

JAVA 프로그램 실습

연산자

경북대학교
소프트웨어융합과
교수 배 희 호

JAVA의 연산자

구분		기능	연산자의 예
산술 연산자	이항 연산자	2개의 자료에 대해 산술 연산	+, -, *, /, %
	단항 연산자	1개의 자료에 대해 산술 연산	++, --
	대입 연산자	왼쪽 변수에 값을 대입	=, +=, -=, *=, /=, %=
관계 연산자		대소, 등가 관계를 비교	<, <=, >, >=, ==, !=
논리 연산자		복수개의 조건을 평가	&&, , !
조건 연산자		조건 만족의 여부에 따라 별개의 값 계산	? 문장1 : 문장2
비트 연산자	비트 이동 연산자	비트 값을 좌우로 이동	<<, >>, >>>
	비트 논리 연산자	비트 값에 대해 논리 연산	&, , ^, ~
기타 연산자	형변환 연산자	자료형 변환	(자료형)
	문자열 연산자	여러개의 문자열 결합	+

관계 연산자

- 이항 연산자로서 2개의 피 연산자 값을 **대소 동등**을 비교
- 관계 연산의 결과값은 항상 **true**가 아니면 **false** 값을 반환함

연산자	연산식	예제		설 명
		연산식	결과	
>	op1 > op2	3 > 7	false	op1이 op2보다 크면 참
>=	op1 >= op2	3 >= 7	false	op1이 op2보다 크거나 같으면 참
<	op1 < op2	3 < 7	true	op1이 op2보다 작으면 참
<=	op1 <= op2	3 <= 7	true	op1이 op2보다 작거나 같으면 참
==	op1 == op2	3 == 7	false	op1과 op2가 같으면 참
!=	op1 != op2	3 != 7	true	op1과 op2가 다르면 참
instance of	op1 instance of op2	추후 설명	true	op1이 op2의 인스턴스이면 참

관계 연산자

■ 선택 조건

	연 산 자	의 미
대	<	보다 작다
	<=	보다 작거나 같다
	>=	크거나 같다
소	>	보다 크다
동	==	같다
등	!=	같지 않다

■ 연산 우선 순위는 대소, 동등 순임

■ 관계 연산자는 두 값을 비교하여 참(true) 또는 거짓(false)을 반환

관계 연산자

- 관계 연산자들은 if문이나 while문의 실행을 제어하는데 사용
- JAVA에서는 정수형, 실수형, 논리형(boolean type) Data를 관계 연산자를 사용하여 비교 가능

```
boolean b = 3 == 1;
```

3과 1을 비교하여 동일하지 않으므로 false값을 변수 b에 배정하게 됨

- C 언어에서는 정수형의 0을 false로, 0이 아닌 모든 값을 true로 간주하지만, JAVA에서 이 규칙은 더 이상 적용되지 않음

관계 연산자

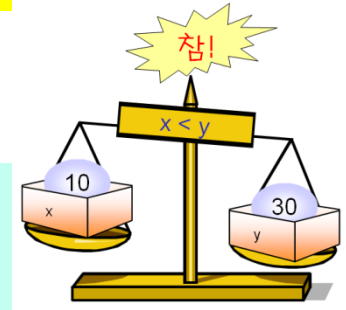
■ 바른 표현 예

$$a > 10$$

$$a \leq b - c$$

$$s \geq 2 * r * 3.141592$$

$$a \neq b$$



■ 틀린 사용 예

$$a <> 10$$

$$a = < b - c$$

$$s > = 2 * r * 3.14159$$

관계 연산자

- $a == b$ 와 $a = b$ 는 유사하지만, 완전히 다른 수식임

```
if (a = 1) {    /* Error의 원인 */  
    . . . . .  
}
```

관계 연산자 예제 1

- 다음과 같은 경우 결과는 ?

```
int test = 7;  
System.out.print(3 < test < 5);
```

1) 위의 문장은 문법적으로는 ?

- JAVA에서 "3 < test < 5" 의 올바른 표현 법은 ?

```
(test > 3) && (test < 5)
```


관계 연산자 예제 2

■ 다음 수식의 결과 값을 설명하여라?

```
char c = 'w';
int i = 1, j = 2, k = -7;
double x = 7e+33, y = 0.001;
```

선언 및 초기화

수 식	동일한 수식(우선 순위)	결과 값
'a' + 1 < c	('a' + 1) < c	true
- i - 5 * j >= k + 1	((-i) - (5 * j)) >= (k + 1)	false
x - 3.333 <= x + y	(x - 3.333) <= (x + y)	true
x < x + y	x < (x + y)	false
'v' == c - 1	'v' == (c - 1)	true
i + j + k == -2 * j	((i + j) + k) == ((-2) * j)	true
x + x != x * y	(x + x) != (x * y)	true
x != x + y	x != (x + y)	false

관계 연산자 예제 2

■ 다음 수식의 결과값을 설명하여라

```
int i = 1, j = 0, k = -1, l = 0;
```

수 식	동일한 수식	결과
$i < j$	$i < j$	false
$i \leq j + 1$	$i \leq (j + 1)$	true
$k > l$	$k > l$	false
$j \geq l$	$j \geq l$	true
$i + k \geq j + l$	$(i + k) \geq (j + l)$	true

논리 연산자

- 논리 연산자는 수식과 수식 사이의 논리 곱(AND) '＆＆' 논리 합(OR) '||'을 평가하는 이항 연산자와 논리 부정(NOT) '!'의 전위 표기 단항 연산자
 - 연산자는 '＆＆', '||'와 같이 연산 기호 사이에 공백 없음
- boolean형 operand에 적용하여, true/false 값을 반환하는 연산자
 - 논리 연산자는 피 연산자를 모두 true과 false으로 평가
- &&(AND)가 ||(OR)보다 우선순위가 높음. 같이 사용되는 경우 괄호를 사용
- JAVA 언어에서는 논리 연산자 중 특이한 사항은 수식 평가 시 실행시간을 단축하기 위하여 단축 연산(short circuit evaluation)을 제공하고 있음
 - 이 연산자는 첫 번째 항을 평가한 결과 두 번째 항을 평가할 필요가 없을 경우, 바로 결과를 반환하는 연산자임

논리 연산자

연산자	연산식	예제		설명
		연산식	결과	
!	!op	!(1==1)	false	op가 거짓(false)이면 참(true)
&&	op1 && op2	(2==1) && (1==1)	false	op1과 op2가 모두 참이면 참 op1이 거짓이면 op2를 평가하지 않음
	op1 op2	(1==1) (2==1)	true	op1이나 op2 중 하나라도 참이면 참 op1이 참이면 op2를 평가하지 않음
&	op1 & op2	(2==1) & (2==1)	false	op1과 op2가 모두 참이면 참 op1이 거짓이어도 op2를 평가함
	op1 op2	(1==1) (2==1)	true	op1이나 op2 중 하나라도 참이면 참 op1이 참이어도 op2를 평가함
^	op1 op2	(1==1) ^ (2==1)	true	op1이나 op2 중 하나만 참이면 참 op1이 참이어도 op2를 평가함

논리 연산자

■ 논리곱 연산자(&& : AND)

- 앞 뒤 두 조건이 모두 true인 경우만 true이고, 나머지 경우는 false

레포트를 잘 해라. 그리고 시험을 잘 보아라.
그러면 JAVA 학점을 잘 주겠다.

논리곱(&&)의 평가		
조건1	조건2	논리합
false	false	false
false	true	false
true	false	false
true	true	true

논리 연산자

■ 논리합 연산자(|| : OR)

- 논리합 기호의 앞 뒤 두 조건이 false인 경우만 false이고, 나머지 경우는 true

레포트를 잘 해라. **또는** 시험을 잘 보아라.
그러면 JAVA 학점을 잘 주겠다.

논리합()의 평가		
조건1	조건2	논리합
false	false	false
false	true	true
true	false	true
true	true	true

논리 연산자

- 논리 연산자(NOT)

- 부정을 의미하는 NOT(!)

- ! 뒤에 있는 조건이 “false”이면 “true”로,
“true”이면 “false”로 바꾸어 줌

논리부정(!)의 평가	
조건	논리부정
false	true
true	false

논리 연산자

■ Short-circuit evaluation

- AND와 OR에서는 첫 번째 조건의 결과에 따라 두 번째 조건을 확인하지 않아도 결과가 정해짐. 따라서, 두 번째 조건 실행을 하지 않는 것이 효율적 임
- 이 방식을 Short-circuit evaluation이라고 함

논리 연산	첫 번째 조건	결과
A && B	A – false	false
A B	A – true	true

논리 연산자 예제 1

■ 다음 Program의 실행 결과를 설명하여라.

```
public static void main(String[] args) {  
    int a = 0, b = 0;  
    boolean x;
```

```
    x = false && ((a = b = 356) != 0);
```

```
    System.out.printf("x = %b, a = %d, b = %d\n", x, a, b);
```

```
    x = true || ((a = ++b) != 0);
```

```
    System.out.printf("x = %b, a = %d, b = %d\n", x, a, b);
```

```
    x = true && ((a = b = 356) != 0);
```

```
    System.out.printf("x = %b, a = %d, b = %d\n", x, a, b);
```

```
    x = false || ((a = ++b) != 0);
```

```
    System.out.printf("x = %b, a = %d, b = %d\n", x, a, b);
```

```
}
```

x = false, a = 0, b = 0

x = true, a = 0, b = 0

x = true, a = 356, b = 356

x = false, a = 357, b = 357



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY

논리 연산자 예제 2

■ 다음 연산 수식의 결과값을 설명하여라

```
char    c = 'B';
int     i = 3, j = 3, k = 3;
double  x = 0.0, y = 2.3;
```

