**TOPICS**

Exception

Unchecked Exception

Checked Exception

# **DESCRIPTION**

**General Description**

In this assignment, you are to create a GUI based Guessing game. At the start, the game randomly selects a number between 1 and 7. The user is asked to guess the selected number. User is allowed unlimited tries for correctly guessing the selected number until the user succeeds. At which point the game ends and displays a summary report. The next game automatically starts and the process is repeated until the user ends the application.�

**GUI**

The GUI for the game consists of the following:� (See PICTURE section below)

**A Text Field**

The game provides a text field that is labeled �Enter Your Guess�. The user enters its guess in the text field. After entering the guess, user presses an enter key while the cursor is still within the text field. This invokes the text field enter key event handler.

**A scrollable Text Area**.

The game provides a scrollable text area. Game uses the text area for displaying to the user messages during the game and a summary report at the end of each game.

**Procedure**

At the start, game randomly selects a number between 1 and 7 and displays a message in the text area �Game Started, Enter Your Guess�.

The user enters the guess in the text field and presses the enter key while the cursor is within the text field.

In response, the program clears the text field and displays one of the following messages in the text area:

Guess Not A Whole Number

Guess Out Of Range

Guess Too Large

Guess Too Small

Correct Guess

If the guess is not correct, the user enters a new guess in the text field and presses the enter key.

Again the program displays its response in the text area.

The above process is repeated until the user enters the correct guess and the game ends.

When a game ends, it displays a summary report in the text area including the following:

Total Guesses:�������������

Indicates the total number of guesses entered in all the games played so far.

Total Games:���������������

Indicates the total number of games played so far.

Average Guesses:��������

Indicates the average number of guesses per game so far.

At the completion of each game, a new game automatically starts, it randomly selects a number between 1 and 7 and displays in the text area �New Game Started. Enter your guess�. The user enters its guess in the text field and the above process is repeated till the user ends the application.

# **Instructions**

In doing this assignment, you must follow the instructions listed below in this section.

**Use An Event Handler For Catching Text Field Enter Key Presses**

Use an event handler for catching text field �enter key� presses.

**Do Not Use Loops/Branches In Text Field Enter Key Event Handler**

Do the above assignment without using any branching or looping statements in the text field enter key event handler. Instead use try/catch statements as needed.

**DESIGN**

Provide classes described below.

**TestJFrameExt class**

Use this class for housing the main method which will create the application frame (JFrameExt object) and display it.

**JFrameExt class**

Use this class as an application frame class. In its constructor, create the application GUI including a text field and a scrollable text area.�

**Guess class**

Provide a class Guess that will handle the functionality for generating game�s selection and comparing user guesses against it.��

Provide a constructor for this class. In its constructor, randomly generate game�s selection between 1 and 7.

Also provide a method checkGuess for this class. In the method checkGuess receive User�s guess as an input parameter. Compare the user�s guess with the game�s selection and respond appropriately. In this method, do not return any value to the caller. Instead, depending upon the situation, generate an exception object of one of the classes listed below. The� caller will catch the exception and will thus determine the response.

**Exception Classes**

Create the following exception classes. Each of these classes will extend Exception class.

# GuessNotWholeNumberException class

GuessOutOfRangeException class

GuessTooSmallException class

GuessTooLargeException class

**Creating Guess Object**

Create a Guess object in two different places in the code. Once in the constructor of JFrameExt when the game starts for the first time. After that, in the text field enter key event handler each time that a game ends and a new game starts.

**Declaring Guess Object**

Since access to a Guess object is needed in multiple places in the code, declare it as an instance variable in frame class (JFrameExt).

**Calling checkGuess method**

Call checkGuess ( ) method from text field enter key handler. Call it from within a try block with multiple catch blocks so that you can determine the exception generated by checkGuess method.

Sample code for a call to checkGuess and handling of exception received is given later in the Sample Code section.

# **IMPLEMENTATION**

Create the following classes:

## **TestJFrameExt class**

Create a class TestJFrameExt that will contain the main method. In the main method, do the following:

Create a JFrameExt object.

Call its setSize method to set its size to 400 by 300.

Call its setVisible method and pass it true.

## **JFrameExt class**

Create a class JFrameExt that extends class JFrame.

This class will provide a constructor and a text field enter key handler.

**A constructor**

The constructor will do the following:

It will create the required GUI.

It will register with the text field for receiving enter key events.

**Text Field Enter Key Event Handler**

This method will do the following:

It will provide the code that will be executed when the user presses the enter key from within the text field.

**Guess class**

This class will provide a constructor and a method checkGuess as described below: .

### **Guess constructor**

The constructor will do the following:

It will randomly select a number between 1 and 7.

It will save the selected number in an instance variable.

**checkGuess method*.***

This method will receive the user�s guess as a String parameter and will do the following:

It will convert the user�s guess received as a String parameter into an int. If the number cannot be converted, it will throw GuessNotWholeNumberException.

�

Otherwise, it will compare the user�s guess with the selected number and will do the following:�

If the guess is not within the range of 1 to 7, it will throw a GuessOutOfRangeException.

