



세종대학교

Report

명품 JAVA 프로그래밍

제출일 2018.09.20

전공 디지털콘텐츠학과

학번 16013093

이름 박상우

Hw 1_1.

두 원의 중심 간 거리가 두 원의 반지름 합보다 작을 때 두 원은 겹친다고 할 수 있습니다.

두 중심의 좌표를 (x,y) (x1,y1)이라고 할 때 $\text{Math.sqrt}((x-x1)*(x-x1) + (y-y1)*(y-y1))$ 로 두 점사이의 거리를 구하고 조건문을 이용해 결과를 출력했습니다.

```
<terminated> Hw1_1 [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018. 9. 20. 오후 8:34:28)
첫번째 원의 중심과 반지름 입력>>10 10 3
두번째 원의 중심과 반지름 입력>>12 12 2
두 원은 겹칩니다.
```

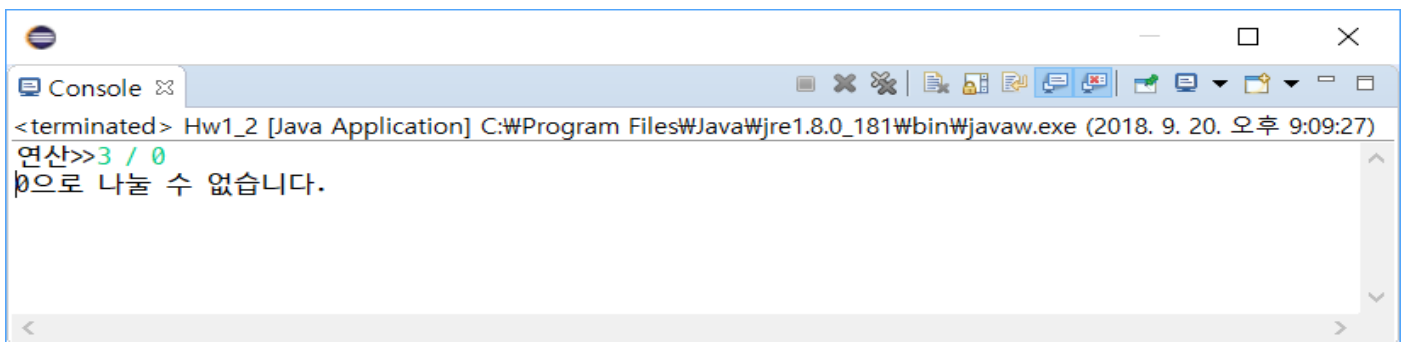
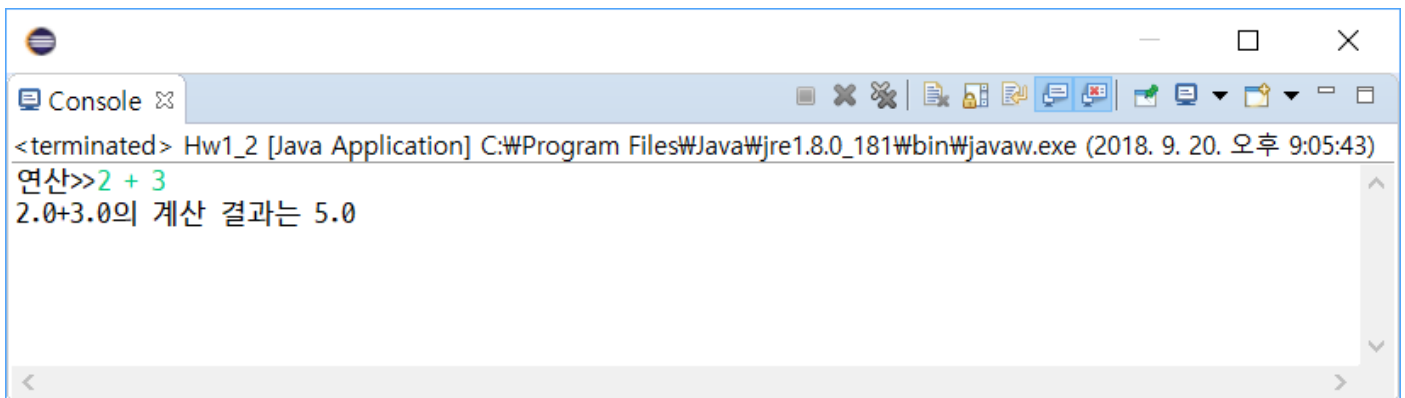
```
<terminated> Hw1_1 [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018. 9. 20. 오후 8:40:15)
첫번째 원의 중심과 반지름 입력>>10 10 3
두번째 원의 중심과 반지름 입력>>30 30 2
두 원은 겹치지 않습니다.
```

