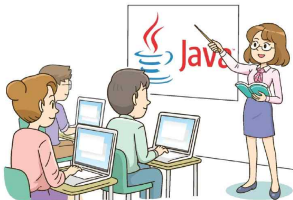


자바 프로그래밍



강의 자 : 오 은 성

목차

- 과제 풀이
- 수식과 연산자
- 산술연산/ 증감연산자/관계연산자
- 논리연산자 / 대입연산자/비트연산자
- 과제

Scanner로 쉽게 키 입력

3

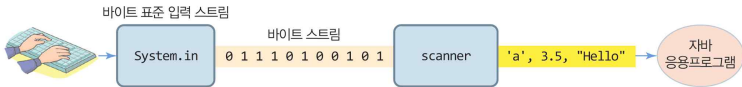
□ Scanner 클래스

- ▣ System.in에게 키를 읽게 하고, 읽은 바이트를 문자, 정수, 실수, 불린, 문자열 등 다양한 타입으로 변환하여 리턴

- java.util.Scanner 클래스

□ 객체 생성

```
import java.util.Scanner; // import 문 필요  
...  
Scanner a = new Scanner(System.in); // Scanner 객체 생성
```



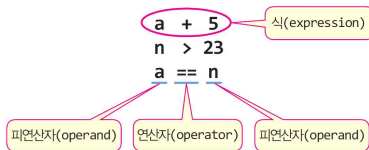
- System.in에게 키를 읽게 하고, 원하는 타입으로 변환하여 리턴

국어 영어 수학 점수를 입력 받아 변수에 저장하고 합계와 평균을 구해 출력해 주세요 반드시 평균은 소수점으로 형 변환 하기

```
1 import java.util.Scanner;
2 public class Home {
3     public static void main(String[] args) {
4         int kor , eng ,math,sum;
5         double avg;
6         Scanner in = new Scanner(System.in);
7         System.out.println("국어 점수입력");
8         kor = in.nextInt();
9         System.out.println("영어 점수입력");
10        eng= in.nextInt();
11        System.out.println("수학점수입력");
12        math = in.nextInt();
13        sum=kor+eng+math;
14        avg=(double)(kor+eng+math)/3;
15        System.out.println("3과목의 총점은 :"+sum);
16        System.out.println("3과목의 평균은 :"+avg);
17    }
18 }
```

식과 연산자

▶ 연산 : 3



연산의 종류	연산자	연산의 종류	연산자
증감	++ --	비트	& ^ ~
산술	+ - * / %	논리	&& ! ^
시프트	>> << >>>	조건	? :
비교	> < >= <= == !=	대입	= *= /= += -= &= ^= = <<= >>= >>>=

연산자 우선순위

6

<div> <div>높음</div> <div>↓</div> <div>낮음</div> </div>	++(postfix) --(postfix)
	+(양수 부호) -(음수 부호) ++(prefix) --(prefix) ~ !
	형 변환(type casting)
	* / %
	+(덧셈) -(뺄셈)
	<< >> >>>
	<> <= >= instanceof
	== !=
	& (비트 AND)
	^ (비트 XOR)
	(비트 OR)
	&& (논리 AND)
	(논리 OR)
	? : (조건)
	= += -= *= /= %= &= ^= = <<= >>= >>>=

- 같은 우선순위의 연산자
 - ▣ 왼쪽에서 오른쪽으로 처리
 - ▣ 예외)오른쪽에서 왼쪽으로
 - 대입 연산자, --, ++, +,-(양수 음수 부호), !, 형 변환은 오른쪽에서 왼쪽으로 처리
- 괄호는 최우선순위
 - ▣ 괄호가 다시 괄호를 포함한 경우는 가장 안쪽의 괄호부터 먼저 처리

산술 연산자

7

산술 연산자

▣ 더하기(+), 빼기(-), 곱하기(*), 나누기(/), 나머지(%)

