2019년 2학기 객체지향프로그래밍
-3차 프로그래밍 과제

Swing을 이용한
좌석 예약 프로그램

실습반	318호 / 남경진 조교
이름	신지수
학번	201621818
학과/학년	심리학과 / 4

Ⅰ. 서론

본 과제는 1차 프로그래밍 과제에서 개발한 좌석 예약 프로그램을 발전시킨 프로그램이다. 이전의 프로그램에서는 콘솔창을 통해 예약을 진행했던 것과는 달리, 이번 프로그램에(이하 프로그램)서는 Swing을 이용하여 그 내용을 시각화하였다.

프로그램은 총 세 가지 기능인 '예약하기'와 '예약 취소하기', 그리고 '예약 보기'를 수행한다. '예약하기'는 연, 월, 일 정보를 입력하고 좌석 버튼을 누르면 사용자가 입력한 정보에 예약 가능한 시간대를 버튼으로 제공한다. 예약은 연속적으로 진행 가능하며, 반드시 같은 날짜나 좌석이 아니어도 예약을 진행할 수 있다. 모든 예약을 마치면 메뉴화면으로 돌아간다. '예약 취소하기'와 '예약 보기' 기능은 함께 수행된다. 연, 월, 일 정보를 입력하고 좌석 버튼을 누르면 입력한 날짜와 좌석에 해당하는 예약 정보를 볼 수 있다. 등록된 예약 정보는 활성화된 버튼으로 표시된다. 활성화된 예약 시간대 중 취소를 원하는 시간대를 누르면 예약을 취소한다.

모든 기능들은 1차 프로그래밍 때 작성한 코드를 수정하여 구현하였으며, GUI는 java에서 제공하는 swing의 components(JFrame, JPanel, JButton, JLabel, JTextField 등)을 이용하여 구현하였다. 기능의 수행은 주로 버튼을 통해 이루어지기에 JButton에 ActionListener를 구현하여 맡은 기능을 수행하게 하였다.

II. 본론

1. 프로그램 설명

본 좌석 예약 시스템은 5x5 행렬 형식의 좌석을 제공한다. 각 좌석의 번호는 <i, j>로 나타내며, 이 때 i는 행 번호, j는 열 번호를 나타낸다. 또한 i와 j는 1부터 5 사이의 정수이다. 예약은 9:00부터 18:00까지 가능하며, 매 시 정각을 기준으로 한 시간 단위로 예약이 가능하다. 또한 사용자는 한 번에 한 좌석의 하나의 시간대에 대하여 예약과 취소가 가능하다.

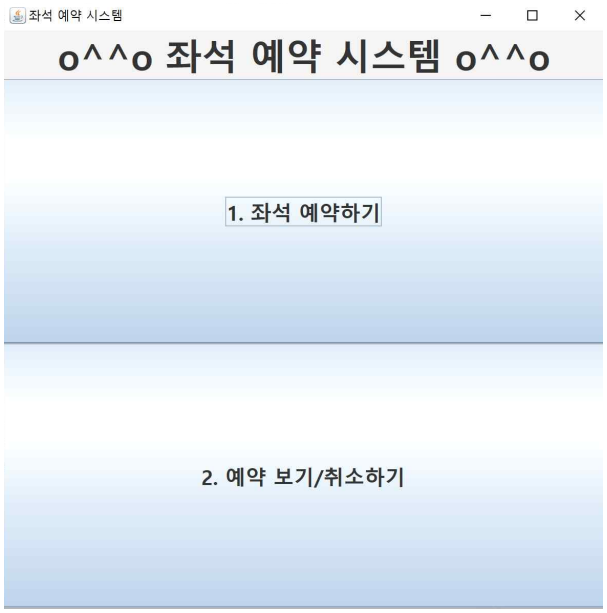

그림 1 프로그램의 메뉴화면

프로그램은 시작과 동시에 초기화면인 메뉴화면을 출력한다. 메뉴화면에서는 기능을 선택하는 버튼들을 함께 출력한다. 원하는 기능의 버튼을 누를 경우, 해당 기능을 수행하는 창으로 넘어간다.