Hw1_2

Scanner 내장 함수 중에 char로 반환 받는 함수가 없기 때문에 String으로 저장하고 '+' 가 아닌 "+" 으로 비교했습니다. 조건문은 두 번째 인자가 0일 경우 "0으로 나눌 수 없습니다." 를 출력하고 아닐 경우 다시 조건문을 이용해 String 변수에 저장된 연산자를 확인하고 값을 출력했습니다.

```
if (b == 0) { // 0으로 나누는 경우
    System.out.println("0으로 나눌 수 없습니다.");
} else { // 아닐경우
    System.out.print(a + op + b + "의 계산 결과는 ");

    if (op.equals("+") == true) {
        System.out.println(a + b);
    } else if (op.equals("-") == true) {
        System.out.println(a - b);
    } else if (op.equals("*") == true) {
        System.out.println(a * b);
    } else { // 연산자가 총 4종류만 들어오기에 나누기는 else를 사용했습니다.
        System.out.println(a / b);
    }
}
```



Hw1_3

정수 n을 입력 받고 `int ar[] = new int[n];`를 이용해 배열을 만들었습니다.

0~n-1까지 반복하며 `ar[i] = (int) (Math.random() * 100 + 1);`로 배열에 랜덤 값을 저장했습니다.

그 이후

```
for (int j = 0; j < i; j++) {  
    if (ar[j] == ar[i]) {  
        ar[i] = (int) (Math.random() * 100);  
    }  
}
```

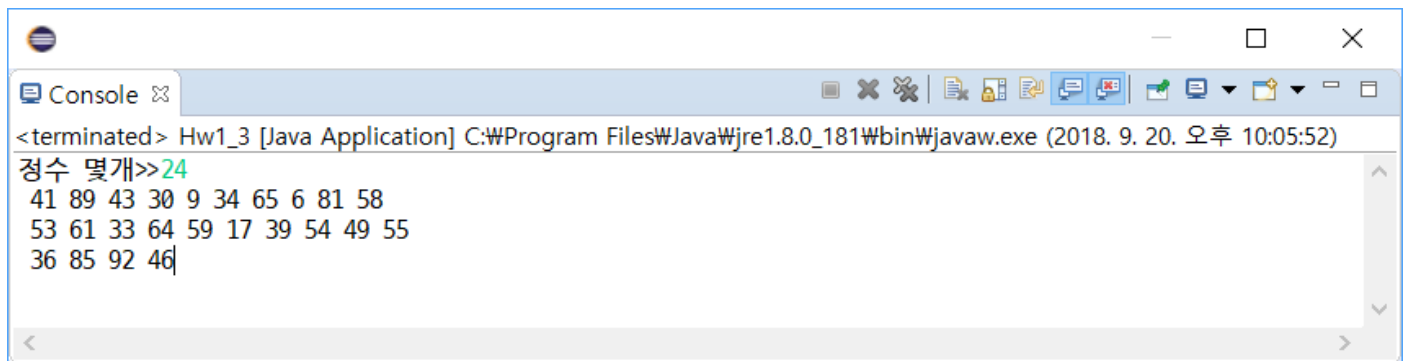
이중반복문을 이용해 중복된 값이 추가된 경우 새로 랜덤 값을 저장해 중복된 값을 피하려 했습니다.

