Collections, Inner Classes and GUI

50 points

Deal or No Deal is a popular television game show which aired in the United States from 2005 until 2009. If you're not familiar with this game, you can find more information at http://en.wikipedia.org/wiki/Deal_or_No_Deal. You can also view an episode on YouTube at http://www.youtube.com/watch?v=z5JcbsUZmzo.

For this assignment, you will create a modified version of this game using the provided program code skeletons. You are not required to use the provided skeletons if you prefer to design your own classes. The skeleton code is only provided to give you a starting point if you want a head start in where to begin.

Your program must use the standard Java Swing API. You cannot use other third party GUI APIs or frameworks. Additionally, you cannot use a GUI builder / code generator such as NetBeans to create the user interface.

Original Television Version

Deal or No Deal is played by one contestant at a time. The game starts with 26 closed briefcases that hold a varying amount of cash – from a penny to one million dollars. Each briefcase displays a number on the outside of the case from 26 to 1. The game begins with the contestant selecting one briefcase to be their briefcase to keep at the end of the game if they choose. Through a series of rounds, the contestant selects a predetermined number of briefcases from those still in play. The cash value is revealed and the case is removed from play. After removing the number of cases for that round, the Banker makes an offer of cash in exchange for the contestant's chosen briefcase. At this point, the contestant must decide whether to accept the Banker's offer (Deal) or continue removing briefcases from play (No Deal.) In each round, the contestant removes fewer cases from play; the first round begins with six cases to be removed, the second round with five, and decreasing until the final rounds require the removal of one case at a time. If lower dollar amounts are removed, the Banker's offer amount will increase; likewise if higher amounts are removed, the offer value will decrease. If the contestant refuses the Banker's final offer, they have the option of trading their briefcase for the one remaining briefcase and winning that amount.

