C# 문장의 종류

배정문: 값을 변수에 저장하는 데 사용

예) a= b= exp;

- 형 변환

묵시적 형 변환: 컴파일러에 의해 자동으로 변환됨

명시적 형 변환: 프로그래머가 캐스트(cast) 연산자를 사용하여 변형해야 함

혼합문: 여러 문장을 한데 묶어 하나의 문장으로 나타냄

예) {<선언> 또는 <문장>}

지역변수 – 블록 내부에서 선언된 변수, 선언된 불록 안에서만 참조 가능

제어문: 프로그램의 실행 순서를 바꾸는 데 사용

-조건문: 주어진 조건에 따라 수행되는 부분이 다를 때 사용하는 문장 예)if, switch

-반복문: 프로그램의 특정한 부분을 주어진 조건이 만족될 때까지 반복해서 실행하는 문

장 예)for, while, do-while, foreach

-분기문: 지정된 곳으로 제어를 옮기는 역할 예)break, continue, return, goto

오버플로우 검사문: 정수식 연산의 결과가 정수형이 표현할 수 있는 범위를 넘어가는 것

* 정수식에서 오버플로우가 일어나는 지를 명시적으로 검사하여 처리하는 것이 바람직

-checked문: 오버플로우를 명시적으로 검사하는 문장

- unchecked문 : 오버플로우 의도적으로 검사하지 않는 문장

표준 입출력 문

-입력: 외부 입력 장치로부터 데이터를 프로그램으로 읽어 들이는 행위 예) Console.Read()

-출력: 프로그램상의 어떤 값을 출력 장치로 내보내는 행위 예) Console.Write()

-> {N[,W][:formatCharacter]}

N:매개변수를 위치적으로 자칭하는 정수 (0부터 시작)

W: 출력될 자리수의 폭을 나타내며 선택으로 명시, ‘-’ 기호를 붙이면 좌측 정렬

로 출력

formatCharacter: 한 문자로 이루어직 형식 지정 문자를 의미

클래스

|  |  |
| --- | --- |
| 수정자 | 설명 |
| Public | 다른 프로그램에서 사용가능 |
| Internal | 같은 프로그램에서만 사용가능  수정자가 생략된 경우 |
| Static | 정적 클래스, 클래스의 모든 멤버가 정적 멤버  객체 단위로 존재하는 것이 아니라 클래스 단위로 존재 |
| Partial | 부분 클래스  여러 파일에 나누어서 같은 이름의 클래스를 정의 |
| Abstract, sealed | 파생 클래스 |
| Protected, private | 친구외에는 참조 불가( 외부에서 참조 불가) |
| New | 중첩 클래스에서 사용되는 베이스 클래스의 멤버를 숨김 |
| Readonly | 읽기 전용 필드  값이 변할 수 없는 속성  실행 중 값에 값이 결정 |
| Const | 값이 변할 수 없는 속성  컴파일 시간에 값이 결정  상수 멤버의 선언 형태 |

필드: 클래스의 형태에서 객체의 구조를 기술하는 자료

메소드: 객체 행위를 기술하는 방법

* 객체의 상태를 검색하고 변경하는 작업
* 특정한 행동을 처리하는 프로그램 코드를 포함하고 있는 함수의 형태

매개변수 전달 방법

-값 호출(call by value): 실 매개변수의 값이 형식 매개 변수로 전달

-참조 호출(call by reference): 주소 호출(call by address), 실 매개변수의 주소가 형식 매개변수로 전달

시그니처: 메소드를 구별하는데 쓰이는 정보(메소드의 이름, 매개변수의 개수, 매개변수의 자료형,)