몇 번 예시를 돌리는 도중 새로 저장되는 값이 이미 검사된 값과 중복되는 경우를 발견했고

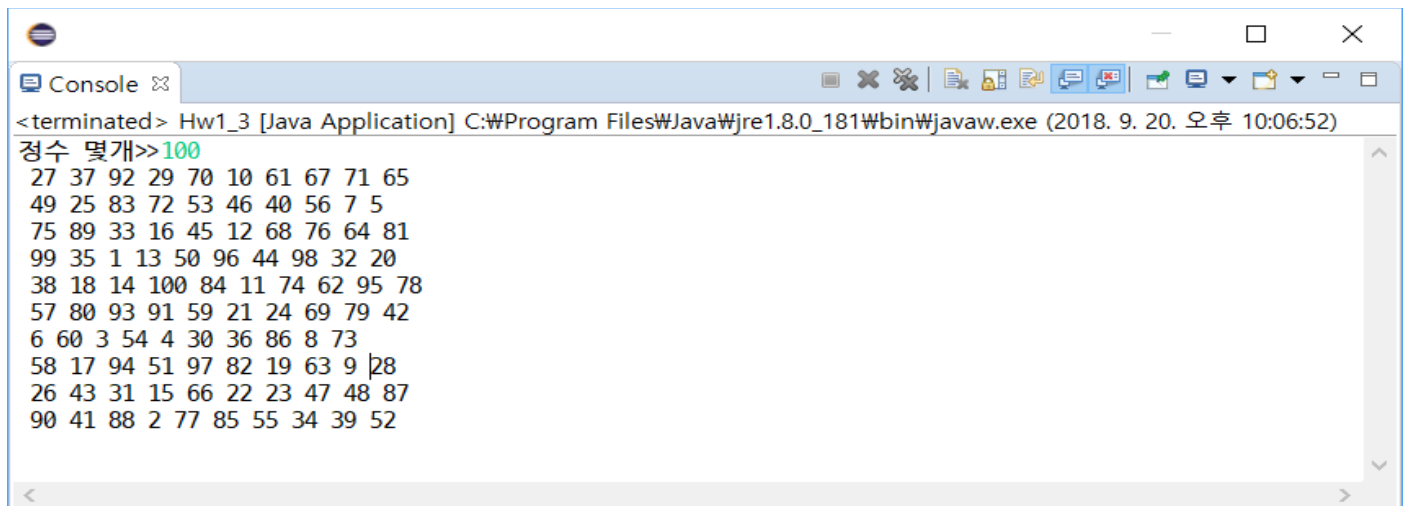
```
for (int i = 0; i < n; i++) {  
    ar[i] = (int) (Math.random() * 100 + 1);  
    for (int j = 0; j < i; j++) {  
        if (ar[j] == ar[i]) {  
            i--;  
            break;  
        }  
    }  
}
```

중복된 값이 발견되는 경우 인덱스를 1줄여 그 자리에 다시 랜덤 값을 저장하게 했습니다.

또 중복된 값이 저장되더라도 아래에서 계속 검사되므로 중복되지 않고 랜덤 값을 저장할 수 있었습니다.



```
<terminated> Hw1_3 [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018. 9. 20. 오후 10:05:52)  
정수 몇개>>24  
41 89 43 30 9 34 65 6 81 58  
53 61 33 64 59 17 39 54 49 55  
36 85 92 46
```



```
<terminated> Hw1_3 [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018. 9. 20. 오후 10:06:52)  
정수 몇개>>100  
27 37 92 29 70 10 61 67 71 65  
49 25 83 72 53 46 40 56 7 5  
75 89 33 16 45 12 68 76 64 81  
99 35 1 13 50 96 44 98 32 20  
38 18 14 100 84 11 74 62 95 78  
57 80 93 91 59 21 24 69 79 42  
6 60 3 54 4 30 36 86 8 73  
58 17 94 51 97 82 19 63 9 28  
26 43 31 15 66 22 23 47 48 87  
90 41 88 2 77 85 55 34 39 52
```

