Lab 5. Chapter7

실습 1> 인터페이스 실습

- 다음은 도형의 구성을 묘사하는 인터페이스이다.

```
interface Shape {
    final double PI = 3.14;
    void draw(); // 도형을 그리는 추상 메소드
    double getArea(); // 도형의 면적을 리턴하는 추상 메소드
    default public void redraw() { // 디폴트 메소드
        System.out.print("--- 다시 그립니다. ");
        draw();
    }
}
```

실습 1> 인터페이스 실습

Shapes.java, Circle.java, Oval.java, Rect.java, Shape.java

- 다음 main() 메소드와 실행 결과를 참고하여, 인터페이스 Shape을 구현한 클래스 Circle, Oval, Rect를 작성하고 전체 프로그램을 완성하시오.

```
public class Shapes {
    static public void main(String [] args) {
        Shape [] list = new Shape[3]; // Shape을 상속받은 클래스
                                      // 객체의 레퍼런스 배열
        list[0] = new Circle(10);     // 반지름이 10인 원 객체
        list[1] = new Oval(20, 30);  // 20x30 사각형에 내접하는 타원
        list[2] = new Rect(10, 40);  // 10x40 크기의 사각형

        for(int i=0; i<list.length; i++)
            list[i].redraw();
        for(int i=0; i<list.length; i++)
            System.out.println("면적은 " + list[i].getArea());
    }
}
```

실습 1> 인터페이스 실습

- 실행 결과

```
---다시 그립니다. 반지름이 10인 원입니다.
---다시 그립니다. 너비: 20, 높이:30에 내접하는 타원입니다.
---다시 그립니다. 너비: 10, 높이:40의 사각형입니다.
면적은 314.0
면적은 471.00000000000006
면적은 400.0
```

Shapes.java의 실행 결과가 pdf와 다르게 나와도 됩니다. (double 타입의 문제)
계산 결과는 471.0 또는 471.0..... 6으로 나와도 됩니다.

실습 2> 인터페이스 실습

- java.util 패키지에 있는 Iterator 인터페이스를 구현하는 클래스 CardDeck을 작성하라.
- Iterator 인터페이스는 다음 메소드를 가지고 있다.
  - boolean hasNext(): 반환할 요소가 있으면 true를 반환한다.
  - Object next(): 반복의 다음 요소를 반환한다.
  - void remove(): 이 반복자가 반환한 마지막 요소를 기본 컬렉션에서 제거한다.
- CardDeck 클래스는 내부에 13장의 카드가 저장된 객체 배열을 가지고 있다. 첫 번째 next() 호출은 카드 2를 반환하고 이어서 카드 3, 카드 4, … 카드 Ace 까지를 반환한다.
  - hasNext(), next() 메소드의 내부 기능을 구현하라.
  - remove() 메소드는 내부 기능은 구현하지 않고 { } 로만 메소드를 구현한다.

```java
import java.util.Iterator;

class CardDeck implements Iterator {
    // 13장의 카드가 저장된 문자열 객체 배열 생성 >> "2", .... "Ace" 까지 저장

    // 현재 위치를 위한 변수 선언 및 초기화

    public boolean hasNext() {
        // 현재 위치가 카드 배열의 끝까지 봤다면 (즉, 카드배열의 길이와 같다면)
        // 더이상 다음 값은 존재하지 않으므로 false
        // 그렇지 않으면 true
    }
    public Object next() {
        // 문자열 배열로부터 현재 위치의 값을 반환함.
        // 현재 위치값 증가
    }
    public void remove() {

    }
}
```

```java
public class IteratorTest{
    public static void main(String[] args) {
        CardDeck i = new CardDeck();
        while (i.hasNext()) {
            System.out.println("next()가 반환하는 값:"+ i.next());
        }
    }
}
```

```
<terminated> IteratorTest [Java
next()가 반환하는 값:2
next()가 반환하는 값:3
next()가 반환하는 값:4
next()가 반환하는 값:5
next()가 반환하는 값:6
next()가 반환하는 값:7
next()가 반환하는 값:8
next()가 반환하는 값:9
next()가 반환하는 값:10
next()가 반환하는 값:Jack
next()가 반환하는 값:Queen
next()가 반환하는 값:King
next()가 반환하는 값:Ace
```

실습 3> 추상 클래스 실습

- 다음은 키와 값을 하나의 아이템으로 저장하고 검색 수정이 가능한 추상 클래스가 있다.

```
abstract class PairMap {
    protected String keyArray []; // key들을 저장하는 배열
    protected String valueArray []; // value 들을 저장하는배열
    abstract String get(String key); // key 값으로 value를 검색
    abstract void put(String key, String value); // key와 value를 쌍으로 저장
    abstract String delete(String key); // key 값을 가진 아이템(value와 함께)을 삭제.
                                        //       삭제된 value 값 리턴
    abstract int length(); // 현재 저장된 아이템의 개수 리턴
}
```

