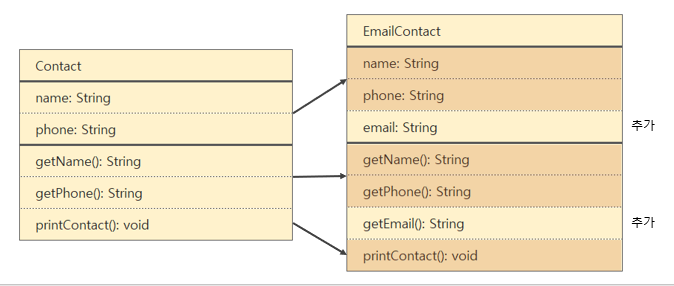
**상속**

* 부모 클래스의 속성과 기능을 자식 클래스에게 그대로 물려주는 것

클래스의 멤버변수나 메소드의 추가가 필요한 경우

1. 클래스를 직접 수정하는 방법
   1. 연결 돼 있던 코드를 모두 수정해야할 가능성 높음
   2. 하나의 기능 추가 및 변경을 위해 모든 코드를 수정해야 할 수도 있음
2. 기존의 클래스를 상속받아 필요한 변수나 기능만 추가하는 방법
   1. 연결 돼 있던 기존의 코드에 영향을 주지 않고 안전하게 수정사항을 반영할 수 있음

SOLID 원칙 지키기 -> 확장 가능



이메일 contact로의 확장

This는 나의 인스턴스

Super는 나의 인스턴스의 상위 클래스

상속은 기존의 클래스에서 확장한 것으로 부모 클래스를 수정할 경우 자식 클래스에서 수정된 내용을 즉시 사용할 수 있음

만약, 상속을 하지 않고 일일이 기능을 추가할 경우, 필요한 클래스를 모두 찾아 수정을 해야하는 불편함이 생김

* 상속은 클래스 속성이나 기능의 추가/수정/삭제를 유연하게 처리할 수 있게 함

Java의 모든 클래스는 상속이 가능하나 final 클래스는 상속이 불가능

객체지향프로그래밍의 4가지 특징

1. 캡슐화

여러 기능을 메소드(캡슐)로 묶어 처리한다.

2. 추상화

기능의 정의만 존재하고 구현은 하지 않는다

3. 다형성

상속 및 구현 대상(부모클래스) 타입에 포함되는 것을 허가한다.

즉, 하나의 타입으로 여러가지 타입을 표현할 수 있다.

4. 상속

부모클래스의 모든 속성, 기능을 확장한다.