2 학기
JAVA Class

'이것이 자바다'

2025.09

Cover Description

made by Lewis

01 **변수란?**

데이터 저장 공간

02 **정수, 문자, 실수, 논리, 문자열**

변수란 무엇인가?

03 **자동 / 강제 타입 변환**

변수 종류

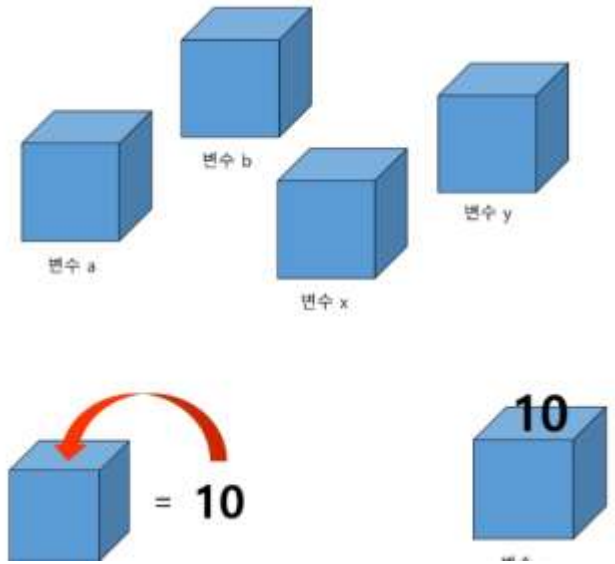
04 **변수값 출력 및 입력 데이터 저장**

-

컴퓨터가 이해하는 코드는
누구라도 작성할 수 있습니다.
뛰어난 프로그래머는 사람이
이해하는 코드를 작성합니다.

변수란 ?

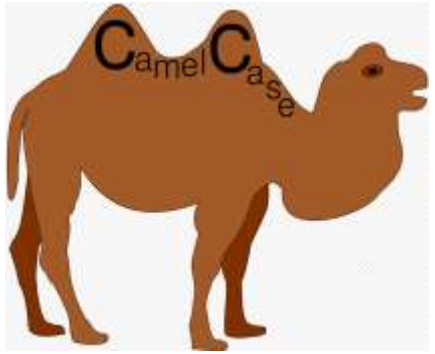
- 하나의 값을 저장할 수 있는 메모리 번지에 붙여진 이름



변수명 생성

- 첫 글자는 문자로 중간에 $, _ 포함 가능
- 변수 명은 카멜케이스로 이루어진 영문자

*카멜케이스 (Camel Case)
단어 연결 시 첫 글자를 제외한 각 단어의 첫 글자를 대문자로 표기하는 명명 규칙
int iCheckPoint; bool bCanMove;



*파스칼 케이스(pascal case)
카멜케이스와 비슷하지만 첫 단어의 첫 글자도 대문자로 표기
class FirstClass;

변수 초기화

- int iValue;
- int iResult = iValue + 10; iResult 결과 값은 ??
- p. 39 VariableInitalizationExample.java
- VariableUseExample.java

*VariableInitializationExample.java

```java
package ch02.sec01;

public class VariableInitializationExample {
    public static void main(String[] args) {
        //변수 value 선언
        int value;

        //연산 결과를 변수 result의 초기값으로 대입
        int result = value + 10;

        //변수 result 값을 읽고 콘솔에 출력
        System.out.println(result);
    }
}
```

VariableUseExample.java

```java
package ch02.sec01;

public class VariableUseExample {
    public static void main(String[] args) {
        int hour = 3;
        int minute = 5;
        System.out.println(hour + "시간 " + minute + "분");

        int totalMinute = (hour*60) + minute;
        System.out.println("총" + totalMinute + "분");
    }
}
```

- 지역변수는 반드시 초기화 필요
- 전역변수는 자동으로 기본값 초기화 됨

변수 초기화

- 변수 스왑핑
- p.40 VariableExchangeExample.java

VariableExchangeExample.java

```java
package ch02.sec01;

public class VariableExchangeExample {
    public static void main(String[] args) {
        int x = 3;
        int y = 5;
        System.out.println("x:" + x + ", y:" + y);

        int temp = x;
        x = y;
        y = temp;
        System.out.println("x:" + x + ", y:" + y);
    }
}