▣ /와 % 응용

■ 10의 자리와 1의 자리 분리

$69/10 = 6$	← 몫 6
$69\%10 = 9$	← 나머지 9

■ n이 홀수인지 판단

<code>int r = n % 2;</code>	// r이 1이면 n은 홀수, 0이면 짝수
-----------------------------	-------------------------

연산자	의미	예	결과
+	더하기	25.5 + 3.6	29.1
-	빼기	3 - 5	-2
*	곱하기	2.5 * 4.0	10.0
/	나누기	5/2	2
%	나머지	5%2	1

예제2: 두개의 정수를 입력 받아 사칙연산을 수행해 보자

```
import java.util.*;
public class InputString
{
    public static void main(String[] args)
    {
        int a,b,result;
        double result1;
        Scanner s = new Scanner(System.in);

        System.out.println("첫번째 정수를 입력하세요");
        a=s.nextInt();
        System.out.println("두번째 정수를 입력하세요");
        b=s.nextInt();
        result = a + b;
        System.out.println(a + "+" + b + "=" + result);
        result = a - b;
        System.out.println(a + "-" + b + "=" + result);
        result = a * b;
        System.out.println(a + "*" + b + "=" + result);
        result1 = (double)a / b;
        System.out.println(a + "/" + b + "=" + result1);
        result = a % b;
        System.out.println(a + "%" + b + "=" + result);
    }
}
```


중감 연산자

연산자	의미
++x	x값을 먼저 증가한 후에 다른 연산에 사용한다. 이 수식의 값은 증가된 x값이다.
x++	x값을 먼저 사용한 후에, 증가한다. 이 수식의 값은 증가되지 않은 원래의 x값이다.
--x	x값을 먼저 감소한 후에 다른 연산에 사용한다. 이 수식의 값은 감소된 x값이다.
x--	x값을 먼저 사용한 후에, 감소한다. 이 수식의 값은 감소되지 않은 원래의 x값이다.

예제: 증감 연산자

직접 입력
하여 확인



UnaryOperator.java

```
01 public class UnaryOperator {  
02     public static void main(String[] args) {  
03         int x = 1;  
04         int y = 1;  
05  
06         int nextx = ++x; // x의 값이 사용되기 전에 증가된다. nextx는 2가 된다.  
07         int nexty = y++; // y의 값이 사용된 후에 증가된다. nexty는 1이 된다.  
08         System.out.println(nextx);  
09         System.out.println(nexty);  
10     }  
11 }
```

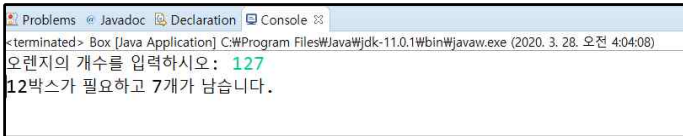
실행결과



```
2  
1
```

확인학습

1. 확인학습
2. 하나의 상자에 오렌지를 10개씩 담을 수 있다고 하자. 오렌지가 127개가 있다면 상자 몇 개가 필요한가? 또 몇 개의 오렌지가 남을 까?
3. 아래와 같이 출력하세요



```
Problems Javadoc Declaration Console
<terminated> Box [Java Application] C:\Program Files\Java\jdk-11.0.1\bin\javaw.exe (2020. 3. 28. 오전 4:04:08)
오렌지의 개수를 입력하시오: 127
12박스가 필요하고 7개가 남습니다.
```

예제 2-5 : /와 % 산술 연산

12

초 단위의 정수를 입력받고, 몇 시간, 몇 분, 몇 초인지 출력하는 프로그램을 작성하라.

```
import java.util.Scanner;

public class ArithmeticOperator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("정수를 입력하세요: ");
        int time = scanner.nextInt();    // 정수 입력
        int second = time % 60;          // 60으로 나눈 나머지는 초
        int minute = (time / 60) % 60;   // 60으로 나눈 몫을 다시 60으로 나눈 나머지는 분
        int hour = (time / 60) / 60;     // 60으로 나눈 몫을 다시 60으로 나눈 몫은 시간

        System.out.print(time + "초는 ");
        System.out.print(hour + "시간, ");
        System.out.print(minute + "분, ");
        System.out.println(second + "초입니다.");

    }
}
```

정수를 입력하세요:5000

5000초는 1시간, 23분, 20초입니다.

