



세종대학교

Report

명품 JAVA 프로그래밍

제출일 2018.11.06

전공 디지털콘텐츠학과

학번 16013093

이름 박상우

Hw 2_1

상속을 받는 pariMap클래스와 main은 책에 나와있는 코드입니다. 추상클래스인 pariMap클래스를 상속받는 Dictionary클래스를 만들었습니다.

우선 생성자의 인자로 배열의 길이를 입력 받고 keyArray, valueArray의 크기를 설정했습니다. 또한 keyArray와 valueArray의 인덱스는 항상 같이 저장되고 수정되므로 아이템 개수를 저장하는 필드변수 cnt를 0으로 초기화 했습니다.

String get(String key) 메소드는 원하는 key의 value값을 찾아오는 메소드입니다. keyArray[i].equals(key)를 사용해 찾는 키가 keyArray 배열에 있으면 valueArray[i]를 반환하고 keyArray배열을 전부 확인해도 찾는 키가 없다면 null을 반환하게 만들었습니다.

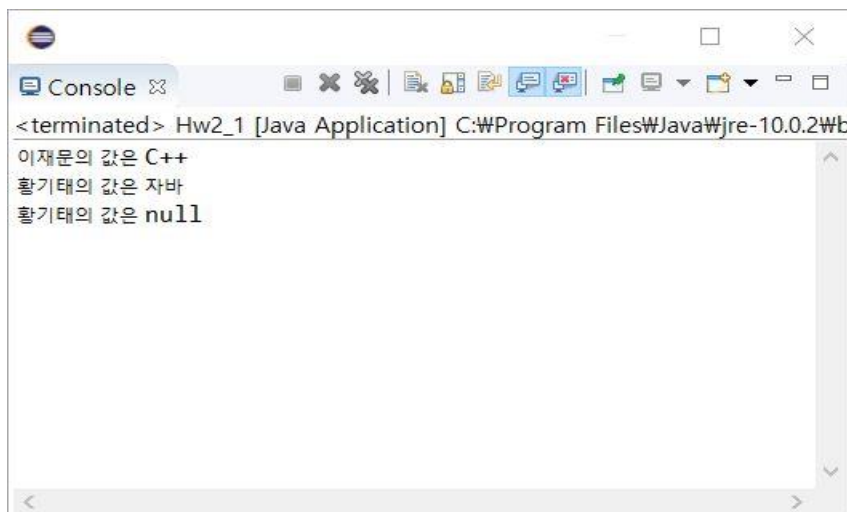
void put(String key, String value) 메소드는 key와 value를 저장하는 메소드입니다. 먼저 keyArray배열을 확인해 key가 중복되는지 확인하고 중복된다면 value값만 저장합니다. 만약 찾는 키가 없다면 keyArray와 valueArray의 마지막에 key와 value를 저장하고 cnt를 1증가시켰습니다.

String delete(String key) 메소드도 위 두 메소드와 마찬가지로 keyArray배열을 반복하며 삭제할 key값이 있는지 확인하고 삭제할 값이 있다면

```
        if (keyArray[i].equals(key)) {
            String str = valueArray[i];
            for (int j = i; j < keyArray.length - 1; j++) {
                keyArray[j] = keyArray[j + 1];
                valueArray[j] = valueArray[j + 1];
            }
            keyArray[cnt] = null;
            valueArray[cnt] = null;
            cnt--;
            return str;
        }
```

위와 같은 방식으로 값을 삭제했습니다.

int length() 메소드의 경우 cnt가 아이템의 개수이므로 cnt를 반환했습니다.



Hw 2_1

Line, Rect, Circle 클래스의 super클래스인 Shape 추상클래스는 아래와 같습니다.

```
abstract class Shape {  
    private Shape next;  
    public Shape() {next = null;}  
    public void setNext(Shape obj) {next = obj;}  
    public Shape getNext() {return next;}  
    public abstract void draw();  
}
```

링크드리스트로 세 도형을 관리하기 위해 next필드변수가 있으며 무조건 구현이 필요한 draw메소드를 추상메소드로 선언했습니다. 실제로 편집기능을 가진 GraphicEditor 클래스는 이름, Shape 리스트, 아이템 개수를 변수로 가지고 있습니다.

void add(int in) 메소드는 공백리스트 일 때

```
Shape head = list;  
if (head == null) {  
    list = tmp; // tmp는 입력된 숫자에 따라 세 도형 중 하나의 객체를 가르키고 있습니다.  
    return;  
}
```

리스트의 처음을 가르키게 합니다. 공백이 아니라면 head를 맨 뒤로 이동시킨 후 맨 뒤에 tmp를 삽입합니다.

boolean delete(int in) 메소드는 삭제가 되면 true, 삭제를 실패하면 false를 반환합니다.

삭제할 위치를 인자로 받기 때문에 in이 cnt 이상이면 리스트의 범위를 벗어나는 것 이므로 false를 반환합니다.

```
Shape head = list;  
if (in == 0) {  
    list = head.getNext();  
    return true;  
}  
Shape tmp = head;  
for (int i = 0; i < in; i++) {  
    tmp = head;  
    head = head.getNext();  
}  
tmp.setNext(head.getNext());  
return true;
```

처음 노드를 삭제하기 위해서 list를 head.next와 연결합니다.