```

변수 타입

변수 구분	기본 타입	메모리 크기
정수	byte, char, short, int, long	1 byte 2 byte 4 byte 8 byte
실수	float, double	4 byte 8 byte
논리 값	boolean	1 byte
문자열	string	

- char : 0 ~ 65,535 , 음수 불가, 유니코드 문자 저장에 사용됨
- char cType = '\u0041'; // 유니코드 (A)
- char cStart = '\uAC00'; // 가

변수 타입

- 에러로그 확인

*ByteExample.java

```
1  package ch02.sec02;
2
3  public class ByteExample {
4      public static void main(String[] args) {
5          byte var1 = -128;
6          byte var2 = -30;
7          byte var3 = 0;
8          byte var4 = 30;
9          byte var5 = 127;
10         //byte var6 = 128;  // Error -> ?
11
12         System.out.println(var1);
13         System.out.println(var2);
14         System.out.println(var3);
15         System.out.println(var4);
16         System.out.println(var5);
17     }
18 }
```

Console × Problems Debug Shell

<terminated> ByteExample [Java Application] C:₩Users₩Lewis₩.p2₩poo

```
-128
-30
0
30
127
```

```
//byte var6 = 128; // Error -> ?
```

- long 변수 타입 설정

*LongExample.java

```
1  package ch02.sec02;
2
3  public class LongExample {
4      public static void main(String[] args) {
5          long var1 = 10;
6          long var2 = 20L;
7          //long var3 = 1000000000000;  // Error --> ??
8          long var4 = 1000000000000L;
9
10         System.out.println(var1);
11         System.out.println(var2);
12         System.out.println(var4);
13     }
14 }
15
16
```

```
//long var3 = 1000000000000; //
Error --> ??
```

문자 타입

- 문자 리터럴 : ‘ 작은 따옴표로 감싼 것 -> 유니코드로 변환되어 저장

*CharExample.java

```java
package ch02.sec03;

public class CharExample {
    public static void main(String[] args) {
        char c1 = 'A';          //문자 저장
        char c2 = 65;           //유니코드 직접 저장

        char c3 = '가';          //문자 저장
        char c4 = 44032;        //유니코드 직접 저장

        System.out.println(c1);
        System.out.println(c2);
        System.out.println(c3);
        System.out.println(c4);
    }
}
```

Console | Problems | Debug Shell

```
<terminated> CharExample [Java Application] C:₩Users₩Lewis₩.p2₩pool₩plu
A
A
가
가
```

char 변수 초기화

⇒ char cInit = ‘’; (X)

⇒ char cInit = ‘ ‘; 공백추가

문자 타입

* 아스키 코드표

* DEC : 10진수
  HEX : 16진수
  OCT : 8진수

DEC	HEX	OCT	Char	DEC	HEX	OCT	Char	DEC	HEX	OCT	Char
0	00	000	Ctrl-@ NUL	43	2B	053	+	86	56	126	V
1	01	001	Ctrl-A SOH	44	2C	054	,	87	57	127	W
2	02	002	Ctrl-B STX	45	2D	055	-	88	58	130	X
3	03	003	Ctrl-C ETX	46	2E	056	.	89	59	131	Y
4	04	004	Ctrl-D EOT	47	2F	057	/	90	5A	132	Z
5	05	005	Ctrl-E ENQ	48	30	060	0	91	5B	133	[
6	06	006	Ctrl-F ACK	49	31	061	1	92	5C	134	₩
7	07	007	Ctrl-G BEL	50	32	062	2	93	5D	135	]
8	08	010	Ctrl-H BS	51	33	063	3	94	5E	136	^
9	09	011	Ctrl-I HT	52	34	064	4	95	5F	137	_
10	0A	012	Ctrl-J LF	53	35	065	5	96	60	140	`
11	0B	013	Ctrl-K VT	54	36	066	6	97	61	141	a
12	0C	014	Ctrl-L FF	55	37	067	7	98	62	142	b
13	0D	015	Ctrl-M CR	56	38	070	8	99	63	143	c
14	0E	016	Ctrl-N SO	57	39	071	9	100	64	144	d
15	0F	017	Ctrl-O SI	58	3A	072	:	101	65	145	e
16	10	020	Ctrl-P DLE	59	3B	073	;	102	66	146	f
17	11	021	Ctrl-Q DCI	60	3C	074	<	103	67	147	g
18	12	022	Ctrl-R DC2	61	3D	075	=	104	68	150	h
19	13	023	Ctrl-S DC3	62	3E	076	>	105	69	151	i
20	14	024	Ctrl-T DC4	63	3F	077	?	106	6A	152	j
21	15	025	Ctrl-U NAK	64	40	100	@	107	6B	153	k
22	16	026	Ctrl-V SYN	65	41	101	A	108	6C	154	l
23	17	027	Ctrl-W ETB	66	42	102	B	109	6D	155	m
24	18	030	Ctrl-X CAN	67	43	103	C	110	6E	156	n
25	19	031	Ctrl-Y EM	68	44	104	D	111	6F	157	o
26	1A	032	Ctrl-Z SUB	69	45	105	E	112	70	160	p
27	1B	033	Ctrl-[ESC	70	46	106	F	113	71	161	q
28	1C	034	Ctrl-₩ FS	71	47	107	G	114	72	162	r
29	1D	035	Ctrl-] GS	72	48	110	H	115	73	163	s
30	1E	036	Ctrl-^ RS	73	49	111	I	116	74	164	t
31	1F	037	Ctrl_ US	74	4A	112	J	117	75	165	u
32	20	040	Space	75	4B	113	K	118	76	166	v
33	21	041	!	76	4C	114	L	119	77	167	w
34	22	042	"	77	4D	115	M	120	78	170	x
35	23	043	#	78	4E	116	N	121	79	171	y
36	24	044	$	79	4F	117	O	122	7A	172	z
37	25	045	%	80	50	120	P	123	7B	173	{
38	26	046	&	81	51	121	Q	124	7C	174	|
39	27	047	'	82	52	122	R	125	7D	175	}
40	28	050	(	83	53	123	S	126	7E	176	

