

# 자바 입문 과제\_4

박은종



# 고객 클래스 만들기

회사에서 고객 관리 프로그램을 만들고자 합니다. 고객은 아이디, 이름, 고객 등급이 있습니다. 고객이 물건을 구매하게 되면 보너스 포인트를 적립해 줍니다.

고객의 등급에서는 일반고객과 VIP고객이 있습니다. 일반고객의 경우에는 구매금액의 0.01%를 보너스 포인트로 적립해주고, VIP 고객의 경우에는 0.05%를 적립해줍니다. 담당 전문 상담원이 있습니다.

일반 고객은 구매 제품에 대한 할인이 없지만, VIP고객은 구매 제품의 10%의 가격을 할인해줍니다.

일반 고객 1명과 VIP 고객 1명이 있습니다. 일반 고객의 이름은 이순신, 아이디는 10010입니다. 이순신 고객은 현재 보너스 포인트 1000 점이 있습니다.

VIP고객의 이름은 김유신 아이디는 10020입니다. 김유신 고객은 10000점 보너스 포인트를 가지고 있습니다.

두 고객의 정보를 다음과 같이 출력해보세요



# 고객 클래스 만들기

```
public class CustomerTest {  
    public static void main(String[] args) {  
        Customer customerLee = new Customer(10010, "이순신");  
        customerLee.bonusPoint = 1000;  
        System.out.println(customerLee.showCustomerInfo());  
  
        VIPCustomer customerKim = new VIPCustomer(10020, "김유신");  
        customerKim.bonusPoint = 10000;  
        System.out.println(customerKim.showCustomerInfo());  
    }  
}
```

```
이순신님의 등급은 SILVER이며, 보너스 포인트는 1000입니다  
김유신님의 등급은 VIP이며, 보너스 포인트는 10000입니다
```

#상속 #하위클래스의생성순서 #상위클래스생성자호출 #super#protected



# 물건 구매

이순신 고객, 김유신 고객, 그리고 또 한 명의 VIP 나몰라 고객이 10000 원짜리 물건을 다음과 같이 물건을 구매했습니다.

```
public class CustomerTest {  
    public static void main(String[] args) {  
        Customer customerLee = new Customer(10010, "이순신");  
        customerLee.bonusPoint = 1000;  
        System.out.println(customerLee.showCustomerInfo());  
  
        VIPCustomer customerKim = new VIPCustomer(10020, "김유신");  
        customerKim.bonusPoint = 10000;  
        System.out.println(customerKim.showCustomerInfo());  
  
        int priceLee = customerLee.calcPrice(10000);  
        int priceKim = customerKim.calcPrice(10000);  
  
        System.out.println(customerLee.showCustomerInfo() + " 지불금액은 " + priceLee + "원 입니다.");  
        System.out.println(customerKim.showCustomerInfo() + " 지불금액은 " + priceKim + "원 입니다.");  
  
        Customer customerNo = new VIPCustomer(10030, "나몰라");  
        customerNo.bonusPoint = 10000;  
        int priceNo = customerNo.calcPrice(10000);  
        System.out.println(customerNo.showCustomerInfo() + " 지불금액은 " + priceNo + "원 입니다.");  
    }  
}
```

```
이순신님의 등급은 SILVER이며, 보너스 포인트는 1000입니다  
김유신님의 등급은 VIP이며, 보너스 포인트는 10000입니다  
이순신님의 등급은 SILVER이며, 보너스 포인트는 1100입니다 지불금액은 10000원 입니다.  
김유신님의 등급은 VIP이며, 보너스 포인트는 10500입니다 지불금액은 9000원 입니다.  
나몰라님의 등급은 VIP이며, 보너스 포인트는 10500입니다 지불금액은 9000원 입니다.
```

#overriding #업캐스팅(upcasting) #묵시적형변환 #상속에서의상위클래스와하위클래스의메모리상태 #가상메서드 #다형성(polymorphism)



# 다형성의 예

```
public class AnimalTest {  
    public static void main(String[] args) {  
        Animal hAnimal = new Human();  
        Animal tAnimal = new Tiger();  
        Animal eAnimal = new Eagle();  
  
        AnimalTest test = new AnimalTest();  
        test.moveAnimal(hAnimal);  
        test.moveAnimal(tAnimal);  
        test.moveAnimal(eAnimal);  
  
        ArrayList<Animal> animalList = new ArrayList<Animal>();  
        animalList.add(hAnimal);  
        animalList.add(tAnimal);  
        animalList.add(eAnimal);  
  
        for (Animal animal : animalList) {  
            animal.move();  
        }  
    }  
  
    public void moveAnimal(Animal animal) {  
        animal.move();  
    }  
}
```

사람이 두발로 걷습니다.  
호랑이가 네 발로 뿔니다.  
독수리가 하늘을 날아갑니다.  
사람이 두발로 걷습니다.  
호랑이가 네 발로 뿔니다.  
독수리가 하늘을 날아갑니다.



