4장. 제어문(조건, 반복문)



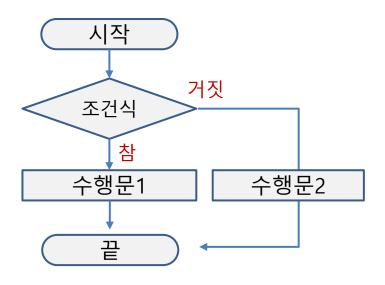
제어문



조건문(if문)

❖ 조건문이란?

주어진 조건에 따라 다른 수행문이 실행되도록 프로그래밍 하는 명령문 if문, switch문이 대표적이다.





조건문(if문)

■ if문

```
if(조건식){
수행문;
}
```

■ if-else 문

```
if(조건식){
수행문1;
}
else{
수행문2
}
```

```
int age = 7;

if (age < 8) {
    printf("학교에 가지 않습니다.\n");
}

printf("나이는 %d세입니다.", age);
```

```
int age = 7;

if (age < 8) {
    printf("학교에 가지 않습니다.\n");
}

else {
    printf("학교에 갑니다.\n");
}

printf("나이는 %d세입니다.\n", age);
```



if ~ 내부 if

■ if ~ 내부 if문

```
int num;
printf("수를 입력하세요: ");
scanf s("%d", &num);
if (num > 10) {
   if (num % 2 == 0) {
       printf("%d는(은) 10보다 큰 짝수입니다.\n", num);
   else {
       printf("%d는(은) 10보다 큰 홀수입니다.\n", num);
else {
   if (num % 2 == 0) {
       printf("%d는(은) 10이하의 짝수입니다.\n", num);
   else {
       printf("%d는(은) 10이하의 홀수입니다.\n", num);
```



if ~ else if ~ else문

■ if - else if - else 문 하나의 상황에 조건이 여러 개인 경우

```
If(조건 1){
   수행문1;
}
else if(조건 2)
   수행문2
}
else{
   수행문3
}
```

```
int age = 9;
int charge; //입장료(요금)
if (age < 8) {
   charge = 1000;
   printf("취학 전 아동입니다.\n");
else if (age < 14) {
   charge = 2000;
   printf("초등학생입니다.\n");
else if (age < 20) {
   charge = 2500;
   printf("중, 고등학생입니다.\n");
else {
   charge = 3000;
   printf("일반인입니다.\n");
printf("입장료는 %d원입니다.\n", charge);
```



<u>if문 - 연습 예제</u>

입장객 수에 따른 좌석 줄 수를 계산하는 프로그램을 작성하세요. (파일이름:seat.c)

```
입장객 수 입력:
23
좌석 열의 수 입력:
5
5개의 줄이 필요합니다.
```

```
int customerNum;
int columnNum;
int rowNum;
printf("입장객 수 입력 : ");
scanf s("%d", &customerNum);
printf("좌석 열의 수 입력 : ");
scanf s("%d", &columnNum);
if (customerNum % columnNum == 0) {
   rowNum = customerNum / columnNum;
else {
   rowNum = customerNum / columnNum + 1;
printf("%d개의 줄이 필요합니다.", rowNum);
```



조건문(SWITCH - CASE)

■ swtich-case문

조건식의 결과가 정수 또는 문자 값이고 그 값에 따라 수행문이

결정될때 if~else if ~ else문을 대신하여 switch-case문을 사용

```
#include <stdio.h>
pint main()
    int ranking = 1;
    char medalColor;
    switch (ranking) {
    case 1:
        medalColor = 'G';
        break;
    case 2:
        medalColor = 'S';
        break;
    case 3:
        medalColor = 'B';
        break;
    default :
        medalColor = 'A';
        break;
    printf("%d등 메달의 색깔은 %c", ranking, medalColor);
    return 0;
```



조건문(SWITCH - CASE)

■ case문 동시에 사용하기

```
int month = 10;
int day;
switch (month) {
case 1: case 3: case 5: case 7: case 8: case 10: case 12:
   day = 31;
    break;
case 2:
   day = 28;
    break;
case 4: case 6: case 9: case 11:
    day = 30;
    break;
default:
   day = 0;
    break;
printf("%d월은 %d일 까지 있습니다.\n", month, day);
```



