

Java 開発演習

目次

1. 演習概要	1
1 目的	1
2 テーマ	1
3 演習形式	1
2. 演習詳細	2
1 Ver1.0 機能要件	2
2 Ver1.1 機能要件	4
3 Ver2.0 機能要件	4
4 Ver3.0 機能要件	5
5 注意事項	7

1. 演習概要

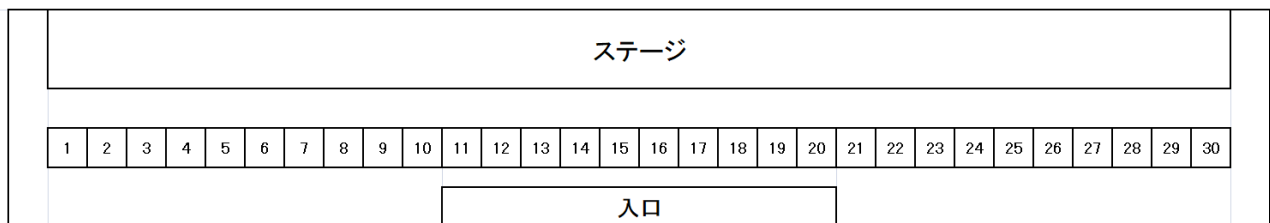
1 目的

Java 研修で学んだ講義内容の理解度、習熟度を測ることを目的とする。

2 テーマ

コンサートホール向け座席予約システムの開発を課題とする。本システムを導入する予定のホール（浅草ジャマイカホール）の座席の総数は 30 席であり、座席には 1 番～30 番までの座席番号が付けられている。座席は横一列に並んでいるものとする。

（※浅草ジャマイカホールの様子は下図に示す通り）



開発は Ver1.0、Ver1.1、Ver2.0、Ver3.0 の 4 段階に分けて行う（必ず前のバージョンを開発してから次のバージョンを開発すること）。各バージョンの仕様は「2. 演習詳細」で示す。

3 演習形式

- ① 本システムの開発は個人で行うものとする（個人演習）。
- ② 作成するプロジェクトの名前は以下の命名規則に従うこと。

reservation_氏名(ローマ字)

（例：東京太郎の場合）

reservation_tokyotaro

- ③ 作成するパッケージの名前は以下の命名規則に従うこと。

Ver1.0 standard
Ver1.1 enhancement
Ver2.0 excellence
Ver3.0 extra

- ④ 作成するファイルは以下の命名規則に従うこと。
※「2. 演習詳細」で示す。
- ⑤ これまでに学習したコーディング規約に従うものとする。

2. 演習詳細

1 Ver1.0 機能要件

ユーザーが座席を選択して予約する機能を実装する。座席の選択は、ユーザーに予約したい座席番号（1～30）を入力してもらい、その座席が既に予約済みであった場合は、「※予約済みの座席です。他の座席を選択してください」と出力しシステムを終了させる。予約可能な座席であれば、「座席の予約が完了しました。」と出力しシステムを終了させる。
（※例外処理は行わないものとする。）

画面（コンソール）のイメージは以下の通り。

① プログラム実行直後の画面（>の右にユーザーが座席番号を入力する）

```
ようこそ浅草ジャマイカホールへ！  
  
何番の座席を予約しますか？（1～30）  
>
```

② 既に予約済みの座席であった場合は以下のメッセージを出力

```
ようこそ浅草ジャマイカホールへ！  
  
何番の座席を予約しますか？（1～30）  
> 15  
  
※予約済みの座席です。他の座席を選択してください。
```

③ まだ予約されていない座席であった場合は以下のメッセージを出力

```
ようこそ浅草ジャマイカホールへ！  
  
何番の座席を予約しますか？（1～30）  
> 16  
  
座席の予約が完了しました。
```

このシステムの次の3つのクラスを連携させて作成する。

- SystemMain クラス (SystemMain.java) …… システムのメインとなるクラス
- InputController クラス (InputController.java) …… 入力処理クラス
- SeatsManager クラス (SeatsManager.java) …… 予約情報を操作するクラス

