



람다식

이것이 C#이다



Contents

- ❖ 람다식, 너는 어디에서 왔니?
- ❖ 처음 만나는 람다식
- ❖ 문 형식의 람다식
- ❖ Func와 Action으로 더 간편하게 무명 함수 만들기
- ❖ 식 트리
- ❖ 식으로 이루어지는 멤버 논리 연산자



14.1 람다식, 너는 어디에서 왔니?

- ❖ 1936년 발표한 알론조 처치의 람다 계산법을 사용하는 식
 - 분명하고 간결한 방법으로 함수를 묘사하기 위해 고안
 - 함수의 정의와 변수, 그리고 함수의 적용으로 구성
- ❖ 제자 존 매카시가 LISP를 만들면서 프로그램에 도입
- ❖ C#, C++, 자바, 파이썬 등의 주류 언어에서 지원



14.2 처음 만나는 람다식

❖ 익명 메소드를 만들기 위해 사용

- 람다식으로 만든 익명 메소드 → 무명 함수(Anonymous Function)

❖ 람다식 선언 형식

- 매개_변수_목록 => 식
- => : 입력 연산자

❖ 람다식 선언 예

- 형식 유추

```
delegate int Calculate(int a, int b);  
  
//.  
static void Main(string[] args)  
{  
    Calculate calc = (a, b) => a + b;  
}
```

C# 컴파일러는 Calculator 대리자의 선언 코드로부터 이 람다식이 만드는 익명 메소드의 매개 변수의 형식을 유추해냅니다.

❖ 데모 예제 - SimpleLambda



14.3 문 형식의 람다식

❖ 식 형식과 문 형식

- 식 형식: $a + b$ 나 $a == b$
- 문 형식: `if (a == b) return 0; else return 1;`

❖ 선언 형식과 사용 사례

```
delegate void DoSomething( );  
// ...  
static void Main(string[] args)  
{  
    DoSomething DoIt = ( ) =>  
    {  
        Console.WriteLine("뭔가를");  
        Console.WriteLine("출력해보자.");  
        Console.WriteLine("이렇게!");  
    };  
    DoIt();  
}
```

당연한 이야기지만, 매개 변수가 없는 경우에는 ()에 아무것도 넣지 않습니다.

문장 형식의 람다식은 {와 }로 둘러쌉니다.

❖ 데모 예제 - StatementLambda



14.4 Func와 Action으로 더 간편하게 무명 함수 만들기

- ❖ 익명 메소드나 무명 함수를 만들기 위해 매번 **별개의 대리자**를 선언해야 할까?
- ❖ **마이크로소프트의 해결책**
 - Func 대리자 - 결과를 반환하는 메소드 참조
 - Action 대리자 - 결과를 반환하지 않는 메소드 참조



14.4.1 Func 대리자

❖ 17가지 버전의 Func 대리자

```
public delegate TResult Func<out TResult>()  
public delegate TResult Func<in T, out TResult>(T arg)  
public delegate TResult Func<in T1, in T2, out TResult>(T1 arg1, T2 arg2)  
public delegate TResult Func<in T1, in T2, in T3, out TResult>(T1 arg1, T2 arg2,  
T3 arg3)  
...  
public delegate TResult Func<in T1, in T2, in T3, ..., in T15, out TResult>(T1  
arg1, T2 arg2, T3 arg3, ..., T15 arg15)  
public delegate TResult Func<in T1, in T2, in T3, ..., in T15, in T16, out  
TResult>(T1 arg1, T2 arg2, T3 arg3, ..., T15 arg15, T16 arg16)
```

❖ 사용 사례

```
Func<int,int,int> func3 = (x,y) => x + y; // 입력 매개 변수는 int 형식 둘, 반환 형식은 int  
Console.WriteLine(func3(2,3));           // 5를 출력
```

❖ 데모 예제 - FuncTest



14.4.2 Action 대리자

- ❖ 결과를 반환하지 않고 일련의 작업을 수행하는 것이 목적
- ❖ 17가지 버전의 Action 대리자

```
public delegate void Action<>()  
public delegate void Action<in T>(T arg)  
public delegate void Action<in T1, in T2>(T1 arg1, T2 arg2)  
public delegate void Action<in T1, in T2, in T3>(T1 arg1, T2 arg2, T3 arg3)  
...  
public delegate void Action<in T1, in T2, in T3, ..., in T14, in T15>(T1 arg1, T2  
arg2, T3 arg3, ..., T14 arg14, T15 arg15)  
public delegate void Action<in T1, in T2, in T3, ..., in T14, in T15, in T16>(T1  
arg1, T2 arg2, T3 arg3, ..., T14 arg14, T15 arg15, T16 arg16)
```

❖ 사용 사례

```
Action<double, double> act3 = (x, y) =>  
{  
    double pi = x / y;  
    Console.WriteLine("Action<T1, T2>({0}, {1}) : {2}", x, y, pi);  
};  
act3(22.0, 7.0);
```