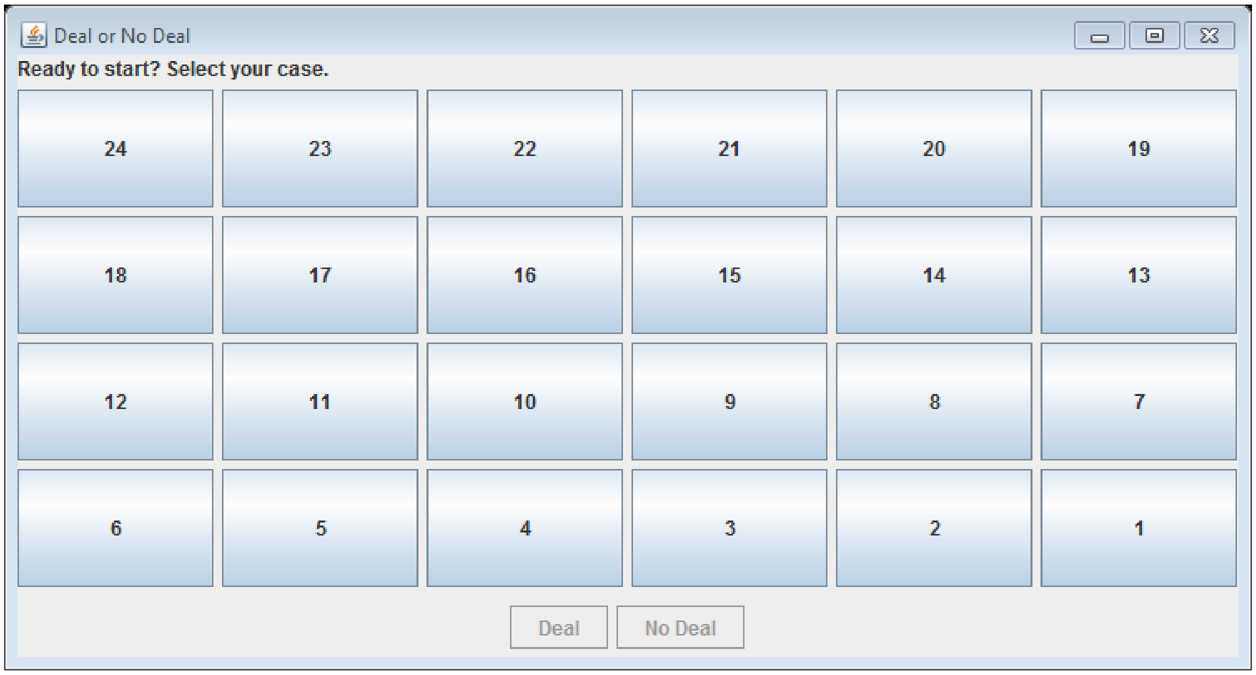
Assignment Requirements

You will create a GUI application of a simplified version of this game. You have the option of using the provided program code skeletons or designing your own program classes.

- Your game will start with 24 briefcases with the following cash values
  - $1, $5, $10, $25, $50, $100, $200, $300, $400, $500, $750, $1000, $5000, $10000, $25000, $50000, $75000, $100000, $200000, $300000, $400000, $500000, $750000, and $1000000
- The game will begin with the player selecting their briefcase. The briefcase value will not be revealed.
- The game rounds continue with the player selecting 6 briefcases for removal, then 5 briefcases, then 4 briefcases, then 3 briefcases, then 2 briefcases, then 1 briefcase at a time until all of the briefcases are selected or a Deal is accepted. After each required number of briefcases for that round are selected and removed, the Banker will make an offer.
- As the cases are selected and removed, the briefcase cash value will be displayed.
- If the player accepts the Banker's offer, the game will display whether the player won (the value of their briefcase is less than the Banker's offer) or lost (the player's case value was more than the Banker's offer.) The game ends at this point.

- If the player rejects the Banker's final offer (when only one case remains to be revealed), the player will be instructed to select the final case. There will not be an option for the player to trade cases, as in the original television version of the game. The Banker's final offer will then be used to determine whether the player won or lost and the result will be displayed. The game ends at this point.
- The Banker's offer will be calculated by taking the average value of the remaining cases * round / 10.

Your interface should look similar to this example:

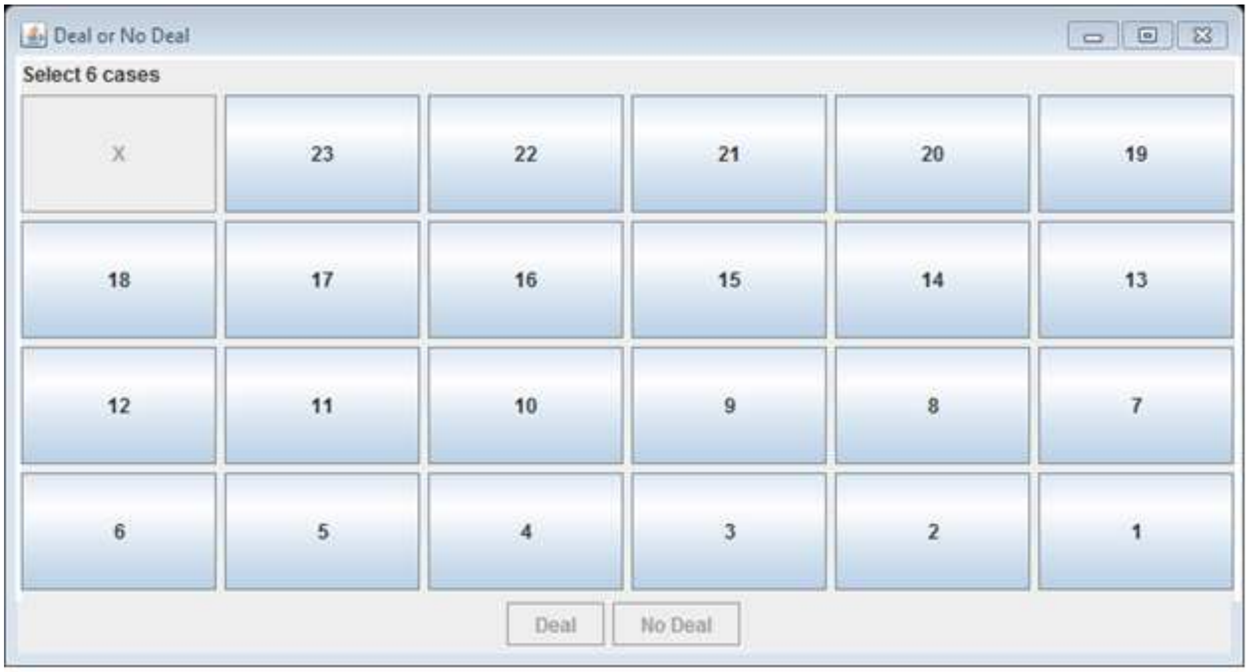


Your interface does not have to look exactly like the example above. However, it does need to meet these minimum requirements:

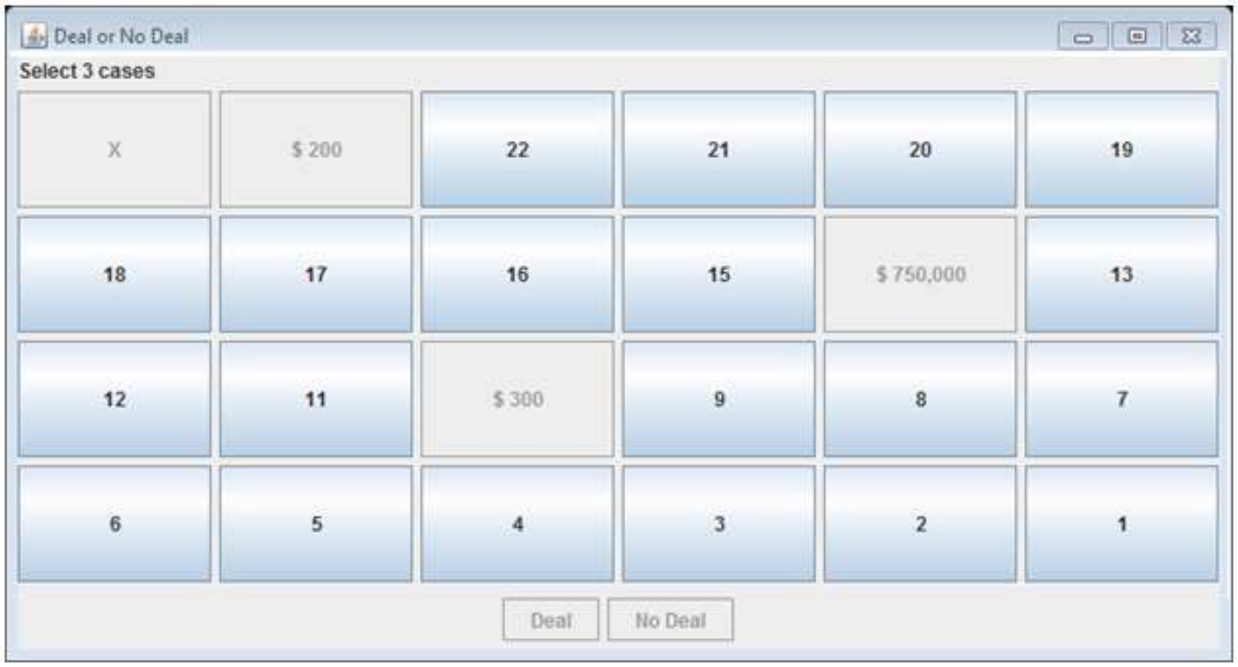
- Your game will use 24 rather than 26 briefcases. When initially displayed, the briefcases will display the briefcase numbers starting with 24 in the upper left corner and decrease to the final case numbered 1.
- The game title, Deal or No Deal must appear in the title bar
- A text instruction area will be used to prompt the contestant of their next step. In the example, the text instruction displays "Ready to start? Select your case."
- As the player selects briefcases, the instruction area will be updated with the remaining number of briefcases to be selected for that round. After the last case for that round is selected, the Banker's offer will be displayed.
- The Deal and No Deal buttons are active only when the Banker has made an offer and a deal is pending.
- Once the Banker makes an offer, the contestant should not be able to select any briefcases until they accept (Deal) or reject (No Deal) the Banker's offer.
- The player's case will be marked with an "X" once selected.
- As the cases are selected, the dollar value will be displayed.
- All dollar amounts will be displayed as comma separated currency without decimal places (ie $1,000 $200,000.)
- When the game is over, clicking a briefcase has no effect.

