

# 3장. 제어문(조건, 반복문)



제어문



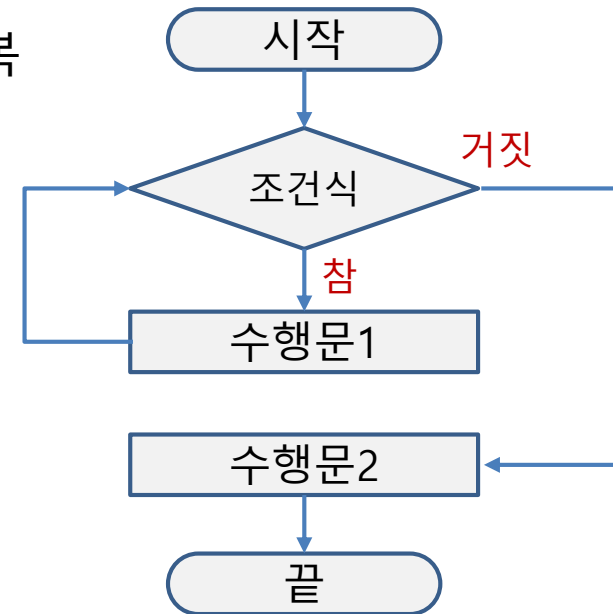
# 반복문(while문)

## ❖ 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

```
//1부터 10까지 출력  
int n = 0;  
  
while (n < 10) {  
    n += 1;  
    cout << n << " ";  
}
```



# 반복문(while문)

## ▪ While문

```
//1부터 20까지 3의 배수와 개수 출력
int n = 0, count = 0;

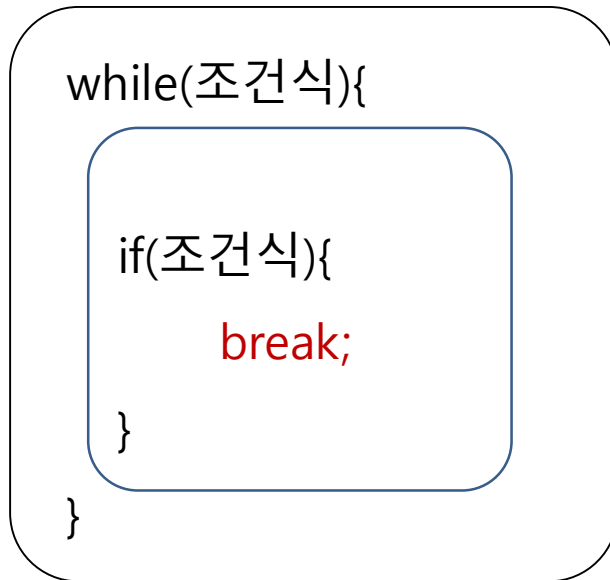
while (n < 20) {
    n += 1;
    if (n % 3 == 0) {
        count++;
        cout << n << " ";
    }
}
cout << endl;
cout << "3의 배수의 개수 : " << count << "개" << endl;
```



# 기타 제어 - break 문

## ▪ break 문

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



```
//break문  
n = 0;  
while (true) {  
    n++;  
    if (n > 10) break;  
    cout << n << endl;  
}
```



# break 예제

- 1부터 더했을때 그 합이 100이 넘는 자연수는?

```
int n = 0, sum = 0;

while (true) {
    n += 1;
    sum += n;
    if (sum > 100) break;
}

cout << "sum = " << sum << endl;
cout << "n = " << n << endl;
```



# break 예제

```
while (true) {  
    char ch;  
    cout << "계속 반복할까요?(y/n) ";  
    cin >> ch;  
  
    if (ch == 'y') {  
        cout << "계속 반복" << endl;  
    }  
    else if (ch == 'n') {  
        cout << "반복 중단!" << endl;  
        break;  
    }  
    else {  
        cout << "키를 잘못 입력했습니다." << endl;  
    }  
}  
cout << "반복 종료!" << endl;
```



# 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;  
    ① → ② ← ④  
for(n = 1; n <= 5; n++) {  
    printf(n);  
    ③ ↗  
}
```



# 반복문(for문)

```
//1부터 5까지 곱하기
int facto = 1;

for (int i = 1; i <= 5; i++) {
    facto *= i;
}

cout << "5! = " << facto << endl;
```



# 기타 제어 – continue문

## ▪ continue 문

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

```
//1부터 10까지의 홀수의 합 계산
int sum = 0;

for (int n = 1; n <= 10; n++) {
    if (n % 2 == 0)
        continue;
    sum += n;
    cout << n << " ";
}
cout << endl;
cout << "합계 : " << sum << endl;
```



# 반복문(중첩 for문)

## ■ 중첩된 반복문(Nested Loop)

	열1	열2	열3	열4	열5
행1					
행2					
행3					
행4					
행5					

```
int i, j;

for (i = 1; i <= 5; i++) {
    for (j = 1; j <= 5; j++) {
        cout << "$";
    }
    cout << endl;
}
```



# 반복문(중첩 for문)

## ■ 삼각형 모양의 별 찍기1

hint) 열(column)이 변하는 것에 주목한다.

```
*  
**  
***  
****  
*****
```

```
for (i = 1; i <= 5; i++) {  
    for (j = 1; j <= i; j++) {  
        cout << "$";  
    }  
    cout << endl;  
}
```

```
for (i = 1; i <= 5; i++) {  
    for (j = 6 - i; j > 0 ; j--) {  
        cout << "$";  
    }  
    cout << endl;  
}
```



# 반복문(for문)

- 중첩된 반복문(Nested Loop)

- 구구단

```
int dan, j;

for (dan = 2; dan < 10; dan++) {
    for (j = 1; j < 10; j++) {
        cout << dan << " x " << j << " = " << dan * j << endl;
    }
    cout << endl;
}
```



# 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

```
for (dan = 2; dan < 10; dan++) {  
    for (j = 1; j <= dan; j++) {  
        cout << dan << " x " << j << " = " << dan * j << endl;  
    }  
    cout << endl;  
}
```



# 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;  
  
for (i = 0; i < 5; i++) {  
    for (j = 1; j <= 5; j++) {  
        int num = i * 5 + j;  
        cout << num << " ";  
    }  
    cout << endl;  
}
```



# 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

seats.h

```
#include <iostream>
using namespace std;

void getSeats(int custom, int col);
```

main.cpp

```
#include "seats.h"

int main() {
    int customNum, colNum, rowNum;

    cout << "입장객 수 입력 : ";
    cin >> customNum;

    cout << "좌석 열 수 입력 : ";
    cin >> colNum;

    getSeats(customNum, colNum);

    system("pause");
    return 0;
}
```



# 자리배치도 프로그램 만들기

get\_seats.cpp

```
#include "seats.h"

void getSeats(int custom, int col) {
    int row; //줄(행) 수

    if (custom % col == 0) { //입장객수가 열수로 나누어 떨어지면
        row = custom / col;
    }
    else {
        row = custom / col + 1;
    }
    //cout << "필요한 줄 수는 " << row << "줄 입니다." << endl;

    for (int i = 0; i < row; i++) {
        for (int j = 1; j <= col; j++) {
            int seatNum = (i * col) + j; //좌석 번호
            cout << "좌석" << seatNum << " ";
            if (seatNum == custom)
                break;
        }
        cout << endl;
    }
}
```