선언 및 초기화

수 식	동일한 수식	결과값
<code>i > 3 && j < 2 && k == 2</code>	<code>(i > 3 && j < 2) && k == 2</code>	false
<code>x > 0.1 i > 3 && j < 3</code>	<code>x > 1.0 (i > 3 && j < 3)</code>	false
<code>i < j && x < y</code>	<code>(i < j) && (x < y)</code>	false
<code>i < j x < y</code>	<code>(i < j) (x < y)</code>	true
<code>'A' <= c && c <= 'Z'</code>	<code>('A' <= c) && (c <= 'Z')</code>	true
<code>c - 1 == 'A' c + 1 == 'Z'</code>	<code>((c - 1) == 'A') ((c + 1) == 'Z')</code>	true
<code>!(c > 'A')</code>	<code>!(c > 'A')</code>	false
<code>!!(x > y)</code>	<code>!!(x > y)</code>	false

관계 논리 연산자 우선 순위

- 논리 연산자의 우선순위는 관계 연산자의 우선순위보다 낮음

 높음	단항 연산자	! (논리 NOT)
	비교	<, <=, >, >=
	등가	==, !=
	논리 AND	&&
	논리 OR	
낮음		

- $a > b \ \&\& \ b > c \ || \ b > d$ 는
((a > b) && (b > c)) || (b > d)로 해석 됨
- $5 > 2 \ \&\& \ 4 > 7$ false
- $5 > 2 \ || \ 4 > 7$ true

조건(?) 연산자

- 조건 연산자는 JAVA의 유일한 3항 연산자 이므로 3개의 피연산자를 취함(3항 연산자)
- 조건 연산자는 C/C++의 연산자와 동일한 기능을 수행함
- ? 연산자의 표현 형식

`boolean_expression ? exp1 : exp2`

- ? 연산자는 먼저 `boolean_expression`을 평가하여 그 결과가 `true`인지 `false`인지 판별하여 `true`이면 `exp1`의 결과를 반환하고, `false`인 경우 `exp2`의 결과를 반환 함
- 조건 연산자는 `if ~ else`문으로 변경 가능
- 조건 연산자는 이해하기는 어렵지만, Program을 간결하게 만드는 이점이 있음
- 조건 연산자는 우 결합성을 가지므로 여러 번 기술하였을 때 맨 우측부터 평가

조건(?) 연산자

- 조건식의 연산결과가 true이면 '식1'의 결과를 반환하고, false이면 '식2'의 결과를 반환

(조건식) ? 식1 : 식2

```
int x = -10;  
int absX = x >= 0 ? x : -x;
```

```
if (x >= 0) {  
    absX = x;  
} else {  
    abxX = -x;  
}
```

```
int score = 50;  
char grade = score >= 90 ? 'A' : (score >= 80 ? 'B' : 'C');
```

cast 연산자(강제 연산자)

- 자료형의 강제 변환에 사용하는 연산자
- Cast Operator는 명시적인 형 변환이 필요 할 때 사용
- Cast Operator는 단항 연산자
- Cast Operator는 원하는 Data Type을 소괄호로 묶어 피 연산자 앞에 둬으로써 Data Type을 변환시키는 역할을 함
- 수식의 Type 변환을 위해 사용할 때는 Cast Operator를 소괄호로 묶지 않아도 됨
- Cast Operator의 결합 방향은 오른쪽에서 왼쪽
- 형식

(형명) 수식;

Expression 작성 예제 1

- 다음의 문장을 JAVA의 수식 표현으로 나타내어라

1. n이 0이 아니다

$n \neq 0$

2. n이 홀수이다

$n \% 2 == 1$

3. x는 1보다 크고 10보다 작다

$x > 1 \ \&\& \ x < 10$

4. $x - 10$ 이 y보다 크거나 같다

$x - 10 \geq y$

5. test 가 짝수이다

$test \% 2 == 0$

Expression 작성 예제 1

6. ch는 SPACE이거나 '₩t' 이다

```
ch == ' ' || ch == '₩t'
```

7. $0 < \text{number} < 10$ 인 number

```
number > 0 && number < 10
```

8. age가 6미만, 또는 60이상일 때

```
age < 6 || age >= 60
```

9. a가 0이상, 100미만이다

```
a >= 0 && a < 100
```

10. a가 0미만이거나 100이상이다

```
a < 0 || a >= 100
```


Expression 작성 예제 1

11. x, y, z 중에서 x가 가장 작은 수이다

```
x <= y && x <= z
```

12. a와 b의 합이 c와 d의 차보다 작다

```
a + b < c - d
```

13. 문자 c가 영문자가 아니다

```
!( c >= 'A' && c <= 'Z' || c >= 'a' && c <= 'z' ) )  
!( c >= 'A' && c <= 'Z' ) && !(c >= 'a' && c <= 'z' ) )
```

14. 평균(average)은 60이상이고 최소점수(min)는 40점 초과한다

```
(average >= 60) && (min > 40)
```

Expression 작성 예제 1

15. 온도(temperature)와 습도(humidity)가 모두 90 이상인
참기 어려운 여름날을 표현

```
temperature >= 90 && humidity >= 90
```

16. n이 0 이상 100 이하 범위에 포함될 때

```
n >= 0 && n <= 100
```

17. 6세 이상 12세 미만이나 60세 이상인 경우

```
age >= 6 && age < 12 || age >= 60
```

Expression 작성 예제 1

18. int형 변수 x가 10보다 크고 20보다 작을 때 true인 조건식

```
x > 10 && x < 20
```

19. char형 변수 ch가 공백이나 탭이 아닐 때 true인 조건식
- ```
!(ch == ' ' || ch == '\t') 또는 ch != ' ' && ch != '\t'
```

20. char형 변수 ch가 'x' 또는 'X'일 때 true인 조건식

```
ch == 'x' || ch == 'X'
```

21. char형 변수 ch가 숫자('0'~'9')일 때 true인 조건식

```
ch >= '0' && ch <= '9'
```

# Expression 작성 예제 2

- 다음 수학 식을 JAVA Expression으로 표현하여라.

$$b^2 - 4ac \longrightarrow b * b - 4.0 * a * c$$

$$\frac{a+b}{c-d} \longrightarrow (a + b) / (c - d)$$

$$2x + \frac{1}{2}y - \frac{1}{z^2} \longrightarrow 2.0 * x + y / 2.0 - 1.0 / (z * z)$$

- 다중 치환문

$x = y = 5;$   $\rightarrow$   $x = (y = 5);$  ... x와 y에 5 저장

$c = 10;$

$a = b = c + 5;$  ... a와 b에 15 저장

# Expression 작성 예제 3

- 영희는 학생회장에 출마하면서 다음과 같은 공약을 하였다.  
논리 연산자를 사용하여 공약 내용을 나타내 보아라



A && (B || C)

# Expression 작성 예제 4

- 다음은 사원 모집에 대한 내용이다. 지원 자격을 논리 연산자를 사용하여 표현하여보아라

－ 회사명 : ○○ 주식회사

— 지원 자격 : 아래 조건 (가), (나)를 모두 만족하는 사람

(가) C언어에 능통하거나 비주얼 베이직에 능통

A

## References

(나) 경력이 2년 이상이고 나이가 20세 이상

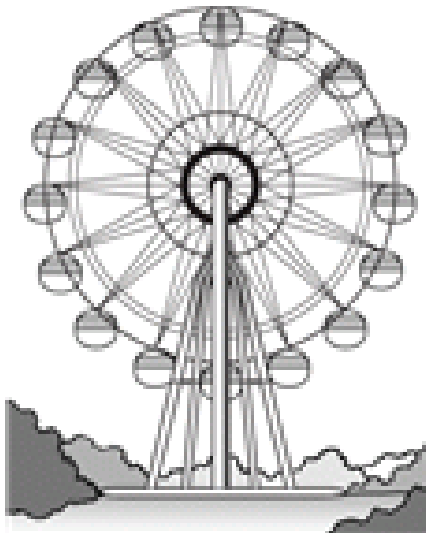
C

D

(A || B) && C && D

# Expression 작성 예제 5

- 그림의 놀이 기구를 탈 수 있는 조건을 논리식으로 바르게 표현한 것은?



<탑승 조건>

이 놀이 기구는 키가 130cm 이상이거나

A

나이가 16세 이상이어야 하며,

B

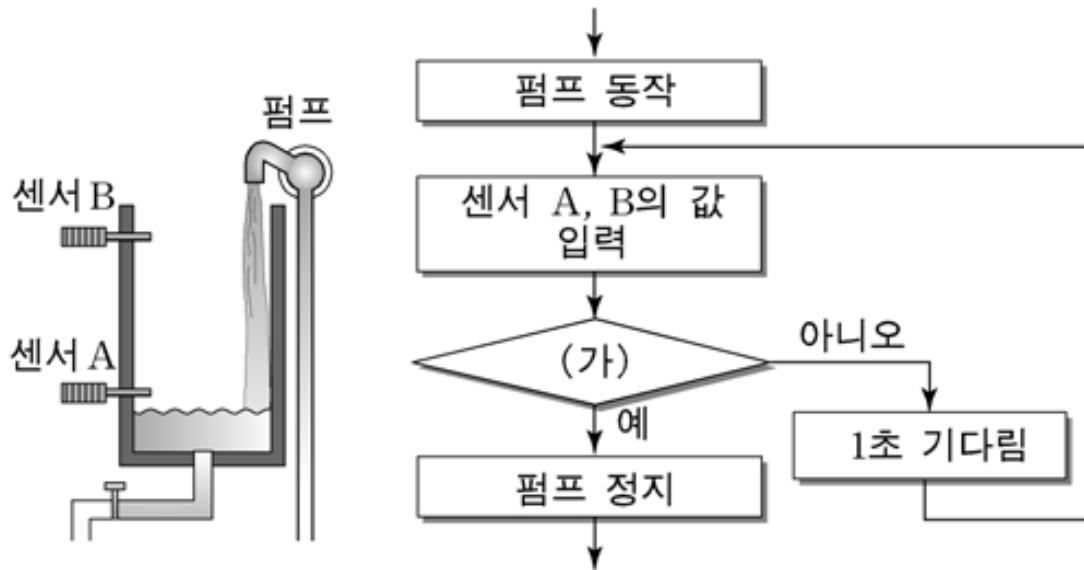
몸무게는 80Kg 이하이어야 탑승할 수 있습니다.