Following are some screen shot examples as the game progresses:

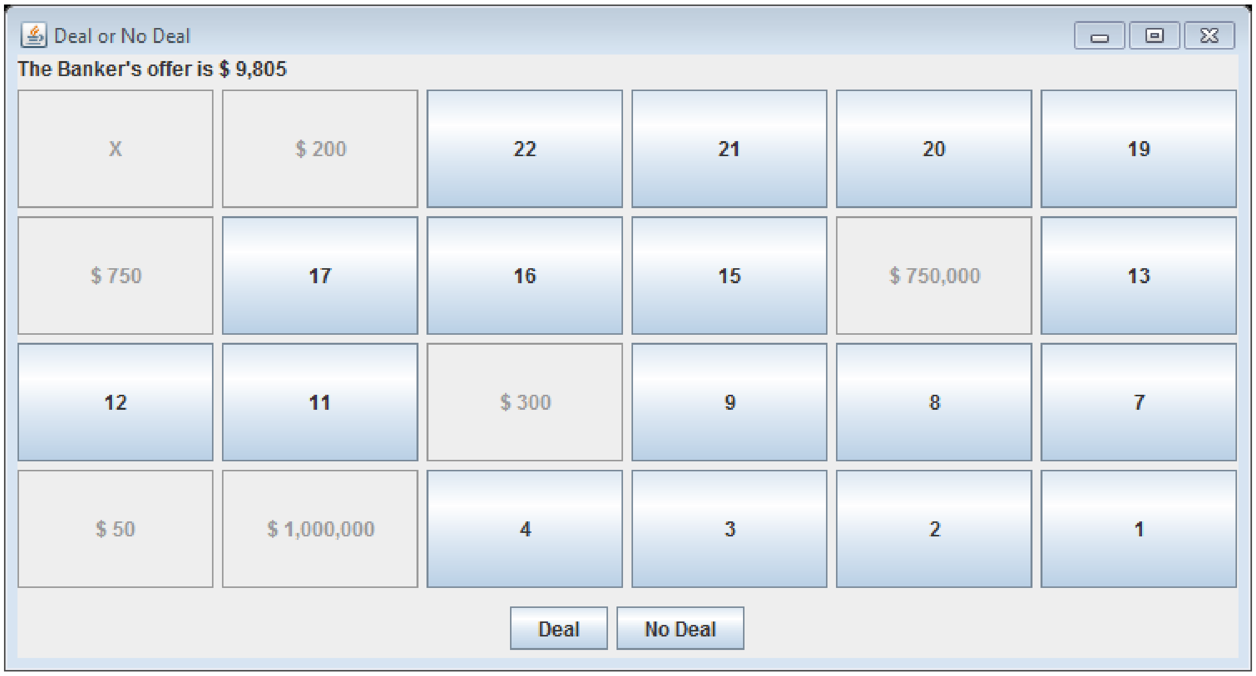
After the player's case is selected



After three cases were selected in the first round



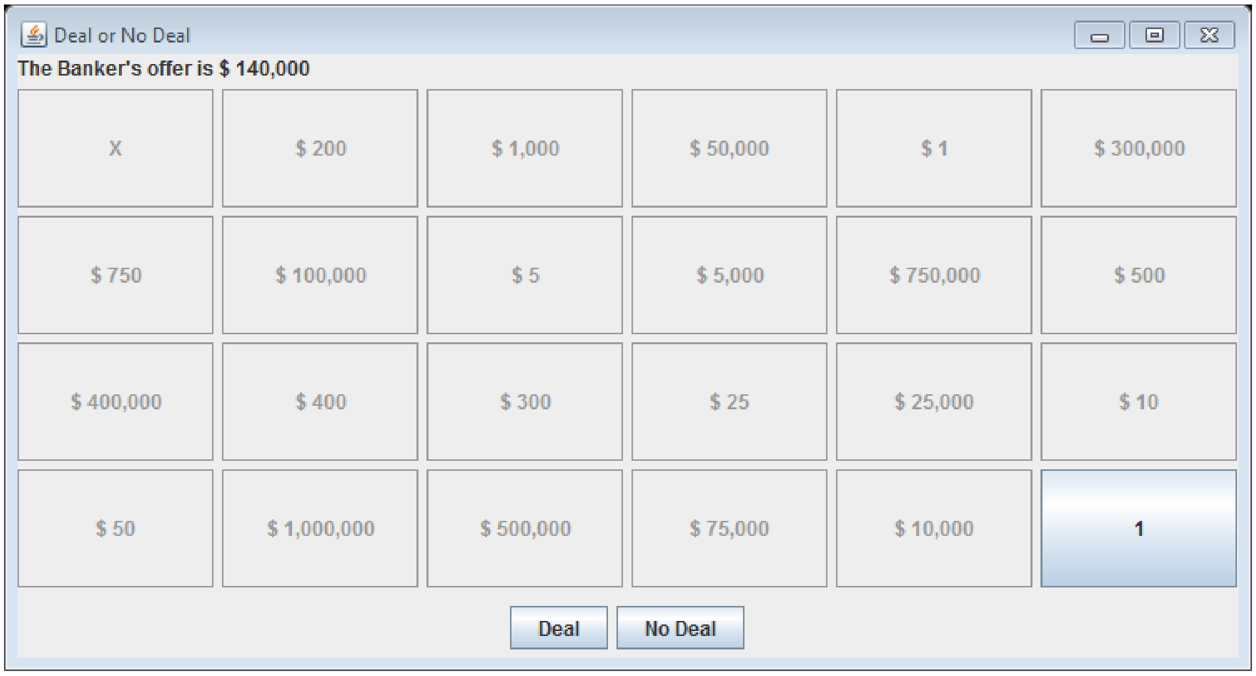
The Banker’s first offer before the player accepts or rejects