그림 2 좌석 예약하기 화면

메뉴화면에서 ‘좌석 예약하기’ 버튼을 클릭할 경우, 예약하기 화면으로 넘어간다. 예약하기 화면의 예시를 위에 그림 2로 제시하였다. 상단에 예약을 희망하는 연, 월, 일을 입력할 수 있다. 프로그램을 실행하는 당일의 날짜가 기본값으로 적혀있으며, 만일 사용자가 날짜를 따로 입력하지 않는다면 당일에 대한 좌석 예약이 진행된다. 날짜를 기입하고 예약을 희망하는 좌석을 클릭하면, 예약이 가능한 시간대를 활성화된 버튼으로 나타낸다. 예약은 한 시간 단위로 등록할 수 있으며, 한 번에 한 시간대에 대해서만 가능하다. 예약을 원하는 시간대를 누르면, 재차 예약 희망 여부를 묻는다. 확실히 예약을 하고자 한다면 예약이 확정된다. 예약은 연속적으로 진행된다. 하나의 예약을 마친 이후 다른 시간대에 대한 예약을 진행할 수 있으며, 다른 날짜 또는 다른 좌석에 대해서도 예약이 가능하다. 모든 예약을 마치고 우측 하단의 ‘예약 완료’ 버튼을 누르면 다시 초기화면으로 돌아간다.

메뉴화면에서 ‘좌석 보기/취소하기’ 버튼을 클릭할 경우. 좌석 보기/취소하기 화면으로 넘어간다. 좌석 보기/취소하기 화면의 예시는 다음 장에서 그림 3으로 제시하였다. 날짜의 입력과 좌석 버튼에 대한 입력은 모두 예약하기 기능의 것과 동일하다. 그러나 좌석 보기/취소하기 화면에서는 등록된 예약이 존재하는 시간대를 활성화된 버튼으로 나타낸다. 특정 날짜와 좌석에 대한 예약 현황을 확인한 뒤 취소하기를 원하는 예약 시간대를 누르면, 취소를 재차 확인한 뒤에 예약을 취소한다. 예약 취소도 연속적으로 수행이 가능하며, ‘취소 완료’ 버튼을 누를 경우, 수행을 모두 마치고 초기화면으로 돌아간다.