관계 연산자

표 2-3 • 관계 연산자

연산자 기호	의미	사용예
==	x와 y가 같은가?	$x == y$
!=	x와 y가 다른가?	$x != y$
>	x가 y보다 큰가?	$x > y$
<	x가 y보다 작은가?	$x < y$
>=	x가 y보다 크거나 같은가?	$x >= y$
<=	x가 y보다 작거나 같은가?	$x <= y$

예제: 관계 연산자

직접 입력
하여 확인



ComparisonOperator.java

```
01 public class ComparisonOperator {  
02  
03     public static void main(String[] args){  
04         int x = 3;  
05         int y = 4;  
06         System.out.print((x == y) + " ");  
07         System.out.print((x != y) + " ");  
08         System.out.print((x > y) + " ");  
09         System.out.print((x < y) + " ");  
10         System.out.print((x <= y) + " ");  
11     }  
12 }
```

실행결과



false true false true true

논리 연산자

연산자 기호	사용예	의미
&&	x && y	AND 연산, x와 y가 모두 참이면 참, 그렇지 않으면 거짓
	x y	OR 연산, x나 y중에서 하나만 참이면 참, 모두 거짓이면 거짓
!	!x	NOT 연산, x가 참이면 거짓, x가 거짓이면 참

예제: 논리 연산자

```
public class LogicalOperator {  
  
    public static void main(String[] args){  
        System.out.println((3 == 3) && (4 == 7)) ;  
        System.out.println((3 == 3 || 4 == 7)) ;  
    }  
}
```

false
true

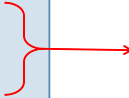
조건 연산자 ?:

17

- `condition ? opr2 : opr3`
 - 세 개의 피연산자로 구성된 삼항(ternary) 연산자
 - `condition`이 `true`이면, 연산식의 결과는 `opr2`, `false`이면 `opr3`
 - if-else를 간결하게 표현할 수 있음

```
int x = 5;  
int y = 3;
```

```
int s;  
if(x>y)  
    s = 1;  
else  
    s = -1;
```



```
int s = (x>y)?1:-1;
```

예제 2-8 : 조건 연산

18

다음은 조건 연산자의 사례이다. 실행 결과는 무엇인가?

```
public class TernaryOperator {  
    public static void main (String[] args) {  
        int a = 3, b = 5;  
  
        System.out.println("두 수의 차는 " + ((a>b)?(a-b):(b-a)));  
    }  
}
```

두 수의 차는 2

확인학습

1. 당신의 나이를 입력 받아 성년인지 미성년자인지 출력해주세요(단 조건연산자를 사용하여 프로그램 작성 19살이상이면 성년 그렇지 않으면 미성년 출력)

비트 연산

20

□ 비트 개념

byte x = 10;