The Banker's offer is $ 9,805

X	$ 200	22	21	20	19
$ 750	17	16	15	$ 750,000	13
12	11	$ 300	9	8	7
$ 50	$ 1,000,000	4	3	2	1

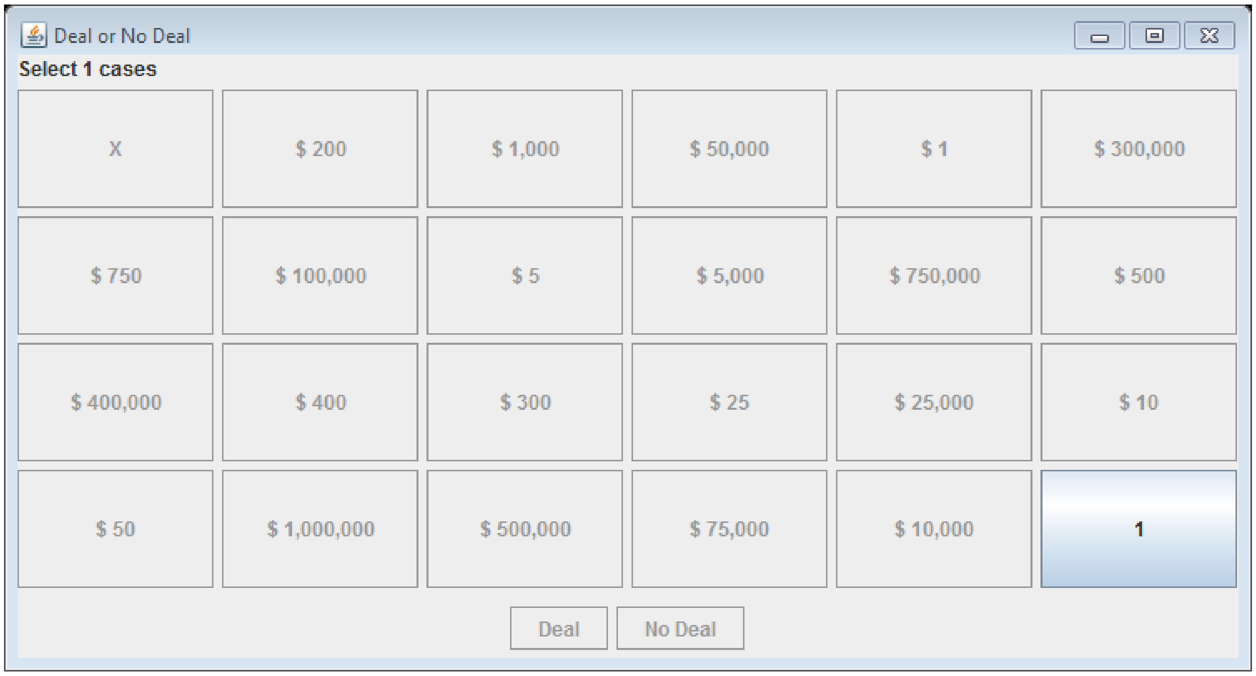
After playing all rounds and rejecting all of the Banker’s offers. This is the final offer.



The Banker's offer is $ 140,000

X	$ 200	$ 1,000	$ 50,000	$ 1	$ 300,000
$ 750	$ 100,000	$ 5	$ 5,000	$ 750,000	$ 500
$ 400,000	$ 400	$ 300	$ 25	$ 25,000	$ 10
$ 50	$ 1,000,000	$ 500,000	$ 75,000	$ 10,000	1