그림 3 좌석 보기/취소하기 화면

2. 클래스

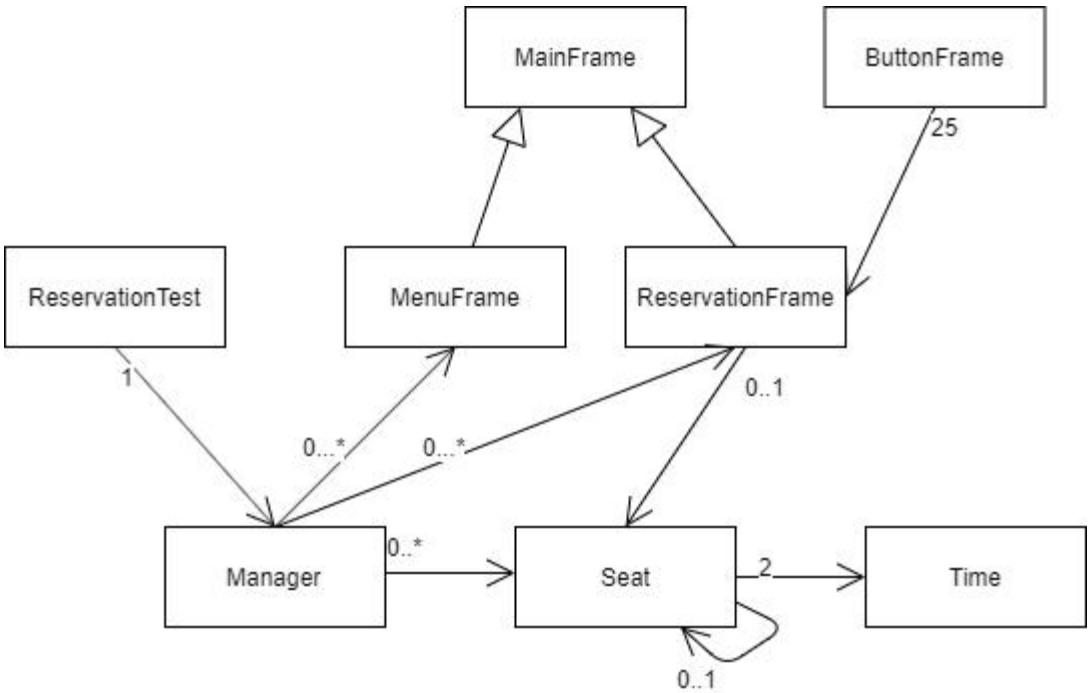

그림 4 클래스 다이어그램

기능을 구현하는 클래스인 Time, Seat, Manager, ReservationTest는 이전의 프로그램의

것을 가져와 수정하여 사용하였다. Time은 시간 정보를 변수로 갖는 클래스이며, 날짜의 유효성을 판단하는 static 함수를 가진다. Seat 클래스는 예약정보를 저장하는 클래스로, 예약시간과 좌석 정보를 저장한다. Manager 클래스는 실질적으로 예약을 진행하는 클래스이다. 사용자로부터 메뉴 항목을 입력 받는 함수 printMenu() 메소드와 예약 및 예약 취소를 담당하는 reserve() 메소드를 가진다. ReservationTest 클래스는 메인함수를 가진다.

위의 네 클래스를 제외하고, 추가적으로 GUI를 담당하는 네 클래스를 더 생성하였다. MainFrame, MenuFrame, ReservationFrame, ButtonFrame이 그것이다.

MainFrame

MenuFrame과 ReservationFrame의 super class로서, 두 하위 클래스의 기본틀이 된다. 이 프로그램에서는 메뉴화면과 예약화면을 각각 클래스로 정의하여 구현하였다. 따라서 두 화면에 대해 공통적으로 필요한 getFrame(), addButton() 등의 메소드를 가진다.

MenuFrame

MenuFrame은 메뉴화면을 구현하는 JFrame인 menuFrame을 객체로 가지며 화면을 구성하는 패널, 라벨, 버튼들로 구성되어있다. 또한 사용자가 메뉴를 버튼을 통하여 선택하는데 이 정보를 타 클래스로 보내기 위한 getChoice() 메소드를 가진다.

ReservationFrame

ReservationFrame은 예약/예약취소 화면을 구현하는 JFrame인 rsrvFrame을 객체로 가진다. 예약하기 화면과 예약 취소하기 화면을 따로 만들지 않았는데, 그 이유는 두 화면의 UI 디자인이 같고, 기능만 다르기 때문이다. 따라서 동일한 프레임과 구성요소들을 이용하여 화면을 구성하고, 메뉴에서 선택한 옵션에 따라 기능을 다르게 동작하도록 하였다. ReservationFrame은 Frame과 여러 패널, 버튼, 라벨, 텍스트 필드로 구성되어있다. 또한 예약 화면에서 사용자가 입력한 옵션들을 저장하고 타 클래스로 전달하기 위해 여러 필드변수들과 메소드들을 가진다.

MenuFrame과 ReservationFrame에 속한 swing components들에 대한 자세한 설명은 본문 3장에서 설명한다.

ButtonFrame

ButtonFrame은 좌석 한 칸을 의미하는 클래스로 특정 좌석에 해당하는 버튼을 생성할 수 있다. 예약화면의 중앙에 위치하는 5x5 그리드 버튼들은 JButton이 아닌 ButtonFrame 객체이다. 이는 좌석의 위치를 나타내는 행, 열 정보를 버튼과 함께 저장하기 위함이다.