□ 비트 연산

□ 비트 논리 연산

- 비트끼리 AND, OR, XOR, NOT 연산

□ 비트 시프트 연산

- 비트를 오른쪽이나 왼쪽으로 이동

비트 논리 연산

21

□ 피 연산자의 각 비트들의 논리 연산

$$\begin{array}{r} 01101010 \\ \& 11001101 \\ \hline 01001000 \end{array}$$

모두 1이므로
결과는 1

둘 중 하나라도
0이면 결과는 0

$$\begin{array}{r} 01101010 \\ | 11001101 \\ \hline 11101111 \end{array}$$

모두 0이므로
결과는 0

둘 중 하나라도
1이면 결과는 1

$$\begin{array}{r} 01101010 \\ ^ 11001101 \\ \hline 10100111 \end{array}$$

두 비트가 같으므로
결과는 0

두 비트가 다르므로
결과는 1

$$\begin{array}{r} \sim 01101010 \\ 10010101 \end{array}$$

1은 0으로 변환

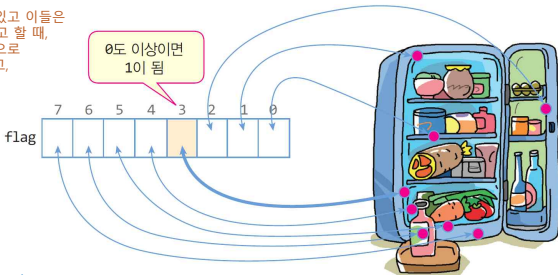
0은 1로 변환

연산자	별칭	내용
$a \& b$	AND 연산	두 비트 모두 1이면 1, 그렇지 않으면 0
$a b$	OR 연산	두 비트 모두 0이면 0, 그렇지 않으면 1
$a ^ b$	XOR 연산	두 비트가 다르면 1, 같으면 0
$\sim a$	NOT 연산	1을 0으로, 0을 1로 변환

비트 논리 연산 응용

22

냉장고에는 8개의 센서가 있고 이들은 flag 변수와 연결되어 있다고 할 때, 냉장고의 온도가 0도 이상으로 올라가면 비트 3이 1이 되고, 0도 이하이면 비트 3이 0을 유지한다.



문제) 현재 냉장고의 온도가 0도 이상인지 판단하는 코드를 작성하라.

```
byte flag = 0b00001010; // 각 비트는 8개의 센서 값을 가리킴
if(flag & 0b00001000 == 0)
    System.out.print("온도는 0도 이하");
else
    System.out.print("온도는 0도 이상");
```

온도는 0도 이상

	x	x	x	x	Y	x	x	x
&	0	0	0	0	1	0	0	0
	0	0	0	0	Y	0	0	0

Y비트가 0이면
& 결과는 0

시프트 연산자의 사례

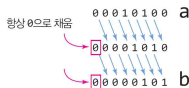
23

□ 피 연산자의 비트들을 이동 연산

```
byte a = 5; // 5
byte b = (byte)(a << 2); // 20
```



```
byte a = 20; // 20
byte b = (byte)(a >> 2); // 5
```



```
byte a = 20; // 20
byte b = (byte)(a >> 2); // 5
```



```
byte a = (byte)0xf8; // -8
byte b = (byte)(a >> 2); // -2
```



시프트 연산자	내용
a >> b	a의 각 비트를 오른쪽으로 b번 시프트한다. 최상위 비트의 빈자리는 시프트 전의 최상위 비트로 다시 채운다. 산술적 오른쪽 시프트라고 한다.
a >>> b	a의 각 비트를 오른쪽으로 b번 시프트한다. 최상위 비트의 빈자리는 항상 0으로 채운다. 논리적 오른쪽 시프트라고 한다.
a << b	a의 각 비트를 왼쪽으로 b번 시프트한다. 최하위 비트의 빈자리는 항상 0으로 채운다. 산술적 왼쪽 시프트라고 한다.

& 연산자 : 비트 단위 AND

0 & 0 → 0을 반환

0 & 1 → 0을 반환

1 & 0 → 0을 반환

1 & 1 → 1을 반환

```
public class Ex04 {  
    public static void main(String[] args) {  
        int a=15;    // 00000000 00000000 00000000 00001111  
        int b=20;    // 00000000 00000000 00000000 00010100  
        int c = a&b;  
        System.out.println(a+"(" +Integer.toBinaryString(a)+")");  
        System.out.println(b+"(" +Integer.toBinaryString(b)+")");  
        System.out.println(c+"(" +Integer.toBinaryString(c)+")");  
    }  
}
```


□ & 연산자 : 비트 단위 AND

	00000000	00000000	00000000	00001111
AND	00000000	00000000	00000000	00010100
<hr/>				
	00000000	00000000	00000000	00000100

