

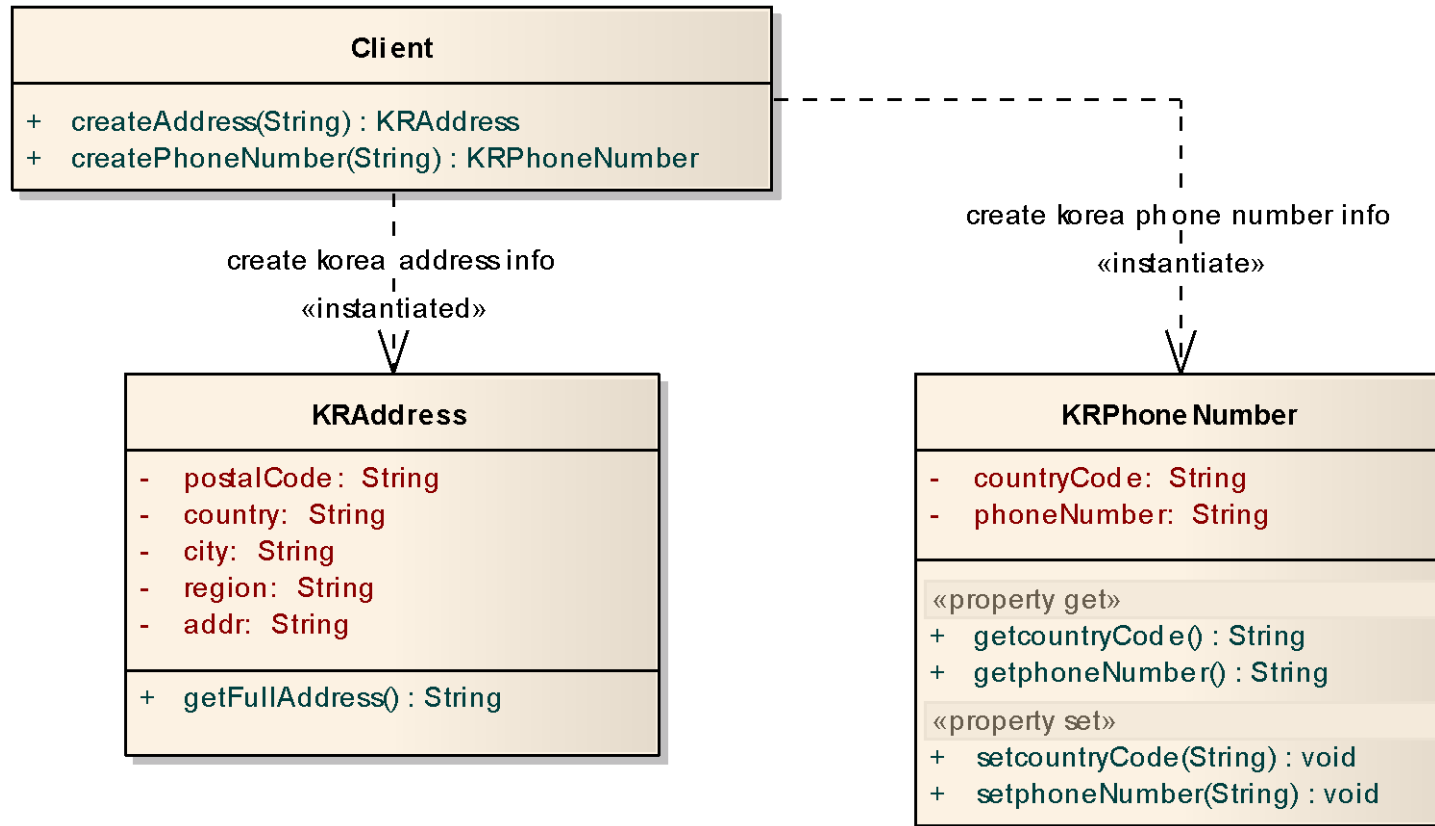
Korea
Software
Technology
Association



UML2.x 기초 다루기 실습

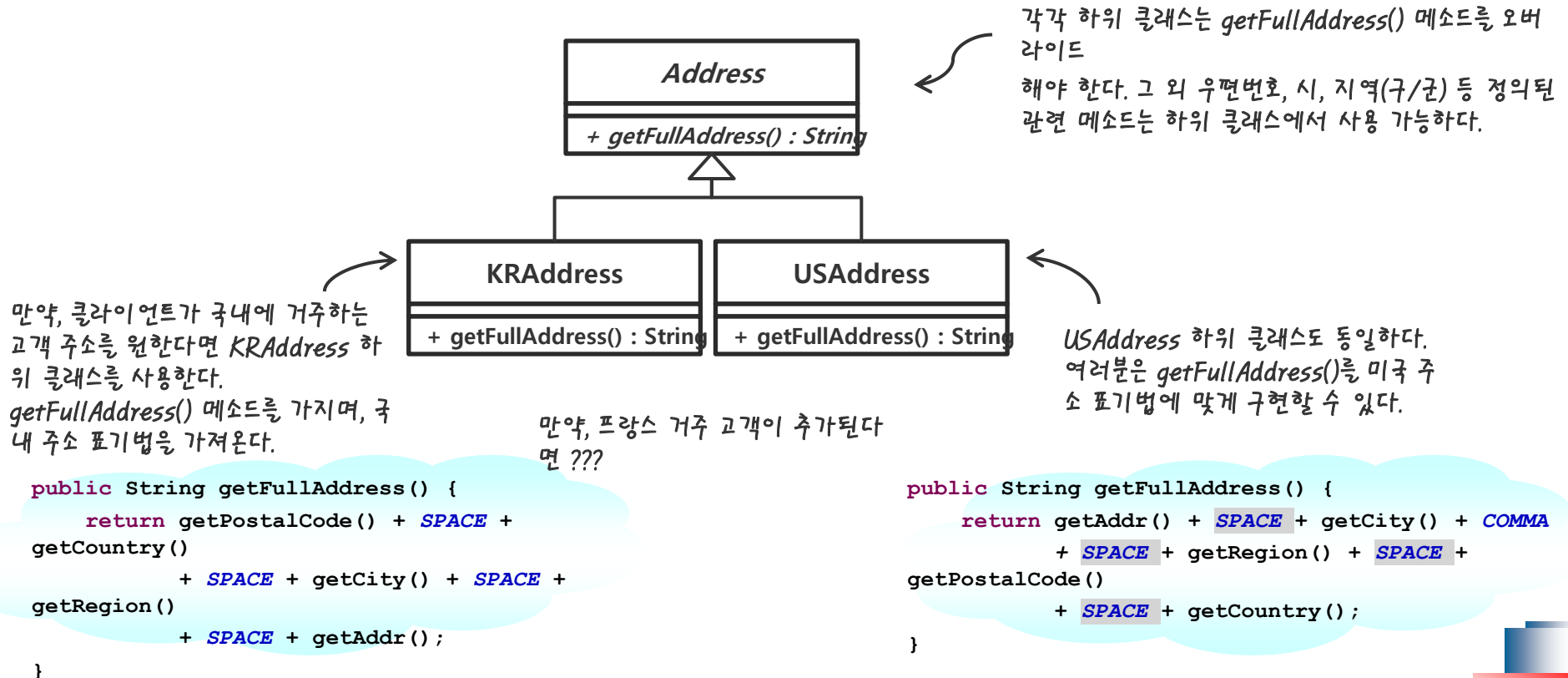
훈련기간: 2010.01.25 ~ 02.05

강사명: 손재현 -넥스트트리소프트
-jhsohn@nexttree.co.kr



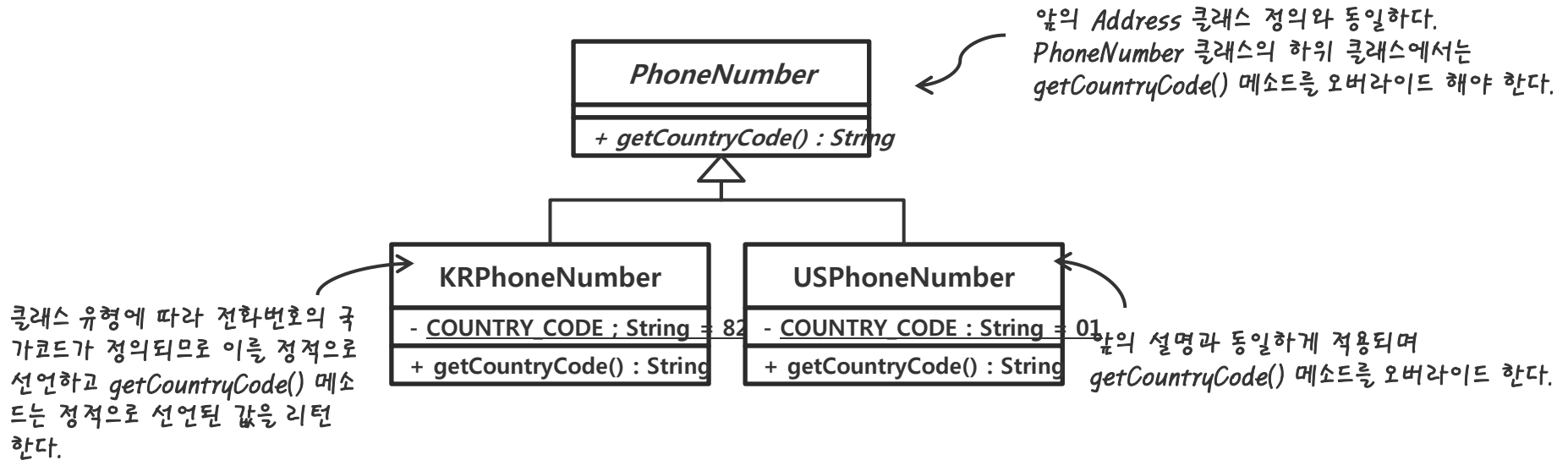
□ 추상화 결정하기

- 이미 국내에 거주하는 고객 주소에 대해서는 잘 정의되어 있음
- 하지만, 다양한 형태의 고객 주소 변화를 가지기 위해서 추상 클래스를 도출해야 함



□ 추상화 결정하기

- 고객의 전화번호도 위와 동일함. 단, 각각의 전화번호의 국가 코드는 하위 클래스 유형에 따라 결정되며 이를 정적인 형태로 정의해야 함



```
private static final String COUNTRY_CODE = "82";
```

```
public String getCountryCode() {
    return COUNTRY_CODE;
}
```

```
private static final String COUNTRY_CODE = "01";
