1. 객체지향, 절차지향

* 절차지향 : 함수를 기반으로 만들겠다! (Text RPG)
* 프로잭트가 작으면 쉽고 빨리빨리 만들 수 있지만, 유지보수가 힘들다.
* 객체지향(OOP) : 설계도를 만들어서 붕어빵 틀에 붕어빵을 찍어내듯이 설계한다.

1. Class, Struct

* Class : ref(참조) 진짜 변수를 가지고 값을 변경
* Struct : copy(복사) 복사한 변수를 가지고 값을 변경

1. 스택, 힙 (메모리)

* 스택(stack) : 불완정하고 임시적임. 함수를 실행하기 위한 메모장. 매우 빠르게 접근과 할당이 가능하며 메모리에 낭비되는 공간이 없이 사용 가능
* 힙(heap) : 사용자가 따라 할당, 관리를 해서 사용해야 하는 공간, 용량은 크지만 단편화의 위험성
* C#같은 경우 자동차의 오토 변속처럼 new 를 사용해도 따로 삭제 처리를 하지 않아도 되지만, C++같은 경우 delete를 사용해 사용자가 지워줘야 한다.

텍스트이(가) 표시된 사진

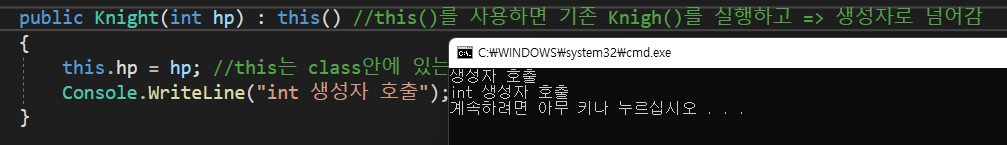
자동 생성된 설명

1. 생성자

텍스트이(가) 표시된 사진

자동 생성된 설명

* This()



1. Static

* 정적, class에 종속적인 필드 또는 함수가 되는 것.

1. 객체지향(OOP) 3대장 : 은닉성, 상속성, 다형성

* 상속성

텍스트이(가) 표시된 사진

자동 생성된 설명

* 은닉성
* Public : 모두가 확인할 수 있다. 접근한정자. 보안에 취약
* Private : 나만 사용하겠다.
* Protected : private와 비슷하지만 상속을 받았으면 사용할 수 있다.
* Class 형식 변환

텍스트이(가) 표시된 사진

자동 생성된 설명

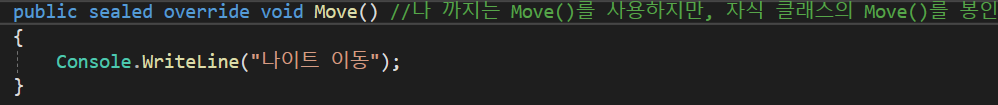
* 다형성
* Virtual, override : 어떤 함수에다 실제 타입에 따라 다양한 형태로 동작을 하겠다!

텍스트이(가) 표시된 사진

자동 생성된 설명

* 부모 클래스의 기능도 사용하고 싶을 때 (base.)

텍스트이(가) 표시된 사진

자동 생성된 설명

* Virtual은 성능의 부하를 주기 때문에 많은 사용은 자제

1. 문자열 둘러보기

텍스트이(가) 표시된 사진

자동 생성된 설명

1. textRPG(객체 지향 적으로 만들기) : state 패턴