C# 교실

CLASS 4
- OOP Part2

고주형
2019/06/06

오늘 할 것

- Encapsulation (캡슐화)
- Information Hiding (정보 은닉)
- Property
- Inheritance
- Method Overriding 복습
- Type의 조상: System.Object
- Array의 조상: System.Array
- this
- base

객체 지향의 철학

#캡슐화
#정보 은닉
#객체지향의 철학
#객체지향프로그래밍(OOP)

캡슐화가 무엇일까요?

감기약(캡슐) 안에는 어떤 성분들이 들어있을까?

설탕? 비타민? 커피?

성적표? 미세먼지? 해열제?

캡슐화가 무엇일까요?

캡슐화

관련된 데이터와 함수를 묶는 것.
다른 말로, bundling.

캡슐화를 왜 할까?

추상화 때문!

== 더 나은 모듈화

== 편하게 프로그래밍 가능!

== 모듈을 선택해서 편하게 프밍 가능

참고
(출처-https://frontierdev.tistory.com/93)

(c언어) 구조체에서는 데이터밖에 못 묶기 때문에 그와 관련된 데이터와 함수를 추적하기 어려웠다.

정보 은닉 전 접근 제한자 알아보기

접근제한자

- private: 클래스 내부에서만 접근 가능
- protected: 내부에서 접근 및 파생 클래스에서만 접근 가능
- public: 내부, 파생클래스, 외부에서 모두 접근 가능
- internal: 동일 어셈블리 내에서는 public에 준한 접근 가능, 다른 어셈블리에서는 접근 불가
- internal protected: 동일 어셈블리 내에서 정의된 파생 클래스까지만 접근 가능

정보 은닉은 무엇일까요?

정보 은닉: 캡슐 속에 있는 데이터와 함수를 외부에 노출시키지 않는 것.

(오해ㄴ ㄴ) 여기서 노출시키지 않는다는 것은 다가 아니라 굳이 노출 시킬 필요가 없을 때, 숨긴다는 것이다.

따라서, 이 객체가 외부와 소통하기 위한 통로(공개 인터페이스)를 빼곤 다 private(또는 protected)으로 설정 한다는 것!
그러면 우리는 구현에 대한 세부사항은 알 수 없게 된다.

정보 은닉은 왜 필요할까요?

정보 은닉은 **관심 원리(소.공.에서 배움)**를 반영하는 것이다.

이는 하드웨어에서 많이 찾아 볼 수 있기 때문에 하드웨어를 예를 들어서 설명해 보겠다.

마우스의 내부 구현사항은 다 숨겨져 있고 인터페이스인 버튼이나 휠만 공개 인터페이스로 제공된다.

정보 은닉은 왜 필요할까요?

1. 추상화!!
   - More 정보은닉 More 추상화
2. 내부 데이터 or 알고리즘을 변경하기 쉽다.
   - 내가 공개 인터페이스 말고 객체 내부의 데이터를 직접적으로 사용했다고 하면 내부 데이터를 변경할 때 다 수정해야한다.
3. 모듈의 독립성을 높여준다.
   - 다른 모듈과의 의존도를 낮춘다.

정보 은닉과 캡슐화의 차이

캡슐화는 관련 요소들을 묶어 줌으로써, 캡슐 내부와 외부를 구분하는 것인 반면, **정보은닉**은 캡슐 내의 요소들에 대한 세부 구현 사항을 외부에 숨기는 것이다.

Property

Property는 C#에서 있는 get/set의 shortcut이다.

get/set은 뭘까?
우리가 멤버에 접근하는 방법은 크게 두가지다.

```
Ex)  a = 3;                          //a를 set
     Console.WriteLine(a);           //a를 get
```

나중에 공부할 때 참고하세용 https://swconsulting.tistory.com/35

Property

(주의)
위의 프로퍼티는 클래스에서 말하는 속성이랑 다른 말이다.
클래스의 속성은 필드와 메서드를 뜻한다.
Ex) String의 속성은 Length.

둘 다 영어로 property라고 말한다고 혼동하지 말자. 속성과 프로퍼티는 다르다.

Property가 왜 필요한가?

Property가 왜 필요할까?

1. 정보 은닉!
2. 멤버 변수에 접근하는 것을 완벽하게 추적할 수 있다.

예를 들어서, hp가 0이면 die하게하고 싶다.
set하는 부분을 만들면 그 부분에서 hp가 0으로 set되면 die하게 만들면 된다.
set이 없다면 공격을 받을 때(스킬, 기본공격, 독 뎀, 등등) hp가 0이 되는지 체크.
Or Update에서 매 프레임마다 hp<1인지 체크.

Inheritance(상속)

상속을 하면 부모(Base or Super) 클래스의 필드 및 메서드들을 자식(파생) 클래스에서 사용할 수 있다!!

