23-여름학기 자바프로그래밍(001) Lab0629
소프트웨어학부 컴퓨터과학전공 2016133 이유진

1. 실습1 Shapes.java

```java
//이유진 (2016133)
//2023-06-29
//Lab0629-1: 도형의 구성을 묘사하는 인터페이스를 이용해 도형을 그리는 프로그램

interface Shape {
    final double PI = 3.14;

    void draw(); // 도형을 그리는 추상 메서드

    double getArea(); // 도형의 면적을 리턴하는 추상 메서드

    default public void redraw() { // 디폴트 메서드
        System.out.print("--- 다시 그립니다. ");
        draw();
    }
}

class Circle implements Shape {
    private int radius;

    public Circle(int radius) { // 생성자
        super();
        this.radius = radius;
    }

    @Override
    public void draw() {
        System.out.println("반지름이 " + radius + "인 원입니다.");
    }

    @Override
    public double getArea() { // 면적
        return radius * radius * PI;
    }

}

class Oval implements Shape {
    private int x, y; // 가로 x, 세로 y인 직사각형에 내접하는 타원

    public Oval(int x, int y) { // 생성자
        super();
        this.x = x;
        this.y = y;
    }

    @Override
    public void draw() {
        System.out.println("너비: " + x + ", 높이: " + y + "에 내접하는 타원입니다.");
    }

    @Override
    public double getArea() { // 면적
        return x * y / 4 * PI; // 타원의 넓이 공식: pi*반지름1*반지름2
    }

}

class Rect implements Shape {

    private int x, y;

    public Rect(int x, int y) {
        super();
        this.x = x;
        this.y = y;
    }

    @Override
    public void draw() {
        System.out.println("너비: " + x + ", 높이: " + y + "의 사각형입니다.");
    }

    @Override
    public double getArea() {
        return x * y;
    }

}

public class Shapes {
    public static void main(String[] args) {
        Shape[] list = new Shape[3]; // Shape을 상속받은 클래스

        // 객체의 레퍼런스 개별
        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());
    }

}
```

```
Console
<terminated> Shapes [Java Application] C:₩Program Files₩Java₩jdk-17₩bin₩javaw.exe (2023. 6. 29. 오후 5:4
--- 다시 그립니다. 반지름이 10인 원입니다.
--- 다시 그립니다. 너비: 20, 높이: 30에 내접하는 타원입니다.
--- 다시 그립니다. 너비: 10, 높이: 40의 사각형입니다.
면적은 314.0
면적은 471.0
면적은 400.0
```

2. 실습2 IteratorTest.java

```java
//이유진 (2016133)
//2023-06-29
//Lab0629-2: java.util의 Iterator 인터페이스를 구현해 카드덱배열을 출력하는 프로그램

import java.util.Iterator;

class CardDeck implements Iterator {
    // 13장의 카드가 저장된 문자열 객체 배열 생성 >> "2", ... "Ace" 까지 저장
    String[] card = new String[] { "2", "3", "4", "5", "6", "7", "8", "9", "10",
            "Jack", "Queen", "King", "Ace" };

    // 현재 위치를 위한 변수 선언 및 초기화
    private int now = 0;

    // 생성자
    public String[] getCard() {
        return card;
    }

    public void setCard(String[] card) {
        this.card = card;
    }

    public int getNow() {
        return now;
    }

    public void setI(int now) {
        this.now = now;
    }

    public boolean hasNext() { // 반환할 요소가 있으면 true를 반환하는 메서드
        if (now == card.length) // 현재 위치가 카드 배열의 끝까지 봤다면 (즉, 카드배열의 길이와 같다면)
            return false; // 더이상 다음 값은 존재하지 않으므로 false
        else // 그렇지 않으면 true
            return true;
    }

    public Object next() { // 반복의 다음 요소를 반환하는 메서드
    public Object next() { // 반복의 다음 요소를 반환하는 메서드
        // 문자열 배열로부터 현재 위치의 값을 반환함
        // 현재 위치값 증가
        return card[now++]; // 반환 후 증가
    }

    public void remove() { // 이 반복자가 반환한 마지막 요소를 기본 컬렉션에서 제거하는 메서드
        // 구현하지 않고 {}로만 메서드를 구현함
    }
}

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());
        }
    }

}
```

```
Console
<terminated> IteratorTest [Java Application] C:₩Progr
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
```

