

.NET Starter 2023

#7. C#을 사용하여 .NET 클래스 라이브러리의 메서드 호출

한상곤(-sd-)

마이크로소프트 MVP / 우분투 한국 커뮤니티



REVIEW

- 5가지 데이터 타입

 - 암시적 형변환

- 변수 선언, 할당 그리고 연산

- 문자열 결합과 보간

학습목표

- .NET 클래스 라이브러리 소개
- .NET 클래스 라이브러리에서 여러 종류의 메서드 호출
- 반환 값 및 입력 매개 변수 작업

.NET 클래스 라이브러리 소개

- .NET 클래스 라이브러리는 수만 개의 메서드를 포함하는 수천 개의 클래스 컬렉션
 - Microsoft에서 만들어 애플리케이션에서 사용할 수 있도록 제공
- C# 데이터 형식도 실제로 .NET 클래스 라이브러리에서 제공되는 항목
 - 내부적으로는 데이터 형식도 .NET 클래스 라이브러리의 다른 모든 클래스와 동일한 방식으로 구현
 - 변수에서 사용할 수 있는 유용한 기본 제공 메서드도 제공

상태 저장 메서드 대 상태 비저장 메서드

- 상태는 특정 시점의 실행 환경 조건을 설명
 - 코드가 줄 단위로 실행됨에 따라 값이 변수에 저장
 - 애플리케이션의 현재 상태는 메모리에 저장된 모든 값의 모음
- 일부 메서드는 애플리케이션의 현재 상태와 관계없이 제대로 작동
 - 즉, 상태 비저장 메서드는 메모리에 저장된 값을 참조하거나 변경하지 않고 작동
 - 상태 비저장 메서드를 정적 메서드라고도 함
- 애플리케이션 상태에 대한 액세스 권한이 있어야 하는 메서드도 있음
 - 즉, 상태 저장 메서드는 이미 실행된 이전 코드 줄에 의해 메모리에 저장된 값을 사용하거나 값을 업데이트 또는 메모리에 새 값을 저장하여 애플리케이션의 상태를 수정하는 방식으로 작성 ➡ 이 메서드를 인스턴스 메서드라고도 함

반환 값 및 입력 매개 변수 작업

- 반환 값
 - 일부 메서드는 함수를 완료하고 “조용히” 종료하도록 설계하며 즉, 완료 시 값을 반환하지 않음 ➔ 이러한 메서드를 void 메서드라고 함
 - 완료 시 값을 반환하도록 설계된 메서드는 일반적으로 연산의 결과, 반환 값은 메서드가 해당 메서드를 호출하는 코드와 다시 통신할 수 있는 주요 방법
- 입력 매개 변수
 - 일부 메서드는 입력 매개 변수를 허용하지 않도록 설계, 이러한 메서드는 작업을 완료하기 위해 추가 입력이 필요하지 않음
 - 다른 메서드는 하나 이상의 입력 매개 변수를 허용하도록 설계, 입력 매개 변수는 메서드가 작업을 수행하는 방법을 구성하거나 직접 작동될 수 있음

Summary

- 5가지 데이터 타입
 - 암시적 형변환
- 변수 선언, 할당 그리고 연산
- 문자열 결합과 보간
- .NET 클래스 라이브러리
 - 상태 저장과 비저장
 - 반환값과 매개변수