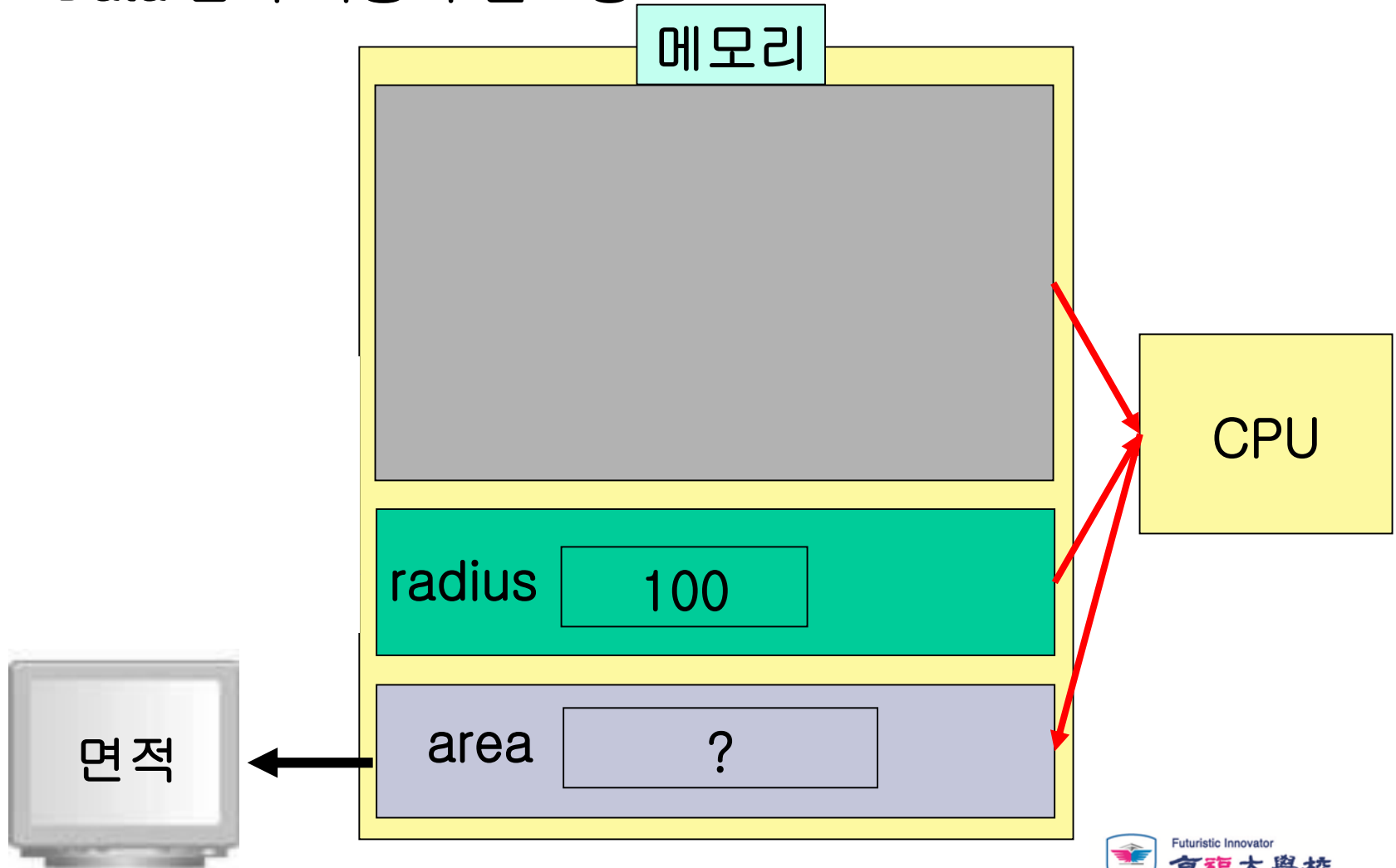


# 원의 면적

```
public static void main(String[] args) {  
    int radius = 100;        // 상수를 사용  
    double area;  
  
    area = 3.141592 * radius * radius;  
  
    System.out.println(" 반지름이 " + radius +  
                        "Cm인 원의 면적은 " + area + "Cm2 입니다.");  
}
```

