**Practice03 프로젝트를 다운받아 아래의 프로그램을 완성하세요**

**[문제] com.javaex.problem01 패키지**

다음 설명을 읽고 Member 클래스를 작성하세요.

**회원(Member) 클래스는 회원 아이디를 나타내는 id(문자열)와 회원 이름을 나타내는 name(문자열), 회원의 point를 나타내는 point(정수) 필드를 가지고 있다.**

**메소드는 회원의 이름 변경 setName메소드와 이름을 반환하는 getName, 포인트를 변경하는 setPoint와 포인트를 반환하는 getPoint, 회원아이디를 변경하는 setId, 반환하는 getId 메소드를 가지고 있다.**

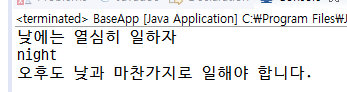
**모든 필드는 외부에서 직접 접근하여 값을 변경하거나 얻어가지 못하도록 private로 접근 제한을 막고, 메소드는 제한 없이 모두 호출할 수 있는 접근제한자를 사용한다.**

**[문제] com.javaex.problem02 패키지**

다음과 같이 2가지 클래스가 제공된다.

BaseApp클래스를 실행했을 때 아래와 같이 나올 수 있도록

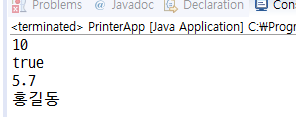
Base 클래스를 수정하세요.



**[문제] com.javaex.problem03 패키지**

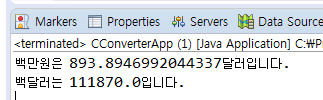
PrintApp 클래스에서 Print 인트턴스를 생성하고 print() 메소드를 호출해서 파라미터 값을 출력하려고 합니다.

출력 결과에 맞게 Print 클래스를 구현하세요.

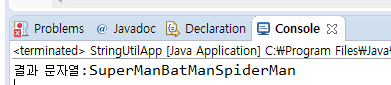


**[문제] com.javaex.problem04 패키지**

Static 필드와 메소드를 이용하여 달러와 우리나라 원화 사이의 변환을 해주는 환율 계산기를 만들어보세요



**[문제] com.javaex.problem05 패키지**

**아래와 같이 출력이 나오도록 StringUtil 클래스를 완성하세요**

1 String 배열 타입의 파라미터를 받습니다.

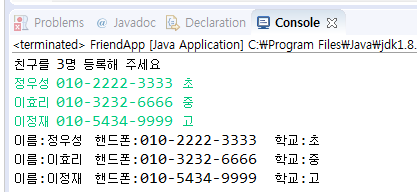
2 인자로 받은 문자열 배열 요소를 하나의 문자열로 결합하고, 결과 문자열을 리턴 합니다.

3. 주어진 클래스명과 메소드는 변경하지 않습니다.

**[문제] com.javaex.problem06 패키지**

java.util.Scanner 클래스를 이용하여 친구정보를 입력받아 Friend 객체를 생성하고 이들을 Friend 객체 배열에 저장하세요.

친구 즉, 친구(Friend)객체를 3개 입력 받으면 이들을 모두 화면에 출력하세요.



**[문제] com.javaex.problem07 패키지**

**아래와 같이 은행 프로그램을 작성하세요**

“1.예금” 선택후 금액을 입력하면 예금액이 합산됩니다.

“2.출급” 선택후 금액을 입력하면 예금액이 차감됩니다.

“3.잔고” 선택시 현재 잔고가 출력됩니다.

“4.종료” 선택시 종료됩니다.

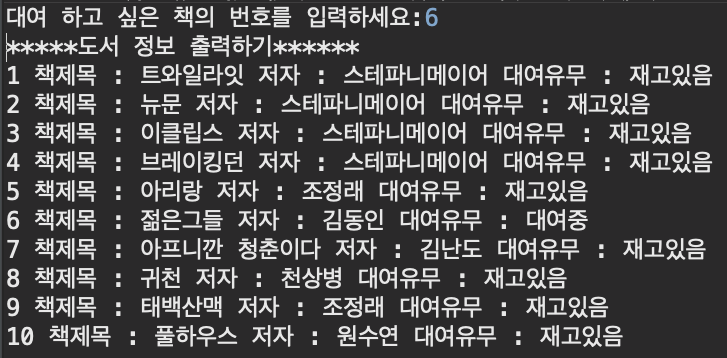
“1,2,3,4 이외의 숫자” 다시입력해주세요 메시지 출력됩니다.



**[문제] com.javaex.problem08 패키지**

**주어진 BookShop.java 를 실행했을 때 다음과 같은 실행결과가 나타나도록 아래의 문제설명을 참고하여 Book.java 를 작성하고 BookShop.java 를 완성하세요.**

**실행결과**

****

**문제설명**

다음과 같은 조건을 만족하도록 작성합니다.

|  |
| --- |
| **Book 클래스** |
| int bookNo  String title  String author  int stateCode |
| Book(int bookNo,String title,String author)  rent() : void  print() : void |

1. 멤버변수는 다음과 같으며 모든 멤버 변수는 초기값을 지정하지 않습니다.

int bookNo: 번호

String title: 제목

String author: 작가

int stateCode: 상태코드(대여유무를 나타내는 상태코드)

1. 생성자는 BookShop의 main메소드를 참고하여 각 멤버변수를 초기화할 수 있도록 작업합니다. 단, stateCode는 생성자 호출 시에 1로 세팅 되도록 설정합니다.
2. 각각의 메소드는 아래의 기능을 구현하도록 작성합니다.

* 모든 메소드는 public으로 작성합니다.
* bookNo, title, author 에 대해서는 모든 setter/getter메소드를 추가하세요.
* rent() : 대여기능을 수행하는 메소드로   
  Book클래스의 stateCode를 0으로 변경하며 “\_\_\_\_\_\_\_이(가) 대여 됐습니다.”메시지가 출력되도록 작성합니다.이때 \_\_\_\_\_\_는 책제목이 출력되도록 작성합니다.
* print(): 책의 정보가 출력형태와 같이 출력될 수 있도록 작성합니다. 이때 대여 유무는 Book클래스의 stateCode가 1이면 “재고있음”으로 0이면 “대여중”으로 출력되도록 설정합니다.

BookShop클래스

주어진 소스에 표시된 (1)과 (2)에 해당하는 내용을 아래의 조건에 맞게 작성합니다.

1. (1)의 부분은 BookShop.java에서 입력받은 책 번호에 해당 책을 배열에서 찾아 rent메소드가 호출되도록 작성합니다.
2. (2)의 부분은 메소드의 매개변수로 전달된 배열에 저장된 모든 Book 객체의 정보를 출력할 수 있도록 작성합니다.

**<<주의사항>>**

1. 주어진 BookShop.java는 수정하지 않습니다.
2. 멤버변수에 초기값을 지정하지 마십시오
3. Book클래스의 rent메소드에서 stateCode를 변경해야 합나다.
4. Book클래스의 print메소드의 실행결과가 주어진 조건과 맞게 출력하도록 하세요