비트 단의 OR

0 | 0 → 0을 반환

0 | 1 → 1을 반환

1 | 0 → 1을 반환

1 | 1 → 1을 반환

```
public class Ex04 {  
    public static void main(String[] args) {  
        int a=15;    // 00000000 00000000 00000000 00001111  
        int b=20;    // 00000000 00000000 00000000 00010100  
        int c = a|b;  
        System.out.println(a+"(" +Integer.toBinaryString(a)+")");  
        System.out.println(b+"(" +Integer.toBinaryString(b)+")");  
        System.out.println(c+"(" +Integer.toBinaryString(c)+")");//31(11111)  
    }  
}
```

^ 연산자 : 비트 단위 XOR

	00000000	00000000	00000000	00001111
XOR	00000000	00000000	00000000	00010100
<hr/>				
	00000000	00000000	00000000	00011011

~ 연산자 : 비트 단위 NOT

~ 0 → 1을 반환

~ 1 → 0을 반환

```
public class Ex04 {  
    public static void main(String[] args) {  
        int a=15;  
        int b=~a;  
        System.out.println(a+"(" +Integer.toBinaryString(a)+")");  
        System.out.println(b+"(" +Integer.toBinaryString(b)+")");//-16  
    }  
}
```

~ 연산자 : 비트 단위 NOT

NOT 00000000 00000000 00000000 00001111

11111111 11111111 11111111 11110000

부호 비트도 반전

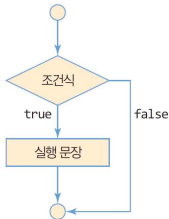
단순 if문

30

□ 단순 if 문

- ▣ if의 괄호 안에 조건식(논리형 변수나 논리 연산)
 - 실행문장이 단일 문장인 경우 둘러싸는 { } 생략 가능

```
if (조건식) {  
    ...실행 문장... // 조건식이 참인 경우  
}
```



```
if(n%2 == 0) {  
    System.out.println(n + "은 짝수입니다.");  
}
```

예제 2-10 : if문 사용하기

31

시험 점수가 80점이 이상이면 합격 판별을 하는 프로그램을 작성하시오.

```
import java.util.Scanner;

public class SuccessOrFail {
    public static void main (String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("점수를 입력하시오: ");
        int score = scanner.nextInt();
        if (score >= 80)
            System.out.println("축하합니다! 합격입니다.");
    }
}
```

점수를 입력하시오: 95
축하합니다! 합격입니다.

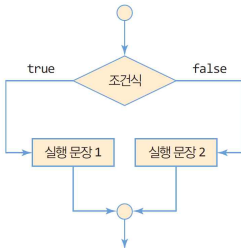
조건문 – if-else

32

□ if-else 문

- 조건식이 true면 실행문장1 실행 후 if-else문을 벗어남
- false인 경우에 실행문장2 실행후, if-else문을 벗어남

```
if (조건식) {  
    ...실행 문장 1...  
}  
else {  
    ...실행 문장 2...  
}
```



예제 2-11 : if-else 사용하기

33

입력된 수가 3의 배수인지 판별하는 프로그램을 작성하시오.

```
import java.util.Scanner;

public class MultipleOfThree {
    public static void main (String[] args) {
        Scanner in = new Scanner(System.in);

        System.out.print("수를 입력하시오: ");
        int number = in.nextInt();

        if (number % 3 == 0)
            System.out.println("3의 배수입니다.");
        else
            System.out.println("3의 배수가 아닙니다.");

        scanner.close();
    }
}
```

수를 입력하시오: 129
3의 배수입니다.

예제: 짝수, 홀수 구별하기

키보드에서 입력 받은 정수가 홀수인지 짝수
인지를 말해 주는 프로그램을 작성하여 보자.



정수를 입력하시오: 2

입력된 정수는 짝수입니다.

프로그램이 종료되었습니다.

