# 05. 프로그래밍 기초 with C#

# 6. 실행 흐름 제어하기(반복문)

---

## 학습목표

- 반복문을 사용할 수 있다.

## 들어가며

기계가 잘하는 것은 역시 어떤 일을 반복하는 것이다. 이번엔 특정한 구문을 어떤 조건을 달성할 때까지 반복할 수 있는 **반복문**(Iteration Statement)에 대해서 알아보자. 반복문의 경우도 선택문과 마찬가지로 여러 가지 구문을 제공하는데 때에 따라 가독성의 차이만 있을 뿐 반복한다는 본질은 바뀌지 않는다는 걸 명심하자.

## while

while문은 반복 횟수가 정해져 있지 않을 때 유용하다.

```
bool isGameOver = false;



// ()의 식이 참일 동안 아래 구문이 실행된다.

while (false == isGameOver) // 게임이 종료되지 않았다면
```

```
{
  PlayGame(); // 게임을 실행한다.
}
```

## for

for문은 반복 횟수가 정해져 있을 때 유용하다.

```
int[] arr = { 1, 2, 3, 4, 5 };
int arrSize = arr.Length;

// for문의 구조는 아래와 같다.
// for (initializer; condition; iterator) body

for (int i = 0; i < arrSize; ++i)
{
  System.Console.WriteLine(arr[i]);
}
```

## do

do문은 위의 구문과는 다르게 조건에 대한 평가를 나중에 한다. 조건식을 나중에 평가해야 하는 경우 유용하다.

```
// do 문의 구조는 아래와 같다.
int selectedMenu = -1;
do
{
```

```
  selectedMenu = SelectMenu(); // 메뉴 선택

} while (IsValidMenu(selectedMenu)); // 유효한 메뉴를 선택했는지 확인합니다.

// 조건을 나중에 평가합니다.
```

# 반복을 제어하기

## break

break문은 반복문을 중단시킨다.

```
int[] arr = { 1, 2, 3, 4, 5 };

int arrSize = arr.Length;


// 아래의 for문은 1, 2만 출력하고 끝난다.

for (int i = 0; i < arrSize; ++i)

{

  if (i == 2)

  {

    break;

  }


  System.Console.WriteLine(arr[i]);

}
```

## continue

continue문은 아래 구문의 실행을 생략하고 다음 루프로 진행시킨다.

```
// 아래의 for문은 짝수만 출력한다.
for (int i = 1; i <= 10; ++i)
{
  if (i % 2 != 0)
  {
    continue;
  }

  System.Console.WriteLine(i);
}
```

## 다중 반복문에서 탈출하기

반복문 안에 반복문을 사용하는 일도 흔하다. 문제는 다중 반복문에서 탈출하는 경우다.

```
for (int i = 0; i < 10; ++i)
{
  for (int j = 0; j < 10; ++j)
  {
    if (i == 5 && j == 4)
    {
      // 반복문 전체를 멈추고 싶지만
      // break문과 continue문은 가장 가까운 반복문에만 적용된다.
      break;
    }
  }
}
System.Console.WriteLine("루프 끝");
```

```

```

이 경우 변수를 사용한다.

```
for (int i = 0; i < 10; ++i)
{
  bool canExit = false;

  for (int j = 0; j < 10; ++j)
  {
    if (i == 5 && j == 4)
    {
      canExit = true;

      break;
    }
  }

  if (canExit)
  {
    break; // 완전 탈출
  }
}
System.Console.WriteLine("루프 끝");
```

## 더 나아가기

1. 아래의 문제를 풀어봅시다..

1.1. 2739번: 구구단

1.2. 10950번: A+B - 3

1.3. 8393번: 합

1.4. 25304번: 영수증

1.5. 15552번: 빠른 A+B

1.6. 11021번: A+B - 7

1.7. 11022번: A+B - 8

1.8. 2438번: 별 찍기 - 1

1.9. 2439번: 별 찍기 - 2

1.10. 10952번: A+B - 5

1.11. 10951번: A+B - 4

1.12. 1110번: 더하기 사이클

2. 프로그래머스에서 아래의 문제를 풀어봅시다.

2.1. 코딩테스트 연습 - 짝수 홀수 개수 | 프로그래머스 스쿨

2.2. 코딩테스트 연습 - 최댓값 만들기(1) | 프로그래머스 스쿨

2.3. 코딩테스트 연습 - 배열 원소의 길이 | 프로그래머스 스쿨

2.4. 코딩테스트 연습 - 배열 자르기 | 프로그래머스 스쿨

2.5. 코딩테스트 연습 - 문자 반복 출력하기 | 프로그래머스 스쿨

2.6. 코딩테스트 연습 - 짝수는 싫어요 | 프로그래머스 스쿨

2.7. 코딩테스트 연습 - 순서쌍의 개수 | 프로그래머스 스쿨

2.8. 코딩테스트 연습 - 배열의 유사도 | 프로그래머스 스쿨

2.9. 코딩테스트 연습 - 자릿수 더하기 | 프로그래머스 스쿨

2.10. 코딩테스트 연습 - 숨어있는 숫자의 덧셈 (1) | 프로그래머스 스쿨

2.11. 코딩테스트 연습 - 대문자와 소문자 | 프로그래머스 스쿨

2.12. 코딩테스트 연습 - 암호 해독 | 프로그래머스 스쿨

2.13. 코딩테스트 연습 - 가위 바위 보 | 프로그래머스 스쿨

2.14. 코딩테스트 연습 - 가장 큰 수 찾기 | 프로그래머스 스쿨

2.15. 코딩테스트 연습 - 피자 나눠 먹기 (2) | 프로그래머스 스쿨

2.16. 코딩테스트 연습 - 369게임 | 프로그래머스 스쿨

2.17. 코딩테스트 연습 - A로 B 만들기 | 프로그래머스 스쿨

2.18. 코딩테스트 연습 - 2차원으로 만들기 | 프로그래머스 스쿨

2.19. 코딩테스트 연습 - 가까운 수 | 프로그래머스 스쿨

## 참고자료

- C# 교과서