Else If the guess is within range but is larger than the selected number, it will throw a GuessTooLargeExcepion.

Else If the guess is within range but is smaller than the selected number, it will throw a GuessTooSmallExcepion.

Else If the guess matches the selected number, it will simply return.

# **Exception Classes**

Create the following exception classes. Each of the classes below will extend the class Exception. (See sample code later below).

# **class GuessNotWholeNumberException**

**class GuessOutOfRangeException**

**class GuessTooSmallException**

**class GuessTooLargeException**

From within the constructor of each of the above classes, call the parent constructor and pass it a message indicating the reason for exception. The constructor will pass this message to the parent constructor. The parent constructor will save this message in a parent instance variable called message.

The catch receiving the Exception object will call Exception object�s getMessage method to obtain the message stored in it.

**Creating Text Field Enter Key Handler Using JBuilder**

Select Project | Project Properties menu item.

(Project Properties dialog box will show).

Select Code Style Tab.

(Code Style tab will show)

In Event Handling section of Code Style tab, select anonymous adapter.

Select Ok.

Select Frame class (JFrameExt) in JBuilder left pane.

Select design tab.

Select the text field.

Double Click in the text field.

(Partially coded handler will show in the source code).

Enter the remaining code.

**Creating A Scrollable Text Area**

Drop A JScrollPane in a JPanel.

Set its HorizontalScrollBarPolicy property to ALWAYS

Set its VerticalScrollBarPolicy property to ALWAYS

Drop a JTextArea in JScrollPane.

Set its columns property to 40.

Set is rows property to 10.

**SAMPLE CODE**

**Sample exception class**

Here is how you may write TooSmallException class. Other exception classes can be written the same way.

public class GuessTooSmallException extends Exception

{

� public GuessTooSmallException()

� {

��� super ("Guess Is Too Small");

� }

}

**Sample Guess class**

//Here is a sample Guess class.

//It generates above mentioned exceptions in its checkGuess method.

public class Guess�

{

� private int selection;

� public Guess()

� {

��� //randomly select a number between 1 and 7 and save it in selection.

��� selection = ((int) (Math.random() \* 7)) + 1;

� }

� public void checkGuess (String guess ) throws GuessOutOfBoundsException, GuessTooLargeException, GuessTooSmallException, GuessNotWholeNumberException

� {

��� int n = 0;

��� //Convert User�s Guess received as a String into an int.

��� try

�� {

����� //convert String guess into an int

����� n = Integer.parseInt (guess);

��� }

��� catch (NumberFormatException ex)

��� {

����� //The user�s guess is not an int

����� throw new GuessNotWholeNumberException( );

��� }

��� //User�s guess is now converted into an int

��� if ( n > 7 || n < 1)

��� {

����� //User�s guess is outside the bounds of 1 and 7.

����� throw new GuessOutOfBoundsException( );

��� }

��� else if ( n > selection )

�� {

����� //User�s guess is larger than the game�s selection.

����� throw new GuessTooLargeException ( );

��� }

��� else if (n < selection)

��� {

����� //User�s guess is smaller than the game�s selection.

����� throw new GuessTooSmallException ( );

��� }

���� //User�s guess matches selection, simply return.

���� return;

� }

}

**Sample Code For Handling Text Field Enter KeyEvent**

//Here is a sample code for handling a text field enter key event.

� try

���������� {

������������������ //Get the user guess from the text field

������������������ userGuess = jtfMessage.getText( ).trim ( );

������������������ //Call checkGuess and� pass it userGuess.

������������������ g.checkGuess ( userGuess);

������������������ //If it got here, user guess is correct.

������������������ message = �Correct Guess\n�;

������������������ //Increment gameCount

������������������ //Accumulate� Summary Report in message.

������������������ //Start a new Game

������������������ g = new Guess ( );

�������������

����������� }

����������� catch (GuessNotWholeNumberException ex)

���������� {

������������������ //Extract message from exception object

������������� ���� message = ex.getMessage ( );

����������� }

������� ����catch (GuessTooSmallException ex)

���������� {

������������������ //Extract message from exception object

����������� }

����������� catch (GuessTooLargeException ex)

���������� {

������������������ //Extract message from exception object

����������� }

����������� catch (GuessOutOfBoundsException ex)

� {

������������� //Extract message from exception object

���������� }

��������� � finally

��������� � {

������������������ //display message in text area.

������������������ jtaMessage.append (message + �\n�);

������������������ //display prompt for the next guess

������������������ jtaMessage.append (�Enter your guess above� + �\n�);

��������� �� }

��

# **Getting Random Numbers**

## **Method 1**

The code below will provide a random whole number in the range 1 to 7.

double dnum;

int inum

dnum = Math.random ( ); //dnum is in the range: 0 <= dnum < 1

dnum = dnum \* 7; //dnum in in the range : 0 <= dnum < 7

inum = (int) dnum; //inum in the range: 0 <= n <= 6

inum = inum + 1; //inum in the range : 1 <= n <= 7

## **Method 2**

The code below will also provide a random whole number in the range 1 to 7.

int� n;

n = ( (int) (Math.random ( ) \* 7) ) + 1;

# **PICTURE**

****