

'상속' 강의를 끝으로 클래스 세 개가 있습니다: **BankAccount**, **SavingsAccount**, **MinimumBalanceAccount**. 이제 각 클래스의 인스턴스를 만들어봅시다.

```
BankAccount ba = new BankAccount();
MinimumBalanceAccount mba = new MinimumBalanceAccount();
SavingsAccount sa = new SavingsAccount();
```

각각 다른 클래스의 인스턴스이기 때문에 각 계좌에 1000원씩 넣어주려면 **deposit** 메소드를 따로따로 호출해줘야 합니다.

```
ba.deposit(1000);
mba.deposit(1000);
sa.deposit(1000);
```

만약 계좌가 100개면 100줄이나 써야하기 때문에 문제가 있죠? 'DRY(Don't Repeat Yourself; 중복 배제)'의 법칙을 어겨버립니다.

## 타입 캐스팅 (Type Casting)

다행히도 이 문제를 해결할 방법이 있습니다. **MinimumBalanceAccount** 와 **SavingsAccount** 를 부모 클래스인 **BankAccount** 처럼 다루는 것인데요.

```
ArrayList<BankAccount> accounts = new ArrayList<>;

accounts.add(ba);
accounts.add(mba);
accounts.add(sa);

for (BankAccount account : accounts) {
    account.deposit(1000);
}
```

이렇게 하면 각 계좌가 **BankAccount** 타입으로 '캐스팅(Casting)'되고, 한꺼번에 묶어서 다룰 수 있습니다.

## instanceof

**sa** 에게는 이자를 붙여주고 싶은데, **BankAccount** 클래스에는 **addInterest** 메소드가 없습니다. 만약 여기서 **SavingsAccount** 만 골라서 **addInterest** 메소드를 쓰고 싶으면 **instanceof** 키워드를 사용하면 됩니다.

```

for (BankAccount account : accounts) {
    account.deposit(1000);

    if (account instanceof SavingsAccount) {
        ((SavingsAccount) account).addInterest();
    }
}

```

이렇게 하면 **SavingsAccount** 로 변환이 가능한 경우에만 **addInterest** 메소드를 호출하게 되는 것이죠.



수업을 완료하셨으면 체크해주세요.



수강생 Q&A 보기



(/questions?  
질문하기

assignment\_id=419&sort\_by=popular)  
(/questions/new?

assignment\_id=419&op1=%EA%B0%9D%EC%B2%B4+%EC%A7



이전 강의 (/assignments/414)  
캐스팅

다음 강의  
super



(/assignments/415)