C

(A || B) && C

# Expression 작성 예제 6

- 다음 조건으로 센서 B가 물을 감지할 때까지 물탱크에 물을 채우려 한다. 순서도의 (가)에 해당하는 논리식으로 옳은 것은?



A && B

[조 건]

- ✓ 센서 A, B의 초기값은 거짓(false)이다.
- ✓ 센서가 물을 감지하면 센서의 값은 참(true)이 된다.
- ✓ 센서 A, B의 값이 모두 참이면 펌프는 정지한다



# Expression 작성 예제 7

- 다음 [조건]에 따라 작동되는 정수기 Y의 논리식으로 옳은 것은?



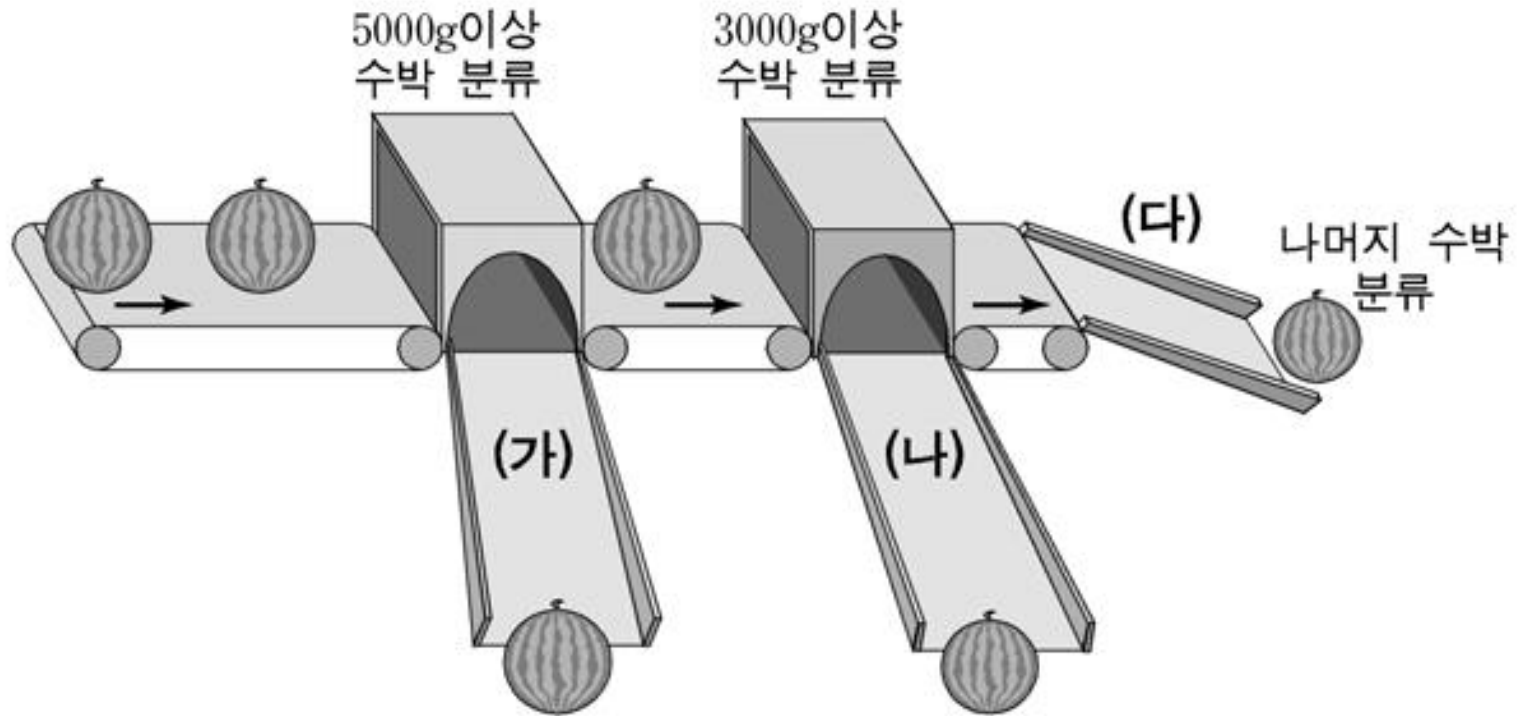
## [조건]

- ✓ 컵 감지센서 A는 컵을 감지하면 1이 되고, 그렇지 않으면 0이 된다.
- ✓ 작동버튼 B는 눌러진 상태이면 1이 되고, 그렇지 않으면 0이 된다.
- ✓ 정수기 Y는 컵 감지센서 A가 컵을 감지하고 작동버튼 B가 눌러진 상태이면 1이 되어 작동하고, 그렇지 않으면 0이 되어 작동하지 않는다.

$$Y = A \ \&\& \ B$$

# Expression 작성 예제 8

- 그림은 수박을 무게에 따라 단계로 분류하는 선별기이다. (나)로 배출되는 수박의 조건으로 옳은 것은? (단 변수  $w$ 는 수박 무게)



$W < 5000 \ \&\& \ w \geq 3000$

# Expression 작성 예제 9

■ 다음 Expression의 의미를 설명하여라.

1.  $x \leq 0$

x는 0보다 작거나 같다

2.  $\text{power} < \text{MAX\_POWER}$

power는 MAX\_POWER 보다 작다

3.  $\text{item} > \text{MIN\_ITEM}$

item은 MIN\_ITEM 보다 크다

4.  $x \geq y$

x는 y보다 크거나 같다

5.  $\text{mom\_or\_dad} == 'M'$

mom\_or\_dad는 'M'과 같다

# Expression 작성 예제 9

- 다음 Expression의 의미를 설명하여라.

6. num != SENTIN

num은 SENTIN과 같지 않다

# 연산자 예제 1

- 다음 Program의 실행 결과를 설명하여라.

- 문제1 (Short-circuit Evaluation)

```
int left = 7, right = 7;
```

```
if ((left++ < 7) && (++right < 7)){
 left = 10;
}
```

left의 값 : 8 right의 값 : 7

```
System.out.println("left의 값 : " + left + " right의 값 : " + right);
```

- 문제2

```
int left = 7, right = 7;
```

```
if ((left++ < 7) & (++right < 7)){
 left = 10;
}
```

left의 값 : 8 right의 값 : 8

```
System.out.println("left의 값 : " + left + " right의 값 : " + right);
```

# 연산자 예제 1

## ■ 문제3

```
int left = 7, right = 7;
```

```
if ((left++ < 7) || (++right < 7)){
 left = 10;
}
```

left의 값 : 8 right의 값 : 8

```
System.out.println("left의 값 : " + left + " right의 값 : " + right);
```

## ■ 문제4

```
int left = 7, right = 7;
```

```
if ((++left > 7) && (right++ < 7)){
 left = 10;
}
```

left의 값 : 8 right의 값 : 8

```
System.out.println("left의 값 : " + left + " right의 값 : " + right);
```

# 연산자 예제 1

## ■ 문제5

```
int left = 7, right = 7;
```

```
if ((++left > 7) && (--right < 7)){
 left = 10;
}
```

left의 값 : 10 right의 값 : 6

```
System.out.println("left의 값 : " + left + " right의 값 : " + right);
```

## ■ 문제6

```
int left = 7, right = 7;
```

```
if ((--left < 7) || (right-- < 7)){
 left = 10;
}
```

left의 값 : 10 right의 값 : 7

```
System.out.println("left의 값 : " + left + " right의 값 : " + right);
```

# 연산자 예제 2

■ 다음 Program의 실행 결과를 설명하여라

```
public static void main(String[] args) {
 boolean b1 = true;
 boolean b2 = false;
 boolean b3 = true;

 System.out.println(b1 && b2);
 System.out.println(b1 && b3);
 System.out.println(b1 || b2);
 System.out.println(b1 ^ b2);
 System.out.println(b1 ^ b3);
 System.out.println(!b1);
 System.out.println(!b2);
}
```

false  
true  
true  
true  
false  
false  
false



# 연산자 예제 3

- 다음 Program의 실행 결과를 설명하여라

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

```
 int a = 5;
```

```
 int b = -3;
```

```
 System.out.printf("%d < %d is %b\n", a, b, a < b);
```

```
 System.out.printf("%d == %d is %b\n", a, b, a == b);
```

```
 System.out.printf("%d != %d is %b\n", a, b, a != b);
```

```
 System.out.printf("%d > %d is %b\n", a, b, a > b);
```

```
 System.out.printf("%d <= %d is %b\n", a, b, a <= b);
```

```
 System.out.printf("%d >= %d is %b\n", a, b, a >= b);
```

```
}
```

5 < -3 is false

5 == -3 is false

5 != -3 is true

5 > -3 is true

5 <= -3 is false

5 >= -3 is true

# 연산자 예제 4

- 다음 Program의 실행 결과를 설명하여라

```
public static void main(String[] args) {
 boolean a = true;
 boolean b = true;
 boolean c = false;

 System.out.printf("%b && %b is %b\n", a, b, a && b);
 System.out.printf("%b && %b is %b\n", a, c, a && c);
 System.out.printf("%b && %b is %b\n", c, a, c && a);
 System.out.printf("%b || %b is %b\n", a, c, a || c);
 System.out.printf("%b || %b is %b\n", c, a, c || a);
 System.out.printf("%b || %b is %b\n", c, c, c || c);
}
```

```
true && true is true
true && false is false
false && true is false
true || false is true
false || true is true
false || false is false
```

# 연산자 예제 5

■ 다음 Program의 실행 결과를 설명하여라

```
public static void main(String[] args) {
 int a = 0, b = 0;
 boolean x;
```

```
x = false, a = 0, b = 0
x = true, a = 0, b = 0
x = true, a = 356, b = 356
x = true, a = 357, b = 357
```

```
 x = false && ((a = b = 356) != 0);
 System.out.printf("x = %b, a = %d, b = %d\n", x, a, b);
 x = true || ((a = ++b) != 0);
 System.out.printf("x = %b, a = %d, b = %d\n", x, a, b);
 x = true && ((a = b = 356) != 0);
 System.out.printf("x = %b, a = %d, b = %d\n", x, a, b);
 x = false || ((a = ++b) != 0);
 System.out.printf("x = %b, a = %d, b = %d\n", x, a, b);
}
```

# 도입 예제

- 양의 정수를 입력 받아 **홀수**인지 **짝수**인지를 판단하여 출력하는 Program을 작성하여라.