❖ 데모 예제 - ActionTest

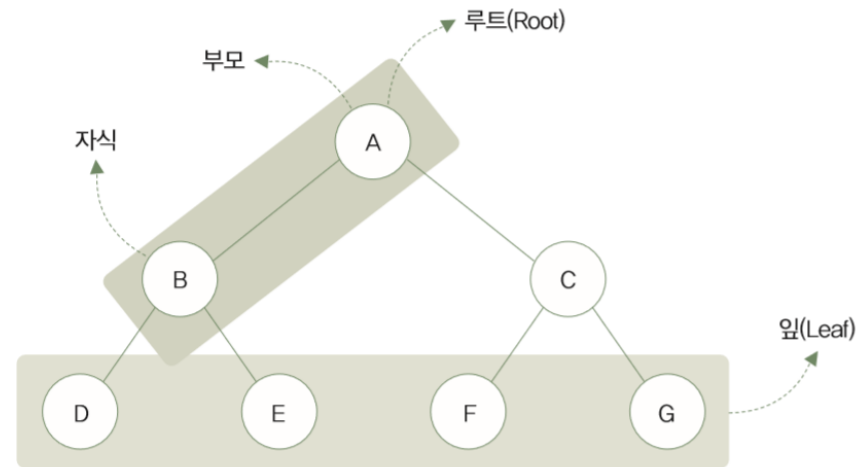


14.5 식 트리(1)

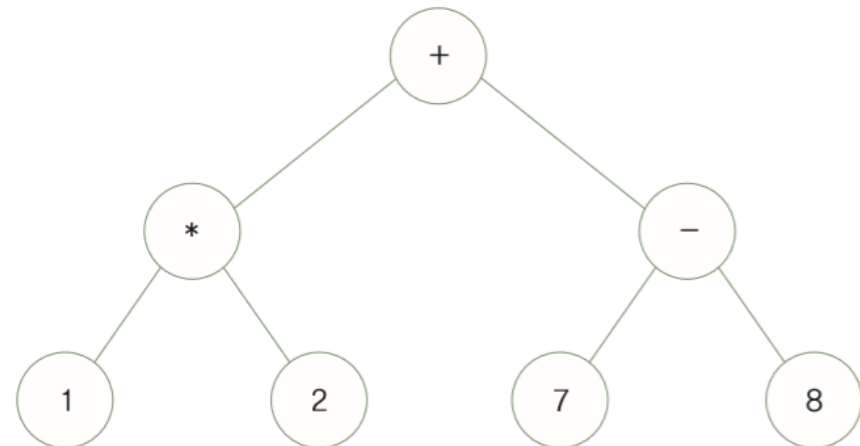
❖ 트리 자료 구조

❖ 식 트리

- 식을 트리로 표현한 자료 구조
- 부모 노드(연산자)가 단 두 개의 자식 노드(피연산자)만 갖는 이진 트리
- 트리의 잎 노드부터 계산해서 루트까지 올라가면 전체 식의 결과



$$1 * 2 + (7 - 8)$$



❖ C#의 식 트리

- C#은 코드에서 직접 식 트리를 조립 및 컴파일해서 사용할 수 있는 기능제공
- → 프로그램 실행 중에 동적으로 무명 함수를 만들어 사용
- System.Linq.Expressions의 Expression 클래스와 파생 클래스



14.5 식 트리(2)

❖ Expression 클래스

- 식 트리를 구성하는 노드 표현
- 파생 클래스들의 객체를 생성하는 역할(팩토리 메소드)

❖ Expression 클래스의 팩토리 메소드 사용 예

```
Expression const1 = Expression.Constant(1);           // 상수 1
Expression param1 = Expression.Parameter(typeof(int), "x"); // 매개 변수 x

Expression exp = Expression.Add( const1, param1 ); // 1 + x
```

❖ 식 트리의 실행

```
Expression<Func<int, int>> lambda1 =
    Expression<Func<int, int>>.Lambda<Func<int, int>>(
        exp,
        new ParameterExpression[] { (ParameterExpression)param1 } );

Func<int, int> compiledExp = lambda1.Compile();
```

실행가능한 코드로 컴파일

컴파일한 무명 함수 실행

```
Console.WriteLine( compiledExp(3) ); // x = 3이면 1+x=4이므로 4를 출력
```

|파일

❖ 데모 예제 – UsingExpressionTree,



14.6 식으로 이루어지는 멤버

❖ 멤버의 본문을 식(Expression)만으로 구현

- Expression-Bodied Member: 식 본문 멤버
- 멤버 => 식;

❖ 사용 사례

```
class FriendList
{
    //...
    public int Capacity // 속성
    {
        get => list.Capacity;
        set => list.Capacity = value;
    }
    public string this[int index] // 인덱서
    {
        get => list[index];
        set => list[index] = value;
    }
}
```

중 인덱서

❖ 데모 예제 - ExpressionBodiedMember





Thank You !

이것이 C#이다

