

Control Structures



2023. Spring

국민대학교 소프트웨어학부

최 준수

break 문

- break 문

- 사용문법


```
break;
```

- 사용되는 경우


- 반복문

- for, while, do-while 반복문의 몸체에서 반복을 즉각 종료시키기 위해서 사용하는 문장
 - break문이 실행되면 반복이 종료되고, 반복문 다음에 있는 문장이 실행됨

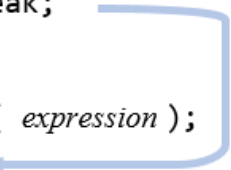
```
for ( for-init ; expression ; for-update ) {  
    ...  
    break;  
    ...  
}  
...
```



```
while ( expression ) {  
    ...  
    break;  
    ...  
}  
...
```



```
do {  
    ...  
    break;  
    ...  
}  
while ( expression );  
...
```



- switch 문

- switch문을 종료

break 문

– 예제

- 입력되는 정수의 합을 구하고 출력하는 프로그램
- 정수의 입력은 입력되는 정수가 0 일 때까지 계속한다.

```
int num, sum = 0;

while (true) {    // 무한 반복문
    cin >> num;
    if (num == 0)
        break;
    sum += num;
}
cout << "입력된 모든 정수의 합은 " << sum << " 입니다." << endl;
```

2 7 9 5 4 0

입력된 모든 정수의 합은 27 입니다.

break 문

– 예제

- 입력되는 정수의 합을 구하고 출력하는 프로그램
- 정수의 입력은 입력되는 정수가 0 일 때까지 계속한다.
 - 입력의 끝을 나타내는 0과 같은 수를 sentinel이라고 부름

```
int num, sum = 0;

while (true) {    // 무한 반복문
    cin >> num;
    if (num == 0)
        break;
    sum += num;
}
cout << "입력된 모든 정수의 합은 " << sum << " 입니다." << endl;
```

2 7 9 5 4 0

입력된 모든 정수의 합은 27 입니다.

break 문

– 예제

- 주어진 정수 n 이 소수인지를 판별하는 프로그램
 - 소수(prime number)
 - » 1과 자기 자신을 제외한 다른 약수가 없는 수

```
int n = 1442968193;
bool isPrime = true;

// n 을 2부터  $\lfloor \sqrt{n} \rfloor$  까지의 모든 정수로 나누어 봄
for (int i = 2; i * i <= n; i++) {
    if (n % i == 0) {
        isPrime = false;
        break;
    }
}

if (isPrime)
    cout << "자연수 " << n << "은 소수입니다." << endl;
else
    cout << "자연수 " << n << "은 소수가 아닙니다." << endl;
```

자연수 1442968193은 소수입니다.

continue 문

- continue 문

- 사용문법

```
continue;
```

- 사용되는 경우

- 반복문

- for, while, do-while 반복문의 몸체에서 사용
 - continue 문 다음에 오는 모든 문장을 건너뛰고 다음 반복 단계를 시작함

```
for ( for-init ; expression ; for-update ) {  
    ...  
    continue;  
    ...  
}  
...
```

```
while ( expression ) {  
    ...  
    continue;  
    ...  
}  
...
```

```
do {  
    ...  
    continue;  
    ...  
}  
while ( expression );  
...
```

continue 문

– 예제

- 입력되는 정수 중에서 모든 양수의 합을 구하고 출력하는 프로그램
- 정수의 입력은 입력되는 정수가 0 일 때까지 계속한다.
 - 입력의 끝을 나타내는 0과 같은 수를 sentinel이라고 부름

```
int num, sum = 0;

while (true) {    // 무한 반복문
    cin >> num;
    if (num == 0)
        break;
    else if (num < 0)
        continue;
    sum += num;
}
cout << "입력된 모든 양수의 합은 " << sum << " 입니다." << endl;
```

2 -5 4 6 -3 9 0

입력된 모든 양수의 합은 21 입니다.