# 도입 예제

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

- 입력

- 양의 정수 (num) – int

- 출력

- 결과(result) – char

- 계산 방법

- 짝수 홀수 검사는 해당 정수를 2로 나누어 나머지가 있으면 “홀수” 그렇지 않으면 “짝수”

- 조건 연산자

## ■ 화면 설계

정수 입력 : 5 <enter>

입력한 정수 5는 홀수 입니다.

# 도입 예제

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 int digit;
 String result;
 char ch;

 System.out.print(" 정수 입력 : ");
 digit = keyboard.nextInt();

 ch = (digit % 2) == 0 ? '짝' : '홀';
 result = (digit % 2) == 0 ? "짝수" : "홀수" ;

 System.out.printf(" 입력한 정수 %d는 %s입니다.\n", digit,
 (digit % 2) == 0 ? "짝수" : "홀수");
 System.out.printf(" 입력한 정수 %d는 %c수입니다.\n", digit, ch);
 System.out.printf(" 입력한 정수 %d는 %s입니다.\n", digit, result);
}
```

# 도입 예제 [심화]

- 정수를 입력 받아 음수, 영, 양수를 구분하여 출력하는 Program을 작성하여 보자
- 출력 형식
  - 입력 받은 숫자 5는 양수입니다.
  - 입력 받은 숫자 0은 영입니다.
  - 입력 받은 숫자 -8은 음수입니다.

# 도입 예제 [심화]

■ 정수를 입력 받아 음수, 영, 양수를 구분하여 출력하고, 짝수와 홀수도 구분하여 출력하여라.

■ 출력 형식

입력 받은 숫자 5는 양수이고, 홀수 입니다.

입력 받은 숫자 0은 영 입니다.

입력 받은 숫자 -8은 음수이고, 짝수입니다.



# 도입 예제 [심화]

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 int digit;
 char ch, test;
 String result;

 System.out.print(" 정수 입력 : ");
 digit = keyboard.nextInt();

 test = (digit > 0) ? '양' : (digit == 0) ? '영' : '음';
 ch = (digit % 2) == 0 ? '짝' : '홀';
 result = (test == '영') ?
 String.format(" 입력한 정수 %d는 %c 입니다.\n", digit, test) :
 String.format(" 입력한 정수 %d는 %c수이고, %c수 입니다.\n",
 digit, test, ch);

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

# 사각형 넓이 계산

- 다음과 같이 사각형의 가로 길이와 세로 길이를 입력 받아 넓이를 계산하여라.

- 출력

```
**** 정사각형 넓이 ****
가로 길이 : 3 Cm
세로 길이 : 3 Cm
넓이 : 9 Cm2
```

# 사각형 넓이 계산

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 사각형 가로 길이(garo) – int

- 사각형 세로 길이(sero) – int

## ■ 출력

- 사각형 넓이(area) – long

## ■ 계산 방법

- 사각형 넓이 = 사각형 가로 길이 \* 사각형 세로 길이

- 입력 길이가 음수이거나 0이면 계산하지 않으며, 메시지를 출력

- 사각형의 가로 길이와 세로 길이가 같으면 **정사각형**  
그렇지 않으면 **직사각형**

# 사각형 넓이 계산

```
public static void main(String[] args) {
 Scanner input = new Scanner(System.in);
 int garo, sero, area;
 String result;

 System.out.printf("\n 사각형의 가로길기와 세로 길이를 입력 : ");
 garo = input.nextInt();
 sero = input.nextInt();

 area = garo * sero;
 result = area > 0 ?
 String.format("\n**** %c사각형 넓이 ****\n 가로 : %d Cm\n + 세로 : %d Cm\n 넓이 : %d Cm2\n garo == sero ? '정' : '직', garo, sero, area) :
 String.format("가로, 세로 길이를 정확하게 입력하세요\n");
 System.out.printf(result);
}
```

# 올바른 문자

- 문자형 변수 ch가 입력 받은 문자가 영문자(대문자 또는 소문자)이거나 숫자일 때만 “올바른 문자”를 출력하는 Program을 만들어보자

```
('a' <= ch && ch <= 'z') ||
('A' <= ch && ch <= 'Z') ||
('0' <= ch && ch <= '9')
```

# 올바른 문자

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 문자(ch) - char

## ■ 출력

- 올바른 문자 또는 틀린 문자

## ■ 계산 방법

- 조건 연산자 사용하기

- `( 'a' <= ch && ch <= 'z' ) ||`  
`( 'A' <= ch && ch <= 'Z' ) ||`  
`( '0' <= ch && ch <= '9' )`

# 올바른 문자

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 char ch;
 String result;

 System.out.print(" 문자 입력 : ");
 ch = keyboard.next().charAt(0);
 result = ('a' <= ch && ch <= 'z') ||
 ('A' <= ch && ch <= 'Z') ||
 ('0' <= ch && ch <= '9') ? "올바른 문자" : "올바르지 않은 문자";

 System.out.printf("%c는 %s 입니다\n", ch, result);
}
```



# 큰수 찾기

- Keyboard로부터 같지 않은 두 개의 정수를 입력 받아 큰 수를 출력하는 Program을 작성하여라 (단, 조건 연산자 사용)



# 큰수 찾기

- 문제 분석

- Data는 무엇이고, Information는 무엇인가?

- 입력

- 숫자 1(num1) – int

- 숫자 2(num2) – int

- 출력

- 두수 중 큰수

- 계산 방법

- 조건 연산자

- 같은 수는 어떻게 할까 ?

# 큰수 찾기

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

 System.out.printf(" 2개의 정수 입력 : ");
 num1 = keyboard.nextInt();
 num2 = keyboard.nextInt();

 max = (num1 >= num2) ? num1 : num2;

 System.out.printf(" %d, %d 중에서 가장 큰 수는 %d 입니다.\n",
 num1, num2, max);
}
```

# 큰수 찾기

```
Scanner keyboard = new Scanner(System.in);
int num1, num2;
String max;

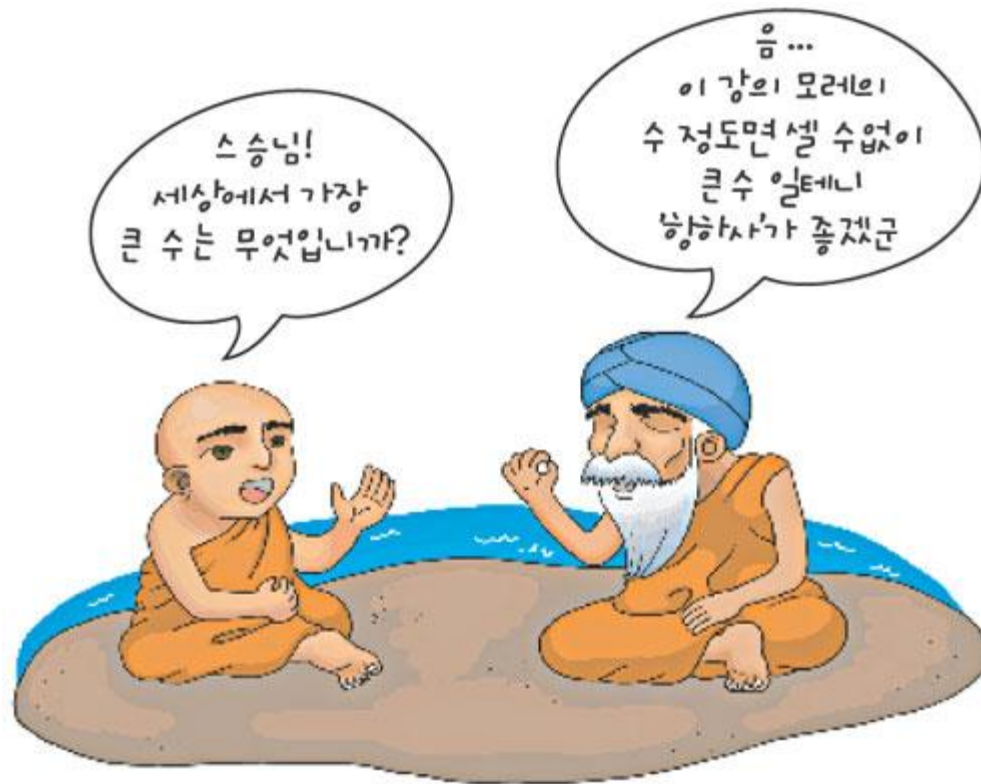
System.out.printf(" 2개의 정수 입력 : ");
num1 = keyboard.nextInt();
num2 = keyboard.nextInt();

max = (num1 > num2) ? " " + num1 :
 (num1 == num2) ? "같은 수" : " " + num2;