3. 실습 3 DictionaryApp.java

```java
//이유진 (2016133)
//2023-06-29
//Lab0629-3: key와 value값을 쌍으로 가지는 Dictionary를 구현하는 프로그램

abstract class PairMap {
    protected String keyArray[]; // key들을 저장하는 배열
    protected String valueArray[]; // value들을 저장하는 배열

    abstract String get(String key); // key값으로 value를 검색

    abstract void put(String key, String value); // key와 value를 쌍으로 저장

    abstract String delete(String key); // key값을 가진 아이템(value와 함께)을 삭제, 삭제된 value값 리턴

    abstract int length(); // 현재 저장된 아이템의 개수 리턴
}

class Dictionary extends PairMap {
    private int cnt = 0; // 현재 배열에 저장된 개수

    // 현재 저장된 아이템의 개수를 위한 변수 선언 및 초기화
    public Dictionary(int capacity) {
        // capacity만큼의 각 배열 생성
        keyArray = new String[capacity]; // key들을 저장하는 배열
        valueArray = new String[capacity]; // value들을 저장하는 배열
    }

    @Override
    String get(String key) { // key값으로 value를 검색
        // 현재 배열에 저장된 원소 개수만큼 반복하면서
        for (int i = 0; i < cnt; i++) {
            if (keyArray[i].equals(key)) // key와 같은 값이 있는지 key배열에서 찾아서
                return valueArray[i]; // 해당 value값 반환
        }

        // for문을 빠져나왔는데 반환값이 없어 함수가 종료되지 않음
        return null; // key를 발견할 수 없다면 null 리턴
    }
```

```java
    @Override
    void put(String key, String value) { // key와 value를 쌍으로 저장
        boolean exist = false;

        for (int i = 0; i < cnt; i++) {// 현재 배열에 저장된 원소 개수만큼 반복하면서
            if (keyArray[i].equals(key)) {// key값이 이미 배열에 저장되어 있는 경우
                valueArray[i] = value;// 그 위치에 value값 저장
                exist = true;
                break;
            }
        }
        if (exist == false) {// key값이 배열에 저장되어 있지 않은 경우
            keyArray[cnt] = key; // 현재 배열에 마지막으로 저장된 원소 다음에 key값과 value값 저장
            valueArray[cnt] = value;
            cnt++; // 현재 배열에 저장된 개수 증가
        }
    }

    @Override
    String delete(String key) { // key값을 가진 아이템(value와 함께)을 삭제, 삭제된 value값 리턴
        for (int i = 0; i < cnt; i++) { // 현재 배열에 저장된 원소 개수만큼 반복하면서
            if (keyArray[i].equals(key)) { // key값이 이미 배열에 저장되어 있는 경우
                String temp = valueArray[i]; // 반환할 값 저장
                for (int j = i; j < cnt - 1; j++) {// 삭제된 위치의 뒤에 있는 원소들을 앞으로 한 칸씩 이동
                    keyArray[j] = keyArray[j + 1];
                    valueArray[j] = valueArray[j + 1];
                }
                cnt--;
                return temp;// value값 반환
            }
        }

        return null; // key값이 배열에 저장되어 있지 않은 경우 null 반환
    }

    @Override
    int length() { // 현재 저장된 아이템의 개수 리턴
        return cnt;
    }

    @Override
    int length() { // 현재 저장된 아이템의 개수 리턴
        return cnt;
    }
}

public class DictionaryApp {

    public static void main(String[] args) {

        Dictionary dic = new Dictionary(10);
        dic.put("김숙명", "자바");
        dic.put("이순헌", "파이썬");
        dic.put("이순헌", "C++"); // key값이 이미 배열에 저장되어 있는 경우 그 위치에 value값 저장
        dic.put("정명신", "react");
        System.out.println("이순헌의 값은 " + dic.get("이순헌"));
        System.out.println("김숙명의 값은 " + dic.get("김숙명"));
        dic.delete("김숙명");
        System.out.println("김숙명의 값은 " + dic.get("김숙명"));
        System.out.println("정명신의 값은 " + dic.get("정명신"));

    }

}
```

```
Console
<terminated> DictionaryApp [Java Application
이순헌의 값은 C++
김숙명의 값은 자바
김숙명의 값은 null
정명신의 값은 react
```