실습 예제

operator 값이 +, -, *, / 인 경우에 사칙연산을 수행하는 프로그램을 if문과 switch문을 사용해 작성하기

System.out.println("결과 값은 " + result + "입니다.");

```
int num1 = 10;
int num2 = 2;
char operator = '+';
int result = 0:
switch(operator) {
case '+':
    result = num1 + num2;
    break;
case '-':
    result = num1 - num2;
    break;
case '*':
    result = num1 * num2;
    break;
case '/':
    result = num1 / num2;
    break:
default:
    System.out.println("연산자 오류입니다.");
    return;
```



실습 예제

```
int num1 = 10;
int num2 = 2;
char operator = '+';
int result = 0;
if(operator=='+') {
    result = num1 + num2;
}else if(operator=='-') {
    result = num1 - num2;
}else if(operator=='*') {
    result = num1 * num2;
}else if(operator=='/') {
    result = num1 / num2;
}else {
    System.out.println("연산자 오류입니다.");
    return;
System.out.println("결과 값은 " + result + "입니다.");
```



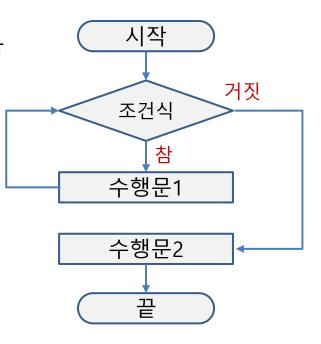
반복문(while문)

❖ 반복문

- 주어진 조건이 만족할 때까지 수행문을 반복적으로 수행함
- while, for 문이 있음

■ while 문

```
조건식이 참인 동안 반복 수행
while(조건식){
수행문1;
...
```





반복문(while문)

■ While문 예제

"Hello"를 5번 반복하기

```
printf("Hello~\n");
printf("Hello~\n");
printf("Hello~\n");
printf("Hello~\n");
printf("Hello~\n");
```

While문 사용

```
int i = 0;
while (i < 5)
{
    printf("Hello~\n");
    i++;
}
return 0;</pre>
```



반복문(while문)

• While문

```
int n = 0; //초기값 설정
n += 1;
n += 2;
n += 3;
n += 4;
n += 5;
n += 6;
n += 7;
n += 8;
n += 9;
n += 10;
printf("합계 = %d\n", n);
```

```
int i = 0, sum = 0;
while (i <= 10)
{
    sum += i; //sum = sum + i;
    printf("i=%d, sum=%d\n", i, sum);
    i++;
}
printf("합계 = %d\n", sum);
printf("------반복문 종료-----\n");
```



기타 제어 -break 문

■ break 문

반복문에서 break 문을 만나면 더 이상 반복을 수행하지 않고 반복문을 빠져 나옴

```
while(조건식){

if(조건식){

break;
}
}
```

while(**1**){ } – 무한 반복문 1이면 참, 1이 아니면 거짓

```
//무한 반복(반복 조건문)
int i = 1;
while (1)
{
    printf("hello\n");
    if (i == 5)
        break;
    i++;
}
```



break 예제

■ 1부터 더했을때 그 합이 100이 넘는 자연수는?

```
int n = 1;
int sum = 0;
while (1)
    sum += n;
    printf("n=%d, sum=%d\n", n, sum);
    if (sum >= 100)
        break;
    n++;
printf("n : %d\n", n);
printf("sum : %d\n", sum);
```



반복문(Dowhile문)

■ do ~ while문

while문이 반복문을 수행하기 전에 제어조건을 먼저 검사하지만, do-while문은 먼저 반복문을 실행한 후 검사한다.

```
char ch;

printf("아무 키나 눌러주세요. q를 누르면 종료합니다.\n");

do
{
    scanf s("%c", &ch);
} while (ch != 'q');
printf("프로그램을 종료합니다.");

e

Q
교로그램을 종료합니다.
```