실수 타입

- float -> 소수점 7자리, 변수값 뒤에 F, f 를 붙여 명확히 한다.
- double -> 소수점 15자리

```
FloatDoubleExample.java
1  package ch02.sec04;
2
3  public class FloatDoubleExample {
4      public static void main(String[] args) {
5          //정밀도 확인
6          float var1 = 0.12345678901234567890f;
7          double var2 = 0.12345678901234567890;
8          System.out.println("var1: " + var1);
9          System.out.println("var2: " + var2);
10
11         //10의 거듭제곱 리터럴
12         double var3 = 3e6;
13         float var4 = 3e6F;
14         double var5 = 2e-3;
15         System.out.println("var3: " + var3);
16         System.out.println("var4: " + var4);
17         System.out.println("var5: " + var5);
18     }
19 }
20
21
22
```

```
Console  Problems  Debug Shell
<terminated> FloatDoubleExample [Java Application] C:₩Users₩Lewis₩.p2₩
var1: 0.12345679
var2: 0.12345678901234568
var3: 3000000.0
var4: 3000000.0
var5: 0.002
```

- 소수점 7자리까지 만 확실하고 그 뒤는 근사치로 보여 짐

논리 타입

- 참 거짓 판단 값

BooleanExample.java

```
1  package ch02.sec05;
2
3  public class BooleanExample {
4      public static void main(String[] args) {
5          boolean stop = true;
6          if(stop) {
7              System.out.println("중지합니다.");
8          } else {
9              System.out.println("시작합니다.");
10         }
11
12         int x = 10;
13         boolean result1 = (x == 20);    //변수 x의 값이 20?
14         boolean result2 = (x != 20);    //변수 x의 값이 20이 아니면 true
15         System.out.println("result1: " + result1);
16         System.out.println("result2: " + result2);
17     }
18 }
```

Console | Problems | Debug Shell

```
<terminated> BooleanExample [Java Application] C:₩Users₩Lewis₩.p2₩pool₩plugins₩org.eclipse.justj
중지합니다.
result1: false
result2: true
```

문자열 타입

- "" 로 감싼 여러 개의 문자 값

StringExample.java

```
1  package ch02.sec06;
2
3  public class StringExample {
4      public static void main(String[] args) {
5          String name = "홍길동";
6          String job = "프로그래머";
7          System.out.println(name);
8          System.out.println(job);
9
10         String str = "나는 \"자바\"를 배웁니다..";
11         System.out.println(str);
12
13         str = "번호\t이름\t직업 ";
14         System.out.println(str);
15
16         System.out.print("나는\n");
17         System.out.print("자바를\n");
18         System.out.print("배웁니다.");
19     }
20 }
21
```