3. 프로그램의 진행과 Swing Components

프로그램의 시퀀스 다이어그램을 그림 5로 제시하였다. 프로그램을 위하여 선언한 클래스는 총 8개이지만 보다 간단하고 직관적인 이해를 위하여 몇 클래스는 그림 5에 나타내지 않았다. Time 객체의 경우 Seat 객체가 생성됨과 동시에 Seat의 필드 객체로 선언된다. 따라서 Time 객체는 Seat 객체와 그 운명을 같이 하기 때문에 다이어그램에서 생략하고 Seat 객체로만 나타냈다. ReservationTest 클래스는 메인함수만 존재하기에, 클래스 대신 사용자(actor)를 두었다. 이를 통해 보다 사용자 관점에서 다이어그램을 이해하도록 하였다. 또한 MainFrame은 MenuFrame과 ReservationFrame의 super class이고 따로 사용되거나 하지 않기 때문에 다이어그램 상에 나타내지 않았다. 마지막으로 ButtonFrame은 ReservationFrame의 생성과 동시에 스물다섯 개의 객체를 생성한다. 그러나 프로그램의 진행과 메시지 전달에서 객체가 큰 역할을 하지 않기 때문에 다이어그램 상에서 제외하였다.

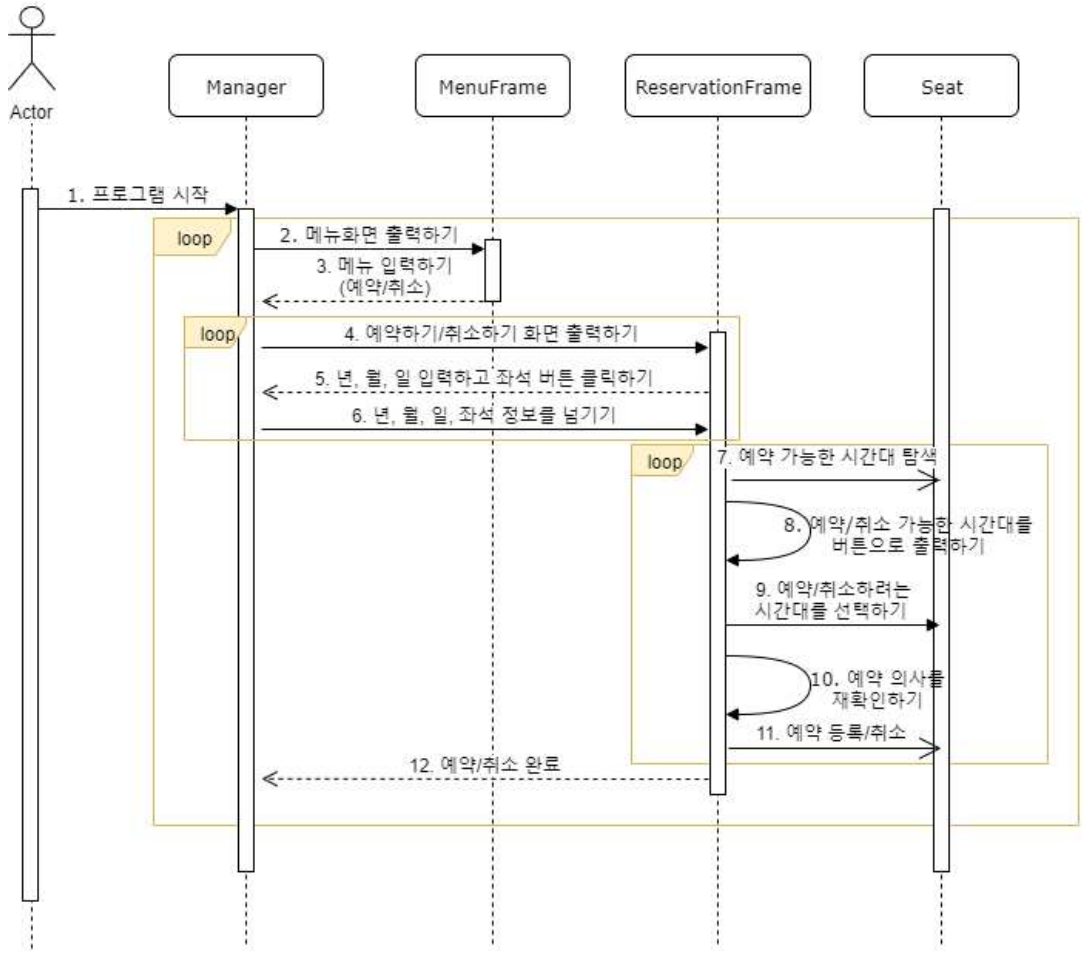

그림 5 시퀀스 다이어그램

그림 6 메뉴화면의 swing components

프로그램은 예약 또는 예약 취소를 무한 반복하는 프로그램으로, Manager 객체가 사용자의 요청에 대한 수행을 담당하는 역할을 한다. 가장 처음 프로그램이 시작할 때, 메뉴화면이 뜨고 사용자로부터 원하는 기능을 버튼을 통해 입력받는다. 메뉴화면의 구성은 다음과 같다.