System.out.printf(" %d, %d 중에서 가장 큰 수는 %s 입니다.\n",
 num1, num2, max);
}
```

# 큰수 찾기[심화]

- Keyboard로부터 같지 않은 3개의 정수를 입력 받아 가장 큰 수를 출력하는 Program을 작성하여라
- (단, 조건 연산자 사용)



# 큰수 찾기[심화]

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 int num1, num2, num3, max;

 System.out.printf("3개의 정수 입력 : ");
 num1 = keyboard.nextInt();
 num2 = keyboard.nextInt();
 num3 = keyboard.nextInt();

 max = num1 > num2 ? num1 : num2;
 max = max > num3 ? max : num3;

 System.out.printf(" %d, %d 그리고 %d에서는 %d(이)가 가장 크다.\n",
 num1, num2, num3, max);
}
```

# 큰수 찾기[심화]

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 int num1, num2, num3, max;
 String result;

 System.out.printf("3개의 정수 입력 : ");
 num1 = keyboard.nextInt();
 num2 = keyboard.nextInt();
 num3 = keyboard.nextInt();

 result = (num1 - num2 == 0 || num1 - num3 == 0 ||
 num2 - num3 == 0) ?
 String.format("같은 수가 있습니다") :
 String.format("");
}
```



# 큰수 찾기[심화]

```
max = (max = num1 > num2 ? num1 : num2) > num3 ? max : num3;

result = result.equals("") ?
 String.format("%d(이)가 가장 크다.%n", max) : result;
System.out.printf(" %d, %d 그리고 %d에서는 " + result,
 num1, num2, num3);
}
```

# 성별 판별하기

- 성별을 남자면 'M', 여자면 'F'를 받아 남자, 여자를 출력하는 Program을 작성하자





# 성별 판별하기

- 문제 분석

- Data는 무엇이고, Information는 무엇인가?

- 입력

- 성별(gender) -char (대 소문자 입력 가능)

- 출력

- 결과(result) - char

- 계산 방법

- 대 소문자 입력 가능

- 조건 연산자 사용

# 성별 판별하기

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 char gender;
 char result;

 System.out.print(" 성별 입력(M or F) : ");
 gender = keyboard.next().charAt(0);

 result = (gender == 'M' || gender == 'm') ? '남' :
 (gender == 'F' || gender == 'f') ? '여' : '?';

 System.out.printf(" 당신의 성별은 %c자 입니다.\n", result);
}
```

# 성별 판별하기

```
public static void main(String[] args) throws IOException {
 int gender;
 char result;

 System.out.print(" 성별 입력(M or F) : ");
 gender = System.in.read();

 result = ((char) gender == 'M' || (char) gender == 'm') ? '남' :
 ((char) gender == 'F' || (char) gender == 'f') ? '여' : '?';

 System.out.printf(" 당신의 성별은 %c자 입니다.\n", result);
}
```

# 8의 배수

- 양의 정수를 입력 받아 8의 배수인가를 판정하는 Program을 작성하여라

# 8의 배수

- 문제 분석

- Data는 무엇이고, Information는 무엇인가?

- 입력

- 양의 정수(num) – long

- 출력

- 결과(result) – String

- 계산 방법

- 8의 배수 판정

- 양의 정수를 8로 나누어 나머지가 0이면 8의 배수임

- 조건 연산자 사용

# 8의 배수

## ■ 화면 설계

정수 입력 : 288 <enter>  
288는 8의 배수 입니다.

정수 입력 : 284 <enter>  
284는 8의 배수가 아닙니다.

# 8의 배수

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 final int BASE = 8;
 long num;
 String result;

 System.out.print("정수 입력 : ");
 num = keyboard.nextLong();

 result = (num % BASE == 0L) ?
 String.format("%d는 %d의 배수 입니다\n", num, BASE) :
 String.format("%d는 %d의 배수가 아닙니다\n", num, BASE);

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



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# 숫자 판단하기

- 정수 값 두 개를 읽어 들여 앞의 정수가 뒤의 정수보다 크면 '앞'을, 같으면 '0'을, 작으면 '뒤'을 출력시키는 Program을 작성하여라



# 숫자 판단하기

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 숫자1(num1) – int

- 숫자2(num2) – int

## ■ 출력

- 결과(result) – char

## ■ 계산 방법

- 관계 연산자 이용

- 조건 연산자 이용

- $(\text{num1} > \text{num2}) ? \text{'앞'} : ((\text{num1} == \text{num2}) ? \text{'0'} : ((\text{num1} < \text{num2}) ? \text{'뒤'} : \text{' '}))$

# 숫자 판단하기

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

두 정수 입력 : 4 5

```
 System.out.printf(" 두 정수 입력 : ");
 num1 = keyboard.nextInt();
 num2 = keyboard.nextInt();
```

입력한 두 수는 4와 5이다.  
결과는 뒤

```
 result = (num1 > num2) ? '앞' : ((num1 == num2) ? '0' :
 ((num1 < num2) ? '뒤' : ' '));
 System.out.printf("\n 입력한 두 수는 %d와 %d이다. \n",
 num1, num2);
 System.out.printf(" 결과는 %c\n", result);
}
```



# 표준체중 구하기

■ 성별에 따른 표준 체중과 미용 체중을 구하여라.

■ 표준 체중과 미용 체중

■ 남자 표준체중 = 키(m) \* 키(m) \* 22

■ 여자 표준체중 = 키(m) \* 키(m) \* 21

■ 여자 미용체중 = 키(m) \* 키(m) \* 20

# 표준체중 구하기

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 성별(gender) – char (1 : 남자, 2 : 여자)

- 키(신장 : height) – float (Cm)

## ■ 출력

- 표준 체중(standard) – float

- 미용 체중(beautiful) – float

## ■ 계산 방법

- 남자 표준체중 = 키(m) \* 키(m) \* 22

- 여자 표준체중 = 키(m) \* 키(m) \* 21

- 여자 미용체중 = 키(m) \* 키(m) \* 20

# 표준체중 구하기

## ■ 문제 분석

### ■ 남자 170Cm 이면

■ 표준체중 =  $1.7(m) * 1.7(m) * 22 = 63.58 \text{ Kg}$

### ■ 여자 170Cm 이면

■ 표준체중 =  $1.7(m) * 1.7(m) * 21 = 60.69 \text{ Kg}$

■ 미용체중 =  $1.7(m) * 1.7(m) * 20 = 57.80 \text{ Kg}$

# 표준체중 구하기

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 char gender;
 float height, standard, beautiful;
 String result;

 System.out.print(" 당신의 성별 입력 (남자 = 1, 여자 = 2) : ");
 gender = keyboard.next().charAt(0);
 System.out.print(" 당신의 키 입력 : ");
 height = keyboard.nextFloat();

 standard = (gender == '2') ? (height / 100) * (height / 100) * 21 :
 (height / 100) * (height / 100) * 22;
 beautiful = (gender == '2') ? (height / 100) * (height / 100) * 20 : 0;
```

# 표준체중 구하기

```
result = (gender == '1') ?
 String.format("키가 %.1fCm인 남성의 표준체중은 %.1fKg 입니다.\Wn",
 height, standard) :
 String.format("키가 %.1fCm인 여성의 표준체중은 %.1fKg 입니다.\Wn"
 + "키가 %.1fCm인 여성의 미용체중은 %.1fKg 입니다.\Wn",
 height, standard, height, beautiful);

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



# 거스름 돈 계산

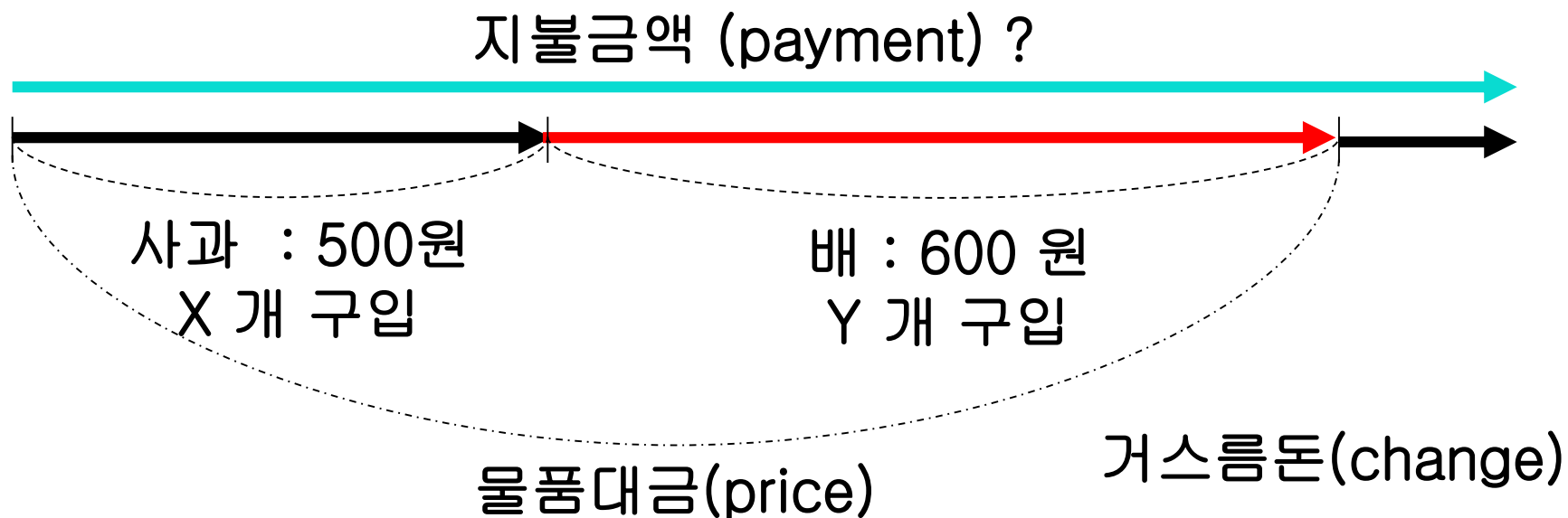
- 사과와 배의 개당 가격은 500원, 배의 개당 가격은 600원이다. 구입하고자 하는 사과와 배의 개수와 지불한 금액을 입력 받아 거스름 돈을 계산하는 Program을 만들어 보자
  - 이 문제를 해결하기 위하여 Programming 절차에 따라 Program을 설계하고 가상 언어로 표현하시오.
- 문제 분석
  - 문제 해결 방법
  - 입출력 설계
  - 가상 언어 표현





# 거스름 돈 계산

## ■ 문제 분석



## 문제 해결 방법

거스름 돈 계산을 위한 수식

$$\text{change} = \text{payment} - (500 \times X \text{ 개} + 600 \times Y \text{ 개})$$

# 거스름 돈 계산

- 입출력 설계
  - 입력 변수
    - apple : 사과 구입 수량 (int)
    - pear : 배 구입 수량 (int)
    - payment : 대금 지불 금액 (long)
  - 출력 변수
    - change : 거스름 돈 (int)

# 거스름 돈 계산

## ■ 가상 언어 표현

1. 사과와 배의 구입 수량 입력
2. 배의 구입 수량 입력
3. 물품 구입 가격을 계산
4. 받은 금액을 입력
5. 거스름 돈을 계산한다.
6. 계산 결과를(거스름 돈을) 출력

>> 좀더 생각하자 <<

- 1) 만약 받은 금액이 더 작은 경우도 반드시 생각하자.
- 2) 배와 사과를 적당하게 X 금액 만큼 주세요.