여기서 퀴즈!! 자식(파생) 클래스에서 상속 받은 클래스의 필드 및 메서드를 사용하려면 부모 클래스에서 그 멤버들의 접근 제어자는 뭘로 설정해야 될까요?

1. public 2. protected 3. private

답은 다음 페이지에

자식(파생) 클래스는 부모 클래스의 멤버들 외에 또 필요한 필드와 메서드를 추가해서 사용한다. (실습 했을 때의 기억을 떠올려봅시다.)

퀴즈!! 자식(파생) 클래스에서 상속 받은 클래스의 필드 및 메서드를 사용하려면 부모 클래스에서 그 멤버들의 접근 제어자는 뭘로 설정해야 될까요?

1. public 2. protected 3. private

답) 1. public 2. protected

부모 클래스의 멤버가 1., 2.여야지만 자식 클래스에서 접근할 수 있다.

Method Overriding 복습

- Overriding? 최우선 되는
- Method Overriding: 클래스간 상속 관계에서 메서드를 재정의하는 것

- virtual(가상): '부모' Class에서 가상 메소드로 정의한다.
- <주의> 모든 메소드가 가상 메소드인 자바와 달리 C#은 virtual키워드로 가상 메서드로 만들어줘야 된다.

- override(짓밟다, 우선하다, 최종 결정권을 갖다): '자식' Class에서 재정의!

Type조상: System.Object

생성자

Object()	Object 클래스의 새 인스턴스를 초기화합니다.

메서드

Equals(Object)	지정한 개체와 현재 개체가 같은지 여부를 확인합니다.
Equals(Object, Object)	지정한 개체 인스턴스가 동일한지 여부를 확인합니다.
Finalize()	가비지 컬렉션이 회수하기 전에 개체가 리소스를 해제하고 다른 정리 작업을 수행할 수 있게 합니다.
GetHashCode()	기본 해시 함수로 작동합니다.
GetType()	현재 인스턴스의 Type을 가져옵니다.
MemberwiseClone()	현재 Object의 단순 복사본을 만듭니다.
ReferenceEquals(Object, Object)	지정한 Object 인스턴스가 동일한지 여부를 확인합니다.
ToString()	현재 개체를 나타내는 문자열을 반환합니다.

Array조상: System.Array

Methods/Properties

Clear	Public shared method that sets a range of elements in the array to zero or to a null reference.
Copy	Overloaded public shared method that copies a section of one array to another array
GetLength	Returns a 32 bit Integer representing the number of elements in the specified dimension
GetLongLength	Returns a 64 bit Integer representing the number of elements in the specified dimension
GetLowerBound	Get the upper bound of the specified dimension. Starting at the number 0.
GetUpperBound	Get the lower bound of the specified dimension. Starting at the number 0.
GetValue	Gets the value of the specified element in the array. Starting at the number 1.
IndexOf	Overloaded public shared method that returns the index (offset) of the first instance of a value in a one-dimensional array.
IsFixedSize	Returns a value indicating whether the array has a fixed size
IsReadOnly	Returns a value indicating if an array is readonly or not
LastIndexOf	Overloaded public shared method that returns the index of the last instance of a value in a one-dimensional array.
Length	Public property that returns the length of the array. Returns a 32 bit Integer representing the total number of elements in all the dimensions of the array.
LongLength	Returns a 64 bit Integer representing the total number of elements in all the dimensions of the array.
Rank	Returns the number of dimensions of the array.
Resize	
Reverse	Overloaded public shared method that reverses the order of elements in a one dimensional array
Sort	Overloaded public shared method that sorts the values in a one-dimensional array.

this

자기 자신의 클래스를 지칭

주의) 정적 메서드 내에서 this 키워드를 사용하는 것은 오류이다.

base

부모의 클래스를 지칭

주의) 정적 메서드 내에서 base 키워드를 사용하는 것은 오류이다.

this/base 연습해보기

정리가 잘 된 사이트를 발견해서 이 사이트로 연습해 봅시당!!

- https://nowonbun.tistory.com/102

실전 코딩!

#실습
#캡슐화
#킹체지향적으로

실습 - 캐릭터 클래스 심화

1. Property를 이용하여 내가 범위 이상으로 넘어가면 알려주는 코드를 짜보자.

2. Object의 멤버 오버라이딩 해보기.
 Character.ToString()을 하면 모든 필드를 하도록 만들어보자.

출처

- 캡슐화와 정보 은닉
  https://frontierdev.tistory.com/93
- Object
  https://docs.microsoft.com/ko-kr/dotnet/api/system.object?view=netframework-4.8
- Array
  https://bettersolutions.com/csharp/arrays/system-array.htm
- Property
  https://swconsulting.tistory.com/35
- this/base
  https://nowonbun.tistory.com/102