Console | Problems | Debug Shell

```
<terminated> StringExample [Java Application] C:₩Users₩Lewis₩.p2₩pool₩
홍길동
프로그래머
나는 "자바"를 배웁니다..
번호    이름    직업
나는
자바를
배웁니다.
```

- 이스케이프 문자

\" : "문자 포함
\' : ' 문자 포함
\\ : \ 문자 포함
\t : 탭 만큼 띄움
\n : 줄바꿈
\r : 캐리지 리턴

문자열 타입

- 텍스트 블록 설정

큰따옴표 3개로 감싸면 작성한 그대로
\n 줄바꿈
\ 줄바꿈 없이 다음줄에 이어서 작성

```java
package ch02.sec06;

public class TextBlockExample {
    public static void main(String[] args) {
        String str1 = "" +
        "{\n" +
        "\t\"id\":\"winter\",\n" +
        "\t\"name\":\"눈송이\"\n" +
        "}";

        String str2 = """
        {
            "id":"winter",
            "name":"눈송이"
        }
        """;

        System.out.println(str1);
        System.out.println("-----------------------------------");
        System.out.println(str2);
        System.out.println("-----------------------------------");
        String str = """
        나는 자바를 \
        학습합니다.
        나는 자바 고수가 될 겁니다.
        """;
        System.out.println(str);
    }
}
```

```
<terminated> TextBlockExample [Java Application] C:₩Users₩Lewis₩.p2₩pool₩plugins₩org.ecli
{
        "id":"winter",
        "name":"눈송이"
}
-----------------------------------
{
        "id":"winter",
        "name":"눈송이"
}

-----------------------------------
나는 자바를 학습합니다.
나는 자바 고수가 될 겁니다.
```

자동 타입 변환

- 변수의 허용 범위가 작은 타입이 허용 범위가 큰 타입으로 자동 대체

Byte < short, char < int < long < float < double

- byte 타입은 char 타입으로 자동변환 안됨 => char 타입은 음수 값이 없다

  byte bType = 65;

  char chType = bType; → Error
- long -> float 자동 형변환 가능 ➔ 크기상 손실 발생

자동 타입 변환

PromotionExample.java

```java
package ch02.sec07;

public class PromotionExample {
    public static void main(String[] args) {
        //자동 타입 변환
        byte byteValue = 10;
        int intValue = byteValue;
        System.out.println("intValue: " + intValue);

        char charValue = '가';
        intValue = charValue;
        System.out.println("가의 유니코드: " + intValue);

        intValue = 50;
        long longValue = intValue;;
        System.out.println("longValue: " + longValue);

        longValue = 100;
        float floatValue = longValue;
        System.out.println("floatValue: " + floatValue);

        floatValue = 100.5F;
        double doubleValue = floatValue;
        System.out.println("doubleValue: " + doubleValue);
    }
}
```

Console | Problems | Debug Shell

```
<terminated> PromotionExample [Java Application] C:₩Users₩Lewis₩.p2₩pool₩plugins
intValue: 10
가의 유니코드: 44032
longValue: 50
floatValue: 100.0
doubleValue: 100.5
```

강제 타입 변환

- 큰 허용 범위의 변수를 작은 허용범위 타입으로 쪼개어서 저장하는 것
- 기존의 값은 유지하면서 값의 타입만 바꾸는 것

Byte < short, char < int < long < float < double

int -> byte

```
int iVal = 65;
byte bVal = (byte)iVal;
```

long -> int

```
long lValue = 300;
int iVal = (int)lValue;
```

int -> char

```
int iVal = 65;
char cValue = (char)iVal;    // cValue == 'A';
```

실수 -> int

```
doble dVal = 3.14;
int iVal = (int) dVal;   // iVal == 3;
```

강제 타입 변환

CastingExample.java

```java
package ch02.sec08;

public class CastingExample {
    public static void main(String[] args) {
        int var1 = 10;
        byte var2 = (byte) var1;
        System.out.println(var2);    //강제 타입 변환 후에 10이 그대로 유지

        long var3 = 300;
        int var4 = (int) var3;
        System.out.println(var4);    //강제 타입 변환 후에 300이 그대로 유지

        int var5 = 65;
        char var6 = (char) var5;
        System.out.println(var6);    //'A'가 출력

        double var7 = 3.14;
        int var8 = (int) var7;
        System.out.println(var8);    //3이 출력
    }
}

```

Console | Problems | Debug Shell

```
<terminated> CastingExample [Java Application] C:₩Users₩Lewis₩.p2₩pool₩plugins₩org.eclipse.just
10
300
A
3
```

연산식에서 타입 변환

* 변수타입 변수1 = 변수2 연산식 변수3;

1. 변수2, 변수3의 변수 타입이 변수1 의 타입 보다 작은 경우 변수1의 타입으로 자동 변환 된다.