# 일반 고객과 VIP고객 중간 등급 만들기

고객이 늘어서 VIP 고객만큼 충성도가 높지 않지만, 단골고객에 대한 관리가 필요합니다. GOLD 고객등급을 하나 추가하도록 합니다.

- 제품을 살때는 항상 10%를 할인해줍니다.
- 보너스 포인트는 2% 적립해줍니다.



# ArrayList 활용하기

이 회사의 고객은 총 5명입니다. VIP 1명, GOLD 2명, SILVER 2명입니다. 각각 10000 원짜리 물건을 구입했을때의 결과를 다음과 같이 출력하세요

```
===== 고객 정보 출력 =====
이순신 님의 등급은 SILVER이며, 보너스 포인트는 0입니다.
신사임당 님의 등급은 SILVER이며, 보너스 포인트는 0입니다.
홍길동 님의 등급은 GOLD이며, 보너스 포인트는 0입니다.
이율곡 님의 등급은 GOLD이며, 보너스 포인트는 0입니다.
김유신 님의 등급은 VIP이며, 보너스 포인트는 0입니다. 담당 상담원 번호는 12345입니다
===== 할인율과 보너스 포인트 계산 =====
이순신 님이 10000원 지불하셨습니다.
이순신 님의 현재 보너스 포인트는 100점입니다.
신사임당 님이 10000원 지불하셨습니다.
신사임당 님의 현재 보너스 포인트는 100점입니다.
홍길동 님이 9000원 지불하셨습니다.
홍길동 님의 현재 보너스 포인트는 200점입니다.
이율곡 님이 9000원 지불하셨습니다.
이율곡 님의 현재 보너스 포인트는 200점입니다.
김유신 님이 9000원 지불하셨습니다.
김유신 님의 현재 보너스 포인트는 500점입니다.
```



# 다운 캐스팅

다음 테스트 프로그램이 수행되도록 testCasting() 을 구현하세요

```
public class AnimalTest {  
    public static void main(String[] args) {  
        Animal hAnimal = new Human();  
        Animal tAnimal = new Tiger();  
        Animal eAnimal = new Eagle();  
  
        ArrayList<Animal> animalList = new ArrayList<Animal>();  
        animalList.add(hAnimal);  
        animalList.add(tAnimal);  
        animalList.add(eAnimal);  
  
        AnimalTest test = new AnimalTest();  
        test.testDownCasting(animalList);  
    }  
}
```

사람이 책을 읽습니다.  
호랑이가 사냥을 합니다.  
독수리가 날개를 쭉 펴고 멀리 날아갑니다

#다운캐스팅 #instanceof #명시적형변환



# Think about...

상속을 언제 사용할 것인가?

상속이 코드의 재사용을 위한 일반적인 방법이 아니라고 하면 코드를 재사용하기 위한 방법은 무엇이 있을까?

IS-A 와 HAS-A 관계

하위 클래스가 상위 클래스 형으로 묵시적 형변환이 되는 이유를 메모리로 설명하세요



# 다음 코드를 상속을 활용하여 리팩토링 해보세요

```
public class Player {

    public static final int BEGINNER_LEVEL = 1;
    public static final int ADVANCED_LEVEL = 2;
    public static final int SUPER_LEVEL = 3;

    int level;

    public Player() {
        level = BEGINNER_LEVEL;
    }

    public void jump() {

        if(level == BEGINNER_LEVEL) {
            System.out.println("Jump 할 줄 모르지롱.");
        }
        else if(level == ADVANCED_LEVEL) {
            System.out.println("높이 jump 합니다.");
        }
        else if(level == SUPER_LEVEL) {
            System.out.println("아주 높이 jump 합니다.");
        }
    }

    public void run() {
        if(level == BEGINNER_LEVEL) {
            System.out.println("천천히 달립니다.");
        }
        else if(level == ADVANCED_LEVEL) {
            System.out.println("빨리 달립니다.");
        }
        else if(level == SUPER_LEVEL) {
            System.out.println("엄청 빨리 달립니다.");
        }
    }

    public void turn() {
        if(level == BEGINNER_LEVEL) {
            System.out.println("Turn 할 줄 모르지롱.");
        }
        else if(level == ADVANCED_LEVEL) {
            System.out.println("Turn 할 줄 모르지롱.");
        }
        else if(level == SUPER_LEVEL) {
            System.out.println("한 바퀴 돌니다.");
        }
    }

    public void play(int time) {

        run();
        for(int i =0; i<time; i++) {
            jump();
        }
        turn();
    }

    public void upgradeLevel(int level) {
        this.level = level;
    }
}
```



```
public class MainBoard {  
  
    public static void main(String[] args) {  
  
        Player player = new Player();  
        player.play(1);  
        player.upgradeLevel(Player.ADVANCED_LEVEL);  
        player.play(2);  
        player.upgradeLevel(Player.SUPER_LEVEL);  
        player.play(3);  
    }  
}
```