# 거스름 돈 계산

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 final int APPLE = 500;
 final int PEAR = 600;
 int apple, pear, payment;
 int price, change;
 String result;

 System.out.print("구인한 사과와 배의 개수 입력 : ");
 apple = keyboard.nextInt();
 System.out.print("구인한 배의 개수 입력 : ");
 pear = keyboard.nextInt();
 System.out.print("대금 지급을 위한 지불 금액 : ");
 payment = keyboard.nextInt();
}
```

# 거스름 돈 계산

```
price = apple * APPLE + pear * PEAR;
change = payment - price;
```

```
result = (change < 0) ?
```

```
 String.format("대금 지급을 위해 %,d 원을 더 주세요\n", -change) :
```

```
 String.format("거스름돈은 %,d 원 입니다\n", change);
```

```
System.out.print(result);
```

```
}
```

# 자판기 거스름돈

- 자판기에서 동전으로 거스름돈을 거슬러 줄 때 동전의 수량이 가정 적게 거슬러 주는 방법을 계산하여라.
- 단, 자판기는 500원, 100원, 50원짜리 동전만 있다고 가정하자
- 예) 1200원  
500원 : 2개  
100원 : 2개



# 자판기 거스름돈

- 문제 분석

- Data는 무엇이고, Information는 무엇인가?

- 입력

- 거스름 돈 (money) – int

- 50원 으로 나누어 져야 함

- 출력

- 최소의 동전으로 거스름돈 출력

- 알고리즘

- 나머지 연산자와 나눗셈 연산자 사용

# 자판기 거스름돈

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 final int WON500 = 500;
 final int WON100 = 100;
 final int WON50 = 50;
 int money;
 int coin500, coin100, coin50;
 String result;

 System.out.print("거스름 돈 입력 : ");
 money = keyboard.nextInt();

 coin50 = money;
 coin500 = coin50 / WON500;
 coin50 %= WON500;
 coin100 = coin50 / WON100;
 coin50 %= WON100;
 coin50 = coin50 / WON50;
```



# 자판기 거스름돈

```
System.out.printf("\n 거스름 돈 : %,d 원", money);
result = (money % WON50 == 0) ?
 String.format("\n %d원 동전 : %,3d 개 (%,10d 원)",
 WON500, coin500, coin500 * WON500) +
 String.format("\n %d원 동전 : %,3d 개 (%,10d 원)",
 WON100, coin100, coin100 * WON100) +
 String.format("\n %d원 동전 : %,3d 개 (%,10d 원)",
 WON50, coin50, coin50 * WON50) :
 String.format("\n 거스름 돈을 지불할 수 없음");
System.out.println(result);
}
```

# Leap Year

- Keyboard로부터 특정 연도를 입력 받아 윤년인지 평년인지를 판별하는 Program을 작성하라

**4년만에 돌아온**

**2월 29일**

# Leap Year

- 문제 분석

- Data는 무엇이고, Information는 무엇인가?

- 입력

- 년도(year) – int

- 출력

- 결과(result) – String

- 계산 방법

- 윤년은 4년마다 돌아오는데 100년으로 나누어 떨어지는 해는 평년이다. 하지만 400으로 나누어 떨어지는 해는 윤년

- $((\text{year} \% 4 == 0) \ \&\& \ (\text{year} \% 100 != 0)) \ || \ (\text{year} \% 400 == 0)$

# Leap Year

## ■ 문제 분석

✓  $1980/4 = 495$ ,  $1980/100 = 1.98$  : 윤년

✓  $2100/4 = 525$ ,  $2100/100 = 21$ ,  $2100/400 = 5.25$  : 평년

✓  $2000/4 = 500$ ,  $2000/100 = 20$ ,  $2000/400 = 5$  : 윤년

# Leap Year

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

 System.out.printf("년도를 입력 : ");
 year = keyboard.nextInt();

 int check1 = year % 4;
 int check2 = year % 100;
 int check3 = year % 400;
 result = ((check1 == 0) && (check2 != 0)) || (check3 == 0) ?
 "윤년" : "평년";

 System.out.printf("%d 년은 %s 입니다.\n", year, result);
}
```

# 정보처리산업기사

- 정보처리산업기사 필기시험의 합격 여부를 판별하는 Progreame를 작성하여라

- 과목

- 데이터베이스
- 전자계산기구조
- 시스템분석설계
- 운영체제
- 정보통신개론

- 합격 기준

- 100점을 만점으로 하여 과목당 40점 이상, 전 과목 평균 60점 이상

# 정보처리산업기사

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 데이터베이스 – int
- 컴퓨터 구조 – int
- 시스템 분석 설계 – int
- 운영 체제 – int
- 정보통신 개론 – int

## ■ 출력

- 합격 여부 – String

# 정보처리산업기사

- 문제 분석

- 계산 방법

- 100점을 만점으로 하여 과목당 40점 이상, 전 과목 평균 60점 이상
    - 과목당 40점 이상
    - 전 과목 평균 60점 이상 (AND 연산)



# 정보처리산업기사

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 int database, computer, system, operating, communication;
 final int CUTLINE = 40;
 String result;

 System.out.println("필기시험 합격 판정 프로그램");
 System.out.print(" 데이터베이스 점수 : ");
 database = keyboard.nextInt();
 System.out.print(" 전자계산기구조 점수 : ");
 computer = keyboard.nextInt();
 System.out.print(" 시스템분석설계 점수 : ");
 system = keyboard.nextInt();
 System.out.print(" 운영체제 점수 : ");
 operating = keyboard.nextInt();
 System.out.print(" 정보통신개론 점수 : ");
 communication = keyboard.nextInt();
}
```

# 정보처리산업기사

```
boolean check1 = database < CUTLINE || computer < CUTLINE ||
 system < CUTLINE
 || operating < CUTLINE || communication < CUTLINE;
int check2 = (database + computer + system +
 operating + communication) / 5;

result = check1 ? " 불 합격(과락)\n" :
 (check2 >= 60) ? "합격\n" : "불 합격(평균 점수 미달)\n";

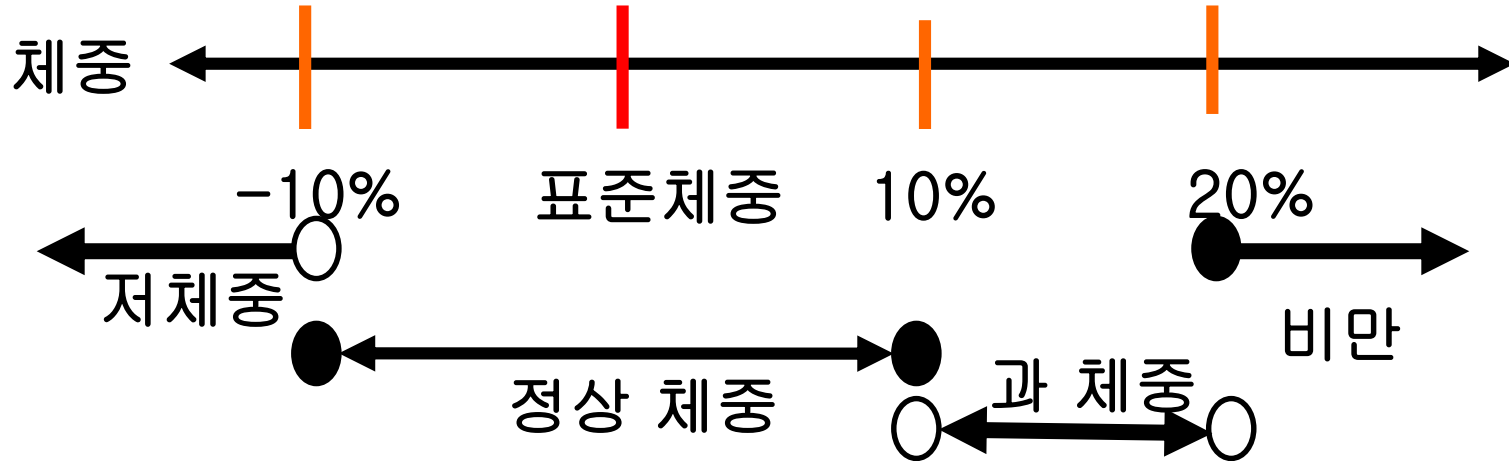
System.out.print(result);
}
```