各クラスの仕様を以下に示す。

SystemMain クラス

フィールド	なし	利用しない。
メソッド	main	システムを開始した後、InputController クラスの inputSeatNum メソッドを呼び出して座席番号を取得し、その座席番号が予約可能であるかを SeatsManager クラスの checkVacantSeat メソッドを呼び出して調べる。予約可能であった場合はさらに SeatsManager クラスの reserveSeat メソッドを呼び出して座席を予約済みに変えた後、システム終了させる。予約不可であった場合はその旨のメッセージを表示してシステムを終了させる。
戻り値	void	利用しない。
引数	String[] args	利用しない。

InputController クラス

フィールド	なし	利用しない。
メソッド	inputSeatNum	標準入力を利用し、1～30 の範囲でユーザーに数値を入力させる。その後、入力値から1を引き、0～29 の範囲に補正した後、呼び出し元に補正後の数値を返す。
戻り値	int	ユーザーが入力した座席番号から-1 をした数値。
引数	なし	利用しない。

ArrayListでもOK

SeatsManager クラス

フィールド	boolean[] seatArray	座席の予約情報。要素数 30 の boolean 型の配列。true の場合はその座席は予約済み、false の場合は空席とする。
メソッド 1	checkVacantSeat	引数として受け取った番号の座席が空席であった場合は、true を返す。予約済みであった場合は false を返す。
戻り値	boolean	空席 : true 予約済み : false
引数	int seatNum	その座席が空席かを確認する座席番号(0～29)
メソッド 2	reserveSeat	引数として受け取った番号の座席を予約済み(true)に変える。
戻り値	void	利用しない。
引数	int seatNum	予約済みにする座席番号(0～29)

2 Ver1.1 機能要件

Exceptionをキャッチし処理

Ver1.0 で作成した SystemMain クラスの main メソッドから throws IOException を外し、main メソッド内で例外処理を行うように変更する。また、ユーザーが入力した番号の座席が既に予約済みであった場合は、再度入力させるように変更する。
(※変更するクラスは SystemMain のみ、それ以外のクラスは Ver1.0 と同じものを使う)

2-1 例外発生時に出力するメッセージ

システムエラーが発生しました。
システムを終了します。

2-2 座席番号を再度入力させる時に出力するメッセージ

※予約済みの座席です。他の座席を選択してください。

3 Ver2.0 機能要件

Ver1.1 に座席情報を表す Seat クラスを追加し、1 座席あたりの予約情報を Seat クラスのオブジェクトで表現する。このオブジェクト 30 個を List で扱うように変更する。(配列は使用しない) また、ソースコード中に埋め込まれている即値(ソースコード中に直に書き込まれた数値や文字列)を定数に直し、定数はまとめて Constant クラス(Constant.java)に記述する。この変更に伴い、SystemMain クラス、InputController クラス、SeatsManager クラス にも必要な修正を適宜加える。また、ユーザーに入力してもらう値は、Ver1.1 までの座席番号に加え、ユーザーの名前(name)も追加となる。この情報も Seat クラスのオブジェクト内に格納する。ユーザーの名前を入力するタイミングは座席番号を入力し、対象の席が空席だった時である。

- Seat クラス (Seat.java) の仕様を以下に示す。

Seat クラス

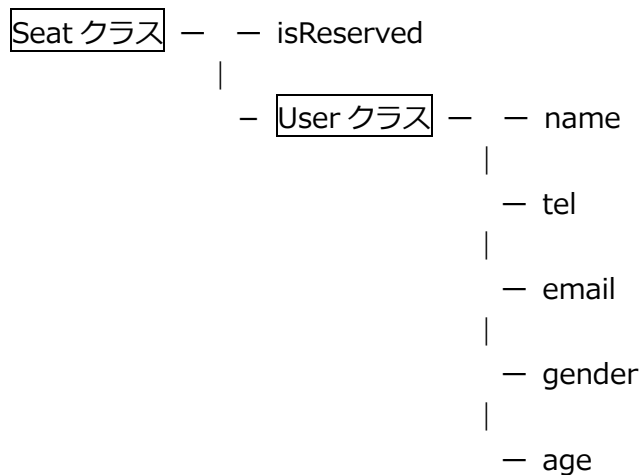
フィールド	String name	予約するユーザーの名前
フィールド	boolean isReserved	予約済み:true 空席:false