중첩 반복문

- 중첩 반복문(nested loop)
 - 반복문(for, while, do-while)의 몸체에 반복문이 사용된 경우
 - 중첩의 깊이에는 한도가 없음 (2중, 3중, ... 반복문)

```

외부
반복문
[
    while ( value < 100 ) {
        // 외부 반복문 몸체
        ...
        ...
        내부
        반복문
        [
            for ( i = 0; i < 100; i++ ) {
                // 내부 반복문 몸체
                ...
                ...
            }
        ]
    }
]
```


중첩 반복문

– 예제: 구구단 출력

```
#include <iostream>
#include <iomanip> // setw(), left(alignment)
using namespace std;

int main()
{
    const int N = 9;
    for (int i = 1; i <= N; i++) {
        for (int j = 1; j <= N; j++) {
            cout << j << "x" << i << "=" << setw(2) << left << i * j << " ";
        }
        cout << endl;
    }

    return 0;
}
```

```
1x1=1 2x1=2 3x1=3 4x1=4 5x1=5 6x1=6 7x1=7 8x1=8 9x1=9
1x2=2 2x2=4 3x2=6 4x2=8 5x2=10 6x2=12 7x2=14 8x2=16 9x2=18
1x3=3 2x3=6 3x3=9 4x3=12 5x3=15 6x3=18 7x3=21 8x3=24 9x3=27
1x4=4 2x4=8 3x4=12 4x4=16 5x4=20 6x4=24 7x4=28 8x4=32 9x4=36
1x5=5 2x5=10 3x5=15 4x5=20 5x5=25 6x5=30 7x5=35 8x5=40 9x5=45
1x6=6 2x6=12 3x6=18 4x6=24 5x6=30 6x6=36 7x6=42 8x6=48 9x6=54
1x7=7 2x7=14 3x7=21 4x7=28 5x7=35 6x7=42 7x7=49 8x7=56 9x7=63
1x8=8 2x8=16 3x8=24 4x8=32 5x8=40 6x8=48 7x8=56 8x8=64 9x8=72
1x9=9 2x9=18 3x9=27 4x9=36 5x9=45 6x9=54 7x9=63 8x9=72 9x9=81
```

중첩 반복문

- 3중첩 반복문

- 예제:

- 세자리 정수중에서 자신과 각 자리수를 앞뒤로 뒤집은 수의 합이 888인 수를 모두 찾기
 - 예: $147 + 741 = 888$

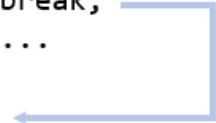
```
for (int a = 1; a <= 9; a++)  
    for (int b = 0; b <= 9; b++)  
        for (int c = 1; c <= 9; c++)  
        {  
            int abc = a * 100 + b * 10 + c;  
            int cba = c * 100 + b * 10 + a;  
  
            if (abc + cba == 888)  
                cout << abc << " ";  
        }
```

147 246 345 444 543 642 741

중첩 반복문 내에서의 break

- 중첩 반복문내에서의 break
 - 모든 중첩 반복문을 종료하지 않음
 - break문을 포함하는 가장 안쪽 반복문 만을 종료함
 - 이 반복문 다음에 오는 문장을 실행함

```
while( value < 100 ) {  
    // 외부 반복문 몸체  
    ...  
    ...  
    for ( i = 0; i < 100; i++ ) {  
        // 내부 반복문 몸체  
        ...  
        break;  
        ...  
    }  
    ...  
}
```



중첩 반복문 내에서의 continue

- 중첩 반복문내에서의 continue
 - 모든 중첩 반복문을 종료하지 않음
 - continue문을 포함하는 가장 안쪽 반복문의 현재 반복 단계의 실행을 종료함
 - 이 반복문 다음 반복을 실행함

```
while( value < 100 ) {  
    // 외부 반복문 몸체  
    ...  
    ...  
    for ( i = 0; i < 100; i++ ) {  
        // 내부 반복문 몸체  
        ...  
        continue;  
        ...  
    }  
    ...  
}
```