2. 변수2, 변수3의 변수 타입이 변수1 의 타입 보다 큰 경우 각 항목의 값을 변수1의 타입으로 강제 형변환 하여 연산 한다.

3. byte, short, char 타입은 산술 연산(+,-,*,/) 을 할 때 자동으로 int 로 형변환 된다.
   * byte + byte => int

```
byte x = 10;
byte y = 10;
byte sum = (byte)(x + y);
```

연산식에서 타입 변환

OperationPromotionExample.java

```java
package ch02.sec09;

public class OperationPromotionExample {
    public static void main(String[] args) {
        byte result1 = 10 + 20; //컴파일 단계에서 연산
        System.out.println("result1: " + result1);

        byte v1 = 10;
        byte v2 = 20;
        int result2 = v1 + v2;      //int 타입으로 변환후 연산
        System.out.println("result2: " + result2);

        byte v3 = 10;
        int v4 = 100;
        long v5 = 1000L;
        long result3 = v3 + v4 + v5;    //long 타입으로 변환후 연산
        System.out.println("result3: " + result3);

        char v6 = 'A';
        char v7 = 1;
        int result4 = v6 + v7;  //int 타입으로 변환후 연산
        System.out.println("result4: " + result4);
        System.out.println("result4: " + (char)result4);

        int v8 = 10;
        int result5 = v8 / 4;       //정수 연산의 결과는 정수
        System.out.println("result5: " + result5);

        int v9 = 10;
        double result6 = v9 / 4.0;  //double 타입으로 변환후 연산
        System.out.println("result6: " + result6);

        int v10 = 1;
        int v11 = 2;
        double result7 = (double) v10 / v11;    //double 타입으로 변환후 연산
        System.out.println("result7: " + result7);
    }
}
```

Console | Problems | Debug Shell

<terminated> OperationPromotionExample [Java Application] C:₩Users₩Lewis₩.p2₩pool₩plugins₩org.e

```
result1: 30
result2: 30
result3: 1110
result4: 66
result4: B
result5: 2
result6: 2.5
result7: 0.5
```

p. 64 OperationPromotionExample.java

연산식에서 타입 변환

JAVA 에서의 '+' 기능

int iVar = 3 + 5; ➔ ?

String sVal = "3" + 7; ➔ ?

p. 66 OperationPromotionExample.java

StringConcatExample.java

```
1  package ch02.sec09;
2
3  public class StringConcatExample {
4      public static void main(String[] args) {
5          //숫자 연산
6          int result1 = 10 + 2 + 8;
7          System.out.println("result1: " + result1);
8
9          //결합 연산
10         String result2 = 10 + 2 + "8";
11         System.out.println("result2: " + result2);
12
13         String result3 = 10 + "2" + 8;
14         System.out.println("result3: " + result3);
15
16         String result4 = "10" + 2 + 8;
17         System.out.println("result4: " + result4);
18
19         String result5 = "10" + (2 + 8);
20         System.out.println("result5: " + result5);
21     }
22 }
```

Console | Problems | Debug Shell

<terminated> StringConcatExample [Java Application] C:₩Users₩Lewis₩.p2

```
result1: 20
result2: 128
result3: 1028
result4: 1028
result5: 1010
```

문자열 변환

String -> byte : Byte.parseByte();
String -> short : Short.parseShort();
String -> int : Integer.parseInt();
String -> long : Long.parseLong();
String -> float : Float.parseFloat();
String -> double : Double.parseDouble();
String -> boolean : Boolean.parseBoolean();

* 기본 타입의 값을 String 으로 변환

```
String.valueOf(기본타입값);
```

int sStr = Integer.parseInt("가"); => ??