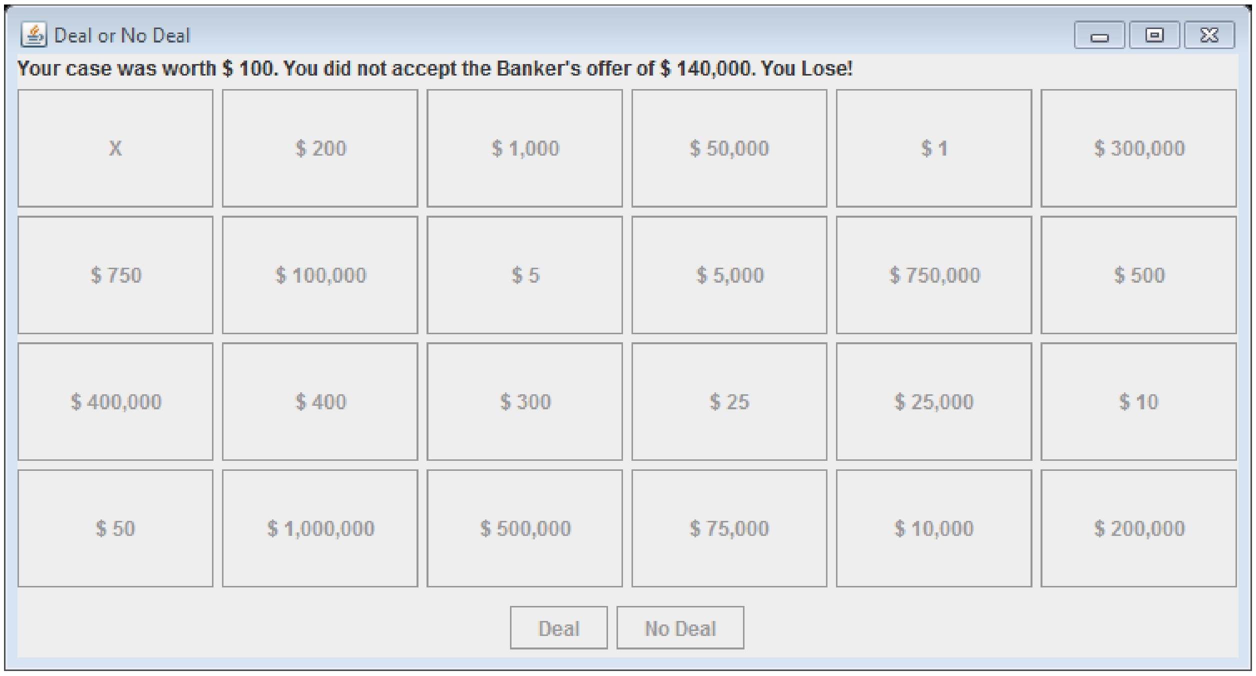
After rejecting the Banker's final offer



Select 1 cases

X	$ 200	$ 1,000	$ 50,000	$ 1	$ 300,000
$ 750	$ 100,000	$ 5	$ 5,000	$ 750,000	$ 500
$ 400,000	$ 400	$ 300	$ 25	$ 25,000	$ 10
$ 50	$ 1,000,000	$ 500,000	$ 75,000	$ 10,000	1

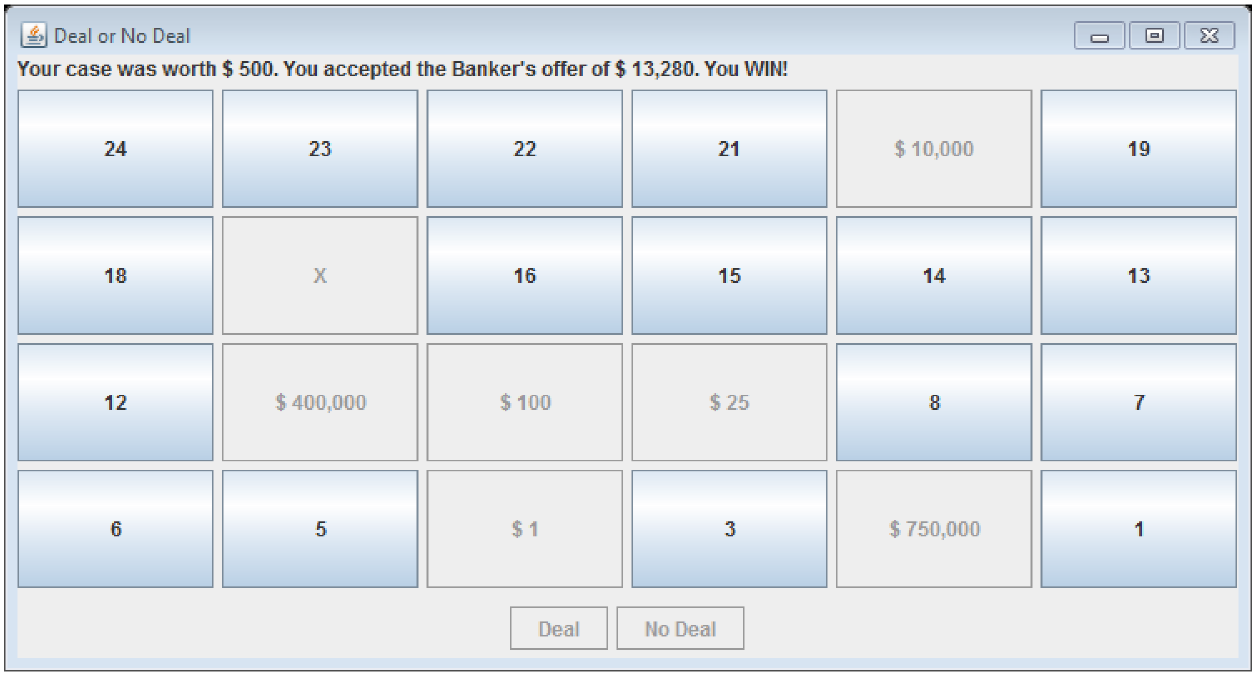
Final result after selecting the last briefcase



