



**클래스 - 정의 및 인스턴스 생성**

## 클래스 - 정의 및 인스턴스 생성

### ● 자바의 클래스

- 자바가 객체지향언어인 이유 : 클래스를 기반으로 프로그래밍하며, 클래스로부터 객체를 만들어 사용
- 클래스(class) = 멤버 변수(=인스턴스 변수, 필드) + 메소드
  - 멤버 변수(= 인스턴스 변수, 필드)
    1. 클래스가 가지고 있는 데이터 및 스펙(ex : 너비, 색상, 나이 등)
    2. 클래스 바로 아래 변수로 정의하며, 클래스 내에서 어디서든 사용 가능
    3. 클래스 외부에서 직접 접근을 막기 위해 일반적으로 접근자 private를 붙여 사용
    4. 멤버 변수의 데이터를 호출하거나 설정하기 위해 getter / setter를 사용
    5. 클래스 내에서 사용될 때에는 this 키워드를 붙여 사용
  - 메소드
    1. 데이터의 처리나 조작을 하는 기능
    2. 클래스 내에서만 사용할 때에는 private 접근자를, 외부에서도 사용할 수 있도록 할 때는 public 접근자를 붙여 사용
    3. 메소드 내에서 선언하는 변수는 멤버 변수가 아니며, 메소드 내에서만 사용 가능
- 클래스 정의 시 관련 있는 멤버 변수와 메소드만 정의(예 : 자동차 클래스에 "숨을 쉰다"와 같은 메소드 정의 X)
- 클래스 정의 시 데이터는 클래스의 명사를, 메소드는 클래스의 동사를 정의
- 클래스 정의 시 어플리케이션의 범위 설정 중요(예: 병원을 방문하는 사람 클래스, 즉 환자를 정의할 때의 멤버 변수와 메소드)

## 클래스 - 정의 및 인스턴스 생성

```
public class BankAccount {  
    private int balance = 0;  
  
    public void deposit(int amount) {  
        this.balance += amount;  
        System.out.println(amount + "원이 입금처리 되었습니다.");  
    }  
  
    public void withdraw(int amount) {  
        this.balance -= amount;  
        System.out.println(amount + "원이 출금처리 되었습니다.");  
    }  
  
    public int getBalance() {  
        return this.balance;  
    }  
  
    public void setBalance(int balance) {  
        this.balance = balance;  
    }  
}
```

## 클래스 - 정의 및 인스턴스 생성

```
public class ClassTest {  
    public static void main(String[] args) {  
        // 두 개의 인스턴스 생성  
        BankAccount park = new BankAccount();  
        BankAccount yoon = new BankAccount();  
  
        // 각 인스턴스를 대상으로 예금 진행  
        park.deposit(10000);  
        yoon.deposit(7000);  
  
        // 각 인스턴스를 대상으로 출금 진행  
        park.withdraw(3000);  
        yoon.withdraw(1000);  
  
        // 각 인스턴스를 대상으로 잔액 조회  
        System.out.println("현재 잔액 : " + park.getBalance() + "원");  
        System.out.println("현재 잔액 : " + yoon.getBalance() + "원");  
    }  
}
```

## 클래스 - 정의 및 인스턴스 생성

- BankAccount 클래스

- 멤버 변수(데이터 및 스펙) : 정수형 변수 balance
- 메소드(기능) : deposit, withdraw, getBalance, setBalance

- ClassTest 클래스

- 클래스의 인스턴스(객체) 생성
  - new 키워드를 사용하며 클래스에 정의한 멤버 변수와 메소드를 인스턴스에서 모두 사용 가능
  - BankAccount park = new BankAccount();
    1. BankAccount 클래스의 참조 변수 park을 선언하고 BankAccount 클래스의 새로운 인스턴스를 생성하여 저장
    2. BankAccount park : BankAccount 클래스의 인스턴스를 참조하는 참조 변수 선언
    3. new BankAccount() : BankAccount 클래스의 새로운 인스턴스 생성
- 정의한 클래스로부터 여러 개의 인스턴스(객체) 생성 가능
- "객체명.메소드명()"으로 객체의 메소드 호출

## 클래스 - 정의 및 인스턴스 생성

- 참조 변수의 특성

```
public class ReferenceVariable {  
    public static void main(String[] args) {  
        BankAccount park = new BankAccount();  
        BankAccount yoon = new BankAccount();  
  
        park.deposit(10000);  
        yoon.deposit(7000);  
  
        System.out.println("현재 잔액 : " + yoon.getBalance() + "원");  
  
        yoon = new BankAccount();    // 새로운 인스턴스 참조  
        System.out.println("현재 잔액 : " + yoon.getBalance() + "원");  
  
        yoon = park;    // park이 참조하던 인스턴스 참조  
        System.out.println("현재 잔액 : " + yoon.getBalance() + "원");  
    }  
}
```

## 클래스 - 정의 및 인스턴스 생성

- 참조 변수의 매개변수 전달

```
public class ReferenceVariableAsParameter {  
    public static void main(String[] args) {  
        BankAccount ref = new BankAccount();  
        ref.deposit(3000);  
        ref.withdraw(300);  
        viewRA(ref);  
    }  
    public static void viewRV(BankAccount acc) {  
        System.out.println(acc.getBalance());  
    }  
}
```

- 참조 변수에 null 대입

```
public class ReferenceVariableNull {  
    public static void main(String[] args) {  
        BankAccount yoon = new BankAccount();  
        yoon.deposit(3000);  
        System.out.println(yoon.getBalance());  
        yoon = null;  
        System.out.println(yoon.getBalance());  
    }  
}
```

## 클래스 - 정의 및 인스턴스 생성

1. 아래 지시 사항에 맞추어 Human, Solution1 클래스를 정의하시오.

- Human 클래스

- 정수형 멤버 변수 tall = 180;
- 정수형 멤버 변수 weight = 75;
- 문자열 멤버 변수 faceType = "옥동자";
- 정수형 멤버 변수 money = 0;
- 정수형 멤버 변수 stress = 0;
- "일을 하다" 메소드
  - 만약 스트레스가 300 이상이면 "에이 귀찮아~일 안해!" 메시지 출력
  - 아니라면 일을 하고 있다는 메시지 출력 후 money, stress를 각각 1000, 100 증가
- "우유를 마시다" 메소드
  - 만약 돈이 없다면 일을 하라는 메시지 출력
  - 아니라면 우유를 마시고 있다는 메시지 출력 후 money, stress를 각각 1000, 100 감소
- 각각의 멤버 변수의 getter / setter



## 클래스 - 정의 및 인스턴스 생성

- Solution1 클래스

- main 메소드

- Human 클래스의 참조 변수 wonBin을 선언하고 Human 클래스의 인스턴스를 생성하여 저장
    - Human 클래스의 참조 변수 jangDongGun을 선언하고 Human 클래스의 인스턴스를 생성하여 저장
    - wonBin, jangDongGun의 현재 money와 stress 출력
    - 인스턴스 wonBin, jangDongGun의 "일하다" 메소드 호출
    - wonBin, jangDongGun의 현재 money와 stress 출력
    - 인스턴스 wonBin, jangDongGun의 "우유를마시다" 메소드 호출
    - wonBin, jangDongGun의 현재 money와 stress 출력
    - 인스턴스 wonBin, jangDongGun의 "우유를마시다" 메소드 호출