

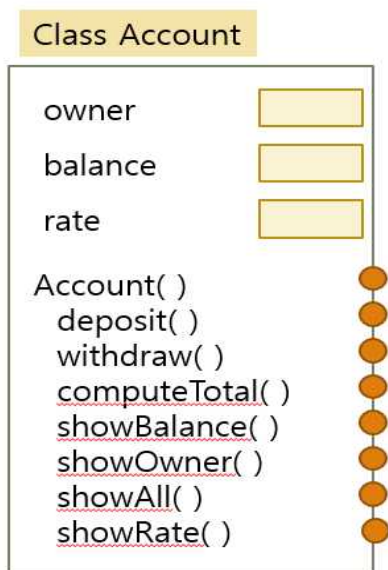
객체지향 프로그래밍 및 실습 : 과제 #2

사용자의 은행 계정을 관리해 주는 프로그램을 작성하려고 한다. 프로그램은 사용자 1명의 정보를 담고 있는 Account 클래스와 아래 제시되어 있는 AccountEx 클래스로 구성된다. AccountEx 클래스의 구성과 실행 결과를 참고로 하여, **Account** 클래스와 **AccountEx** 클래스를 작성하여 제출하시오. 과제의 프로그램 구성과 실행예제를 참조하여 프로그램을 작성합니다.

과제 제출은 **작성한 클래스 프로그램 소스**와 **실행 결과를 캡처한 파일**을 제출해야 합니다.

프로그램 구성

#2-1. 은행 계정 클래스 프로그램



[그림-1] Account 의 클래스의 구성

프로그램 구성에 대한 참고사항

1. Account 클래스에는 사용자의 이름(Owner)을 저장하는 String 변수가 있고, 은행 잔고(balance)를 저장할 int변수와 은행의 이율(rate연이율)을 저장하는 double형 변수가 있다.
2. 사용자 1명의 객체는 (1) 사용자 이름과 예금액, 연이율, (2)사용자 이름과 연이율, (3)사용자 이름과 예금액으로도 생성할 수도 있다.
3. 클래스 내부에 this, this()메소드를 사용하여 처리한다.
4. deposit() 메소드는 입금액을 처리하는 기능을 수행하고 withdraw() 메소드는 출금액을 처리하는 기능을 수행한다. 또한 computeTotal()메소드는 이자계산액을 처리하는 기능을 수행한다.
5. showBalance() 메소드는 잔액을 출력하고, showOwner() 메소드는 사용자 계정을 출력하고, showRate() 메소드는 은행이율(연이율)을 출력하고, showAll() 메소드는 사용자 계정과 잔액, 연이율을 모두를 출력하는 기능을 수행한다.

#2-2. 은행 계정 실행 프로그램

프로그램 실행 예제

```
public class AccountEx {  
    public static void main(String[] args){  
  
        Account a = new Account("Kim", 1000, 0.03);  
        Account b = new Account("Lee", 0.04);  
        Account c = new Account("Park", 2000);  
  
        a.showAll();  
        a.deposit(300);  
        a.withdraw(100);  
        a.computeTotal(5);  
        a.showOwner();  
        a.showBalance();  
  
        b.showAll();  
        b.deposit(300);  
        b.withdraw(100);  
        b.computeTotal(5);  
        b.showOwner();  
        b.showBalance();  
  
        c.rate(0.05);  
        c.showAll();  
        c.deposit(300);  
        c.withdraw(100);  
        c.computeTotal(5);  
        c.showOwner();  
        c.showBalance();  
  
    }  
}
```

실행 결과

```
Owner = Kim  
Balance = 1000  
Rate = 0.03  
  
Owner = Kim  
Balance = 1215  
  
Owner = Lee  
Balance = 0  
Rate = 0.04  
  
Owner = Lee  
Balance = 203  
  
Owner = Park  
Balance = 2000  
Rate = 0.05  
  
Owner = Park  
Balance = 2245
```

#2-3. 은행 계정 배열 클래스 프로그램

AccountEx는 계정 3개를 독립된 변수 a, b, c 에 저장하여 관리한다. 배열을 사용하여 계정들을 저장하고 관리해 보는 프로그램을 작성하시오. 앞에서 작성한 Account 클래스는 그대로 사용하고 AccountEx 클래스를 수정하여 **AccountModEx** 클래스를 작성하시오.

실행 결과

```
Owner = Kim  
Balance = 1000  
Rate = 0.03  
  
Owner = Kim  
Balance = 1215  
  
Owner = Lee  
Balance = 0  
Rate = 0.04  
  
Owner = Lee  
Balance = 203  
  
Owner = Park  
Balance = 2000  
Rate = 0.05  
  
Owner = Park  
Balance = 2245
```

#2-4. 개인 클래스 작성하기

자신이 구축하고 싶은 대상을 클래스로 작성해 봅니다. 클래스 콘텐츠 및 내용은 종류제한 없이 모두 가능합니다. (예시:자동차,도서관리,주유관리,성적관리,병원관리,사람정보,학교정보,회사정보 등) 필드 선언,메소드 생성,기본 생성자, 매개변수를 받는 생성자, 메소드 오버로딩을 보이는 생성자, 생성자를 호출하는 this()와 자기자신을 가리키는 참조변수인 this도 사용되어야 하며, getter 메소드와 setter 메소드, 최종 캡슐화를 진행되는 프로그램 구현으로 본인의 클래스를 구축합니다.

1. 프로그램 내부에 주석을 달아 기본 생성자, 메소드 오버로딩 및 프로그램 로직을 설명하는 내용을 표시해야 합니다.
2. 반드시 라이브러리 클래스와 실행 클래스로 나누어 작성하여야 합니다.
3. 라이브러리 클래스에는 필드 선언, 메소드 2개 이상 작성, 기본 생성자, 추가 생성자를 포함하여 메소드 오버로딩을 보여야만 합니다.
4. 실행 클래스에는 라이브러리 클래스의 내용을 이용하는 객체 생성 및 사용을 이용한 프로그램을 구현합니다.
5. 객체 생성시 생성자를 이용해서 초기화하는 코드도 작성되어야 합니다.
6. 클래스 구축시 클래스의 캡슐화가 진행되어야 합니다.
7. 클래스 제출은 라이브러리 클래스와 실행 클래스로 제출되어야 하며, 클래스 개수에 제한은 없습니다. 실행 클래스를 1개 이상 작성하거나 라이브러리 클래스를 여러개 작성하셔도 됩니다!

제출 안내

- 프로그램을 이해하기 쉽도록 주석을 반드시 포함합니다. (배점 10점)
주석에는 자신의 전공, 학번, 이름을 포함합니다. (//, /*~*/ 주석 이용)
- 각각의 클래스 프로그램은 hw2.txt 파일로 저장하고, 실행결과를 캡처한 결과화면도 파일로 함께 제출해야 합니다.
- 사이버 강의실의 (과제2) 제출함에 프로그램과 캡쳐파일을 제출합니다.
- 프로그램 소스과 캡쳐파일은 하나의 압축파일로 제출하거나 각각 제출해도 됩니다.
- 프로그램 제출 후에 다시 다운로드 받아서 제대로 제출했는지 반드시 확인해야 하며, 제출이 잘못된 경우는 0점 처리 될 수 있습니다.
- 제출 기한 : 5월 9일(일) 저녁 11:00 까지(늦은 제출 허용, 단 감점 있음)