# 창의적 소프트웨어 설계 실습 문제 06 - hw06-1

**제출 기한**

- 10월 16일 화 16:59까지

1. hw6-1(mkdir hw6-1)라는 폴더를 만들고 GitLab에 push
2. hw6-1 디렉토리에 Makefile과 각 문제에서 요구하는 파일들을 작성
3. **‘make’명령을 수행하여 1~2 숙제가 모두 빌드**
4. 최종 버전을 GitLab에 commit
5. **시간과 파일명, 입력과 출력 방식 반드시 지키기**

## 과제1. 도형 클래스 구현

원과 직사각형의 클래스를 정의하고, 각 클래스는 넓이와 둘레를 구하는 멤버 함수를 갖는다

* 원은 중심과 반지름을 멤버변수로 갖는다
* 직사각형은 왼쪽 위와 오른쪽 아래를 맴버변수로 갖는다
* 각 도형의 넓이와 둘레를 구하는 멤버함수
* 입력은 항상 유효한 값이며, 모든 종류의 잘못된 입력(명령어 오타 또는 definition에 어긋나는 입력)에 대한 예외처리는 하지 않는다

파일명 : simple\_shape (simple\_shape.h, simple\_shape.cc, simple\_shape\_main.cc)

입력 :

* 입력은 두가지 형태가 가능하다.
* 원은 4개의 입력 (C, 중심의 x 좌표, 중심의 y좌표, 반지름)을 제공한다.
* 직사각형은 5개의 입력 (R, 왼쪽 위 꼭지점의 x 좌표, 왼쪽 위 꼭지점의 y좌표, 오른쪽 아래 꼭지점의 x 좌표, 오른쪽 아래 꼭지점의 y 좌표)를 제공한다.

출력 :

|  |
| --- |
| **$ ./simple\_shape**  **shape?**  **C 1 1 1**  **area: 3.14, perimeter: 6.28**  **shape?**  **R 1 5 5 1**  **area: 16, perimeter: 16**  **shape?**  **Q**  **$** |

## 과제 2. 계좌 클래스 구현

* 계좌 클래스는 계좌 주인의 ID와 잔고를 멤버변수로 갖는다
* 계좌 관리자 클래스는 계좌 클래스와 관리하는 계좌 개수를 멤버변수로 갖는다
* 계좌 관리자는 입금, 출금, 이체, 잔고확인 함수를 갖는다
* 계좌가 보유할 수 있는 최대 금액은 100만원, 최저 금액은 0원이며, 1회 입금/출금/이체 한도 금액은 없다
* 한도 초과로 입금/출금/이체가 불가능한 경우의 예외처리 필요(이체는 양쪽에서 확인)
* 각 작업 이후에는 잔고를 표기한다
* 처음 생성된 계좌의 잔고는 반드시 0원이다
* 계좌는 최대 10개까지 생성 가능하며, ID는 계좌 생성 순서대로 0~9가 할당된다
* 10개 초과 계좌 생성에 대한 예외처리 필요
* 명령어 오타, type error 등은 예외처리하지 않는다.

파일명: simple\_account (simple\_account.h, simple\_account.cc, simple\_account\_main.cc)

입력 :

* 입력은 네가지 형태가 가능하다.
* 계좌 생성을 위해서는 1개의 입력 (N)을 제공한다.
* 입금을 위해서는3개의 입력 (D, 계좌ID, 입금금액)을 제공한다
* 출금을 위해서는 3개의 입력 (W, 계좌ID, 입금금액)을 제공한다
* 이체를 위해서는 4개의 입력 (T, 이체하려는 계좌 ID (from), 이체 대상이 되는 계좌 ID (to), 금액)을 제공한다.
* 마치려면 Q를 입력한다.

출력 :

|  |
| --- |
| **$ ./simple\_account**  **Job?**  **D 0 50000**  **Account does not exist**  **Job?**  **N**  **Account for user 0 registered**  **Balance of user 0: 0**  **Job?**  **D 0 50000**  **Success: Deposit to user 0 50000**  **Balance of user 0: 50000**  **Job?**  **W 0 100000**  **Failure: Withdraw from user 0 100000**  **Balance of user 0: 50000**  **Job?**  **N**  **Account for user 1 registered**  **Balance of user 1: 0**  **Job?**  **T 1 0 30000**  **Failure: Transfer from user 1 to user 0 30000**  **Balance of user 0: 50000**  **Balance of user 1: 0**  **Job?**  **T 0 1 30000**  **Success: Transfer from user 0 to user 1 30000**  **Balance of user 0: 20000**  **Balance of user 1: 30000**  **Job?**  **Q**  **$** |