

✓ 상속

- ✓ Inheritance, Polymorphism, Encapsulation, Abstraction 을 이해한다.
 - ✓ 멤버변수와 메소드에 Encapsulation 을 적용한다.
 - ✓ Method Overriding을 적용해 본다.
-
- ✓ KOSTA은행은 새로운 고객 관리 시스템을 구축하고 있다. 이전 코드에 이어 추가적으로 AccountDto 클래스에 상속 개념을 적용하려고 한다. 또한 AccountDto 및 UserDto 에 Access Modifier 를 통해 Encapsulation 을 적용해 본다.

✓ 상속

1. AccountDto는 종류별로 하위 클래스를 각각 아래와 같이 작성한다.
 - 저축계좌 (SavingAccount) 클래스를 만들고, 이체 수수료 항목을 추가한다.
 - 대출계좌 (LoanAccount) 클래스를 만들고, 담보 항목을 추가한다.
 - 정기예금계좌 (InstallAccount) 클래스를 만들고, 기간, 금액 항목을 추가한다.
2. AccountDto 및 UserDto 에 Encapsulation 을 적용한다.
4. BankService 는 accountList 를 그대로 유지하되, 등록되는 객체는 세분화된 하위 클래스의 객체를 담도록 코드를 변경한다.
5. AccountDto 의 하위 클래스는 toString() 메소드를 Overriding 한다.
6. 위 변화에 따른 다른 코드의 오류가 발생하는 경우 적절하게 수정한다.

✓ 상속

dto.UserDto
<ul style="list-style-type: none">accountSeq: intname: Stringemail: Stringphone: StringisSleep: boolean
<ul style="list-style-type: none">UserDto()UserDto(userSeq: int, name: String, email: String, phone: String, isSleep: boolean)getUserSeq(): intsetUserSeq(userSeq: int): voidgetName(): StringsetName(name: String): voidgetEmail(): StringsetEmail(email: String): voidgetPhone(): StringsetPhone(phone: String): voidisSleep(): booleansetSleep(isSleep: boolean): voidtoString(): String