다중 if-else 문

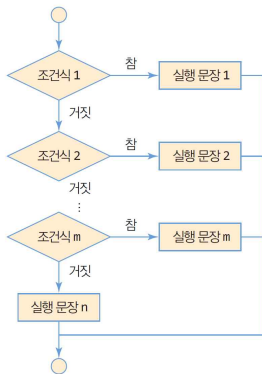
35

□ 다중 if-else 문

▣ if-else가 연속되는 모양

- 조건문이 너무 많은 경우, switch 문 사용 권장

```
if (조건식 1) {  
    실행 문장 1; // 조건식 1이 참인 경우  
}  
else if (조건식 2) {  
    실행 문장 2; // 조건식 2가 참인 경우  
}  
else if (조건식 m) {  
    ..... // 조건식 m이 참인 경우  
}  
else {  
    실행 문장 n; // 앞의 모든 조건이 거짓인 경우  
}
```



예제: 다중 if-else 문

- 사용자로부터 받은 정수가 양수인지, 0인지, 음수인지를 검사해보자.

정수를 입력하시오: 10
양수입니다.

예제:

```
import java.util.Scanner;

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

        Scanner sc = new Scanner(System.in);
        System.out.print("정수를 입력하시오: ");
        int number1 = sc.nextInt();
        if(number1 > 0) {
            System.out.println("양수입니다.");
        }
        else if (number1 == 0) {
            System.out.println("0입니다.");
        }
        else {
            System.out.println("음수입니다.");
        }
    }
}
```

확인학습

1. 사용자로부터 두 개의 정수를 입력 받아서 둘 중에 큰 수를 출력하는 프로그램을 작성하여 보자.(두개의 정수가 같으면 같습니다 출력)



```
첫번째 정수: 10  
두번째 정수: 20  
큰 수는 20
```

LAB: 성적에 따라 학점 결정하기



종종 우리는 조건에 따라서 다중으로 분기되는 결정을 내려야 하는 경우가 있다. 학생들의 성적을 받아서 학점을 출력하는 프로그램을 작성하여 실행하여 보자. 성적이 90점 이상이면 A학점, 80점 이상이고 90점 미만이면 B학점, 70점 이상이고 80점 미만이면 C학점과 같이 결정하는 것이다.



성적을 입력하시오: 92

학점 A

Grading.java

```
01 import java.util.Scanner;
02
03 public class Grading {
04     public static void main(String[] args) {
05         int score;
06
07         Scanner input = new Scanner(System.in);
08         System.out.print("성적을 입력하십시오: ");
09         score = input.nextInt();
10         if (score >= 90)           // ①
11             System.out.println("학점 A");
12         else if (score >= 80)      // ②
13             System.out.println("학점 B");
14         else if (score >= 70)
15             System.out.println("학점 C");
16         else if (score >= 60)
17             System.out.println("학점 D");
18         else
19             System.out.println("학점 F");
20
21     }
22 }
```

문장 ②에서 if(grade >= 80 && grade < 90)이라고 할 필요가 없음에 유의하라. grade가 90보다 크거나 같은 경우에는 앞의 문장 ①에서 이미 걸렸기 때문이다.

switch 선택 구조

전체적인 구조



형식

```
switch(변수)
{
    case c1:
        처리문장1;
        break;
    case c2:
        처리문장2;
        break;
    ...
    default:
        처리문장d;
        break;
}
```

'A'

변수와 일치하는 값을 가진 case 절이 실행됩니다.



예제: 정수-> 한글

- 사용자가 값을 입력하면 화면에 "하나", "둘"과 같이 출력하는 코드를 작성하여 보자.



숫자를 입력하시오: 1

하나

사용자가 number에
1을 입력한다.



```
switch(number)
```

```
{
```

```
case 0:
```

```
    System.out.println("없음");
```

```
    break;
```

```
case 1:
```

```
    System.out.println("하나");
```

```
    break;
```

```
case 2:
```

```
    System.out.println("둘");
```

```
    break;
```

```
default:
```

```
    System.out.println("많음");
```

```
    break;
```

```
}
```

1

값이 0이면
이쪽으로 오세요!

