

이전 섹션들에서는 자바의 아주 기초적인 문법만 보았습니다. 이번 섹션부터는 본격적으로 '객체 지향 프로그래밍'에 대해서 배워봅시다.

개요

김모씨가 은행 계좌에 입금을 하고 출금을 하고, 또 최모씨의 계좌에 20,000원을 이체해주려고 합니다. 이런 상황을 어떻게 프로그램으로 표현할 수 있을까요?

일단 '사람'이라는 것을 표현할 방법이 필요하고, '계좌'라는 것을 표현할 방법이 필요합니다. 그런데 현재까지 배운 자료형들(숫자형, 불린형, 배열 등)로는 완벽하게 표현해내기 어렵겠죠? 이런 상황을 쉽게 정리할 수 있는 프로그래밍 방법이 바로 **객체 지향 프로그래밍**입니다.

이 경우에는 '사람'을 표현하는 객체와 '은행 계좌'를 표현하는 객체를 만들 수 있습니다. '**객체**'는 '속성'들과 '동작'들의 묶음이라고 할 수 있는데요. 속성은 '**변수**'로 나타내고, 동작은 '**메소드(함수)**'로 나타냅니다.

'사람' 객체에는 이름, 나이, 보유 현금, 소유 계좌 등의 속성들이 있겠죠? 반면에 '은행 계좌'에는 잔액과 소유인이 속성으로 필요하고, 입금, 출금, 이체 동작들이 필요합니다.

클래스 (Class)

사람 (Person)

'클래스'는 앞서 말한 객체들에 대한 설계도, 또는 설명서입니다. '사람' 객체에 대한 설명서를 먼저 써봅시다.

```
public class Person {  
    String name;  
    int age;  
    int cashAmount;  
    // 은행 계좌?  
}
```

Person이라는 클래스를 만들었습니다. 이 클래스에는 '이름'을 표현하는 **name**이라는 문자열 변수, '나이'를 표현하는 **age**라는 정수 변수, 그리고 '보유 현금'을 표현하는 **cashAmount**라는 정수 변수가 있습니다. '은행 계좌'를 표현하는 변수는 일단 비워두겠습니다.

은행 계좌 (BankAccount)

이번에는 '은행 계좌' 객체에 대한 설명서를 써보겠습니다.

```
public class BankAccount {  
    int balance;  
    // 소유인?
```

}

BankAccount 라는 클래스를 만들고, 이 클래스에는 '잔액'을 표현하는 **balance** 라는 정수 변수가 있습니다. '소유인'을 표현하는 변수는 일단 비워두겠습니다.

연결하기

이제 '사람'과 '은행 계좌'의 관계를 표현해봅시다.

```
public class Person {
    String name;
    int age;
    int cashAmount;
    BankAccount account;
}
```

BankAccount 클래스를 마치 자료형처럼 사용한 걸 보실 수 있습니다. **account** 변수는 **BankAccount** 타입을 갖게 됩니다.

마찬가지로:

```
public class BankAccount {
    int balance;
    Person owner;
}
```

owner 변수는 **Person** 타입을 갖게 됩니다.

이런 식으로 프로그램을 객체 단위로 정리하고, 객체들끼리 엮으면서 프로그래밍 하는 방식이 바로 '객체 지향 프로그래밍'입니다!



수업을 완료하셨으면 체크해주세요.



수강생 Q&A 보기



(/questions?assignment_id=328&sort_by=popular)
질문하기

(/questions/new?)

assignment_id=328&op1=%EA%B0%9D%EC%B2%B4+%EC%A7



이전 강의

객체 지향 프로그래밍 입문

(/assignments/254)

다음 강의

메소드



(/assignments/256)