메뉴화면의 프레임은 JFrame을 사용하였다. 또한 Toolkit 객체를 이용하여 프레임의 사이즈와 위치를 설정하였다. 화면에는 환영 인사말을 출력하는 JLabel과 사용자의 액션을 입력받는 JButton 두 개로 구성된다. 두 버튼은 JPanel에 추가한 뒤, 해당 패널을 프레임에 추가하는 식으로 구성하였다. 버튼은 GridLayout을 통해 세로로 배치하였고, JLabel과 JPanel은 BorderLayout을 통해 북쪽과 중앙에 배치하

```
btn.addActionListener(event->{
    choice = n;
});
```

그림 7 버튼에 등록된 ActionListener

였다.

사용자가 버튼을 눌렀을 때, 그에 반응하기 위해 버튼에 ActionListener를 추가하였다. 버튼을 생성하면서 좌석 예약하기 메뉴를 1로, 예약 보기/취소하기 메뉴를 2로 설정하여 사용자의 입력을 숫자로 이해하게끔 하였다. 버튼이 눌릴 경우, 버튼의 ActionListener로 인해 MenuFrame 클래스의 choice 변수가 해당 입력값(1 또는 2)으로 변경된다. 이후 Manager에서 choice의 값을 가져와 사용자의 입력을 이해한다.

사용자가 예약하기 메뉴를 클릭한 경우에 manager 객체는 reserve(true)를 호출하고, 예약 보기/취소하기 메뉴를 클릭한 경우에는 reserve(false)를 호출한다. reserve() 메소드를 통해 예약화면은 동일하게 띄우지만 매개변수를 달리함으로 기능과 기능에 걸맞은 UI를 달리한다.

