Homework – Java 007

Swing

Root folder: `java_training/assignments/java_007/`

Background

You will be inspiring yourself from the code samples provided to you and may have to do a little bit of research on your own.

Assignments

- Head or Tail (GUI style)
- Game of Dice (GUI style)
- Refactoring Sample Code to use Lambdas and/or Anonymous Inner Classes
- Grid Layout Challenge

Head or Tail

folder: headortail

Quick Description

Based on your java_006/lambda assignment, you will refactor the game of Head of Tail to use Swing.

App features

Create a title “Head of Tail”

Have a button that goes “Click to Flip!”

Have a label which initially starts with “waiting…”

When the button is clicked, it will provide an output of the result

Have another button that goes “Reset” and when you press it, it resets the label to “waiting…”

Game of Dice

folder: gameofdice

Quick Description

Based on your java_006/lambda assignment, you will refactor the game of dice using Swing

App features

Inspire yourself from the previous Head or Tail assignment in this document to get your app design and working.

Refactoring Exercise to use Lambdas

folder: refactoring

Quick Description

The projects provided to you will be refactored to use lambdas as required.

Within the 'refactoring' folder, ensure you have a folder of the same name as what is provided to you.

- BasicSwingDemo
- BasicSwingDemo2
- BasicSwingDemo3
- BasicSwingDemo4
- BasicSwingJButtonDemo
- BasicSwingJCheckBoxDemo
- BasicSwingJListDemo
- BasicSwingJTextFieldDemo
- BasicSwingCaseStudy

Grid Layout Challenge

folder: gridlayoutchallenge

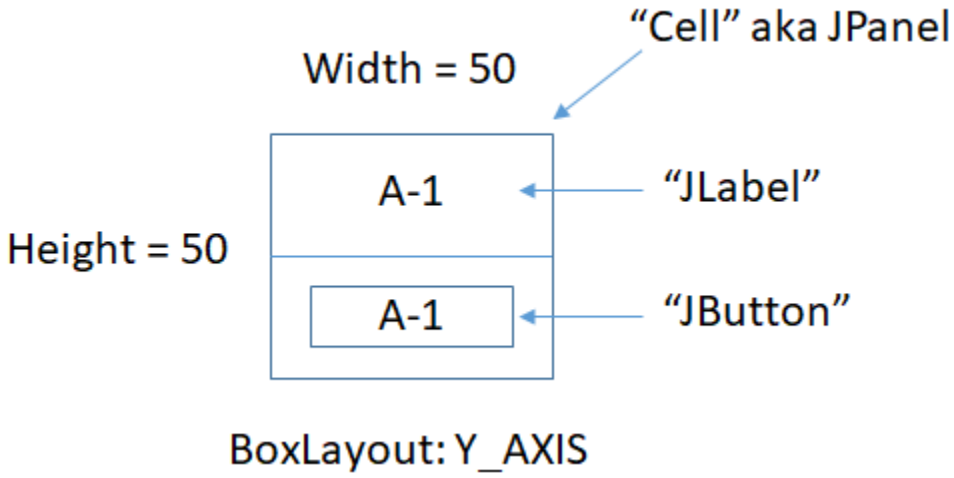
Quick Description

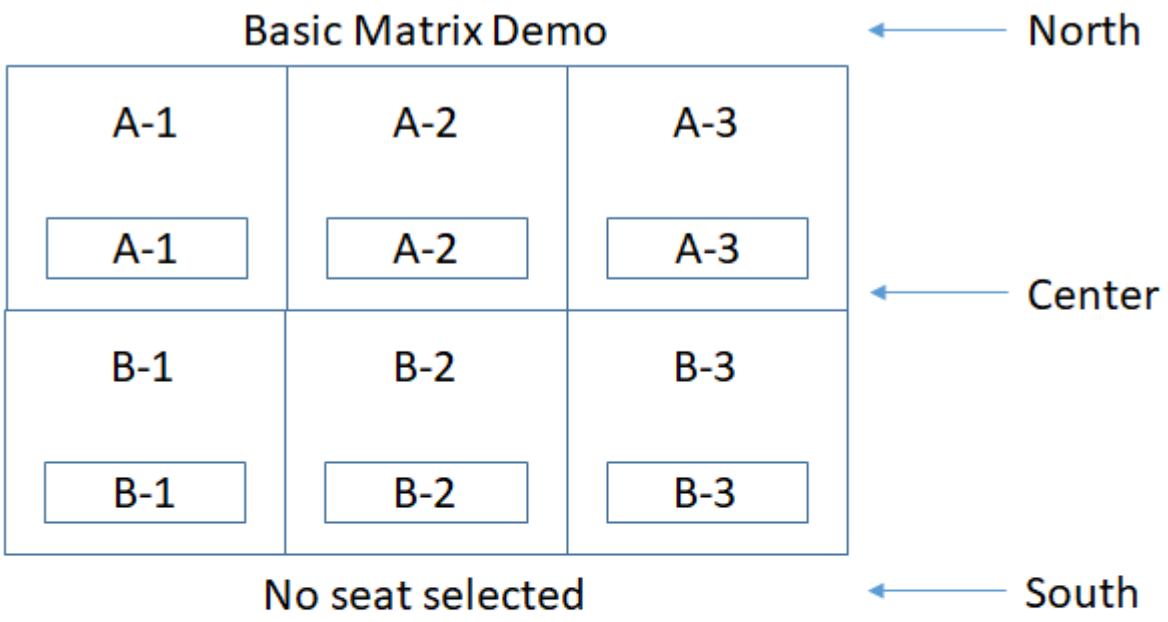
One core requirement of your pre-term assignment is to create a matrix of seats for a room (10x10).

Using Swing, here's the challenge

- Generate a random number for a count of rows and columns.
- rows between 2 and 5
- columns between 2 and 5
- Use a JFrame to hold your app.
- Create a border layout where you have a top, center and bottom region
- The top region is nothing more than a centered label with the words: "Basic Matrix Demo"
- The bottom region will contain a label that will contain "No seat selected"
- The center region will contain a panel that will host a grid layout.
- Your 'grid layout' will be dynamic based on the random values you generated
    - this means that you will allocate 50 pixels per columns and 50 pixels per row
    - Add to your JFrame an extra 80 pixels in height to accommodate your top/bottom region
- Your output should be coordinates based on letters A to .. for your rows and 1 to .. for your columns.
- Each "cell" so to speak will be a panel which will contain a BoxLayout with Y_AXIS, the top part is the label of the "cell" and the bottom part is a button which will contain the same content as the above created label.

Visual Specs





Randomly generated 3 columns and 2 rows example above
Expect a Border Layout
User Interaction: click on a button and the label in the south
Will display the selected seat: “B-1” for example.
Height of your JFrame should be:

```
(Count of Rows * 50 pixels) + 80 pixels
```