KNU 강원대학교
KANGWON NATIONAL UNIVERSITY

자바프로그래밍2

4주차 실습

클래스와 메소드 심층 탐구

2024-2학기 자바프로그래밍2 3분반
TA 강지호

실습 평가 방법

- 점수 : 100점 만점 기준(문제 별 난이도에 따라 점수 부여)
- 채점 기준 : 완성도, 작동 유무, 일부 오류 등에 따라 감점
  - 프로그램이 동작하지 않거나, 코드 공유, ChatGpt 사용 등의 부정행위 적발 시 0점
  - 소스 코드에 허점, 잘못된 들여쓰기, 일부 입출력 오작동 시 정도에 따라 감점
- 제출 기한 : 실습 당일 23시 59분까지(이후 제출 불가능)
  - 실습 시간(14:00-15:50) 내 제출 시 감점X
  - 18:00 까지 제출 시 채점 점수의 5% 감점
  - 20:00 까지 제출 시 채점 점수의 10% 감점
  - 23:59 까지 제출 시 채점 점수의 20% 감점

실습 제출 방법

- 압축 파일명 : n주차_학번_이름.zip
- 소스 파일 : Eclipse에서 Export한 zip 파일 내 소스 파일(**.java**)
- 보고서 : 각 문제별 문제 번호 및 소스 코드 실행 결과 화면 캡처한 **pdf 파일** - 부재 시 감점
- 소스 파일과 보고서를 압축하여 주차별로 위 압축 파일명과 같이 e-루리에 제출
- e-루리 접속 오류 등 특별한 사유로 인해 제출하지 못하는 경우
  - rkdwlgh01@naver.com 해당 e-mail을 통해 제출

실습 조교 및 질의응답

- e-루리 Q&A 게시판 활용 (작성 후 e-루리 메시지 시 빠른 응답 가능)
- 실습 TA의 e-mail 활용
  - 강지호 : rkdwlgh01@naver.com

[자바프로그래밍2]
4주차 실습 문제

제출 기한 9.26(목) 23:59 전까지

문제 1 (20점)

- 클래스 **Pizza**를 아래의 조건에 맞게 구현하고 **Exercise01**에서 테스트하세요.

Pizza
- size : String - type : String - price : int - count : int
+ Pizza() + Pizza(size: String) + Pizza(size: String, type: String) + Pizza(size: String, type: String, price: int) + toString() : String + getCount() : int

Pizza 클래스

- 필드는 외부에서 접근할 수 없도록 설정
- **정적 변수** count는 총 생성된 피자 객체의 수 (따라서, 초기값은 0)
- 생성자 **오버로딩**을 통해 4개의 생성자 구현
- toString() 메소드를 **오버라이딩**하여 피자 객체의 상태를 출력
- getCount()는 **정적 메소드**로 정적 변수 count를 반환

문제 1 (20점)

- 클래스 명 : **Pizza, Exercise01**

Pizza.java
Exercise01.java

```
public class Exercise01 {

    public static void main(String[] args) {

        Pizza p0 = new Pizza();
        Pizza p1 = new Pizza("medium");
        Pizza p2 = new Pizza("large", "페퍼로니");
        Pizza p3 = new Pizza("small", "치즈", 8000);

        System.out.println(p0);
        System.out.println(p1);
        System.out.println(p2);
        System.out.println(p3);

        System.out.println("총 생성된 피자의 수: " + Pizza.getCount());

    }

}
```

실행결과 예시 :

```
Pizza [피자 종류: 기본, 사이즈: 기본, 가격: 10000]
Pizza [피자 종류: 기본, 사이즈: medium, 가격: 10000]
Pizza [피자 종류: 페퍼로니, 사이즈: large, 가격: 10000]
Pizza [피자 종류: 치즈, 사이즈: small, 가격: 8000]
지금까지 생성된 피자 수: 4
```

문제 2 (80점)

- 클래스 **Account**와 **BankSystem**을 조건에 맞게 구현하고 **Exercise02**에서 테스트하세요.