p. 67 PrimitiveAndStringConversionExample.java

PrimitiveAndStringConversionExample.java

```
1  package ch02.sec10;
2
3  public class PrimitiveAndStringConversionExample {
4      public static void main(String[] args) {
5          int value1 = Integer.parseInt("10");
6          double value2 = Double.parseDouble("3.14");
7          boolean value3 = Boolean.parseBoolean("true");
8
9          System.out.println("value1: " + value1);
10         System.out.println("value2: " + value2);
11         System.out.println("value3: " + value3);
12
13         String str1 = String.valueOf(10);
14         String str2 = String.valueOf(3.14);
15         String str3 = String.valueOf(true);
16
17         System.out.println("str1: " + str1);
18         System.out.println("str2: " + str2);
19         System.out.println("str3: " + str3);
20     }
21 }
```

Console | Problems | Debug Shell

<terminated> PrimitiveAndStringConversionExample [Java Application] C:₩Users₩

```
value1: 10
value2: 3.14
value3: true
str1: 10
str2: 3.14
str3: true
```

변수의 사용 범위

- 자신의 블록 내부에서만 사용 가능

*VariableScopeExample.java

```
1  package ch02.sec11;
2
3  public class VariableScopeExample {
4      public static void main(String[] args) {
5          int v1 = 15;
6          if(v1>10) {
7              int v2 = v1 - 10;
8          }
9          //int v3 = v1 + v2 + 5;  // Error
10     }
11 }
12
13
```

p. 69 VariableScopeExample.java

콘솔에 변수값 출력

- System.out.println(“출력 후 줄바꿈”);
- System.out.print(“출력 후 줄 변환 없음”);
- System.out.printf(“형식 문자열”, 값1, 값2…..);

%b	**boolean** 형식으로 출력
%d	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	줄바꿈 기능
%e 또는 %E	지수 표현식의 형식으로 출력

콘솔에 변수값 출력

- System.out.println(“출력 후 줄바꿈”);
- System.out.print(“출력 후 줄 변환 없음”);
- System.out.printf(“형식 문자열”, 값1, 값2…..);

PrintfExample.java

```
1  package ch02.sec12;
2
3  public class PrintfExample {
4      public static void main(String[] args) {
5          int value = 123;
6          System.out.printf("상품의 가격:%d원\n", value);
7          System.out.printf("상품의 가격:%6d원\n", value);
8          System.out.printf("상품의 가격:%-6d원\n", value);
9          System.out.printf("상품의 가격:%06d원\n", value);
10
11         double area = 3.14159 * 10 * 10;
12         System.out.printf("반지름이 %d인 원의 넓이:%10.2f\n", 10, area);
13
14         String name = "홍길동";
15         String job = "도적";
16         System.out.printf("%6d | %-10s | %10s\n", 1, name, job);
17     }
18 }
19
```

Console | Problems | Debug Shell

```
<terminated> PrintfExample [Java Application] C:₩Users₩Lewis₩.p2₩pool₩plugins₩org.eclipse.ju
상품의 가격:123원
상품의 가격:   123원
상품의 가격:123   원
상품의 가격:000123원
반지름이 10인 원의 넓이:    314.16
     1 | 홍길동       |         도적
```

키보드 입력

```
// 1. 객체 생성
Scanner scanner = new Scanner(System.in);
// 2. 데이터 입력
String inputData = scanner.nextLine();
// 3. 문자열 비교
data.equals(“q”);
```

p. 73 ScannerExample.java

ScannerExample.java

```
1  package ch02.sec13;
2
3  import java.util.Scanner;
4
5  public class ScannerExample {
6      public static void main(String[] args) throws Exception {
7          Scanner scanner = new Scanner(System.in);
8
9          System.out.print("x 값 입력: ");
10         String strX = scanner.nextLine();
11         int x = Integer.parseInt(strX);
12
13         System.out.print("y 값 입력: ");
14         String strY = scanner.nextLine();
15         int y = Integer.parseInt(strY);
16
17         int result = x + y;
18         System.out.println("x + y: " + result);
19         System.out.println();
20
21         while(true) {
22             System.out.print("입력 문자열: ");
23             String data = scanner.nextLine();
24             if(data.equals("q")) {
25                 break;
26             }
27             System.out.println("출력 문자열: " + data);
28             System.out.println();
29         }
30
31         System.out.println("종료");
32     }
33 }
```

Console | Problems | Debug Shell

```
ScannerExample [Java Application] C:₩Users₩Lewis₩.p2₩pool₩plugins₩org.eclipse.justj.openj
x 값 입력: 7
y 값 입력: 9
x + y: 16

입력 문자열:
```