

프로그래밍	7. do-while 반복문과 for 반복문, 그리고 중첩 반복문의 사용	일자: 2024. . .
학습 목표	<input type="checkbox"/> do-while 반복문 <input type="checkbox"/> for 반복문 <input type="checkbox"/> 중첩 반복문의 사용	하이다:

1. do-while 반복문

1) do-while

- ① do-while문은 무조건 한 번의 루프를 실행한 후, 조건을 체크하여 루프하는 반복문이다.
- ② do-while과 while의 기능적 차이는 무조건 한 번은 실행하느냐와 아니냐의 차이이다.

2) 조건식

- ① do-while의 조건식은 첫 루프에는 체크되지 않는다.
- ② 첫 번째 루프 실행 후로는 항상 체크되며 조건식이 참인 경우에는 상단 실행문 실행(루프), 거짓인 경우 실행 없이 while문을 탈출한다.
- ③ 조건식 안에는 while문과 동일하게 다양한 형태의 조건을 집어넣을 수 있다.

3) break; / continue;

- ① break 키워드는 _____ 하는 역할로 기능한다.
- ② continue 키워드는 _____ 역할로 기능한다.

4) 코드 예시

① 계수기 제어 반복문

```
카운터 초기화;
do {
    명령문;
    카운터++;
} while (카운터에 관한 조건식);
```

② 감시값 제어 반복문

```
do {
    if (감시값 확인 조건문)
        break;
    명령문;
    감시값 변경문;
} while (조건식);
```

2. while과 do-while에서 감시값 제어 반복문의 형태

1) 0번 이상 반복

```
감시값 초기화;
while (감시값 != 목표값) {
    명령문;
    감시값 변경문;
}
```

```
while (1) { // 무한반복
    감시값 변경문;
    if (감시값 == 목표값) break;
    명령문;
}
```

- “_____을 주의하라” : 전자와 같이 반복문을 사용하거나, while문의 조건으로 사용될 변수를 하나 선언하는 것이 바람직하다.

2) 1번 이상 반복

```
do {
    명령문;
    감시값 변경문;
} while (감시값 != 목표값)
```

```
감시값 초기화; // 목표값이 아닌 값으로
while (감시값 != 목표값) {
    명령문;
    감시값 변경문;
}
```

3. for 반복문

1) for

- ① for문은 조건식이 참일 때 루프하는 반복문이다.
- ② for문은 초기화식, 조건식, 증감식을 가지는데, 조건식을 제한 나머지 둘은 작성하지 않아도 된다. 그러나 반드시 "**for(조건식;)**"의 형태를 지켜야 한다.

2) 초기화식

- ① 초기화식은 일반적으로 for문에서 반복 횟수를 제어하는 변수를 초기화하는 용도이다.
- ② for문이 처음 실행될 때 오직 한 번 실행되며 for문에서 탈출할 때까지 실행되지 않는다.

3) 조건식

- ① 조건식은 첫 루프부터 매반복 시마다 체크되어 조건식이 참인 경우에는 하단 실행문 실행, 거짓인 경우 for문을 탈출하도록 한다.
- ② 조건식 안에는 while문과 동일하게 다양한 형태의 조건을 집어넣을 수 있다.

4) 증감식

- ① 일반적으로 조건식에서 탐색하는 변수의 값을 변하게 하는 용도로 사용된다.
- ② 한 루프의 실행이 끝난 후에 실행된다.

5) break; / continue;

- 앞서 설명한 것과 동일한 기능을 수행한다.

6) 코드 예시

```
for (i = 0; i < n; i++) {
    실행문;
}
```

- for문의 반복을 제어하는 변수의 이름은 간단히는 i, j, k, l 등을 사용하여 표기하기도 한다. 물론 각 인덱스가 의미하는 바가 명확한 경우에는 의미 있는 이름으로 사용하는 것이 바람직하다.
- for문에서는 1부터 n까지의 반복 말고도 **0부터 n - 1까지의 반복** 형태가 많이 사용된다. 이는 추후에 배열 배열의 사용과 관련이 있다.
- for문은 일반적으로 _____ 제어 반복문으로 사용되나 _____ 제어 반복문으로도 사용할 수 있다.

4. 중첩 반복문

1) 중첩 반복문

- ① 중첩 반복문은 반복문 내부에 또다른 반복문이 존재하는 경우를 뜻한다.
- ② 반복문의 중첩의 횟수에는 제한이 없고, for와 while을 섞어 중첩할 수도 있다.

2) 주의사항

- ① 중첩되면 중첩될수록 반복의 횟수는 곱해지므로 시간 또는 메모리 차원에서 컴퓨터에 무리가 갈 수 있다.
 - 예를 들어 1을 n번 출력하는 반복문의 시간 소모를 $O(N)$ 이라고 하자. 그렇다면 해당 반복문을 감싸고 n번 반복하는 반복문이 있고, 그 외부에 또 n번만큼 반복하는 반복문이 있다면, 해당 코드의 시간복잡도는 $O(N^3)$ 이다.
- ② 중첩된 반복문 내·외부에서 동일한 변수를 함께 사용한다면, 내부에서 수정된 값이 외부에, 혹은 외부에서 수정된 값이 내부에 의도하지 않은 영향을 미칠 수 있으니 주의하여야 한다.

5. 개념 확인 문제

1) while문과 do-while문, for문의 순서도를 그려보아라.