Account
- name : String - accountNumber : String - password : String - balance : int - count : int
+ **Account**(name: String, accountNumber: String, password: String, balance: int) + **deposit**(amount: int) : boolean + **withdraw**(amount: int) : boolean + **transfer**(other: Account, amount: int) : boolean + **getName**() : String + **getAccountNumber**() : String + **getPassword**() : String + **getBalance**() : int + **getCount**() : int

* Account 클래스 내에는 어떠한 입출력도 이뤄지지 않음

Account 클래스

- 필드는 외부에서 접근할 수 없도록 설정
- **정적 변수** count는 총 생성된 계좌 객체의 수 (초기값은 0)
- **생성자** 구현
- **입금**, **출금**, **이체** 동작을 하는 메소드 각각 구현
  - 이체는 매개변수로 **Account(객체)**를 받음
- 필드의 값을 반환하는 메소드인 **접근자** 각각 구현
- getCount()는 **정적 메소드**로 정적 변수 count를 반환

문제 2 (80점)

BankSystem
- accounts[] : Account //전체 계좌가 저장된 배열 - loggedIn : Account //로그인된 계좌
+ **BankSystem**(size: int) + **getAccount**(number: String) : Account + **createAccount**(name: String, number: String, pwd: String, balance: int) : void + **login**(nunber: String, pwd: String) : void + **process**(idx: int) : void

BankSystem 클래스

- 필드는 외부에서 접근할 수 없도록 설정

- **생성자** 구현(size는 배열의 크기, loggedIn은 null로 초기화)
- 객체 배열 내의 계좌번호(매개변수)를 갖는 Account(객체) 반환 (존재하지 않으면 null 반환)
- Account(객체) 생성(중복되는 계좌 번호가 이미 존재하면 개설X)
- Account(객체) 생성하여 배열에 저장
- idx에 맞는 업무(입금, 출금, 이체, 잔액조회)를 처리(로그인 확인)

문제 2 (80점)

➢ **BankSystem** 내 메소드 추가 설명

+ **createAccount(...)** : 계좌 개설 (중복되는 계좌 번호가 이미 존재하면 개설X)

개설 성공 →

```
이름: 강백호
계좌번호: 1001
비밀번호: 1234
금액: 5000
계좌가 정상적으로 개설되었습니다.
```

개설 실패 →

```
이름: 서태웅
계좌번호: 1001
비밀번호: 1111
금액: 10000
입력한 계좌번호는 이미 존재합니다.
```

+ **login(number: String, pwd: String)** : 계좌 로그인

로그인 성공 →

```
계좌번호를 입력하세요: 1001
비밀번호를 입력하세요: 1234
강백호님, 환영합니다!
```

로그인 실패 →

```
계좌번호를 입력하세요: 1001
비밀번호를 입력하세요: 0000
계좌번호 또는 비밀번호가 일치하지 않습니다.
```

문제 2 (80점)

+ **process(idx: int)** : 입력한 idx에 맞는 은행 업무(0-입금, 1-출금, 2-이체, 3-잔액조회) 처리

(로그인되어 있어야만 업무 처리)

로그인 X →

```
0: 입금    1: 출금    2: 이체    3: 잔액조회
업무를 선택하세요: 3
로그인 먼저 해주세요.
```

로그인 O →

입금

```
0: 입금    1: 출금    2: 이체    3: 잔액조회
업무를 선택하세요: 0
입금할 금액을 입력하세요: 1000
1000원 입금되었습니다. 잔액: 6000
```

```
0: 입금    1: 출금    2: 이체    3: 잔액조회
업무를 선택하세요: 0
입금할 금액을 입력하세요: -5000
잘못된 금액을 입력하였습니다.
```

출금

```
0: 입금    1: 출금    2: 이체    3: 잔액조회
업무를 선택하세요: 1
출금할 금액을 입력하세요: 1000
1000원이 출금되었습니다. 잔액: 5000
```

```
0: 입금    1: 출금    2: 이체    3: 잔액조회
업무를 선택하세요: 1
출금할 금액을 입력하세요: 50000
잔액이 부족하거나 잘못된 금액을 입력하였습니다.
```

이체

```
0: 입금    1: 출금    2: 이체    3: 잔액조회
업무를 선택하세요: 2
이체할 계좌번호를 입력하세요: 1002
이체할 금액을 입력하세요: 1500
서태웅님의 계좌로 1500원을 이체하였습니다.
```

```
0: 입금    1: 출금    2: 이체    3: 잔액조회
업무를 선택하세요: 2
이체할 계좌번호를 입력하세요: 1000
이체할 금액을 입력하세요: 1500
일치하는 계좌를 찾을 수 없어 이체를 실패하였습니다.
```

```
0: 입금    1: 출금    2: 이체    3: 잔액조회
업무를 선택하세요: 2
이체할 계좌번호를 입력하세요: 1001
이체할 금액을 입력하세요: 10000
잔액이 부족하거나 잘못된 금액을 입력하였습니다.
```