Hw1_4

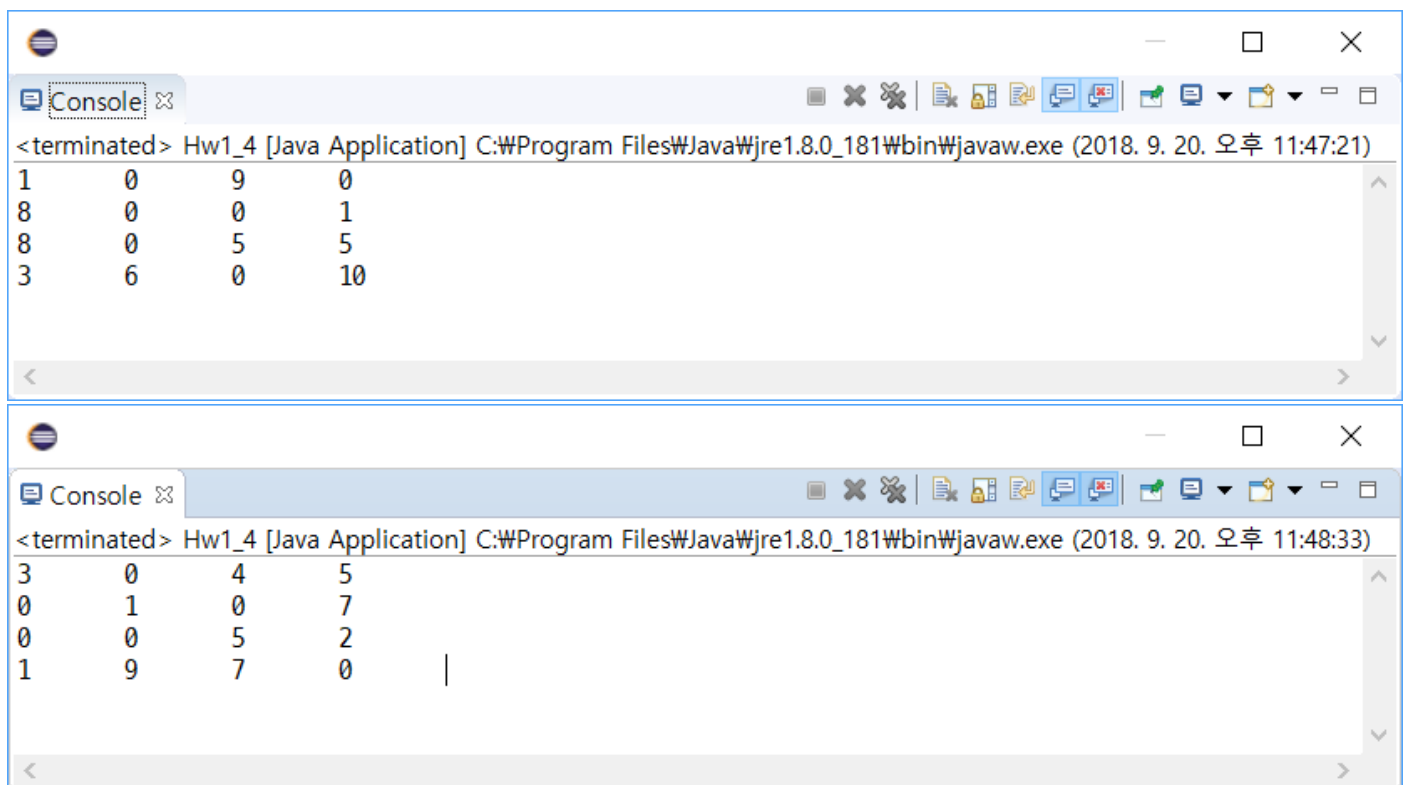
위치, 값 모두 랜덤이므로 아래 방법으로 2차원 배열의 좌표를 찾았습니다.

```
int x = (int) (Math.random() * 4);  
int y = (int) (Math.random() * 4);
```

저장되는 값의 범위가 1~10이므로 배열의 값이 0이라면 수정되지 않은 위치라고 판단했습니다.

이미 값이 저장되어 있는 위치면 다시 랜덤 위치를 저장하도록 했습니다.

```
for (int i = 0; i < 10; i++) {  
    int x = (int) (Math.random() * 4);  
    int y = (int) (Math.random() * 4);  
  
    while (ar[x][y] != 0) {  
        x = (int) (Math.random() * 4);  
        y = (int) (Math.random() * 4);  
    }  
  
    ar[x][y] = (int) (Math.random() * 10 + 1);  
}
```



Hw1_5

메뉴를 관리하는 MyConcertMenu클래스와 각 좌석을 기억하는 MySeat클래스를 만들었습니다.

MyConcertMenu는 MySeat 배열, 예약, 조회, 취소 메소드를 가지고 있습니다

MySeat클래스는 10개의 string 배열을 가지고 있으며 생성자에서 모두 “---“로 초기화 시킵니다.

또한 좌석이 비어 있는지 확인하는 메소드, 좌석을 설정하는 메소드, 좌석을 출력하는 메소드, 좌석을 비우는 메소드가 있습니다.

메인 함수에서는 MyConcertMenu 객체를 생성하고 무한반복문을 돌며 switch()를 이용해 입력에 맞는 메뉴 함수를 호출합니다.

다른 입력이 없이 출력만 하면 되는 조회와 좌석출력 메소드는 void형이고 나머지 메소드들은 전부 boolean형으로 구현해 정상적으로 수행하면 true를 오류가 생기면 false를 반환합니다.

