

# 추상 클래스(Abstract)와 인터페이스(Interface) 차이

2022년 5월 16일 월요일 오전 10:17

## 추상 클래스

1. 미완성된 클래스
2. 오로지 자식의 상속을 통해서만 생성 가능 ( new 생성 불가 )
3. 추상 메소드를 가짐
4. 추상 메소드는 본문이 없음 (선언만 한 상태)

```
참조 1개
abstract class Animal
{
    public string name;
    참조 1개
    public abstract void Move(); //추상 메소드

    참조 0개
    public void Move2()
    {
        Console.WriteLine("일반 메소드");
    }
}
```

```
참조 2개
class Dog : Animal
{
    참조 1개
    public override void Move()
    {
        Console.WriteLine("네 발로 이동한다.");
    }
}
```

## 인터페이스

1. 미완성된 요소
2. 인터페이스에 속한 메서드는 모두 가상 메서드
3. Virtual/override 키워드 사용하지 않음
4. 구현부 없음
5. 모두 public
6. 객체를 생성할 수 없지만(new) 참조 변수는 만들 수 있음
7. 클래스의 표준화 가능
8. 다른 관련 없는 클래스를 연결 가능
9. 다중 상속 가능

```

참조 1개
interface ILogger //인터페이스
{
    참조 1개
    void WriteLog(string log);
}

참조 0개
class ConsoleLogger : ILogger //인터페이스를 구현한 클래스
{
    참조 1개
    public void WriteLog(string log)
    {
        Console.WriteLine("{0} {1}", DateTime.Now.ToLocalTime(), log);
    }
}

```

```

참조 0개
class Program
{
    참조 0개
    static void Main(string[] args)
    {
        ILogger logger = new ConsoleLogger();
    }
}

```

추상 클래스와 인터페이스 차이

1. 다중 상속의 차이점 ( 인터페이스 다중 상속 가능)
2. Virtual, override 키워드 차이 ( 인터페이스는 사용안함)
3. 인터페이스는 모두 가상 메소드로 구성됨
4. 자신의 객체를 생성할 순 없지만 참조 가능 (추상 클래스는 무조건 자식의 상속으로만 생성됨)