- PairMap을 상속받는 Dictionary 클래스를 구현하고, 이를 다음과 같이 활용하는 main() 메소드를 가진 클래스 DictionaryApp도 작성하라. (DictionaryApp.java)
  - PairMap, Dictionary, DictionaryApp 클래스가 모두 DictionaryApp.java 파일안에 있어도 됩니다.
  - 물론 각각 개별로 작성해도 됩니다.

```
class Dictionary extends PairMap {
    // 현재 저장된 아이템의 개수를 위한 변수 선언 및 초기화
    public Dictionary(int capacity) {
        // capacity 만큼의 각 배열 생성
    }
    String get(String key) {
        // 현재 배월에 저장된 원소 개수만큼 반복하면서
        // key와 같은 값이 있는지 key배열에서 찾아서
        // 해당 value 값 반환
        // key를 발견할 수 없다면 null리턴
    }
    void put(String key, String value) {
        // 현재 배열에 저장된 원소 개수만큼 반복하면서
        // key 값이 이미 배열에 저장되어 있는 경우
            // 그 위치에 value 값 저장
        // key 값이 배열에 저장되어 있지 않는 경우
            // 현재 배열에 마지막으로 저장된 원소 다음에 key값과 value 값 저장
    }
    String delete(String key) {
        // 현재 배열에 저장된 원소 개수만큼 반복하면서
        // key 값이 이미 배열에 저장되어 있는 경우
            // 그 위치의 key, value 값 삭제
                // 삭제된 위치의 뒤에 있는 원소들을 앞으로 한칸씩 이동해야 함!!
            // value 값 반환
        // key 값이 배열에 저장되어 있지 않는 경우 null 반환
    }
    int length() { return count; }

}
```

문자열 값의 비교는 a == b 비교가 아닌 a.equals(b) 를 사용

실습 3> 추상 클래스 실습

- 다음 실행 결과와 같이 되도록 DictionaryApp 클래스를 작성하시오.
  - 사용자로부터 메뉴 입력 받아서
  - 해당 메뉴별 기능을 실행하도록 함
  - 앞서 작성한 Dictionary 클래스 이용

```
DictionaryApp [Java Application] D:₩eclipse24₩plugins₩org.eclips
한영 단어 등록 프로그램입니다.
==========================================
1. 단어 등록
2. 단어 검색
3. 단어 삭제
4. 모든 단어 보기
5. 종료
==========================================
메뉴를 선택하세요>> 1
한영 단어를 등록하시오. ex) 과학 science >> 사과 apple
==========================================
1. 단어 등록
2. 단어 검색
3. 단어 삭제
4. 모든 단어 보기
5. 종료
==========================================
메뉴를 선택하세요>> 1
한영 단어를 등록하시오. ex) 과학 science >> 과학 science
==========================================
1. 단어 등록
2. 단어 검색
3. 단어 삭제
4. 모든 단어 보기
5. 종료
==========================================
메뉴를 선택하세요>> 1
한영 단어를 등록하시오. ex) 과학 science >> 달력 calendar
==========================================
1. 단어 등록
2. 단어 검색
3. 단어 삭제
4. 모든 단어 보기
5. 종료
==========================================
메뉴를 선택하세요>> 1
한영 단어를 등록하시오. ex) 과학 science >> 겨울 winter
==========================================
1. 단어 등록
2. 단어 검색
3. 단어 삭제
4. 모든 단어 보기
5. 종료
==========================================
메뉴를 선택하세요>> 4
현재 등록된 모든 단어는 다음과 같습니다.
사과: apple
과학: science
달력: calendar
겨울: winter
==========================================
메뉴를 선택하세요>> 2
검색할 단어를 입력하시오. >> 사과
검색 결과:apple
==========================================
1. 단어 등록
2. 단어 검색
3. 단어 삭제
4. 모든 단어 보기
5. 종료
==========================================
메뉴를 선택하세요>> 2
검색할 단어를 입력하시오. >> 겨울
검색 결과:winter
==========================================
1. 단어 등록
2. 단어 검색
3. 단어 삭제
4. 모든 단어 보기
5. 종료
==========================================
메뉴를 선택하세요>> 2
검색할 단어를 입력하시오. >> 봄
검색 결과:null
==========================================
1. 단어 등록
2. 단어 검색
3. 단어 삭제
4. 모든 단어 보기
5. 종료
==========================================
메뉴를 선택하세요>> 3
삭제할 단어를 입력하시오. >> 달력
삭제 결과:calendar
==========================================
1. 단어 등록
2. 단어 검색
3. 단어 삭제
4. 모든 단어 보기
5. 종료
==========================================
메뉴를 선택하세요>> 4
현재 등록된 모든 단어는 다음과 같습니다.
사과: apple
과학: science
겨울: winter
==========================================
```