그 이후 노드들은 삭제 위치만큼 이동 후 다다음 노드를 연결해 해당 위치를 삭제합니다.

void print() 메소드의 경우 null이 나오지 않을 때까지 이동하며 .draw()를 호출해 출력합니다.

```

121
122 public static void main(String[] args) {
123     Editor myEditor = new Editor("beauty");
124     myEditor.add(1);
125     myEditor.add(2);
126     myEditor.add(3);
127     myEditor.add(1);
128     myEditor.add(2);
129     myEditor.add(3);
130     //System.out.println(myEditor.delete(0));
131     myEditor.print();
132 }
133
134 }
135

```

Problems @ Javadoc Declaration Console

<terminated> Hw2_2 [Java Application] C:\Program Files\Java

Line
Rect
Circle
Line
Rect
Circle

```

118 }
119
120 public class Hw2_2 {
121
122 public static void main(String[] args) {
123     Editor myEditor = new Editor("beauty");
124     myEditor.add(1);
125     myEditor.add(2);
126     myEditor.add(3);
127     myEditor.add(1);
128     myEditor.add(2);
129     myEditor.add(3);
130     System.out.println(myEditor.delete(0));
131     myEditor.print();
132 }
133
134 }
135

```

Problems @ Javadoc Declaration Console

<terminated> Hw2_2 [Java Application] C:\Program Files\Java

true
Rect
Circle
Line
Rect
Circle

```

118 }
119
120 public class Hw2_2 {
121
122 public static void main(String[] args) {
123     Editor myEditor = new Editor("beauty");
124     myEditor.add(1);
125     myEditor.add(2);
126     myEditor.add(3);
127     myEditor.add(1);
128     myEditor.add(2);
129     myEditor.add(3);
130     System.out.println(myEditor.delete(1));
131     myEditor.print();
132 }
133
134 }
135

```

Problems @ Javadoc Declaration Console

<terminated> Hw2_2 [Java Application] C:\Program Files\Java

true
Line
Circle
Line
Rect
Circle

```

119
120 public class Hw2_2 {
121
122 public static void main(String[] args) {
123     Editor myEditor = new Editor("beauty");
124     myEditor.add(1);
125     myEditor.add(2);
126     myEditor.add(3);
127     myEditor.add(1);
128     myEditor.add(2);
129     myEditor.add(3);
130     System.out.println(myEditor.delete(6));
131     myEditor.print();
132 }
133
134 }
135

```

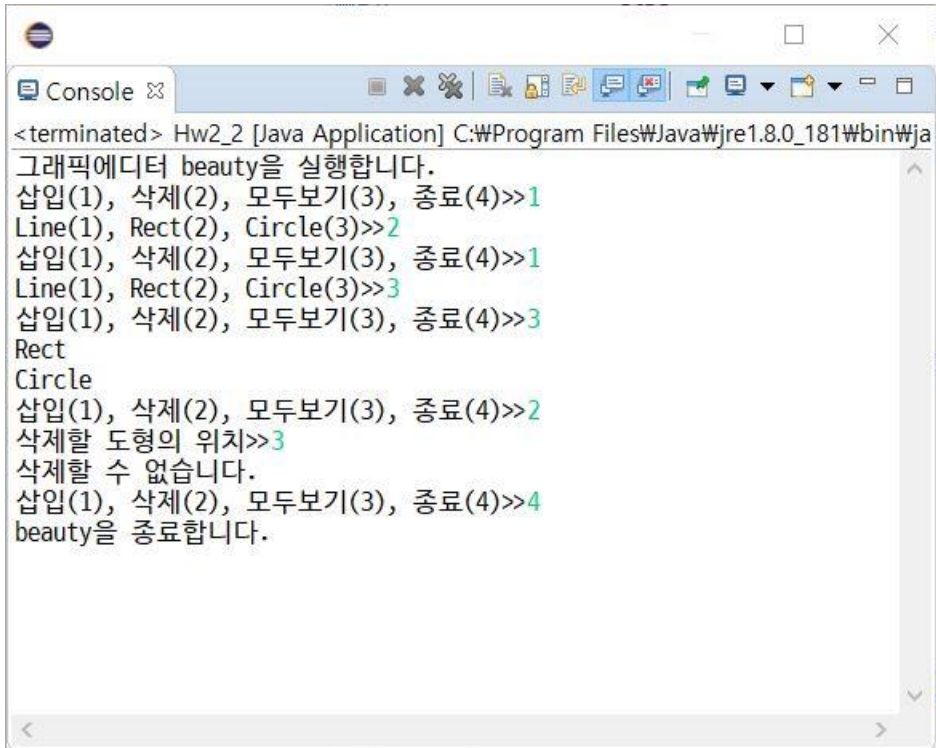
Problems @ Javadoc Declaration Console

<terminated> Hw2_2 [Java Application] C:\Program Files\Java

false
Line
Rect
Circle
Line
Rect
Circle

삽입과 삭제, 출력 예시입니다.

마지막으로 void startProgram() 메소드를 추가해 텍스트 입출력을 가능하게 했습니다.



The screenshot shows a Java console window titled "Console" with a standard toolbar. The text inside the console is as follows:

```
<terminated> Hw2_2 [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\java.exe
그래픽에디터 beauty을 실행합니다.
삽입(1), 삭제(2), 모두보기(3), 종료(4)>>1
Line(1), Rect(2), Circle(3)>>2
삽입(1), 삭제(2), 모두보기(3), 종료(4)>>1
Line(1), Rect(2), Circle(3)>>3
삽입(1), 삭제(2), 모두보기(3), 종료(4)>>3
Rect
Circle
삽입(1), 삭제(2), 모두보기(3), 종료(4)>>2
삭제할 도형의 위치>>3
삭제할 수 없습니다.
삽입(1), 삭제(2), 모두보기(3), 종료(4)>>4
beauty을 종료합니다.
```