

문제해결및실습:JAVA Homework #2

20011844

안수경

< Hw2_1 >

```
Console [Java Application] /Library/Java/JavaVirtualMachines/jdk-16.0.2.jdk/Contents/Home/bin/java
<terminated> Hw2_1 [Java Application] /Library/Java/JavaVirtualMachines/jdk-16.0.2.jdk/Contents/Home/bin/java
명품콘서트홀 예약 시스템입니다.
예약:1, 조회:2, 취소:3, 끝내기:4 >> 1
좌석구분 S(1), A(2), B(3) >> 1
S >> -----
이름 >> 황기태
번호 >> 1
예약:1, 조회:2, 취소:3, 끝내기:4 >> 1
좌석구분 S(1), A(2), B(3) >> 2
A >> -----
이름 >> 김호수
번호 >> 5
예약:1, 조회:2, 취소:3, 끝내기:4 >> 2
S >> 황기태 -----
A >> ----- 김호수 -----
B >> -----
<<< 조회를 완료하였습니다. >>>
예약:1, 조회:2, 취소:3, 끝내기:4 >> 3
좌석 S:1, A:2, B:3 >> 4
Error! Try again!
좌석 S:1, A:2, B:3 >> 2
A >> ----- 김호수 -----
이름 >> 안수경
Error! Try again!
이름 >> 김호수
예약:1, 조회:2, 취소:3, 끝내기:4 >> 2
S >> 황기태 -----
A >> -----
B >> -----
<<< 조회를 완료하였습니다. >>>
예약:1, 조회:2, 취소:3, 끝내기:4 >> 5
Error! Try again!
예약:1, 조회:2, 취소:3, 끝내기:4 >> 4
```

예외 처리 할 게 너무 많았다. 잘못된 (메뉴 번호, 삭제 시 이름, 비어있는데 삭제를 시도한 경우, 좌석 타입, 좌석 번호) 를 예외 처리 했지만 이 밖에도 예러가 떠야하는 경우가 더 있다. 처음에 findName 함수에서 입력된 이름이 이미 예약되어 있는지 안 되어 있는지를 확인하려고 if(name == seat[type-1][i]) 라고 조건문을 작성 했었는데 안 됐는데 String의 비교는 equals 메소드를 사용해서 비교해야 한다는 것을 알고 수정했더니 잘 작동 되었다.

```
import java.util.Scanner;
```

```
class Concert {
```

```
    String[][] seat = new String[3][10];
```

```
    void printSeat(int type) {
```

```
        if (type == 1) {
```

```
            System.out.print("S >> ");
```

```
        } else if (type == 2) {
```

```
            System.out.print("A >> ");
```

```
        } else if (type == 3) {
```

```
            System.out.print("B >> ");
```

```
        }
```

```
        for (int i = 0; i < 10; i++) {
```

```
            if (seat[type - 1][i] == null)
```

```
                System.out.print("---");
```

```
            else
```

```
                System.out.print(seat[type - 1][i]);
```

```
            System.out.print(" ");
```

```
        }
```

```
        System.out.println();
```

```
    }
```

```
    void reserveSeat(int type, String name, int num) {
```

```
        seat[type - 1][num - 1] = name;
```

```
    }
```

```
    void showSeat() {
```

```
        printSeat(1);
```

```
        printSeat(2);
```

```
        printSeat(3);
```

```
        System.out.println("<<< 조회를 완료하였습니다. >>>");
```

```
    }
```

```
    void cancelSeat(int type, String name) {
```

```
        for (int i = 0; i < 10; i++) {
```

```
            if (name.equals(seat[type - 1][i]))
```

```
                seat[type - 1][i] = null;
```

```

    }
}

boolean findName(int type, String name) {
    boolean flag;
    for (int i = 0; i < 10; i++) {
        if (name.equals(seat[type - 1][i])) {
            return true;
        }
    }
    System.out.println("Error! Try again!");
    return false;
}

boolean findNum(int num) {
    if (num < 1 || num > 10) {
        System.out.println("Error! Try again!");
        return false;
    } else
        return true;
}

boolean findType(int type) {
    if (type < 1 || type > 3) {
        System.out.println("Error! Try again!");
        return false;
    } else
        return true;
}

boolean isEmpty(int type) {
    for (int i = 0; i < 10; i++) {
        if (seat[type - 1][i] != null)
            return false;
    }
    System.out.println("Error! Try again!");
    return true;
}
}

public class Hw2_1 {