※これに加えて上記のフィールドに対応するゲッター、セッターも記述する

SeatsManagerクラス		
フィールド	public static final List<Seat> seats = new ArrayList<Seat>(0);	座席オブジェクトを保管する
メソッド	checkVacantSeat	引数として受け取った番号の座席が空席であった場合は、true を返す。予約済みであった場合は false を返す。
戻り値	boolean	空席 : true 予約済み : false
引数	int seatNum	その座席が空席かを 確認する座席番号 (0 ~ 2 9)
メソッド	reserveSeat	引数として受け取った番号の座席を予約済み(true) に変える。
戻り値	void	利用しない
引数	int seatNum, Seat seat	予約済みにする座席番号 (0 ~ 2 9)、座席オブジェクト
メソッド	init	座席オブジェクトを保管するSeatsフィールドの初期化を行う
戻り値	void	利用しない
引数	なし	
種別	static	

4 Ver3.0 機能要件

Ver2.0 にユーザー情報を表す User クラスを追加する。予約するユーザーに名前、電話番号、メールアドレス、性別、年齢の各項目を入力してもらい User クラスのオブジェクトに格納する。Seat クラスを変更し、フィールドにこの User クラスのオブジェクトを持つようにする。これとは別に予約済みか空席かの情報を持つ。つまり以下のようなデータ構造になる。



また、ユーザーによる座席番号の入力処理は行わず、独自のロジックに基づき予約可能な座席を自動的に選択し、予約済みにする処理に変更する。予約する座席の決定には3つのロジックが使われる。この座席選択のロジックを優先度が高い順に以下に示す。

- 1、両側の座席が空いている席を選択する（端の座席（※1番と30番）は選択しない）
- 2、両側の座席に座っている人が同性である席を選択する（端の座席は選択しない）
- 3、どこでもいいので空いている座席を選択する（端の座席も選択の対象とする）

上記のロジックを1→2→3の順に充てて行く。これらのロジックは全て新規に作成する AutoSelectLogic クラス内のメソッドとして実装されることとする（メソッド名は自由）。また、その他のクラスについても必要な変更は行って良いこととする。

座席の予約が出来た場合は、最後に次のメッセージを出力する。

```
予約できた座席は○番です。
座席の予約が完了しました。
```

Ver3.0 の機能をテストする場合は、予め適当な予約状況にするコードを追記しておく。

画面（コンソール）のイメージは以下の通り

ようこそ浅草ジャマイカホールへ！

あなたのお名前を入れてください。

> TIS 太郎

あなたの電話番号を入れてください。

> 00011112222

あなたのメールアドレスを入れてください。

> sample@mail.com

あなたの性別を入れてください。

男性 . . . 1

女性 . . . 2

> 1

あなたの年齢を入れてください。

> 23

予約できた座席は 4 番です。

座席の予約が完了しました。

5 注意事項

- ① プログラム開始時点で、予め 15 番と 21 番の座席が予約されている状態を作るために、`checkVacantSeat` メソッド内に次のコードを記述することとする。

```
seatArray[14] = true;
seatArray[20] = true;
```
- ② ユーザーが入力する値に対するチェック機能は実装しないものとする。
- ③ 演習中に不明な点がある場合は、下記の手順に沿って作業を進めること。
 - 1. まずは自分で調べ、解決を試みる。
 - 2. 自身で調べても解決できない場合、他の受講生に相談や質問する。
 - 3. それでも解決できない場合、サポーターに対して質問する。
- ④ 各パッケージには同名のクラスが作成される構成である。
(例: Ver1.1 の `enhancement` パッケージの中には Ver1.0 の `standard` パッケージ内と同様に `SystemMain.java`、`InputController.java`、`SeatsManager.java` を作成すること)