생성자

- 객체를 생성할 떄 객체의 초기화를 위해 자동으로 호출되는 루틴

- 클래스와 동일한 이름을 갖는 메소드

- 객체 생성시에 필요한 초기 값을 매개변수로 가질 수 있음

소멸자: 클래스의 객체가 소멸될 때 필요한 행위를 기술한 메소드

Finalize() 메소드: 컴파일 시 소멸자를 finalize() 메소드로 변환해서 컴파일

Finalize() 메소드 재정의할 수 없음

객체가 더 이상 참조되지 않을 때 GC(Garbage Collection)에 의해 호출

Dispose() : CLR에서 관리되지 않은 자원을 직접 해제할 때 사용

자원이 스코프를 벗어나면 즉시 시스템에 의해 호출

프로퍼티(property): 클래스의 private 필드를 형식적으로 다루는 일종의 메소드

-set- 접근자: 값을 지정

-get-접근자로: 값을 참조

-get-접근자 혹은 셋: 접근자만 정의할 수 있음

인덱서(Indexer): 배열 연산자인 ‘[]’를 통해서 객체를 다룰 수 있도록함

지정어 this를 사용하고, ‘[]’안에 인덱스로 사용되는 매개 변수 선언

겟-접근자 혹은 셋- 접근자만 정의할 수 있음

델리게이트(delegate): 메소드 참조 기법

* 객체지향적 특징이 반영된 메소드 포인터
* 이벤트와 스레드를 처리하기 위한 방법론

델리게이트 정의 시 주의점

델리게이트 할 메소드의 메소드 반환형 및 매개변수의 개수, 반환형을 일치시켜야 함

델리게이트 사용

델리게이트 객체 생성 -> 대상 메소드를 연결

이벤트: 사용자의 행동에 의해 발생되는 사건, 어떤 사건이 발생한 것을 알리기 위해 보내는 메시지, C#에서는 이벤트 개념을 프로그래밍 언어 수준에서 지원

이벤트 정의 순서

이벤트 처리기의 형태와 일치하는 델리게이트를 정의 -> 델리게이트를 이용하여 이벤트를 선언-> 이벤트 처리기를 작성 -> 이벤트에 이벤트 처리기를 등록 -> 이벤트 발생

연산자 중복의 의미

-시스템에서 제공한 연산자를 재정의하는 것

-클래스만을 위한 연산자로서 자료 추상화가 가능

-문법적인 규칙은 변경 불가 (연산 순위나 결합 법칙등)

구조체: 클래스와 동일하게 객체의 구조와 행위를 정의하는 방법(클래스: 참조형, 구조체: 값형)

클래스와 구조체와 클래스의 차이점

-클래스는 참조형이고 구조체는 값형이다

-클래스는 객체는 힙에 저장되고 구조체 객체는 스택에 저장된다.

-배정 연산에서 클래스는 참조가 복사되고 구조체는 내용이 복사된다

-구조체는 상속이 불가능하다.

-구조체는 소멸자를 가질 수 없다.

-구조체의 멤버는 초기값을 가질 수 없다.

메소드 재정의(Method Overriding)

베이스 클래스에서 구현된 메소들르 파생 클래스에서 구현된 메소드로 대체함

메소드의 시그니처가 동일한 경우: 메소드 재정의

메소드의 시그니처가 다른 경우: 메소드 중복

가상 메소드(Vertual Method)

지정어 virtual로 선언된 인스턴스 메소드임

파생 클래스에서 재쟁의해서 사용한 것임을 알려주는 역할을 함

-new 지정어: 객체 형에 따라 호출

-override 지정어: 객체 참조가 가리키는 개체에 따라 호출

추상 클래스 (Abstract class)

추상 메소드를 갖는 클래스

추상메소드: 실질적인 구현을 갖지 않고 메소드 선언만 있는 경우

클래스 형 변환

상향식 캐스트( up-casting)

-타당한 변환

-파생 클래스형의 객체가 베이스 클래스형의 객체로 변환함

하향식 캐스트(down-casting)

-타당하지 않는 변환

-캐스트 연산자 사용: 예외 발생

인터페이스

-사용자 접속을 기술할 수 있는 프로그래밍 단위

-구현되지 않은 멤버들로 구성된 순수한 설계의 표현

인터페이스의 특징

-지정어 interface 사용함

-멤버로는 메소드, 프로퍼티, 인덱스 이벤트가 올 수 있으며 모두 구현부분이 없음

-다중 사속 가능함

-접근수정자: public, protected, internal, private,new

제네릭: 변수의 형을 매개 변수로 하여 클래스나 메소드의 알고리즘을 자료형과 무관하게 기술하는 기법

제네릭의 장점

-알고리즘의 재사용성을 높임

-자료형에 따른 프로그램의 중복을 줄임

-프로그램의 구조를 단순하게 만듦

애트리뷰트

-어셈블리나 클래스, 필드, 메소드, 프로퍼티 등에 다양한 종류의 속성 정보를 추가하기 위해서 사용

-어셈블리에 메타데이터형식으로 저장되며 이를 참조하는 .NET 프레임 워크나 C# 또는 다른 언어의 컴파일러에 의해 다양한 용도로 사용함

Conditional 애트리뷰트

-조건부 메소드를 작성할 때 사용함

-c/c++ 언어에서 사용했던 전처리기 지시어를 이용하여 명칭을 정의(#define)함

Obsolete 애트리뷰트

-앞으로 사용되지 않을 메소드를 표시하기 위해서 사용함

- 해당 애트리뷰트를 가진 메소드를 호출할 경우 컴파일러는 컴파일 과정에서 애트리뷰트에 설정한 내용이 출력하는 경고를 발생함