HW 11: March Madness Simulator

Release date: April 1st
Due date: April 8th, 11:59pm

Goals:

- Familiarization of the basics of GUI implementation using JOptionPane from Java Swing

Description:

For this homework, you will be creating a simple GUI interface to simulate games in the annual NCAA March Madness tournament. This program will ask for the team names, the win percentage for each team (the chance of each respective team winning, must add up to 100) , and number of simulations that you want to run. The goal of this program is to decide the number of times each team wins based on the number of simulations you will run. You will utilize **JOptionPane** methods in order to complete the homework. Your program should be written in a class named **MarchMadness.java**. Below are the required tasks to be implemented.

Task 1: Implement the team name and seeding pop-up windows
Task 2: Implement the simulation count pop-up window
Task 3: Implement the simulations
Task 4: Implement the result pop-up windows.
Task 5: Implement the repetition pop-up windows.

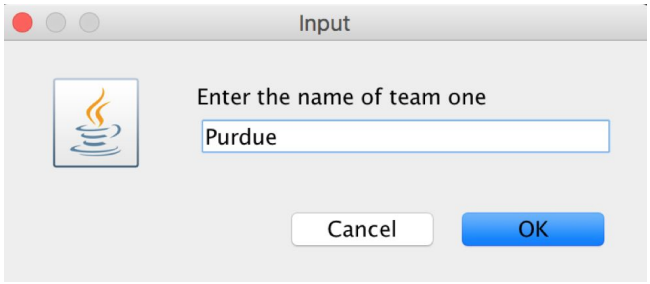
Task 1: Implement the team name and seeding pop-up windows

For this task, you are going to implement a set of pop-up windows to gather the names and win percentages for the two competing basketball teams

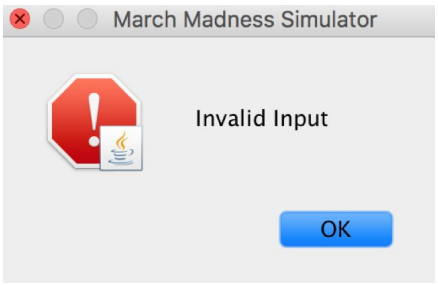
Explanation:
First, you are going to create a pop up window for collecting the first team's name from the user. If the name that the user inputs is empty, you need to send an error dialog and ask for the name again. Keep doing this until the user enters a proper string. You will do the same to get the second team's name.

Once you take in the name inputs, you will need two different dialogs to take in the win percentage for each team. **Remember, the win percentages between the two times must add up to 100 and neither can be above 100 or below 0!** If the percentages don't add up to 100, the user will be notified with an ERROR_MESSAGE message box. Keep asking the user for these inputs, until they fit the constraints.

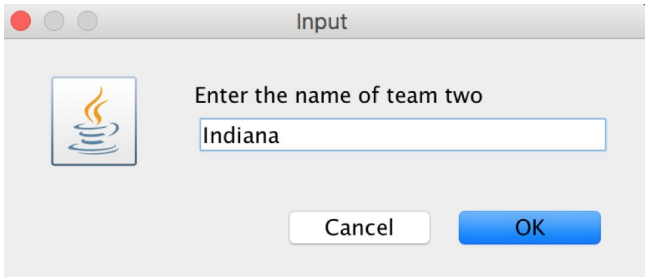
Example/Demonstration:



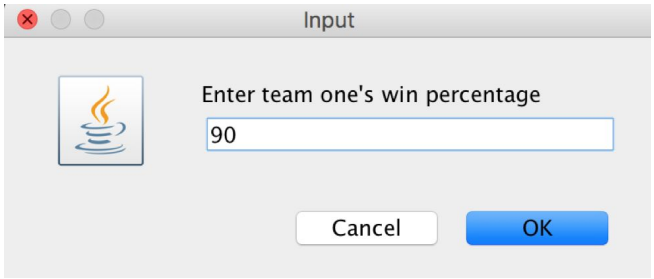
Name one input message



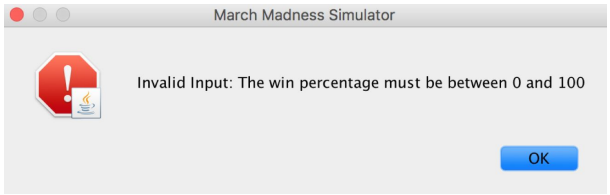
Name error message



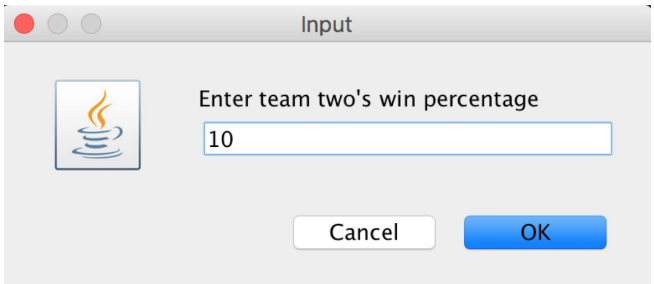
Name two input message



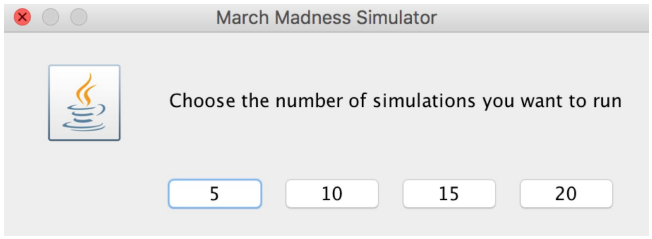
Team one win percentage input



Invalid input for win percentage message



Team two win percentage input



Simulations input message dialog

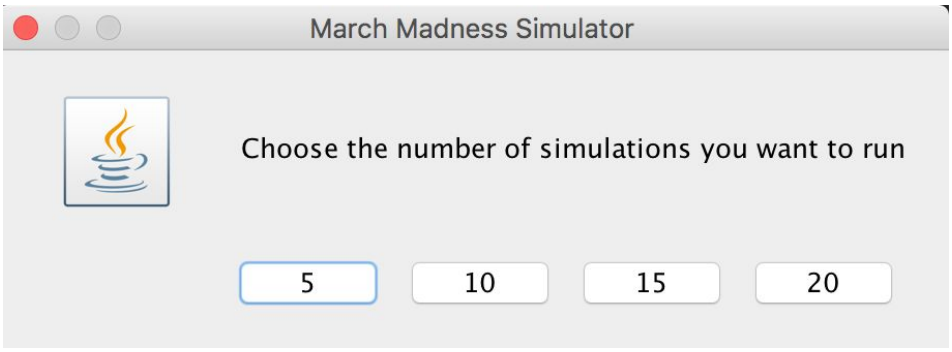
Task 2: Implement the simulation count pop-up window

For this task, you will be implementing the option dialog to choose the number of times the two teams play against each other.

Explanation:

You will need to use the JOptionPane option dialog for this. Look up the Java API for more information. You will be populating the dialog with four different choice buttons. The user will be able to select 5, 10, 15, or 20 simulations to run.

Example/Demonstration:

Main pop-up window


Task 3: Implement the simulations

Explanation:

Now that the user has entered the inputs and the number of simulations that they want to run, now you need to calculate the number of times that each team wins based on their win percentages.

Task 4: Implement the result pop-up windows.

For this task, you are going to implement the message dialog that displays the results of the simulation.

Example/Demonstration (20 simulations):

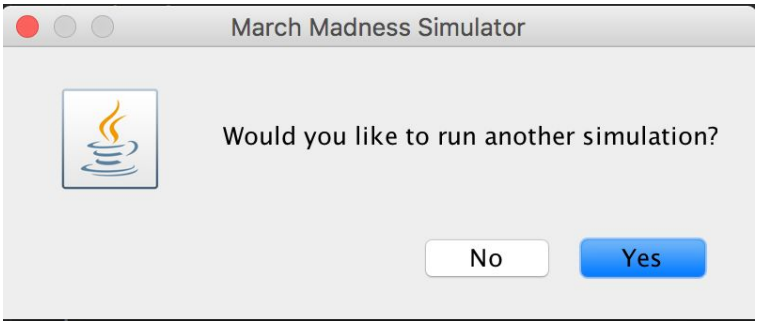
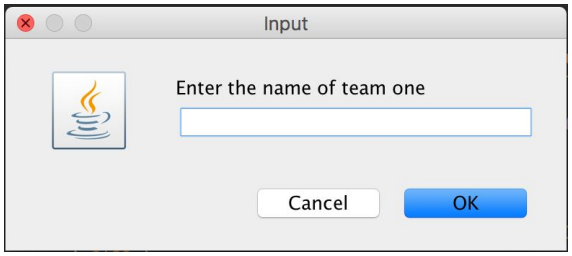
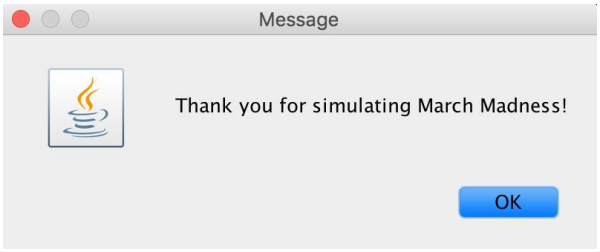
Result windows


Task 5: Implement the repetition pop-up windows.

For this task, you are going to implement the set of pop-up windows asking whether to do the simulations again with new teams.

Explanation:
After showing the result, there will be a **YES_NO_OPTION** window asking the user whether to do it again. If the user chooses yes, send them back to the first screen where you take in the first team's name. If the user chooses the no option instead, send them a thank you dialog and exit the program.

Example/Demonstration:

<table>
<tr><th colspan="2">Repetition sequence</th></tr>
<tr><td colspan="2"> 
</td></tr>
<tr><th>Yes</th><th>No</th></tr>
<tr><td>(Go back to the first window)
 
</td><td> 
</td></tr>
<tr><td>(Go back to the input window)</td><td>--the program ends--</td></tr>
</table>

Starter Code
There is no starter code for this assignment

Submission

Items required for submission via Blackboard:

- **MarchMadness.java**

Rubric

- 20 pts. for input in task 1
- 20 pts. each for results in task 2, 3, 4
- 20 pts. for loop in task 5

NOTE: This homework will be fully manually graded.