# 나의 체중 판단하기

- 키(height)와 체중(weight)을 읽어 들어 표준체중(standard weight)을 판별 하는 Program을 작성하여라.
  - 판단 기준
    - 표준 체중의 10% 미만이면 **저체중**
    - 표준체중의 10% 내외 이면 **정상 체중**
    - 표준체중의 10% 초과에서 20% 미만이면 **과체중**
    - 표준체중의 20% 이상이면 **비만**

# 나의 체중 판단하기

## ■ 문제 분석



- **이내, 이상, 이하** 등등의 '**이**'는 모두 해당 숫자를 포함하는 걸 말함
- 포함하지 않으려면 '**미만**' 혹은 '**초과**' 라고 함

# 나의 체중 판단하기

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 키(height) – float

- 체중(weight) – float

## ■ 출력

- 체중 형태(type) – String

## ■ 계산방법

- 표준 체중 :  $(\text{키(hight)} - 100) * 0.9$

- 표준 체중 10%, -10%, 20% 값 구하기

- 체중이 10% 미만이면 저체중

- 체중이 10% 이내를 정상

- 체중이 10% 초과를 20%미만이면 과 체중

- 체중이 20% 이상을 비만으로 분류

- 산술 연산자와 조건 연산자 사용

# 나의 체중 판단하기

## ■ 가상 언어 표현

1. 키의 크기를 입력 받는다.
2. 몸무게를 입력 받는다.
3. 표준 몸무게를 계산  
$$\text{표준 몸무게} = (\text{키} - 100) * 0.9$$
4. 비만도를 계산 한다.
5. 비만도가 10% 이내를 정상, 10% 이상을 과 체  
중, 20% 이상을 비만으로 분류
6. 결과 값을 출력 한다.

# 나의 체중 판단하기

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 float height, weight, standard, ratio;
 String type;

 System.out.print(" 당신의 키 입력 : ");
 height = keyboard.nextFloat();
 System.out.print(" 당신의 몸무게 입력 : ");
 weight = keyboard.nextFloat();

 standard = (height - 100) * 0.9f;
 ratio = (weight - standard) / standard * 100;
 type = (ratio < -10.0f) ? "저 체중" :
 (ratio >= -10.0f && ratio <= 10.0f) ? "정상 체중" :
 (ratio > 10.0f && ratio <= 20.0f) ? "과 체중" : "비만";
}
```



# 나의 체중 판단하기

```
System.out.printf("\n 당신의 키는 %.2f Cm 입니다.", height);
System.out.printf("\n 당신의 체중은 %.2f Kg 입니다.", weight);
System.out.printf("\nWt 표준 체중 : %.2f Kg", standard);
System.out.printf("\nWt 비만도 : %.2f %%", ratio);
System.out.printf("\n 당신은 %s 상태 입니다.", type);
}
```



# BMI

- 키와 체중을 읽어 들어 정상과 비만을 신체 질량지수 BMI(Body Mass Index)로 판별 하는 Program을 작성하여라
- 조건 연산자 사용

# BMI

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

- 입력 : 키와 체중을 실수로 읽어 들임

- 

- $BMI = \text{체중(kg)} / \text{신장(m)}^2$

- 마른 사람 : BMI 20 미만

- 보통 : BMI 20 ~ 24 미만

- 경 비만 : BMI 24 ~ 26.4 미만

- 비만 : BMI 26.4 이상

-

# 페인트 칠하기

- 페인트를 칠하기를 원하는 면적을 입력하면 필요한 페인트 통(1통 = 180 m<sup>2</sup>)의 수를 구하는 Program을 작성하여라.



# 페인트 칠하기

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 페인트 칠할 면적(area) – int
- 페인트 1통으로 칠할 수 있는 면적(paint) – int (180)

## ■ 출력

- 준비해야 할 페인트 통의 수 (can) – int

## ■ 계산 방법

- 페인트 칠할 면적을 한 통의 페인트로 칠할 수 있는 면적으로 나눔
- 이때 나머지가 남으면 1통을 더 준비해야 함

# 페인트 칠하기

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 int area;
 final int PAINT = 180;
 int can;

 System.out.print(" 페인트 칠할 면적 입력 : ");
 area = keyboard.nextInt();

 can = area / PAINT;
 can += ((area % PAINT) == 0) ? 0 : 1;

 System.out.printf("필요한 페인트 통 : %d 통\n", can);
}
```



# 숫자 읽기

- 천만 이하(8자리 수)의 하나의 양의 정수를 입력 받아 우리가 사용하는 단위인 만, 천, 백, 십, 일 단위로 출력하는 Program을 작성하여라.

- 예)

정수 입력 : 2347653

입력한 수 2,347,653는 234만 7천 6백 5십 3 입니다.

입력한 수 2,347,653는 2백3십4만 7천 6백 5십 3 입니다

# 숫자 읽기

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 돈(money) -long (천만 단위 까지 입력)

## ■ 출력

- 읽는 문자열 (result) - String

## ■ 계산 방법

- 끝자리 숫자와 나머지 숫자 분리 방법

- $\text{money \% 10} \rightarrow$  끝자리

- $\text{money / 10} \rightarrow$  나머지 숫자

예) 1234를 123과 4를 분리

$1234 \% 10 \rightarrow 4$

$1234 / 10 \rightarrow 123$

# 숫자 읽기

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 int num;
 int man, temp;
 int one, ten, hundred, thousand;
 String result;

 System.out.print("숫자 입력 : ");
 num = keyboard.nextInt();
}
```



# 숫자 읽기

```
man = num / 10000;
temp = num % 10000;
thousand = man / 1000;
man %= 1000;
hundred = man / 100;
man %= 100;
ten = man / 10;
one = man % 10;
result = thousand > 0 ? String.format("%d천 ", thousand) : "";
result += hundred > 0 ? String.format("%d백 ", hundred) : "";
result += ten > 0 ? String.format("%d십 ", ten) : "";
result += one > 0 ? String.format("%d만 ", one) : "만 ";
```

# 숫자 읽기

```
thousand = temp / 1000;
temp %= 1000;
hundred = temp / 100;
temp %= 100;
ten = temp / 10;
one = temp % 10;
result += thousand > 0 ? String.format("%d천 ", thousand) : "";
result += hundred > 0 ? String.format("%d백 ", hundred) : "";
result += ten > 0 ? String.format("%d십 ", ten) : "";
result += one > 0 ? String.format("%d원\n", one) : "원\n";

result = (99999999 - num >= 0) ? "8자리 이상의 숫자입니다" : result;

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

# 소대문자 변환

- Keyboard에서 하나의 영문자를 입력하면 대문자이면 소문자로 소문자이면 대문자로 변경하는 Program을 작성하여라.
- ASCII 코드 테이블 상에서 알파벳 대문자는 65~90번, 소문자는 97~122번이라고 앞에서 언급했는데요. 테이블을 보면 정확히 32 크기의 간격으로 서로 떨어져 있습니다

# 소대문자 변환

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- 하나의 영문자(alphabet) -char

## ■ 출력

- 소문자 -> 대문자
- 대문자 -> 소문자

## ■ 계산 방법

- ASCII Code에서 Alphabet 'A' ~ 'Z' : 65 ~ 90,
- 'a' ~ 'z' : 97 ~ 122
- $97 - 65 = 32$ 씩 차이가 발생

# 소대문자 변환

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 char ch;
 int convert;

 System.out.print("하나의 영문자 입력 : ");
 ch = keyboard.next().charAt(0);

 convert = (ch >= 'A' && ch <= 'Z') ? ch + 32 : ch - 32;

 System.out.printf("%c - > %c\n", ch, convert);
}
```

# 문자 변환

- 영문자를 입력 받아 **대문자는 소문자, 소문자는 대문자로 바꾸는** 프로그램을 작성하라
  - 문자 변환은 ASCII Code 표를 이용하여라

# 문자 변환

- 문제 분석
  - 입력
    - 입력 문자(ch) – char
  - 출력
    - 출력 문자(convert) – char
  - 계산 방법
    - 관계 연산자와 조건 연산자를 이용
    - 문자 변경 (ASCII 코드표 참조)
      - 'a' – 32 = 'A'
      - 'A' + 32 = 'a'
- 출력 형식

입력문자 **A**는 **대**문자이고, **소**문자로 변경하면 **a** 입니다.  
입력문자 **a**는 **소**문자이고, **대**문자로 변경하면 **A** 입니다.

# 문자 변환

## ■ 아스키 코드표

| 오른쪽<br>왼쪽 | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0         | nul | soh | stx | etx | eot | enq | ack | bel | bs  | ht  |
| 1         | nl  | vt  | np  | cr  | so  | si  | dle | dc1 | dc2 | dc3 |
| 2         | dc4 | nak | syn | etb | can | em  | sub | esc | fs  | gs  |
| 3         | rs  | us  | sp  | !   | "   | #   | \$  | %   | &   | 48  |
| 4         | (   | )   | *   | +   | ,   | -   | .   | /   | 0   | 1   |
| 5         | 2   | 3   | 4   | 5   | 6   | 7   | 65  | 9   | :   | ;   |
| 6         | <   | =   | >   | ?   | @   | A   | B   | C   | D   | E   |
| 7         | F   | G   | H   | I   | J   | K   | L   | M   | N   | O   |
| 8         | P   | Q   | R   | S   | T   | U   | V   | W   | X   | Y   |
| 9         | Z   | [   | \   | ]   | ^   | _   | `   | a   | b   | c   |
| 10        | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   |
| 11        | n   | o   | p   | q   | r   | s   | t   | u   | v   | w   |
| 12        | x   | y   | z   | {   |     | }   | ~   | del |     |     |



# 문자 변환

## ■ 가상 언어 표현

1. 문자를 입력 받는다.
2. 문자가 영문자 인가를 판단한다.  
영문자가 아니면 종료한다.
3. 소문자인가 판단한다.  
소문자이면 대문자로 변경하여 출력한다.
4. 대문자 이면 소문자로 변경하여 출력한다.

# 문자 변환