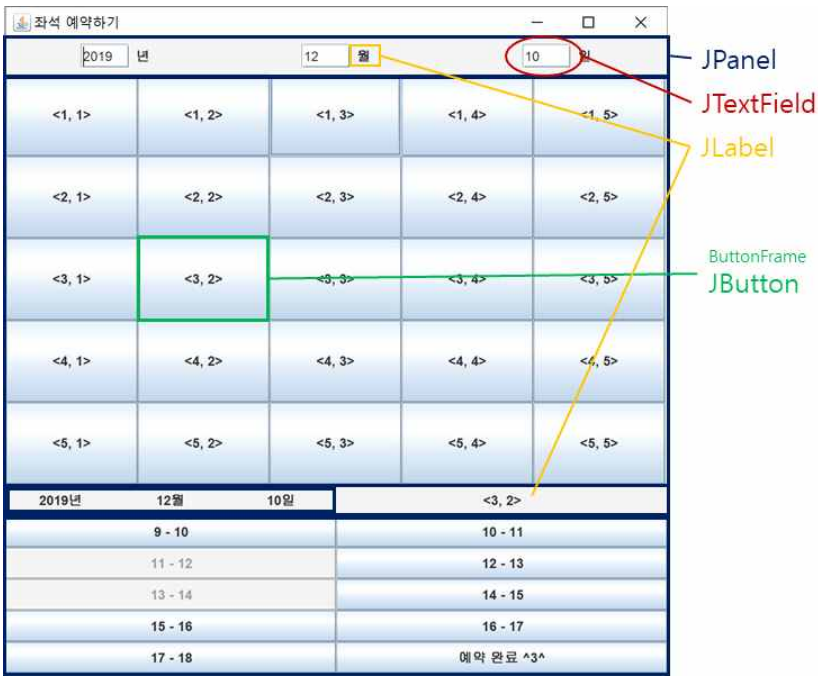

그림 8 예약하기 화면의 swing components

Manager 객체가 예약 관련 기능을 수행하려는 동시에 ReservationFrame 객체가 생성된다. 객체를 통해 출력되는 화면은 그림 8과 같다. 프레임은 JFrame으로 선언하였으며 메뉴화면에서 설정한 바와 같이 타이틀 설정, 프레임 크기 및 위치 설정을 하였다. 많은 정보들을 한 화면에 질서 있게 출력하기 위해 JPanel에 추가하고 GridLayout으로 정렬하는 방법을 사용하였다. 화면 상단은 사용자로부터 연, 월, 일 정보를 입력 받는 JTextField가 있다. 예약일의 기본값은 프로그램을 시작하는 당일로 설정해야 하므로 LocalDate를 이용하여 프로그램 실행일의 연, 월, 일 정보를 받아 텍스트 필드에 함께 출력한다. 만약 사용자가 별도의 연, 월, 일 정보를 입력하지 않는다면 기본 값을 바탕으로 진행한다. 연, 월, 일 정보를 입력한 이후 중앙의 좌석 버튼을 클릭하면 연, 월, 일, 좌석의 행, 열 정보가 ReservationFrame의 필드 변수에 동기화되어 저장된다. 동기화 이후 해당 일자와 좌석에 해당하는 예약을 검색하여

예약 또는 취소가 가능한 시간대를 그림 8 하단의 버튼으로 활성화시킨다. (이 과정 때문에 예약일자나 좌석이 변경될 경우, 시간대 버튼도 갱신된다.) 이 과정을 좌석 버튼 각각에 ActionListener로 구현하였다. ActionPerformed 함수는 pseudo code 형식으로 작성하여 그림 9로 첨부하였다.

```
btn.addActionListener(event->{
    if(연, 월, 일 정보 미입력 또는 올바르지 않을 경우) {
         오류메세지를 알리는 Dialog Box 출력
    }
    else if(날짜, 좌석이 이전 예약의 내용과 다를 경우)
    {
        버튼과 텍스트 필드의 정보를 int 데이터로 변환
        해당 정보를 ReservationFrame 클래스 내 필드 변수에 동기화

        모든 시간 버튼을 활성화

        등록된 예약 내역을 확인하여 해당 날짜, 좌석에 해당하는 예약을 탐색하기
            만일 선택한 기능이 예약이면 버튼을 비활성화
            만일 선택한 기능이 취소이면 버튼을 활성화
    }
});
```

그림 9 좌석 버튼에 추가한 ActionListener

예약을 진행할 경우, 활성화된 버튼 중 원하는 시간대의 버튼을 하나 선택한다. 시간대 버튼의 ActionListener는 그림 10과 같이 예약을 재확인하는 Option Dialog 창을 띄우고, OK를 클릭하면 버튼 스스로를 비활성화 시키는 역할을 한다. 따라서 시간대 버튼을 누르면 ActionListener의 ActionPerformed 메소드를 수행하고, OK 옵션을 입력받은 뒤 예약 및 예약 취소를 성공적으로 수행한다. 만일 Cancel 옵션을 받는다면 예약을 하지 않는다. 예약 수행에 대한 정보는 Reservation 객체에서 Manager 클래스로 전달되며, Manager 클래스에서 Seat 객체를 생성하여 좌석 배열에 저장한다.

