

클래스가 '설계도'라면, **인스턴스**는 그 설계도를 따른 '객체'를 뜻합니다. 예를 들어서 **Song** 이라는 클래스가 있으면, '걱정말아요 그대', '거위의 꿈', '다행이다' 등은 **Song** 클래스를 따르는 인스턴스라고 부를 수도 있고, 편하게는 그냥 **Song** 인스턴스라고 부를 수 있습니다.

우리의 '은행 계좌 시뮬레이션' 프로그램에서는 **Person** 이라는 클래스가 있는데요. '김신의'라는 이름을 가진 **Person** 인스턴스, '문종모'라는 이름을 가진 **Person** 인스턴스, '성태호'라는 이름을 가진 **Person** 인스턴스 등이 있을 수 있겠죠?

예제

실제로 인스턴스를 생성하는 방법을 볼까요? 일단 저희는 **Person.java** 파일과 **BankAccount.java** 파일밖에 없는데, 이 둘에는 **main** 메소드가 없죠? **main** 메소드가 있어야 프로그램이 돌아갈 텐데요. **main** 메소드를 쓰고, 이런 저런 테스트를 하기 위해서 **BankDriver.java** 파일을 만들겠습니다. (테스트용 클래스나 파일을 만들 때 'Driver'라는 단어를 자주 씁니다!)

```
public class BankDriver {
    public static void main(String[] args) {

    }
}
```

Person 인스턴스 생성

먼저 **Person** 인스턴스를 담는 변수를 만들어야겠죠?

```
public class BankDriver {
    public static void main(String[] args) {
        // 사람 선언
        Person p1;
    }
}
```

실제로 **Person** 인스턴스를 만들기 위해서는 '생성자'라는 것을 써야하는데요. 생성자에 대해서는 곧 자세히 배우니까, 일단은 '인스턴스를 생성하는 메소드' 정도로만 알고 계시면 됩니다.

생성자는 **new** 키워드와 함께 클래스의 이름을 써주면 됩니다.

```
public class BankDriver {
    public static void main(String[] args) {
        // 사람 선언
        Person p1 = new Person();
    }
}
```

이렇게 써주면 이제 **p1** 에는 **Person** 인스턴스가 저장됩니다.

기억을 다시 되살려보면 **Person** 객체는 **name** 변수, **age** 변수, 그리고 **cashAmount** 변수가 있죠? 각 변수에 값을 지정해줍니다.

```
public class BankDriver {
    public static void main(String[] args) {
        // 사람 선언
        Person p1 = new Person();
        p1.name = "김신의";
        p1.age = 28;
        p1.cashAmount = 30000;
    }
}
```

제대로 들어갔는지 값들을 출력해볼까요?

- 만약 **BankAccount**의 비어있는 메소드 오류 때문에 실행할 수 없다면, 해당 부분을 주석처리해 주세요! (인스턴스 강의 영상 3:40초 참고)

```
public class BankDriver {
    public static void main(String[] args) {
        // 사람 선언
        Person p1 = new Person();
        p1.name = "김신의";
        p1.age = 28;
        p1.cashAmount = 30000;

        // 테스트
        System.out.println(p1.name);
        System.out.println(p1.age);
        System.out.println(p1.cashAmount);
    }
}
```

```
김신의
28
30000
```

사실 **Person** 인스턴스는 **account** 라는 속성도 있는데요. 아직 **BankAccount** 인스턴스가 없기 때문에 일단 두겠습니다.

BankAccount 인스턴스 생성

그럼 이제 **BankAccount** 인스턴스도 만들어볼까요?

```

public class BankDriver {
    public static void main(String[] args) {
        // 사람 생성
        Person p1 = new Person();
        p1.name = "김신의";
        p1.age = 28;
        p1.cashAmount = 30000;

        // 은행 계좌 생성
        BankAccount a1 = new BankAccount();
        a1.balance = 100000;
    }
}

```

BankAccount 인스턴스를 생성하고 **a1**에 넣어주었습니다. 그리고 계좌 **a1**의 잔액을 **100000**으로 설정해주었습니다.

두 인스턴스 엮기

객체 지향 프로그래밍에서 중요한 부분은 연관된 객체들끼리 서로 엮는 것이죠? 한번 **p1**과 **a1**을 엮어 보겠습니다.

```

public class BankDriver {
    public static void main(String[] args) {
        // 사람 생성
        Person p1 = new Person();
        p1.name = "김신의";
        p1.age = 28;
        p1.cashAmount = 30000;

        // 은행 계좌 생성
        BankAccount a1 = new BankAccount();
        a1.balance = 100000;

        // 두 객체의 관계 설정
        p1.account = a1;
        a1.owner = p1;

        // 테스트
        System.out.print(p1.name + "의 계좌 잔액: " + p1.account.balance);
    }
}

```

김신의의 계좌 잔액: 100000

p1 (**Person** 인스턴스)의 **account** 변수의 자료형은 **BankAccount** 이죠? **p1.account**에 **BankAccount** 인스턴스인 **a1**을 넣어줬습니다. 마찬가지로 **a1.owner**에는 **Person** 인스턴스인 **p1**을 지정해줬습니

다. 즉, **a1** 은 **p1** 의 은행 계좌이고, **p1** 은 **a1** 은 소유자인 셈이죠!



수업을 완료하셨으면 체크해주세요.



수강생 Q&A 보기



[\(/questions?](#)
질문하기

[assignment_id=330&sort_by=popular\)](#)
[\(/questions/new?](#)

[assignment_id=330&op1=%EA%B0%9D%EC%B2%B4+%EC%A7](#)



이전 강의
인스턴스 [\(/assignments/257\)](#)

다음 강의
입금, 출금



[\(/assignments/332\)](#)