값이 1이면
이쪽으로 오세요!

값이 2이면
이쪽으로 오세요!

일치하는 값이 없으면
이쪽으로 오세요!

만약 break 문이 없다면



```
switch(number)
{
    case 0:
        System.out.println("없음");
        break;
    case 1:
        System.out.println("하나");
    case 2:
        System.out.println("둘");
        break;
    default:
        System.out.println("많음");
        break;
}
```

case 1에 break문이 없으므로 case 2의 문장들이 이어서 실행됩니다.



LAB: 월의 일수 출력하기

sun	mon	tue	wed	thu	fri	sat
			1	2	3	4
5	6	7	8	9	10	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

실행결과



일수를 알고 싶은 월을 입력하시오: 12

월의 날수는 31

일수를 알고 싶은 월을 입력하시오: 6

월의 날수는 30



DaysInMonth.java

```
01 import java.util.*;
02
03 public class DaysInMonth {
04     public static void main(String[] args) {
05         int month;
06         int days = 0;
07
08         System.out.print("일수를 알고 싶은 월을 입력하십시오:");
09         Scanner scan = new Scanner(System.in);
10
11         month = scan.nextInt();
12         switch (month) {
13             case 4:
14             case 6:
15             case 9:
16             case 11:
17                 days = 30;
18                 break;
19             case 2:
20                 days = 28;
21                 break;
22             default:
23                 days = 31;
24                 break;
25         }
26         System.out.println("월의 날수는 " + days);
27     }
28 }
```

고의적으로 break문을 생략
하여서 여러 가지 경우를 동일한
문장으로 처리하고 있다.

예제 : switch 문으로 학점 매기기

47

예제 2-12의 성적 매기는
코드를 switch 문으로
작성하라.

```
import java.util.Scanner;
public class GradingSwitch {
    public static void main (String[] args) {
        Scanner scanner = new Scanner(System.in);

        char grade;
        System.out.print("점수를 입력하세요(0~100): ");
        int score = scanner.nextInt();
        switch (score/10) {
            case 10: // score = 100
            case 9: // score는 90~99
                grade = 'A';
                break;
            case 8: // score는 80~89
                grade = 'B';
                break;
            case 7: // score는 70~79
                grade = 'C';
                break;
            case 6: // score는 60~69
                grade = 'D';
                break;
            default: // score는 59 이하
                grade = 'F';
        }
        System.out.println("학점은 "+grade+"입니다");
        scanner.close();
    }
}
```

점수를 입력하세요(0~100): 89
학점은 B입니다.

예제 1 :두개의 정수를 입력 받아서 입력 받은 연산기호의
연산 수행 후 출력해주세요 (if문 사용)

실행결과



```
Problems Javadoc Declaration Console
<terminated> Ex02 (10) [Java Application] C:\Program Files\Java\jdk-11.0.1\bin\javaw.exe (2020. 4. 3. 오전 1:20:36)
첫번째 수 입력 10
두번째 수 입력 5
연산기호 입력 +
10+5=15
```

```
Problems Javadoc Declaration Console
<terminated> Ex02 (10) [Java Application] C:\Program Files\Java\jdk-11.0.1\bin\javaw.exe (2020. 4. 3. 오전 1:23:43)
첫번째 수 입력 10
두번째 수 입력 5
연산기호 입력 -
10-5=5
```


SOLUTION

```
1 import java.util.Scanner;
2
3 public class Ex02 {
4     public static void main(String[] args) {
5         int a,b;
6         char op;
7         Scanner s = new Scanner(System.in);
8         System.out.print("첫번째 수 입력");
9         a = s.nextInt();
10        System.out.print("두번째 수 입력");
11        b = s.nextInt();
12        System.out.print("연산기호 입력");
13        op=s.next().charAt(0);
14
15        if(op=='+')
16            System.out.println(a+" "+b+"="+ (a+b));
17        else if(op=='-')
18            System.out.println(a+" "+b+"="+ (a-b));
19        else if(op=='*')
20            System.out.println(a+" "+b+"="+ (a*b));
21        else if(op=='/')
22            System.out.println(a+" "+b+"="+ (a/b));
23        else
24            System.out.println("잘못된 수식입니다");
25    }
26 }
```