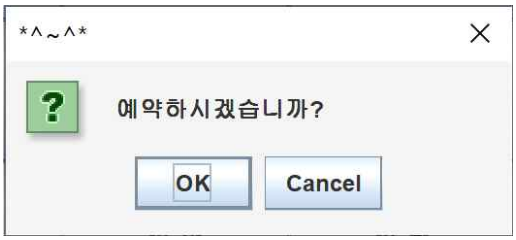

그림 10 예약 의사를 묻는 JOptionPane

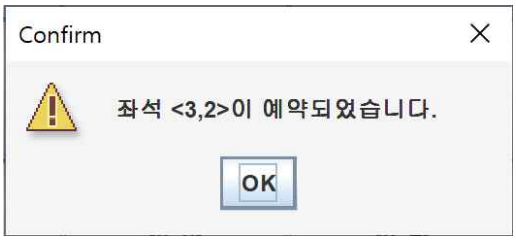

그림 11 예약을 완료한 상태

```
enterBtn.addActionListener(event->{
    isEnable = false;
    rsrvFrame.setVisible(false);
});
```

그림 12 완료 버튼에 추가된 ActionListener

예약 및 취소는 연속적으로 수행할 수 있으며, 다른 일자나 좌석에 대해서도 연속적으로 수행 가능하다. 만일 예약 및 취소를 모두 마쳤다면 우측 하단의 '예약완료/취소완료' 버튼을 통해 초기화면(메뉴화면)으로 돌아간다. 완료 버튼에 추가된 ActionListener는 그림 12와 같다. 완료 버튼을 클릭하면 현재 수행 중인 기능(예약 혹은 예약 취소) 기능을 중지한다는 의미로 isEnable 변수를 false로 변경하고, 예약창을 보이지 않게 한다. isEnable 정보는 Manager 객체의 스코프로 넘어가서 기능을 멈추고 reserve 함수를 반환케 한다.

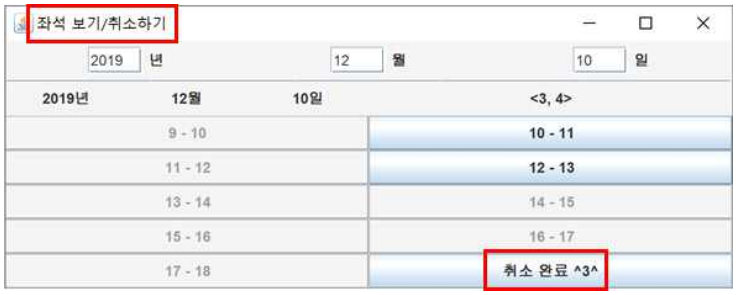

그림 13 예약 보기/취소하기 화면

예약하기 화면과 예약 보기/취소하기 화면의 UI는 거의 동일하지만 약간의 다른 점이 존재한다. 그 다른 점을 그림 13에 표시하였다. 그림 13에서 중앙의 좌석버튼은 생략하였다. 앞서 말했듯, 예약하기와 예약 취소하기 기능은 같은 메소드를 사용하지만, 인자를 달리하여 호출한다. 이 정보는 reserveFlag 변수에 저장하여, 이로써 하나의 ReservationFrame 객체가 두 가지 기능을 수행하게 한다. 그림 14의 코드는 그림 13에 나타난 예약하기 화면과 예약 보기/취소화면의 차이를 구현한 것이다. UI에서 드러난 차이 외에도 기능적인 차이 또한 이 reserveFlag를 이용하여 구현하였다.

```
setFrame(rsrvFrame, "좌석 " + (reserveFlag ? "예약하기" : "보기/취소하기"));
enterBtn = new JButton((reserveFlag ? "예약" : "취소") + " 완료 ^3^");
```

그림 14 reserveFlag를 이용한 예시

III. 결론

이번 과제는 기존에 해왔던 콘솔창을 이용한 프로그램에서 더 나아가 GUI를 이용한 프로그램을 개발하는 데 의의를 둔다. 이번 프로그램 개발을 통해 Swing Component의 종류와 그 쓰임새를 이해하였다. 또한 그래픽 요소를 통한 외부의 입력과 내부의 프로세스의 차이를 이해하고, 그 둘을 ActionListener와 메시지 패싱으로 연결지을 수 있었다. 실생활에서 접하는 대다수의 프로그램은 사용자 친화적인 인터페이스를 구현하여 사용자가 해당 프로그램을 다루기 용이하게끔 한다. 과제를 계기로 그러한 프로그램들을 이해하게 되었다.