은행업무 App

■ 은행 업무 입출금 프로그램 만들기

1.예금 | 2.출금 | 3.잔고 | 4.종료 선택> 1 예금액> 10000 1.예금 | 2.출금 | 3.잔고 | 4.종료 선택> 2 출금액> 5000 1.예금 | 2.출금 | 3.잔고 | 4.종료 선택> 3 잔고> 5000 1.예금 | 2.출금 | 3.잔고 | 4.종료 선택> 5 지원되지 않는 기능입니다. 1.예금 | 2.출금 | 3.잔고 | 4.종료 선택> 4 프로그램 종료

출금액이 잔액을 초과했을 경우"잔액을 초과했습니다. 다시 입력 해주세요" 출력



은행업무 App

```
∃#include <stdio.h>
#include <stdbool.h>
pint main() {
   bool sw = true; //실행, 중단
   //int sw = 1; //bool 사용하지 않을 경우
   int balance = 0; //잔고
   while (sw) {
       int selNo; //선택
       int money; //입.출금
       printf("=======\n");
       printf(" 1.예금 | 2.출금 | 3.잔고 | 4.종료\n");
       printf("=======\n");
       printf("선택> ");
       scanf s("%d", &selNo);
       if (selNo == 1) {
          printf("예금액> ");
          scanf s("%d", &money);
          balance += money;
```



은행업무 App

```
else if (selNo == 2) {
       while (sw) {
           printf("출금액> ");
           scanf s("%d", &money);
           if (money > balance) {
              printf("잔액을 초과하였습니다. 다시 입력해 주세요\n");
           else {
               balance -= money; break;
   else if (selNo == 3) {
       printf("잔고> %d\n", balance);
   else if (selNo == 4) {
       sw = false; //sw = 0
   else {
       printf("지원되지 않는 기능입니다.\n");
printf("프로그램 종료");
```



반복문(for문)

■ for 문

```
주로 조건이 횟수인 경우에 사용하는 반복문이다.
초기화식, 조건식, 증감식을 한꺼번에 작성
```

```
for(초기화식; 조건식; 증감식){
수행문;
}
```

■ for문 수행 과정

```
Int n;
for(n = 1; n <= 5; n++) {
    printf(n);
}</pre>
```



반복문(for문)

1부터 10까지 더하기

```
int n, sum;
for (n = 1; n <= 10; n++) {
    printf("%d\n",n);
}

for (n = 1, sum = 0; n <= 10; n++) {
    sum += n;
}
printf("1부터 10까지의 합 = %d\n", sum);
printf("%d\n", n);
```



기타 제어 - continue문

• continue 문

반복문과 함께 쓰이며, 반복문 내부 continue 문을 만나면 이후 반복되는 부분을 수행하지 않고 조건식이나 증감식을 수행함.

```
int n, sum;
for (n = 0, sum=0; n <= 10; n++)
{
    if (n % 2 == 1)
        continue;
    sum += n;
    printf("%d \n", n);
}
printf("1부터 10까지 짝수의 합 = %d\n", sum);
```



구구단

■ 구구단 프로그램 – 단을 입력받아 출력하기

```
int dan = 3;
int i;

for (i = 1; i < 10; i++) {
    printf("%d x %d = %d\n", dan, i, (dan * i));
}</pre>
```

```
//단을 입력 받아 구구단 출력
int i, dan;

printf("단을 입력하세요 : ");
scanf("%d", &dan);

for (i = 1; i < 10; i++) {
    printf("%d x %d = %d\n", dan, i, (dan*i));
}
```



반복문(중첩 for문)

■ 중첩된 반복문(Nested Loop)

	열1	열2	열3	열4	열5
행1					
행2					
행3					
행4					
행5					

```
가가가가가
가가가가가
가가가가가
가가가가가
가가가가
```

```
int i, j;

for (i = 0; i < 5; i++) {
    for (j = 0; j < 5; j++) {
        printf("7\");
    }
    printf("\n");
}</pre>
```



반복문(for문)

- 중첩된 반복문(Nested Loop)
 - 구구단의 예

```
int dan = 3;
int i, j;

for (i = 2; i < 10; i++) {
    for (j = 1; j < 10; j++) {
        printf("%d x %d = %d\n", i, j, (i * j));
    }
    printf("\n");
}</pre>
```



반복문(중첩 for문)

■ 삼각형 모양의 별 찍기1

hint) 열(column)이 변하는 것에 주목한다.



