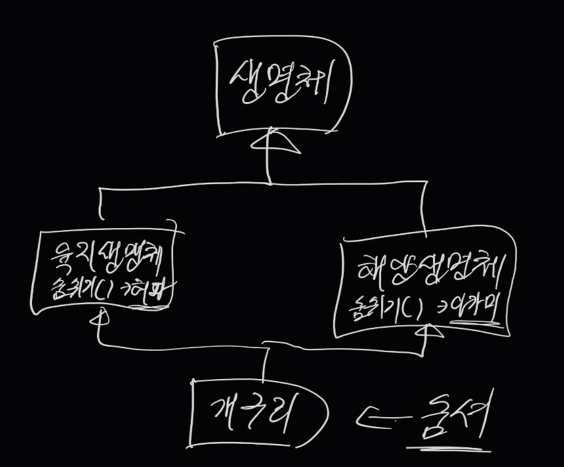
죽음의 다이아몬드

클래스를 다중상속을 못하도록 막는 이유

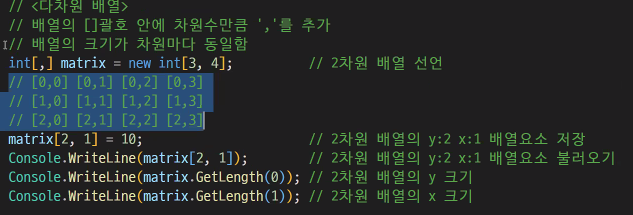


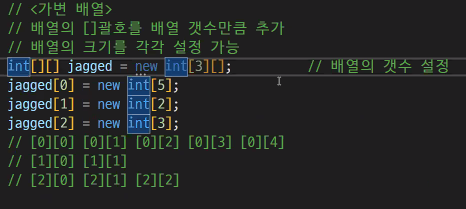
모호한 상속상태가 되어버릴 수 있기떄문에

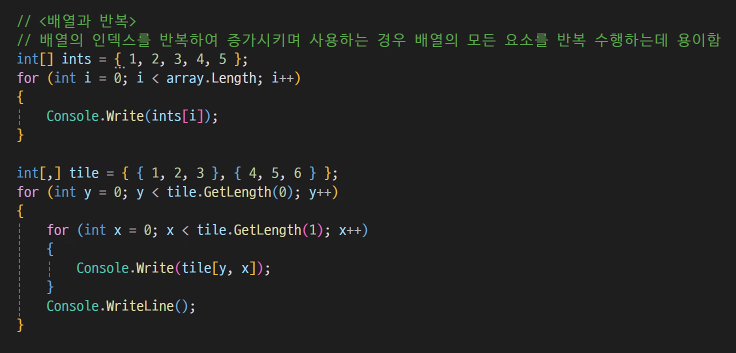
C++에서는 이를 만들 수는 있지만 이는 기계친화적 언어라 잘못 만들더라도 프로그래머 문제로 인식하기때문이다. 죽음의 다이아몬드도 잘사용하면 문제가 없기에 사용 가능하도록 해준다. 이는 즉 C++은 유연성이 있다고 볼 수 있다.

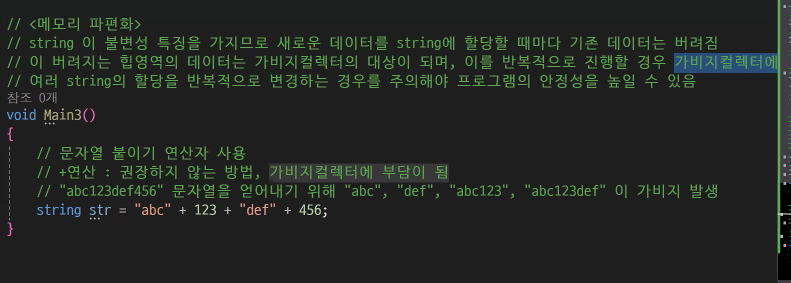
C#는 인간친화적 언어라 문제가 될 수 있는 것은 전부 막아두었다고 보면된다. 즉 C#는 안전하지만 유연성이 없다고 볼 수 있다.

(Bool값을 int로 바꾸는데 가능 불가능이라는 부분을 보더라도 C++과 C#의 유연성을 엿볼 수 있다.)









메모리 파편화로 인하여 가비지 컬렉터를 작동하게 만들고 이를 계속하게 되면 프레임드랍의 원인이 될 수 있다.