① while문

```
while (C) {  
    명령문 S;  
}
```

② do-while문

```
do {  
    명령문 S;  
} while (조건식 C);
```

③ for문의 순서도

```
for (i = 0; i < n; i++) {  
    명령문 S;  
}
```

6. 실습

1) LAB7_1 (do-while문)

나이를 입력받아 미성년이라면 성인(만19세)이 되기까지 남은 시간을, 성년이라면 성인임을 출력하는 프로그램을 작성하라. 사용자의 선택에 따라 프로그램이 종료되는지 이어지는지가 결정된다.

```
Enter your age: 15
You will be an adult in 4 years.
Do you want to continue(1 / 0)? 1

Enter your age: 20
You are an adult.
Do you want to continue(1 / 0)? 0
```

2) LAB7_2 (for문)

다음의 실행결과를 갖도록 프로그램을 작성하라. 입력되는 수는 2 이상이라고 가정한다.

```
Enter a number(>=2): 13
13! = 13 * 12 * 11 * 10 * 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 6227020800
```

3) LAB7_3 (for문)

n개의 정수를 읽어서 짝수의 개수를 세는 프로그램을 작성하라.

```
Enter the # of integers: 5
Enter an integer: 8
Enter an integer: 12
Enter an integer: 9
Enter an integer: 59
Enter an integer: 4
The number of even numbers is 3.
```

```
Enter the # of integers: 0
The number of even numbers is 0.
```

4) LAB7_4 (for문)

정수를 입력받아 소수인지 아닌지를 판별하는 프로그램을 작성하라.

힌트 및 요구사항은 아래와 같다.

- 소수는 1과 자기 자신, 오직 2개의 약수만을 갖는 수를 말한다.
- 반복은 최소화한다.

```
Enter a number: 2
2 is a Prime number
```

```
Enter a number: 10
10 is NOT a Prime number
```

5) LAB7_5 (감시값 제어 for문)

여러 개의 성적을 읽어서 그 총점을 계산하는 프로그램을 작성하라. -1은 입력의 끝을 나타낸다.

```
Enter a score(-1 for exit): 30
Enter a score(-1 for exit): 50
Enter a score(-1 for exit): 30
Enter a score(-1 for exit): -1
You've entered 3 students.
The total is 110.
The average is 36.7.
The best score is 50.
```

```
Enter a score(-1 for exit): -1
You've entered 0 students.
There is no data.
```

6) LAB7_6 (중첩 while문)

중첩된 while문을 사용하여 행과 열의 크기를 받아 다음과 같이 출력하도록 하라.

```
Enter the number of rows: 3
Enter the number of columns: 5
*****
*****
*****
```

```
Enter the number of rows: 2
Enter the number of columns: 7
*****
*****
```

7) LAB7_7 (중첩 for문)

중첩된 for문을 이용해 입력받은 수를 높이로 갖는 '*'로 이루어진 피라미드를 출력하라.

힌트 및 요구사항은 아래와 같다.

- 규칙성을 찾아라. 각 층에서 ' '는 몇 개, '*'은 몇 개인가?
- 각 층을 담당하는 반복문과, ' '를 출력하는 반복문, '*'을 출력하는 반복문, 줄바꿈 출력문의 배치를 고민하여라. 어떻게 중첩시켜야 하는가?

```
Enter a number:5
  *
 ***
*****
*****
*****
```

```
Enter a number:4
  *
 ***
*****
*****
```

```
Enter a number:1
*
```

8) LAB7_8 (중첩 for문)

for문을 세 번만 사용하여 구구단을 출력하는 프로그램을 만들어라.

2 * 1 = 2	3 * 1 = 3	4 * 1 = 4	5 * 1 = 5
2 * 2 = 4	3 * 2 = 6	4 * 2 = 8	5 * 2 = 10
2 * 3 = 6	3 * 3 = 9	4 * 3 = 12	5 * 3 = 15
2 * 4 = 8	3 * 4 = 12	4 * 4 = 16	5 * 4 = 20
2 * 5 = 10	3 * 5 = 15	4 * 5 = 20	5 * 5 = 25
2 * 6 = 12	3 * 6 = 18	4 * 6 = 24	5 * 6 = 30
2 * 7 = 14	3 * 7 = 21	4 * 7 = 28	5 * 7 = 35
2 * 8 = 16	3 * 8 = 24	4 * 8 = 32	5 * 8 = 40
2 * 9 = 18	3 * 9 = 27	4 * 9 = 36	5 * 9 = 45
6 * 1 = 6	7 * 1 = 7	8 * 1 = 8	9 * 1 = 9
6 * 2 = 12	7 * 2 = 14	8 * 2 = 16	9 * 2 = 18
6 * 3 = 18	7 * 3 = 21	8 * 3 = 24	9 * 3 = 27
6 * 4 = 24	7 * 4 = 28	8 * 4 = 32	9 * 4 = 36
6 * 5 = 30	7 * 5 = 35	8 * 5 = 40	9 * 5 = 45
6 * 6 = 36	7 * 6 = 42	8 * 6 = 48	9 * 6 = 54
6 * 7 = 42	7 * 7 = 49	8 * 7 = 56	9 * 7 = 63
6 * 8 = 48	7 * 8 = 56	8 * 8 = 64	9 * 8 = 72
6 * 9 = 54	7 * 9 = 63	8 * 9 = 72	9 * 9 = 81