예제 5-7 : 추상 클래스의 구현 연습

다음 추상 클래스 Calculator를 상속받은 GoodCalc 클래스를 구현하라.

```
abstract class Calculator {
        public abstract int add(int a, int b);
        public abstract int subtract(int a, int b);
        public abstract double average(int[] a);
}
```

예제 5-7 정답


```
public class GoodCalc extends Calculator {
   @Override
   public int add(int a, int b) { // 추상 메소드 구현
      return a + b;
   }
   @Override
   public int subtract(int a, int b) { // 추상 메소드 구현
      return a - b;
   }
   @Override
   public double average(int[] a) { // 추상 메소드 구현
      double sum = 0;
      for (int i = 0; i <a.length; i++)
         sum += a[i];
      return sum/a.length;
   }

   public static void main(String [] args) {
      GoodCalc c = new GoodCalc();
      System.out.println(c.add(2,3));
      System.out.println(c.subtract(2,3));
      System.out.println(c.average(new int [] { 2,3,4 }));
   }
}
```

```
5
-1
3.0
```

예제 5-8 인터페이스 구현

PhoneInterface 인터페이스를 구현하고 flash() 메소드를 추가한 SamsungPhone 클래스를 작성하라.

```
** Phone **
띠리리리링
전화가 왔습니다.
전화기에 불이 켜졌습니다.
```

```
interface PhoneInterface { // 인터페이스 선언
    final int TIMEOUT = 10000; // 상수 필드 선언
    void sendCall(); // 추상 메소드
    void receiveCall(); // 추상 메소드
    default void printLogo() { // default 메소드
        System.out.println("** Phone **");
    }
}
class SamsungPhone implements PhoneInterface { // 인터페이스 구현
    // PhoneInterface의 모든 메소드 구현
    @Override
    public void sendCall() {
        System.out.println("띠리리리링");
    }
    @Override
    public void receiveCall() {
        System.out.println("전화가 왔습니다.");
    }

    // 메소드 추가 작성
    public void flash() { System.out.println("전화기에 불이 켜졌습니다."); }
}
public class InterfaceEx {
    public static void main(String[] args) {
        SamsungPhone phone = new SamsungPhone();
        phone.printLogo();
        phone.sendCall();
        phone.receiveCall();
        phone.flash();
    }
}
```

예제 5-9 : 인터페이스를 구현하고 동시에 클래스를 상속받는 사례


```
interface PhoneInterface { // 인터페이스 선언
    final int TIMEOUT = 10000; // 상수 필드 선언
    void sendCall(); // 추상 메소드
    void receiveCall(); // 추상 메소드
    default void printLogo() { // default 메소드
        System.out.println("** Phone **");
    }
}
interface MobilePhoneInterface extends PhoneInterface {
    void sendSMS();
    void receiveSMS();
}
interface MP3Interface { // 인터페이스 선언
    public void play();
    public void stop();
}
class PDA { // 클래스 작성
    public int calculate(int x, int y) {
        return x + y;
    }
}
// SmartPhone 클래스는 PDA를 상속받고,
// MobilePhoneInterface와 MP3Interface 인터페이스에 선언된 추상 메소드를 모두 구현한다.
class SmartPhone extends PDA implements
MobilePhoneInterface, MP3Interface {
    // MobilePhoneInterface의 추상 메소드 구현
    @Override
    public void sendCall() {
        System.out.println("따르릉따르릉~~");
    }
    @Override
    public void receiveCall() {
        System.out.println("전화 왔어요.");
    }
    @Override
    public void sendSMS() {
        System.out.println("문자갑니다.");
    }
    @Override
    public void receiveSMS() {
        System.out.println("문자왔어요.");
    }
    // MP3Interface의 추상 메소드 구현
    @Override
    public void play() {
        System.out.println("음악 연주합니다.");
    }
    @Override
    public void stop() {
        System.out.println("음악 중단합니다.");
    }
    // 추가로 작성한 메소드
    public void schedule() {
        System.out.println("일정 관리합니다.");
    }
}
public class InterfaceEx {
    public static void main(String [] args) {
    SmartPhone phone = new SmartPhone();
    phone.printLogo();
    phone.sendCall();
    phone.play();
    System.out.println("3과 5를 더하면 " +
        phone.calculate(3,5));
    phone.schedule();
    }
}
```

```
** Phone **
따르릉따르릉~~
음악 연주합니다.
3과 5를 더하면 8
일정 관리합니다.
```