조회

```
0: 입금    1: 출금    2: 이체    3: 잔액조회
업무를 선택하세요: 3
잔액: 3500
```

문제 2 (80점)

```java
import java.util.Scanner;

public class Exercise02 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        BankSystem bank = new BankSystem(100);

        String number, name, pwd;
        int balance, idx;
        boolean flag = true;

        System.out.println("======menu======");
        System.out.println("0. 계좌 개설");
        System.out.println("1. 계좌 로그인(재로그인)");
        System.out.println("2. 입출금/이체/조회");
        System.out.println("3. 종료");
        System.out.println("================");

        while(flag) {
            System.out.print("메뉴를 선택하세요: ");
            int choice = input.nextInt();
            input.nextLine();   //Enter 처리
            switch (choice) {
                case 0:
                    System.out.print("이름: ");
                    name = input.nextLine();
                    System.out.print("계좌번호: ");
                    number = input.nextLine();
                    System.out.print("비밀번호: ");
                    pwd = input.nextLine();
                    System.out.print("금액: ");
                    balance = input.nextInt();
                    input.nextLine();  //Enter 처리
                    bank.createAccount(name, number, pwd, balance);
                    break;
                case 1:
                    System.out.print("계좌번호를 입력하세요: ");
                    number = input.nextLine();
                    System.out.print("비밀번호를 입력하세요: ");
                    pwd = input.nextLine();
                    bank.login(number, pwd);
                    break;
                case 2:
                    System.out.println("0: 입금    1: 출금    2: 이체    3: 잔액조회");
                    System.out.print("업무를 선택하세요: ");
                    idx = input.nextInt();
                    input.nextLine(); //Enter 처리
                    bank.process(idx);
                    break;
                case 3:
                    System.out.println("이용해주셔서 감사합니다.");
                    flag = false;
                    break;
            }
            System.out.println();
        }

        input.close();
    }

}
```

문제 2 (80점)

- 클래스 명 : **Account, BankSystem, Exercise02**

Account.java
BankSystem.java
Exercise02.java

실행결과 예시 :

```
======menu======
0. 계좌 개설
1. 계좌 로그인(재로그인)
2. 입출금/이체/조회
3. 종료
================
메뉴를 선택하세요: 0
이름: 강백호
계좌번호: 1001
비밀번호: 1234
금액: 5000
계좌가 정상적으로 개설되었습니다.

메뉴를 선택하세요: 0
이름: 서태웅
계좌번호: 1002
비밀번호: 1111
금액: 10000
계좌가 정상적으로 개설되었습니다.

메뉴를 선택하세요: 2
0: 입금    1: 출금    2: 이체    3: 잔액조회
업무를 선택하세요: 3
로그인 먼저 해주세요.

메뉴를 선택하세요: 1
계좌번호를 입력하세요: 1002
비밀번호를 입력하세요: 1111
서태웅님, 환영합니다!

메뉴를 선택하세요: 2
0: 입금    1: 출금    2: 이체    3: 잔액조회
업무를 선택하세요: 2
이체할 계좌번호를 입력하세요: 1001
이체할 금액을 입력하세요: 2000
강백호님의 계좌로 2000원을 이체하였습니다.

메뉴를 선택하세요: 2
0: 입금    1: 출금    2: 이체    3: 잔액조회
업무를 선택하세요: 3
잔액: 8000

메뉴를 선택하세요: 3
이용해주셔서 감사합니다.
```

```
======menu======
0. 계좌 개설
1. 계좌 로그인(재로그인)
2. 입출금/이체/조회
3. 종료
================
메뉴를 선택하세요: 0
이름: 강백호
계좌번호: 1001
비밀번호: 1234
금액: 5000
계좌가 정상적으로 개설되었습니다.

메뉴를 선택하세요: 0
이름: 송태섭
계좌번호: 1001
비밀번호: 1111
금액: 10000
입력한 계좌번호는 이미 존재합니다.

메뉴를 선택하세요: 1
계좌번호를 입력하세요: 1002
비밀번호를 입력하세요: 1111
계좌번호 또는 비밀번호가 일치하지 않습니다.

메뉴를 선택하세요: 1
계좌번호를 입력하세요: 1001
비밀번호를 입력하세요: 0000
계좌번호 또는 비밀번호가 일치하지 않습니다.

메뉴를 선택하세요: 3
이용해주셔서 감사합니다.
```