```
public static void main(String[] args) throws IOException {
 int ch, result;
 char origin;
 char convert;

 System.out.print("\n 하나의 문자 입력 : ");
 ch = System.in.read();

 origin = (ch >= 'a' && ch <= 'z') ? '소' : '대';
 convert = (ch >= 'a' && ch <= 'z') ? '대' : '소';
 result = (ch >= 'a' && ch <= 'z') ? (ch + 'A' - 'a') : (ch + 'a' - 'A');

 System.out.printf("\n 입력문자 %c는 %c문자이고,
 %c문자로 변경하면 %c 입니다.\n",
 ch, origin, convert, result);
}
```

# 문자 변환

```
public static void main(String[] args) throws IOException {
 int ch, result;
 boolean flag;
 char origin;
 char convert;
 String message;

 System.out.print("\n 하나의 문자 입력 : ");
 ch = System.in.read();

 flag = ((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z'))
 ? true : false;

 origin = (ch >= 'a' && ch <= 'z') ? '소' : '대';
 convert = (ch >= 'a' && ch <= 'z') ? '대' : '소';
 result = (ch >= 'a' && ch <= 'z') ? (ch + 'A' - 'a') : (ch + 'a' - 'A');
```

# 문자 변환

```
message = flag == true ?
 String.format("\n 입력문자 %c는 %c문자이고,
 %c문자로 변경하면 %c 입니다.\n",
 ch, origin, convert, result) :
 String.format("\n 입력 오류 입니다.");
System.out.printf(message);
}
```

# 학점 계산

- 점수를 입력 받아 학점(A, B, C, D, F)을 계산하는 Program을 작성하여라
- 단, 사칙 연산자만을 사용할 것
- ASCII Code의 특성을 이용할 것

# 학점 계산

- 문제 분석
  - Data는 무엇이고, Information는 무엇인가?
- 입력
  - 점수(jumsu) – int
- 출력
  - 학점 (grade) – char
- 계산 방법

# 학점 계산-1

## ■ 점수와 ASCII Code 관계

| (jumsu / 10)의 값  | ASCII 코드 |
|------------------|----------|
| 10, 9, 8, 7, 6   | 65 (A)   |
| 5, 4, 3, 2, 1, 0 | 70 (F)   |

## ■ 10은 별도로 처리

■  $'A' + 9 - 9 = 'A'$

■  $'A' + 9 - 8 = 'A' + 1 = 'B'$

## ■ 50이하는 'F'

# 학점 계산-1

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 int jumsu;
 final int SCORE = 60;
 final int ASCII = 'A' + 9;
 int grade;

 System.out.print(" 점수 입력 : ");
 jumsu = keyboard.nextInt();

 grade = (jumsu < SCORE) ? 'F' : (jumsu == 100) ? 'A' :
 (ASCII - jumsu / 10);
 System.out.printf("점수 = %d\n학점 = %c\n", jumsu, grade);
}
```



# 학점 계산-2

## ■ 점수와 ASCII Code 관계

| (jumsu / 60)의 값 | (jumsu / 10)                  | ASCII 코드 |
|-----------------|-------------------------------|----------|
| 1               | $0 \leq \text{jumsu} \leq 10$ | 65 (A)   |
| 0               |                               | 70 (F)   |

### ■ 90점인 경우

■  $\text{check} = 1, \text{jumsu} = 9$

■  $X - (\text{jumsu} * \text{check}) = 65 \rightarrow X = 65 + 9 = 74$

### ■ 50점인 경우

■  $\text{check} = 0, \text{jumsu} = 5$

■  $74 - ((\text{jumsu} * \text{check}) + 4 * (1 - \text{check})) = 70$

### ■ 100점은 별도 처리

# 학점 계산-2

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 int jumsu, check, grade;
 String result;

 System.out.print(" 점수 입력 : ");
 jumsu = keyboard.nextInt();

 result = String.format(" 점수 = %d\n", jumsu);
 check = jumsu / 60;
 jumsu /= 10;
 grade = 74 - (jumsu * check + 4 * (1 - check)); //A 학점 수식

 result += (jumsu == 10) ? String.format(" 학점 = %c\n", 'A') :
 String.format(" 학점 = %c\n", grade);
 System.out.println(result);
}
```

# 학점 계산- 3

- 관계 연산자와 조건 연산자를 이용하여 계산하여라.

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 int score;
 int grade;

 System.out.print(" 점수 입력 : ");
 score = keyboard.nextInt();

 grade = 'A' * ((score >= 90) ? 1 : 0) +
 'B' * ((score >= 80 && score <= 89) ? 1 : 0) +
 'C' * ((score >= 70 && score <= 79) ? 1 : 0) +
 'D' * ((score >= 60 && score <= 69) ? 1 : 0) +
 'F' * ((score < 60) ? 1 : 0);

 System.out.printf("\n 입력 점수 : %d", score);
 System.out.printf("\n 학점 : %c\n", grade);
}
```

# 학점 계산(II)

- 점수를 입력 받아 학점(A0, A+, B0, B+, C0, C+, D0, D+, F)을 계산하는 Program을 작성하여라
- 단, 사칙 연산자만을 사용할 것
- ASCII Code의 특성을 이용할 것

# 학점 계산(II)

- +를 표시하기 위한 ASCII 관계

| $(\text{jumsu} \% 10) / 5$ | ASCII 코드 | 출력  |
|----------------------------|----------|-----|
| 1                          | 43       | +   |
| 0                          | 48       | '0' |
| $\text{jumsu} < 60$        | 32       | ' ' |

- 95점의 경우
  - $\text{plus} = 1$
  - $43 = 48 - X * \text{plus} \rightarrow X = 5$
- 90점인 경우
  - $\text{plus} = 0$
  - $48 - (5 * \text{plus}) = 48$

# 학점 계산(II)

- +를 표시하기 위한 ASCII 관계
  - 55점인 경우
    - $\text{plus} = 1$
    - $48 - (5 * \text{plus}) = 43$

60점 이하에 대해서는 +나, 0 기호를 표시하지  
말아야 하므로 60점 이하를 검사하는 별도로  
검사하여 이용

# 학점 계산(II)

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 int jumsu, check, grade, plus;
 String result;

 System.out.print(" 점수 입력 : ");
 jumsu = keyboard.nextInt();

 result = String.format(" 점수 = %d\n", jumsu);
 check = jumsu / 60;
 plus = (jumsu % 10) / 5;
 jumsu /= 10;
 grade = 74 - (jumsu * check + 4 * (1 - check));
}
```

# 학점 계산(II)

```
result += (jumsu == 10) ? String.format(" 학점 = %c", 'A') :
 String.format(" 학점 = %c", grade);
result += (jumsu == 10) ? String.format("%cWn", '+') :
 String.format("%cWn", (jumsu > 5) ? 48 - 5 * plus : 32);
System.out.println(result);
}
```



# 학점 계산(II)[심화]

## ■ Program의 우수한 기준

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 int score;
 int grade;
 char plus;

 System.out.print(" 당신의 JAVA 점수 입력 : ");
 score = keyboard.nextInt();

 grade = 'A' * ((score >= 90) ? 1 : 0) +
 'B' * ((score >= 80 && score <= 89) ? 1 : 0) +
 'C' * ((score >= 70 && score <= 79) ? 1 : 0) +
 'D' * ((score >= 60 && score <= 69) ? 1 : 0) +
 'F' * ((score < 60) ? 1 : 0);
 plus = (score == 100 || score % 10 >= 5) ? '+' : '0';
}
```



# 학점 계산(II)[심화]

## ■ Program의 우수한 기준

```
System.out.printf("\n 입력 점수 : %d", score);
System.out.printf("\n 학점 : %c%c\n",
 grade, grade == 'F' ? ' ' : plus);
}
```

# 학점 계산(II)[심화]

- Program의 우수한 기준
  - program의 크기(line 수)
  - 변수의 수
  - 제어문 사용횟수
    - Program의 크기와 조건문은 Memory와 Program의 실행 속도에 영향을 주기 때문임

# Metropolis

■ 어떤 도시가 Metropolis(거대도시)가 되려면 다음과 같은 2가지 조건 중의 하나를 만족하여야 한다고 가정하자.

- ① 한 나라의 수도이고 인구가 100만 이상이어야 한다.
- ② 연 소득이 1억 이상인 인구가 50만 이상이어야 한다.



# Metropolis

## ■ 문제 분석

- Data는 무엇이고, Information는 무엇인가?

## ■ 입력

- JAVA 실습 성적 (java) – float
- 컴퓨터 개론 성적 (computer) – float
- 디지털 공학 성적 (digital) – float
- 안드로이드 성적 (android) – float
- 이산수학 성적 (math) – float

## ■ 출력

- 평균 평점 (avg) – float
- 백분율 점수 (score) – float

다.

# Metropolis

- 문제 분석

- Data는 무엇이고, Information는 무엇인가?

- 입력

- 수도 인가 – boolean
  - 인구수 – int
  - 부자수 – int

- 출력

- Metropolis 여부 – boolean

- 계산 방법 (OR 조건)

- 한 나라의 수도이고 인구가 100만 이상이어야 한다.
  - 연 소득이 1억 이상인 인구가 50만 이상이어야 한다.

# Metropolis

```
public static void main(String[] args) {
 Scanner keyboard = new Scanner(System.in);
 boolean capital;
 int person, rich;

 System.out.print("수도 입니까 ? (수도 : 1, 수도 아님 0)");
 capital = keyboard.nextInt() == 1;
 System.out.print("인구 (단위 : 만) ");
 person = keyboard.nextInt();
 System.out.print("부자의 수(단위 : 만) ");
 rich = keyboard.nextInt();

 boolean metro = (capital && person >= 100) || rich >= 50;
 System.out.print(metro);
}
```