

문제 1. 다음과 같은 설명을 읽고 Account 클래스를 작성하십시오.

(참조파일 : TestAccount.java)

계좌(Account) 클래스는 계좌 번호를 나타내는 accNo 데이터와 잔고를 나타내는 balance 데이터를 가지고 있습니다.

함수로써는 입금을 의미하는 save 함수와 출금을 의미하는 deposit 함수 및 잔고를 확인하는 getBalance 함수와 계좌 번호를 확인하는 getAccNo 함수를 가지고 있습니다. TestAccount 클래스를 이용하여 Account 클래스의 함수를 호출시켰을 때 콘솔에 출력되는 결과는 다음과 같습니다.

**[TestAccount 실행결과]**

078-3762-293 계좌가 개설되었습니다.

078-3762-293 계좌의 잔고는 0만원입니다.

078-3762-293 계좌에 100만원이 입금되었습니다.

078-3762-293 계좌의 잔고는 100만원입니다.

078-3762-293 계좌에 30만원이 출금되었습니다.

078-3762-293 계좌의 잔고는 70만원입니다.

문제 2. 다음은 FlowerTest 의 소스와 실행 결과입니다. [실행 결과]와 같은 결과가 나오도록 조건에 맞게 Flower 객체를 생성하시오. (참조파일 : FlowerTest.java)

[소스]

```
public class FlowerTest {
    public static void main(String[] args) {
        Flower flower = new Flower("장미", "빨간색", false);
        printInfo(flower);

        flower = new Flower("해바라기", "노란색");
        printInfo(flower);

        flower = new Flower("패랭이꽃");
        printInfo(flower);
    }

    static public void printInfo(Flower flower){
        System.out.println("이름 : "+flower.getName() );
        System.out.println("색깔 : "+flower.getColor() );
        System.out.println("야생 : "+flower.isWild()+"\n" );
    }
}
```

[실행 결과]

```
이름 : 장미
색깔 : 빨간색
야생 : false

이름 : 해바라기
색깔 : 노란색
야생 : true

이름 : 패랭이꽃
색깔 : 알지못함
야생 : true
```

[Flower 객체 생성 조건]

1. 멤버 변수 String name, String color, boolean wild 를 갖는다.

2. 모든 멤버 변수의 접근자는 private으로 한다.
3. FlowerTest 의 main()메서드가 정상적으로 실행될 수 있도록 생성자를 오버로딩한다.
4. 조건에 주어지지 않는 값은 [실행 결과]를 보고 처리한다.

문제 3. 다음과 같은 설명을 읽고 Duck과 Sparrow 클래스를 작성하십시오.

(참조파일 : PlayBird.java)

오리(Duck) 클래스와 참새(Sparrow) 클래스는 이름을 나타내는 name과 다리의 수를 나타내는 legs, 몸길이를 나타내는 length 데이터를 가지고 있다.  
메소드로는 날다 fly, 울다 sing, 이름기록하기 setName, 문자열출력하기 toString 를 가지고 있다.  
PlayBird 클래스를 이용하여 두 클래스의 메소드를 호출시켰을 때 콘솔에 출력되는 결과는 다음과 같다.

[PlayBird 실행결과]

오리(꺅꺅이)는 날지 않습니다.  
오리(꺅꺅이)가 소리내어 읊니다.  
오리의 이름은 꺅꺅이 입니다.  
참새(쩍쩍)가 날아다닙니다.  
참새(쩍쩍)가 소리내어 읊니다.  
참새의 이름은 찻쩍 입니다.

문제 4. 아래 클래스 설명을 만족하고, 배포되는 CellPhoneMain의 실행결과가 나오도록 CellPhone 클래스를 작성하세요. (참조파일 : CellPhoneMain.java)

CellPhone	
String model	// 핸드폰 모델 번호
double battery;	// 남은 배터리 양
CellPhone(String model)	// 모델 번호를 인자로 받는 생성자
void call(int time)	// 통화 시간(분)을 출력하고, 통화 시간에 따라 배터리 양을 감소시킨다. // 감소되는 배터리 양 = 통화시간(분) * 0.5 // 배터리 양은 0보다 작아지지 않는다.

	// 통화 시간(분)이 0보다 작은 경우에는 ("통화시간입력오류")를 출력한다.
<code>void charge(int time)</code>	// 충전한 시간(분)을 출력하고, 충전한 시간에 따라 배터리 양을 증가시킨다.
	// 충전되는 배터리 양 = 충전시간(분) * 3
	// 배터리 양은 100까지만 증가한다.
	// 충전 시간(분)이 0보다 작은 경우에는 ("충전시간입력오류")를 출력한다.
<code>void printBattery()</code>	// 남은 배터리 양을 출력한다.
<code>boolean isEqual(ChlIPhone c)</code>	//인자로 전달된 CellPhone의 모델 번호가 현재 객 체의모델 번호와 같은 경우에 true를 리턴한다.

### <<CellPhoneMain 실행결과>>

```

충전 시간 : 20분
남은 배터리 양 : 60.0
통화 시간 : 300분
남은 배터리 양 : 0.0
충전 시간 : 50분
남은 배터리 양 : 100.0
통화 시간 : 40분
남은 배터리 양 : 80.0
통화시간입력오류
동일 모델입니다.

```