또한 함수 호출부에 조건문으로 false가 반환된 경우 다시 되돌아가 입력 받을 수 있게 했습니다. Hw1_4번에서 사용한 방법과 같은 방법이며 아래는 예약취소 메소드 일부분입니다.

```
default:
    System.out.println("잘못된 좌석입니다.");
    return false;
=====
    boolean tmp = false;
    while (tmp == false) {
        System.out.print("이름>>");
        String name = in.next();
        tmp = seat[classNum - 1].DeleteMySeat(name);

        if (seat[classNum - 1].isEmpty() == true) {
            break;
        }
    }
    return true;
```

잘못된 좌석이 입력되었을 때 오류문장을 출력하고 false를 반환합니다. 이 후 이름을 입력 받았을 때는 좌석을 지우는 메소드에서 정상적으로 지워졌거나 해당 좌석이 전부 비어 있을 때 반복문을 빠져나와 true를 리턴합니다.

```
case 3:
    tmp = menu.CancleMySeat();

    while (tmp == false) {
        tmp = menu.CancleMySeat();
    }
    break;
```

예약취소 메소드를 호출하는 메인함수 부분도 같은 방법입니다. 각 함수를 사용하는 입장에서는 이 함수가 true인지 false이지만 확인하여 false일 때 다시 입력 받습니다.

```

<terminated> Hw1_5 [Java Application] C:\Program Files\Java\jre1.8.0_1
명품콘서트홀 예약 시스템입니다.
예약1, 조회:2, 취소:3, 끝내기:4>>1
좌석 구분 S(1), A(2), B(3)>>2
A>> --- --- --- --- --- --- ---
이름>>박상우
번호>>5
예약1, 조회:2, 취소:3, 끝내기:4>>1
좌석 구분 S(1), A(2), B(3)>>2
A>> --- --- --- 박상우 --- --- ---
이름>>김민수
번호>>1
예약1, 조회:2, 취소:3, 끝내기:4>>1
좌석 구분 S(1), A(2), B(3)>>1
S>> --- --- --- --- --- --- ---
이름>>김영희
번호>>7
예약1, 조회:2, 취소:3, 끝내기:4>>2
S>> --- --- --- 김영희 --- --- ---
A>>김민수 --- --- 박상우 --- --- ---
B>> --- --- --- --- --- --- ---
<<<조회를 완료했습니다>>>
예약1, 조회:2, 취소:3, 끝내기:4>>3
좌석 구분 S(1), A(2), B(3)>>2
A>>김민수 --- --- 박상우 --- --- ---
이름>>김민수
예약1, 조회:2, 취소:3, 끝내기:4>>2
S>> --- --- --- 김영희 --- --- ---
A>> --- --- --- 박상우 --- --- ---
B>> --- --- --- --- --- --- ---
<<<조회를 완료했습니다>>>
예약1, 조회:2, 취소:3, 끝내기:4>>1
좌석 구분 S(1), A(2), B(3)>>1
S>> --- --- --- 김영희 --- --- ---
이름>>이철수
번호>>7
이미 예약된 좌석입니다.
이름>>이철수
번호>>2
예약1, 조회:2, 취소:3, 끝내기:4>>2
S>> --- 이철수 --- --- 김영희 --- --- ---
A>> --- --- --- 박상우 --- --- ---
B>> --- --- --- --- --- --- ---
<<<조회를 완료했습니다>>>
예약1, 조회:2, 취소:3, 끝내기:4>>4

```

같은 줄 결과 설명입니다.

예약입니다.

2번 A석

박상우, 5번째 자리

2번 A석

김민수, 1번째 자리

1번 S석

김명희, 7번째 자리

조회입니다.

삭제입니다.

2번 A석

김민수 삭제

조회 - 김민수가 삭제됨을 확인할 수 있습니다.

예약

1번 S석

이철수, 7번째 자리

//이미 예약된 자리이기에 오류를 표시합니다.//

새로 입력 받고

결과 조회 후

프로그램을 종료했습니다.

맨 위 <terminated>로

종료를 확인할 수 있습니다.

