



05 - 05. 실행 흐름 제어하기

선택문



목차

- 선택문과 진리값
- 불리언 타입
- 단축 평가
- 선택문의 종류
 - if문
 - 삼항 연산자
 - switch문

선택문과 진리값

선택문과 진리값

- 선택문(Selection Statement): 특정 조건을 만족할 때(참일 때)만 코드를 실행
- 진리값(Truth Value)
 - 명제의 내용이 참인지 거짓인지를 나타내는 값
 - 값이 0이면 거짓, 0이 아니라면 모두 참
 - 정수
 - 0 → 거짓
 - 1 → 참
 - 실수
 - 0.0 → 거짓
 - 1.2 → 참

폴리언 타입

불리언 타입

- 불리언(Boolean)

- 참, 거짓의 데이터를 저장할 수 있는 타입으로 `bool`을 사용한다.
- 불리언 타입의 리터럴은 `true`, `false`다.
- 불리언 타입에는 논리 연산자(Logical Operator)를 사용할 수 있다.
- 비교 연산자(Comparison Operator)의 평가 타입이다.

```
bool isTrue = true;  
System.Console.WriteLine(isTrue);
```

불리언 타입

- 이름은 is, has, can, should 등을 사용해 가부(可否)를 표현할 수 있는 형태로 짓는다.

```
// is + 형용사
bool isActive;
bool isValid;

// has + 명사
bool hasPermission;
bool hasAccess;

// can + 동사
bool canAttack;
bool canExecute;

// should + 동사
bool shouldApplySpecialRate;
```

불리언 타입

- 복잡한 조건을 변수로 분리하면 읽기가 쉽다.[1]

```
// ❌ 조건이 한 줄에 다 들어감
if ((day = "SAT" || day = "SUN") && hour ≥ 18 && hour ≤ 21 && user.isPremium)
{
    applySpecialRate();
}
```

```
// ✅ 각 조건에 의미 부여
bool isWeekend = (day = "SAT" || day = "SUN");
bool isPeakHour = (hour ≥ 18 && hour ≤ 21);
bool isSpecialRate = isWeekend && isPeakHour && user.isPremium;

if (isSpecialRate)
{
    applySpecialRate();
}
```


불리언 타입

- 부정적인 의미보다는 **긍정적인 의미의 이름**을 사용한다.

```
bool isActive;  
bool isDisabled;  
bool isSuspended;
```

// 부정문은 인지 부하가 있다.

```
bool canProcessUser = !isActive && !isDisabled && !isSuspended;
```

```
bool isActive;  
bool isEnabled;  
bool isOperational;
```

// 긍정문은 바로 읽을 수 있다.

```
bool canProcessUser = isActive && isEnabled && isOperational;
```

단축평가

단축 평가

- 단축 평가(Short-Circuit Evaluation)

- && 연산자나 || 연산자는 왼쪽 피연산자 평가 후 오른쪽 피연산자를 평가해 최종 결과를 도출한다.
- 그러나 왼쪽 피연산자의 평가 결과에 따라 오른쪽 피연산자를 평가하지 않아도 최종 결과를 알 수도 있다.
 - `false && true` → 오른쪽을 평가하지 않아도 결과는 `false`
 - `true || false` → 오른쪽을 평가하지 않아도 결과는 `true`
- 따라서 단축 평가에 유의하여 효율적으로 식을 작성하자.

선택문의 종류

if문

- if문의 기본 구조는 아래와 같다.

```
// if (boolean-expression) statement
if (true)
{
    System.Console.WriteLine("참일 때 실행된다.");
}
```

if문

- 거짓일 때도 특정 코드를 실행하게 하려면 else 파트를 작성한다.

```
if (false)
{
    System.Console.WriteLine("참일 때 실행된다.");
}
// else 파트에서 if 문을 사용해 분기점을 더 늘릴 수도 있다.
else if (false)
{
    System.Console.WriteLine("위의 조건이 거짓이므로 실행되지 않는다.");
}
else
{
    System.Console.WriteLine("앞의 모든 조건이 거짓이기 때문에 여기가 실행된다.");
}
```

삼항 연산자

- 삼항 연산자(Ternary Conditional Operator)로 (특정 상황에서) if문의 가독성을 높일 수 있다.

```
// 삼항 연산자는 아래와 같이 구성된다.  
// boolean-expression ? expression-if-true : expression-if-false;  
  
// 이것은 if-else로 표현한 것이다.  
if (true)  
{  
    System.Console.WriteLine("참");  
}  
else  
{  
    System.Console.WriteLine("거짓");  
}  
  
// 위의 예시를 삼항 연산자로 표현하면 아래와 같다.  
System.Console.WriteLine(true ? "참" : "거짓");
```

switch문

- switch문은 패턴(Pattern)에 따라 구문을 실행한다. 여러 조건을 나타낼 때 유용하다.

```
double measurement;

// measurement의 값에 따라 4가지 케이스 중 하나가 실행된다.
switch (measurement)
{
    case < 0.0: // measurement < 0.0
        Console.WriteLine($"Measured value is {measurement}; too low.");
        break;

    case > 15.0: // measurement > 15.0
        Console.WriteLine($"Measured value is {measurement}; too high.");
        break;

    case double.NaN: // measurement = double.NaN
        Console.WriteLine("Failed measurement.");
        break;

    default: // 그 외의 경우
        Console.WriteLine($"Measured value is {measurement}.");
        break;
}
```


switch문

- 여러 케이스를 한번에 처리할 수 있다.

```
double measurement;
```

```
switch (measurement)
```

```
{
```

```
    // 여러 경우를 한번에 처리하고 싶다면 break를 쓰지 않으면 된다.
```

```
    // measurement < 0 || measurement > 100
```

```
    case < 0:
```

```
    case > 100:
```

```
        Console.WriteLine($"Measured value is {measurement}; out of an acceptable range.");
```

```
        break;
```

```
    default:
```

```
        Console.WriteLine($"Measured value is {measurement}.");
```

```
        break;
```

```
}
```

부록

더 나아가기

- 백준 온라인 저지에서 아래의 문제를 풀어보자.
 - [1330번: 두 수 비교하기](#)
 - [9498번: 시험 성적](#)
 - [2753번: 윤년](#)
 - [14681번: 사분면 고르기](#)
 - [2884번: 알람 시계](#)
 - [2525번: 오븐 시계](#)
 - [2480번: 주사위 세개](#)

더 나아가기

- 프로그래머스에서 아래의 문제를 풀어보자.
 - [코딩테스트 연습 - 제곱수 판별하기 | 프로그래머스 스쿨](#)
 - [코딩테스트 연습 - 옷가게 할인 받기 | 프로그래머스 스쿨](#)
 - [코딩테스트 연습 - 외계행성의 나이 | 프로그래머스 스쿨](#)
 - [코딩테스트 연습 - 삼각형의 완성조건 \(1\) | 프로그래머스 스쿨](#)
 - [코딩테스트 연습 - 문자열안에 문자열 | 프로그래머스 스쿨](#)

더 나아가기

- 각 타입의 진리값은 어떻게 되나요?
 - int
 - float
 - string
 - bool
- 아래의 불리언 변수의 이름을 지어봅시다.
 - 인벤토리의 아이템이 가득찼는지의 여부
 - 점프할 수 있는지의 여부
 - 로그인을 할 수 있는지의 여부
- 단축 평가란 무엇인가요?

참고자료

- [C# 교과서](#)