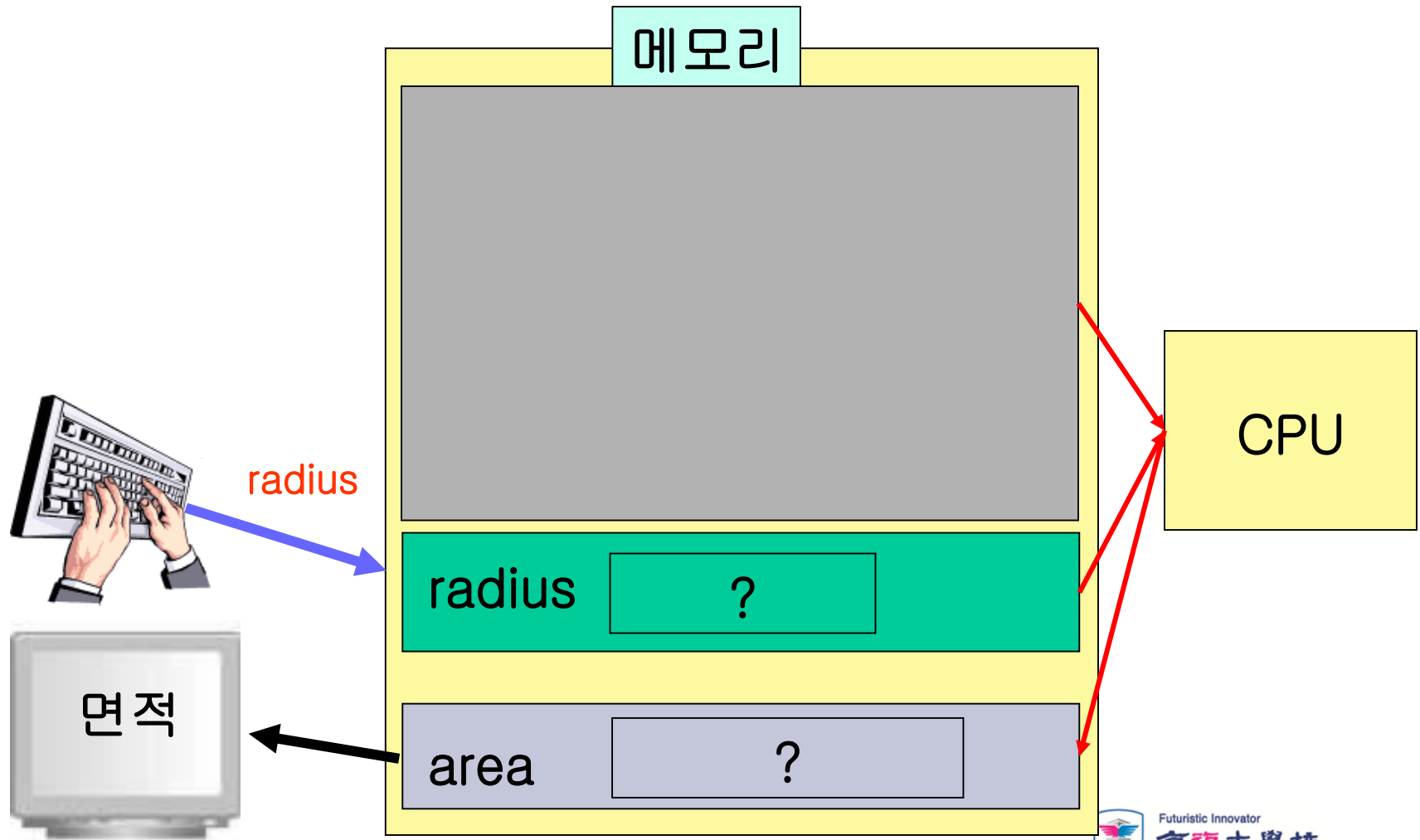
# 원의 면적

## ■ Data 입력 기능의 필요성



# 원의 면적

## ■ Data의 일반화(입력 필요)



# 원의 면적

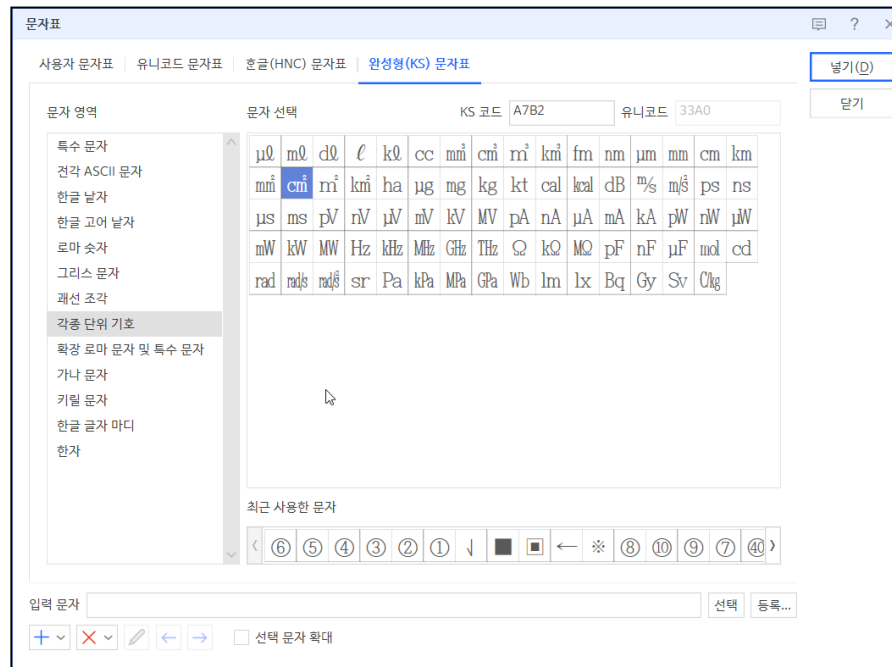
```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    int radius;  
    double area;  
  
    System.out.print("원의 반지름을 입력 : ");  
    radius = keyboard.nextInt();  
  
    area = 3.141592 * radius * radius;  
  
    System.out.println(" 반지름이 " + radius +  
        "Cm인 원의 면적은 " + area + "Cm2 입니다.");  
}
```

# 원의 면적

## ■ 단위 표시법

■ 출력에서 Keyboard에서 입력할 수 없는 것을 출력하려면 어떻게 할까요?

■ [Hwp] - [입력] - [문자표] - [유니코드]



# 원의 면적

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    int radius;  
    double area;  
  
    System.out.print("원의 반지름을 입력 : ");  
    radius = keyboard.nextInt();  
  
    area = 3.141592 * radius * radius;  
  
    System.out.println(" 반지름이 " + radius + "Cm인 원의 면적은 " +  
        area + "Wu33A0 입니다.");  
}
```

# 원의 면적

- 원주율 = 3.141592
- Program 작성시 원주율과 같은 고정된 값은 Program의 이해를 돕기 위해서 필요한 곳에 그때 그때 작성하는 것보다는 final 변수로 저장하여 사용

```
final double PI = 3.141592;