반복문[중첩 for문]

■ 삼각형 모양의 별찍기2

hint) 공백과 별로 나눠서 생각, 반복문과 조건문으로 구현하기

```
*

**

***

****
```



break문 예제

구구단을 단보다 곱하는 수가 작거나 같은 경우까지만 출력하는 프로그램 작성하기

```
2x1=2
2x2=4
3x1=3
3x2=6
3x3=9
4x1=4
4x2=8
4x3=12
4x4=16
```

```
int i, j;
for(i=2; i<10; i++) {
    for(j=1; j<10; j++) {
        if(i < j)
            break;
        System.out.println(i + "x" + j + "=" + (i*j));
      }
    System.out.println();
}</pre>
```



반복문(for문)

■ 중첩된 반복문(Nested Loop)

```
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
```

```
int i, j, num;
for (i = 0; i < 5; i++) {
    for (j = 1; j < 6; j++) {
        num = i * 5 + j;
        printf("%d ", (i * 5 + j));
    }
    printf("\n");
}</pre>
```



자리배치도 프로그램 만들기

입장객 수에 따라 좌석을 배치하는 프로그램을 작성하세요.

(파일이름: Seats.java)

입장객 수 입력: 44

좌석 열의 수: 5

좌석1 좌석2 좌석3 좌석4 좌석5

좌석6 좌석7 좌석8 좌석9 좌석10

좌석11 좌석12 좌석13 좌석14 좌석15

좌석16 좌석17 좌석18 좌석19 좌석20

좌석21 좌석22 좌석23 좌석24 좌석25

좌석26 좌석27 좌석28 좌석29 좌석30

좌석31 좌석32 좌석33 좌석34 좌석35

좌석36 좌석37 좌석38 좌석39 좌석40

좌석41 좌석42 좌석43 좌석44



자리배치도 프로그램 만들기

```
int customerNum;
int columnNum;
int rowNum;
                                        int i, j, seat;
                                        for (i = 0; i < rowNum; i++) {
printf("입장객 수 입력 : ");
                                            for (j = 1; j \leftarrow columnNum; j++) {
scanf s("%d", &customerNum);
                                                 seat = i * columnNum + j;
                                                 printf("좌석%d ", seat);
printf("좌석 열의 수 입력 : ");
                                                 if (seat == customerNum)
scanf s("%d", &columnNum);
                                                     break;
if (customerNum % columnNum == 0) {
                                            printf("\n");
   rowNum = customerNum / columnNum;
else {
   rowNum = customerNum / columnNum + 1;
```



반복문(for문)

■ ASCII 코드 출력하기

10진수 32부터 127까지 출력

```
10: 16: C
10: 16: C
            10: 16: C
                                     10: 16: C
                                                  10: 16: C
                                                              10: 16: C
 32:20:
             48:30: 0
                          64:40: @
                                      80:50: P
                                                   96:60:
                                                              112:70: p
33:21: !
             49:31: 1
                          65:41: A
                                      81:51: Q
                                                   97:61: a
                                                              113:71: q
 34:22:
             50:32: 2
                          66:42: B
                                      82:52: R
                                                   98:62: b
                                                              114:72: r
35:23: #
             51:33: 3
                          67:43: C
                                      83:53: S
                                                   99:63: c
                                                              115:73: s
36:24: $
             52:34: 4
                          68:44: D
                                      84:54: T
                                                  100:64: d
                                                              116:74: t
 37:25: %
             53:35: 5
                          69:45: E
                                      85:55: U
                                                  101:65: e
                                                              117:75: u
             54:36: 6
38:26: &
                          70:46: F
                                      86:56: V
                                                  102:66: f
                                                              118:76: v
 39:27:
             55:37: 7
                          71:47: G
                                      87:57: W
                                                  103:67: g
                                                              119:77: w
40:28:
             56:38: 8
                          72:48: H
                                      88:58: X
                                                  104:68: h
                                                              120:78: x
 41:29: )
             57:39: 9
                          73:49: I
                                      89:59: Y
                                                  105:69: i
                                                              121:79: y
42:2a: *
             58:3a: :
                          74:4a: J
                                      90:5a: Z
                                                  106:6a: i
                                                              122:7a: z
43:2b: +
             59:3b: :
                          75:4b: K
                                      91:5b: [
                                                  107:6b: k
                                                              123:7b:
             60:3c: <
                                      92:5c:
44:2c:,
                          76:4c: L
                                                  108:6c: 1
                                                              124:7c:
 45:2d: -
             61:3d: =
                         77:4d: M
                                      93:5d:
                                                  109:6d: m
                                                              125:7d:
46:2e: .
             62:3e: >
                          78:4e: N
                                      94:5e:
                                                  110:6e: n
                                                              126:7e:
47:2f: /
             63:3f: ?
                          79:4f: 0
                                      95:5f:
                                                  111:6f: o
                                                              127:7f:
```