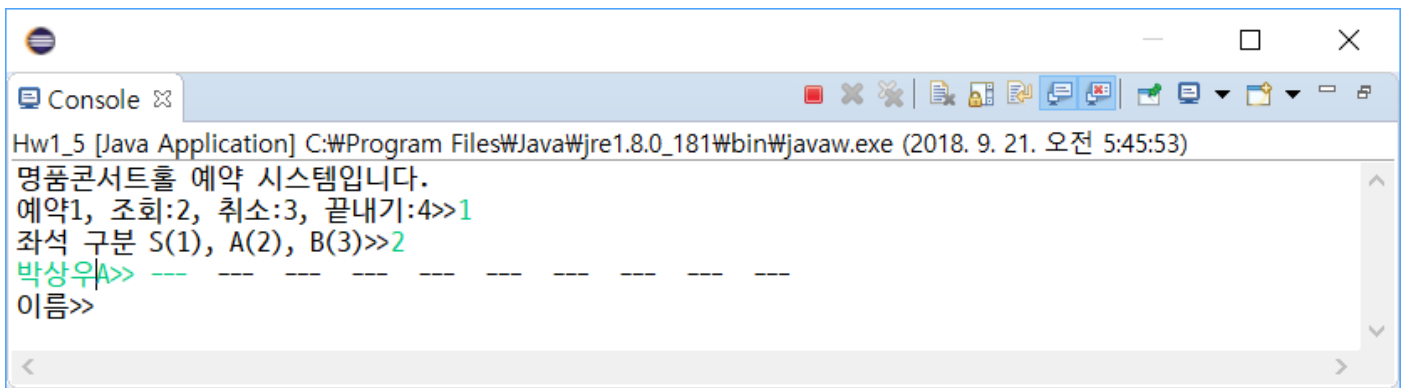
```
Console
<terminated> Hw1_5 [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018. 9. 21. 오전 5:22:55)
명품콘서트홀 예약 시스템입니다.
예약1, 조회:2, 취소:3, 끝내기:4>>6
잘못된 입력입니다.
예약1, 조회:2, 취소:3, 끝내기:4>>1
좌석 구분 S(1), A(2), B(3)>>5
잘못된 좌석입니다.
좌석 구분 S(1), A(2), B(3)>>1
S>> --- --- --- --- --- --- --- --- ---
이름>>박상우
번호>>14
잘못된 좌석번호입니다.
번호>>6
예약1, 조회:2, 취소:3, 끝내기:4>>3
좌석 구분 S(1), A(2), B(3)>>1
S>> --- --- --- --- 박상우 --- --- --- ---
이름>>김철수
이름이 없습니다.
이름>>박상우
예약1, 조회:2, 취소:3, 끝내기:4>>2
S>> --- --- --- --- --- --- --- --- ---
A>> --- --- --- --- --- --- --- --- ---
B>> --- --- --- --- --- --- --- --- ---
<<<조회를 완료했습니다>>>
예약1, 조회:2, 취소:3, 끝내기:4>>3
좌석 구분 S(1), A(2), B(3)>>1
S>> --- --- --- --- --- --- --- --- ---
이름>>박상우
이름이 없습니다.
예약1, 조회:2, 취소:3, 끝내기:4>>2
S>> --- --- --- --- --- --- --- --- ---
A>> --- --- --- --- --- --- --- --- ---
B>> --- --- --- --- --- --- --- --- ---
<<<조회를 완료했습니다>>>
예약1, 조회:2, 취소:3, 끝내기:4>>4
```

잘못된 입력이 들어왔을 때 처리되는 모습들입니다.

메뉴와 좌석번호는 오류를 출력하며 다시 입력하도록 했습니다.

예약취소 시에는 이름이 하나라도 있으면 다시 이름을 입력하도록 하였고 모든 좌석이 빈 상태면 처음 메뉴로 돌아가도록 하였습니다.

과제를 하면서 해결하지 못한 부분입니다.



```
Hw1_5 [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018. 9. 21. 오전 5:45:53)
명품콘서트홀 예약 시스템입니다.
예약1, 조회:2, 취소:3, 끝내기:4>>1
좌석 구분 S(1), A(2), B(3)>>2
박상우A>> ---
이름>>
```

문제를 풀면서 이클립스 콘솔창에서 입력 시 커서가 제 위치에 있지 않아 Scanner가 잘못된 값을 입력 받고 예외를 던지는 경우가 많았습니다. 위 상황을 보면 1 enter, 2 enter 후 A>> ---로 출력됐지만 커서는 움직이지 않아 이름이 엉뚱한 곳에 입력되었습니다.

C언어처럼 입력버퍼가 남아있는 경우라고 생각해 인터넷에서 입력버퍼를 지우는 방법을 찾았고 `in.nextLine()`을 중간에 삽입해서 버퍼를 지우는 방법을 썼지만 계속 엉뚱한 위치에 커서가 있었습니다.

결국 입력이 필요할 때마다 마우스나 아래방향키로 커서를 제 위치로 이동시킨 후 값을 입력받았습니다.