예제2: 두개의 정수를 입력 받아서 입력 받은 연산기호의
연산 수행 후 출력해주세요 (switch문 사용)



첫번째 수 입력 10
두번째 수 입력 5
연산기호 입력 +
 $10+5=15$

첫번째 수 입력 10
두번째 수 입력 5
연산기호 입력 -
 $10-5=5$

```
switch (op)
{
case '+':
    System.out.println(a+" "+b+"="+ (a+b));
    break;
case '-':
    System.out.println(a+" "+b+"="+ (a-b));
    break;
case '*':
    System.out.println(a+" "+b+"="+ (a*b));
    break;
case '/':
    System.out.println(a+" "+b+"="+ (a/b));
    break;
case '%':
    System.out.println(a+" "+b+"="+ (a%b));
    break;
default:
    System.out.println("잘못된 연산 기호입니다");
    break;
}
```

실습 예제: 세개의 정수를 입력 받아 가장 큰 수를 출력 해주세요



```
Problems @ Javadoc Declaration Console
<terminated> Ex02 (10) [Java Application] C:\Program Files\Java\jdk-11.0.1\bin\javaw.exe (2020. 4. 3. 오전 1:33:29)
첫번째 수 입력10
두번째 수 입력20
세번째 수 입력5
가장큰수는 20입니다
```

예제1: 세개의 정수를 입력 받아 가장 큰 수를 출력 해주세요



```
Problems @ Javadoc Declaration Console
<terminated> Ex02 (10) [Java Application] C:\Program Files\Java\jdk-11.0.1\bin\javaw.exe (2020. 4. 3. 오전 1:33:29)
첫번째 수 입력10
두번째 수 입력20
세번째 수 입력5
가장큰수는 20입니다
```

SOLUTION

```
1 import java.util.Scanner;
2
3 public class Ex02 {
4     public static void main(String[] args) {
5         int a,b,c;
6         Scanner s = new Scanner(System.in);
7         System.out.print("첫번째 수 입력");
8         a = s.nextInt();
9         System.out.print("두번째 수 입력");
10        b = s.nextInt();
11        System.out.print("세번째 수 입력");
12        c = s.nextInt();
13        if(a>b){
14            if(a>c)
15                System.out.println("가장큰수는 "+a+"입니다");
16            else
17                System.out.println("가장큰수는 "+c+"입니다"); }
18        else{
19            if(b>c)
20                System.out.println("가장큰수는 "+b+"입니다");
21            else
22                System.out.println("가장큰수는 "+c+"입니다"); }
23    }
24 }
25
```

if else와 조건 연산자

55

- if else 대신 조건 연산자를 이용해서 같은 코드를 작성할 수 있다.

if else를 이용하는 경우

```
if (num < 0)
    abs = -num;
else
    abs = num;
```

=

조건 연산자를 이용하는 경우

if의 조건식

else

```
abs = num < 0 ? -num : num;
```

참일때
수식의 값

거짓일때
수식의 값

```

1 import java.util.Scanner;
2 public class Ex02 {
3     public static void main(String[] args) {
4         int a,b,c,max;
5         Scanner s = new Scanner(System.in);
6         System.out.println("첫번째 수 입력");
7         a = s.nextInt();
8         System.out.println("두번째 수 입력");
9         b = s.nextInt();
10        System.out.println("세번째 수 입력");
11        c = s.nextInt();
12        if(a>b)
13            max=(a>c)?a:c;
14        else
15            max =(b>c)?b:c;
16        System.out.println("가장 큰수는 : "+max+"입니다");
17    }
18 }

```