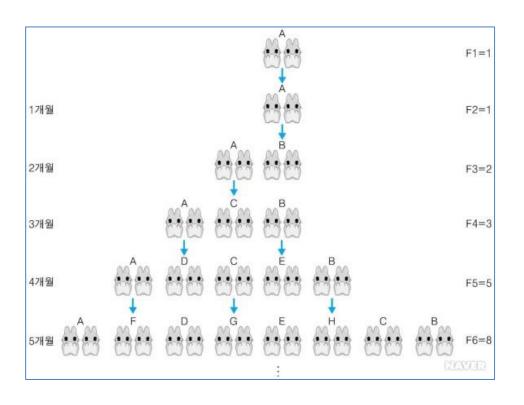
반복문(for문)

```
for (y = 0; y < 6; y++)
   printf("10: 16: C | "); // 제목줄 표시
printf("\n");
for (y = 0; y < 6; y++)
   printf("-----;
printf("\n");
for (x = 0; x < 16; x++)
   for (y = 2; y < 8; y++)
      c = y * 16 + x; // 32부터 시작
       printf("%3d:%2x:%2c | ", c, c, c);
   printf("\n");
```



피보나치 수열

피보나치 수열은 이탈리아 수학자 피보나치가 발견한 수열이며, 토끼의 번식을 보면서 출발했다고 한다.



```
int i = 1; //반복 변수
int n = 12; //12개월 후
int value1, value2, value_new;
printf("피보나치 수열 - 토끼의 번식\n");
value1 = value2 = 1; //1항, 2항
printf("%4d", value1);
printf("%4d", value2);
i = 3; //3항
while (i <= n) {
   value_new = value1 + value2; //1항+2항=3항
   value1 = value2;
   value2 = value new;
   printf("%4d", value new);
   i++;
```

```
■ Microsoft Visual Studio 디버그 콘솔
피보나치 수열 - 토끼의 번식
1 1 2 3 5 8 13 21 34 55 89 144
```



조건문(if문)

■ UpAndDown 게임(숫자 맞히기 게임)

- 먼저 출제자가 두 자리의 숫자를 입력하면 경기자가 이것을 추측하여 맞추는 것이다.(입력한 숫자는 경기자가 볼 수 없도록 ****로 출력)
- 게이머가 숫자를 예측하면 컴퓨터는 답과 비교하여 '더 큰 숫자입니다',
 '더 작은 숫자입니다:', '정답입니다 '를 출력한다.
- 점수는 회수x10으로 계산한다.

```
두 자리 수 입력(1~99): **

[ 1회] 1~99 사이의 값 예측 ->
50
더 큰 숫자입니다.
[ 2회] 50~99 사이의 값 예측 ->80
더 작은 숫자입니다.
[ 3회] 50~80 사이의 값 예측 ->65
더 큰 숫자입니다.
[ 4회] 65~80 사이의 값 예측 ->77
성공 !!! 정답은 77
최종 점수 = 70
```

```
printf("두 자리 수 입력(1~99): ");
char a = _getch();
printf("*");
char b = _getch();
printf("*\n\n");
n = (a - '0') * 10 + (b - '0'); //아스키 코드 '0'은 48

for (i = 0; i < 10; i++)
{
    printf("[%2d회] %2d ~%2d 사이의 값 예측 ->", i + 1, min, max);
    scanf_s("%d", &input);

    if (n == input) break;
    else if (n > input) {
        printf("더 큰 숫자입니다.\n");
        min = input;
    }
```