Your case was worth $ 100. You did not accept the Banker's offer of $ 140,000. You Lose!

X	$ 200	$ 1,000	$ 50,000	$ 1	$ 300,000
$ 750	$ 100,000	$ 5	$ 5,000	$ 750,000	$ 500
$ 400,000	$ 400	$ 300	$ 25	$ 25,000	$ 10
$ 50	$ 1,000,000	$ 500,000	$ 75,000	$ 10,000	$ 200,000

After accepting the Banker's offer in the first round.



Your case was worth $ 500. You accepted the Banker's offer of $ 13,280. You WIN!

24	23	22	21	$ 10,000	19
18	X	16	15	14	13
12	$ 400,000	$ 100	$ 25	8	7
6	5	$ 1	3	$ 750,000	1

Program Code

These classes are provided to get you started. You do not have to use these classes if you prefer to work from your own design.

- dealOrNoDeal.ui.GameFrame - this is the class which will contain the main method and create the GUI user interface. Its primary function is to display the GUI and handle the GUI events. It does not contain any game logic.
- dealOrNoDeal.DealOrNoDeal – this is the game class. It controls the logic and rules of the game and manages the state of the game, ie the collection of briefcases, player case, number of cases selected, the next step in the game, etc.
- dealOrNoDeal.Banker – the Banker calculates the offer
- dealOrNoDeal.Briefcase – this is the data model class. It will store information about the Briefcase itself, such as its value, whether it was selected, and if it's the player case. It is also a subclass of javax.swing.JButton. A Briefcase will also have the functionality and properties of a JButton so it can be displayed and interacted with on the GUI.

Problem Solving Tips

- Use the provided code templates and replace the **/* TODO: */** comments with Java code.
- When creating class fields, be sure to use the appropriate level of access and data type. Use constants (final) fields when applicable. Provide the correct getters and setters only if necessary.
- DealOrNoDeal.java – some methods in this class include:
    - gameSetup()
    - getInstruction()
    - getResult()
    - initiateDeal()
    - isDealPending()
    - isGameContinue()
    - noDeal()
    - deal()
    - selectCase(Briefcase)
    - setPlayerCase(Briefcase)
- Banker.java
    - The Banker class calculates and maintains the Banker's offer. It should have a field for the offer. Other classes should not be able to set the offer by simply passing a value. i.e., public void setOffer(int offer). In order to calculate and set the offer, the Banker needs to know the current round and the Briefcases in play. Other classes will pass that information to the Banker so the Banker can calculate and set the offer.
- Briefcase.java
    - Other classes should not be able to set or modify the Briefcase value. The Briefcase value will be set using the constructor when a Briefcase is created.
- GameFrame,java
    - This is the GUI class for the game. It contains the GUI components, a DealOrNoDeal object, and inner classes to handle the GUI events.

What to Submit

Submit a zip file containing the all of the source files and directories if necessary. Your application must compile when unzipped.