```

```

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);

    System.out.println("명품콘서트홀 예약 시스템입니다.");

    int order;
    int type;
    String name;
    int num;
    Concert concert = new Concert();
    while (true) {
        System.out.print("예약:1, 조회:2, 취소:3, 끝내기:4 >> ");
        order = input.nextInt();
        if (order == 1) {
            while (true) {
                System.out.print("좌석구분 S(1), A(2), B(3) >> ");

                type = input.nextInt();
                if (concert.findType(type))
                    break;
            }
            concert.printSeat(type);
            System.out.print("이름 >> ");
            name = input.next();
            while (true) {
                System.out.print("번호 >> ");
                num = input.nextInt();
                if (concert.findNum(num))
                    break;
            }
            concert.reserveSeat(type, name, num);
        } else if (order == 2) {
            concert.showSeat();
        } else if (order == 3) {
            while (true) {
                System.out.print("좌석 S:1, A:2, B:3 >> ");
                type = input.nextInt();
                if (concert.findType(type)) {
                    if (!concert.isEmpty(type))
                        break;
                }
            }
        }
    }
}

```

```
    }
    concert.printSeat(type);
    while (true) {
        System.out.print("이름 >> ");
        name = input.next();
        if (concert.findName(type, name))
            break;
    }
    concert.cancelSeat(type, name);
} else if (order == 4) {
    break;
} else {
    System.out.println("Error! Try again!");
}
}
}
}
```

< Hw2_2 >

Console

<terminated> Hw2_2 [Java Application] /Library/Java/JavaVirt

이재문의 값은 C++

황기태의 값은 자바

황기태의 값은 null

2번 문제를 풀면서 추상 자료형의 필요성에 대하여 확실히 알게 되었다. 실은 서브클래스에서 접근할 수 있는 protected라는 변수도 이론 공부만 했을 때에는 잘 몰랐는데 이 문제를 풀면서 더 와닿았던 것 같다. 처음에 Dictionary 클래스 안에 있는 len, max 라는 정수형 변수를 그냥 선언했었는데 수퍼 클래스에서 int length() 함수가 추상 자료형으로 선언 된 것을 보고 이 변수들을 private으로 선언하는 게 맞다고 생각해서 수정했다. 이렇게 사소한 데에서 하나하나 알아가니 흥미로웠다.

```
abstract class PairMap {
    protected String keyArray [];
    protected String valueArray [];
    abstract String get(String key);
    abstract void put(String key, String value);
    abstract String delete(String key);
    abstract int length();
}

class Dictionary extends PairMap{
    private int len;
    private int max;
    Dictionary(int _max){
        max = _max;
        keyArray = new String[max];
        valueArray = new String[max];
        len = 0;
    }
    String get(String key) {
        int index = -1;
        for(int i =0;i<len;i++) {
            if (key.equals(keyArray[i])) index = i;
        }
        if (index == -1) return null;
        else return valueArray[index];
    }
    void put(String key, String value) {
        if(this.get(key)==null) {
            keyArray[len] = key;
```

```

        valueArray[len] = value;
        len++;
    }
    else {
        for(int i =0;i<len;i++) {
            if (key.equals(keyArray[i])) valueArray[i] = value;
        }
    }
}

String delete(String key) {
    String value = null;
    boolean flag = false;
    for(int i =0;i<len;i++) {
        if (key.equals(keyArray[i])) {
            if (flag == false) {
                value = valueArray[i];
                flag = true;
            }
            keyArray[i] = keyArray[i+1];
            valueArray[i] = valueArray[i+1];
        }
    }
    return value;
}

int length() {
    return len;
}
}

public class Hw2_2 {

    public static void main(String[] args) {
        Dictionary dic = new Dictionary(10);
        dic.put("황기태", "자바");
        dic.put("이재문", "파이썬");
        dic.put("이재문", "C++");
        System.out.println("이재문의 값은 "+dic.get("이재문"));
        System.out.println("황기태의 값은 "+dic.get("황기태"));
        dic.delete("황기태");
        System.out.println("황기태의 값은 " + dic.get("황기태"));
    }
}