```

```
public String getCountryCode() {
    return COUNTRY_CODE;
}
```

□ 변경의 국지화 (Localization of Changes)

- 앞서 추상화시킨 클래스의 인스턴스는 어떻게 생성할 것인가? 클라이언트에서 인스턴스를 생성한다면 유지보수의 문제가 발생함

```
public Address createAddress(String countryType) {
    Address address = null;
    if (countryType.equals("KR")) {
        address = new KRAAddress();
    }
    else if countryType.equals("US")) {
        address = new UAddress();
    }
    return address;
}
```

Tight coupling & Weak cohesion

앞서 제시한 프랑스 거주 고객이 추가된다면 아래의 소스가 createAddress() 메소드와 createPhoneNumber() 메소드에 추가되어야 한다.

```
else if countryType.equals("FR")) {
    address = new FRAddress();
}
```

변경사항에 대한 유지보수를 위해
서 변경에 대한 국지화가 필요하다.

```
public PhoneNumber createPhoneNumber(String countryType) {
    PhoneNumber phoneNumber = null;
    if (countryType.equals("KR")) {
        phoneNumber = new KRPhoneNumber();
    }
    else if countryType.equals("US")) {
        phoneNumber = new USPhoneNumber();
    }
    return phoneNumber;
}
```

```
else if countryType.equals("FR")) {
    phoneNumber = new FRPhoneNumber();
}
```

□ 변경의 국지화 (Localization Of Changes)

- 새로운 오브젝트를 추가해야 할 경우 클라이언트 소스의 여러 곳에 영향을 미침
- 그래서 새로운 인스턴스를 생성하기 위한 Factory 객체를 사용해야 함