```

- 또 다른 방법은 System의 Math 클래스의 상수를 사용 하는 방법

```
Math.PI
```

# 원의 면적

■ 변하는 데이터는 어떻게 표현할까?

```
public class CircleArea03 {  
    public static void main(String[] args) {
```

```
        final float PI = 3.14159f;
```

부동 소수점형 변수의 선언문

```
        float radius = 5.5f;
```

```
        float area = PI * radius * radius;
```

부동 소수점형 변수에 대한  
대입문

```
        System.out.printf("반지름이 %.1f Cm인 ", radius);
```

```
        System.out.printf("원의 면적 = %f\n", area);
```

```
    }
```

```
}
```

변수 : 부동 소수점수





# 원의 면적[심화]

- 원의 둘레의 길이를 추가로 구해보자
- 문제 분석

원 둘레 구하는 공식

$$= 4 \times 3.141592 \times \text{반지름(radius)}$$

# 원의 면적 [심화]

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    float radius;  
    float area;  
    double 둘레;  
  
    System.out.print(" 원의 반지름 입력 ");  
    radius = keyboard.nextFloat( );  
    area = 3.14f * radius * radius;  
    둘레 = 2 * 3.14 * radius;  
  
    System.out.println(" 반지름 = " + radius);  
    System.out.printf(" 원의 면적 = %5.2f\n", area);  
    System.out.printf(" 원의 둘레 = %5.2f\n", 둘레);  
}
```