```

< Hw2_3 >

```
Console ✕
<terminated> Hw2_3 [Java Application] /Library/Java/Java
그래픽 에디터 beauty을 실행합니다.
삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> 1
Line(1), Rect(2), Circle(3) >> 2
삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> 1
Line(1), Rect(2), Circle(3) >> 3
삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> 3
Rect
Circle
삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> 2
삭제할 도형의 위치 >> 3
삭제할 수 없습니다.
삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> 1
Line(1), Rect(2), Circle(3) >> 4
Error! Try again!
Line(1), Rect(2), Circle(3) >> 1
삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> 3
Rect
Circle
Line
삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> 4
beauty을 종료합니다.
```

링크드리스트를 C언어에서 한 번 배웠지만 자바로 구현하려니 왠지 모르게 더 어려운 느낌이었다.

우선 처음 삽입 할 때부터 에러가 많이 났었다. whatShape를 Shape형으로 선언해서 업캐스팅을 활용했어야 했는데 각 서브 클래스로 선언해서 굉장히 혼란스러운 코드가 짜였었다. 다 알아엮고 차분하게 다시 하니가 정상적으로 삽입이 되는 걸 확인할 수 있었다.

삽입을 끝내고 삭제를 구현하는데 다른 삭제는 다 되는데 맨 마지막으로 삽입된 도형이 자꾸 삭제가 안 되었다. 그래서 마지막을 삭제하는 경우는 if (what == n) sh = p; 로 따로 처리해줬더니 정상 작동하였다.

C언어에서 링크드리스트를 구현할 때에는 p = p.next; 이런 식으로 했었는데 JAVA에서는 next라는 변수가 private으로 선언되어 있어서 메소드로 접근해야해서 p=p.getNext();로 사용되는 점이 생소했다.

그리고 지금 보고서를 작성하며 생각해보니 삽입, 삭제, 출력을 따로 메소드로 빼서 코드를 작성하는 편이 더 깔끔했을 것 같다.


```

import java.util.Scanner;

abstract class Shape {
    private Shape next;

    public Shape() { // next 초기화
        next = null;
    }

    public void setNext(Shape obj) { // next에 obj 저장
        next = obj;
    }

    public Shape getNext() { // next 반환
        return next;
    }

    public abstract void draw();
}

class Line extends Shape {
    public void draw() {
        System.out.println("Line");
    }
}

class Circle extends Shape {
    public void draw() {
        System.out.println("Circle");
    }
}

class Rect extends Shape {
    public void draw() {
        System.out.println("Rect");
    }
}

public class Hw2_3 {

    public static void main(String[] args) {

```

```

Scanner input = new Scanner(System.in);
System.out.println("그래픽 에디터 beauty을 실행합니다.");
int order, what;
int n = 0;
Shape head = null;
Shape sh = null;

while (true) {
    System.out.print("삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> ");
    order = input.nextInt();
    if (order == 1) {

        Shape whatShape;
        while (true) {
            System.out.print("Line(1), Rect(2), Circle(3) >> ");

            what = input.nextInt();
            if (what == 1) {
                whatShape = new Line();
                break;
            } else if (what == 2) {
                whatShape = new Rect();
                break;
            } else if (what == 3) {
                whatShape = new Circle();
                break;
            } else {
                System.out.println("Error! Try
again!");
            }
        }
        if (sh == null) {
            head = whatShape;

        } else {
            sh.setNext(whatShape);
        }
        sh = whatShape;
        n++;
    }
    else if (order == 2) {

```

```

        System.out.print("삭제할 도형의 위치 >> ");
        what = input.nextInt();
        Shape p = head;
        if (what > n)
            System.out.println("삭제할 수 없습니다.");

        else {
            if (what == 1)
                head = p.getNext();
            else {
                for (int i = 1; i < what - 1; i++) {
                    p = p.getNext();
                }
                if (what == n) {
                    sh = p;
                }
                p.setNext(p.getNext().getNext());
                n--;
            }
        } else if (order == 3) {
            Shape p = head;
            while (p != null) {
                p.draw();
                p = p.getNext();
            }
        } else if (order == 4) {
            System.out.println("beuaty을 종료합니다.");
            break;
        } else {
            System.out.println("Error! Try again!");
        }
    }
}

```