# 사칙 연산하기

- 두개의 실수를 Keyboard로부터 입력 받아 사칙연산하는 Program을 작성하여라

컴퓨터가 입력을 유도하는 문장(prompt)을 보인다

두 수를 입력 ? 82 20<enter>

화면을 통하여 직접 입력

82.00 + 20.00 = 102.00  
82.00 - 20.00 = 62.00  
82.00 \* 20.00 = 1640.00  
82.00 / 20.00 = 4.10

끝자리를 잘 맞추어서 출력해 보자

# 사칙 연산하기

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력 (데이터)

- 두수 (num1, num2) – float (double)

## ■ 출력 (정보)

- 사칙연산 결과 출력

- 소수점 이하 2번째 자리까지 표시

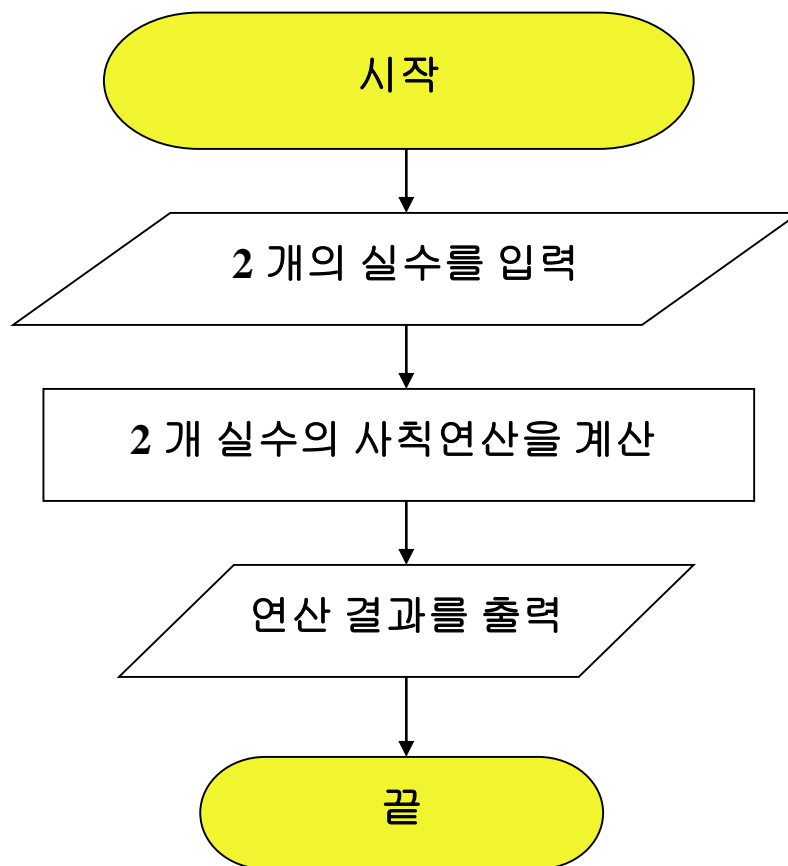
- 결과 줄을 맞추기 위해 가장 큰 수의 자리수는 7자리

## ■ 계산 방법

# 사칙 연산하기

## ■ 문제 분석

### ■ 계산 방법 (FlowChart 작성)



# 사칙 연산하기

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    float num1, num2;
```

Prompt를 활용한 입력

실행 결과 출력 부문

```
}
```

# 연봉 계산

- 10년 동안 월급을 모두 저금할 경우, 얼마나 모을 수 있을까?



# 연봉 계산

- 문제 분석

- Data는 무엇이고, Information는 무엇인가?

- 입력

- 월급(salary) – int

- 몇 년(year) – int

- 출력

- 자산(asset) –int

- 계산 방법

- 저축액(deposit) = 월급(salary) \* 12 달

- 자산(asset) = 저축액(deposit) \* 몇 년(year)



# 연봉 계산

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    int salary, year;  
    int deposit, asset;
```

Prompt를 활용한 입력

계산하기(처리하기)

출력하기

```
}
```