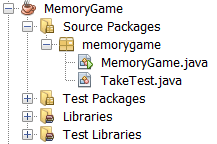
1. **Structure of the solution**

There are two classes were created in MemoryGame project namely MemoryGame and TakeTest. The main class is called MemoryGame and the second class is named as TakeTest as shown in **Figure 1**.



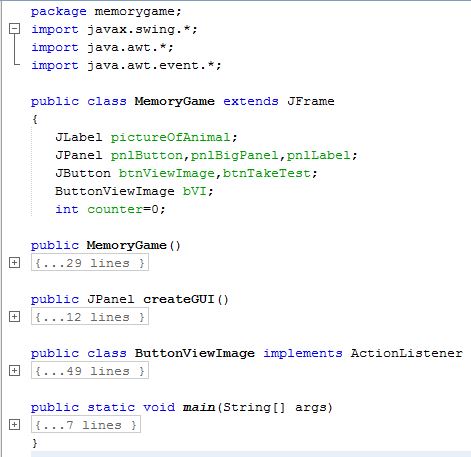
**Figure 1**:Structure of the MemoryGame project

**1.1 MemoryGame class**

**Figure 2** is the methods in the MemoryGame. There is a default constructor and two methods namely createGUI() and main(). Besides the the methods, it also has an inner class named ButtonViewImage.

The reason why MemoryGame is known as a main class because it has a main method. Besides, TakeTest is known as the sub class because it has no main method.

The ButtonViewImage class consists of an overriding method named actionPerformed(). Upon click event detected from the View Image button, the image will be displayed one by one. There are ten images of animals in total. A switch case is used to set the path to ten different images accordingly. The sequence of animal to be displayed is based on a counter that was set from 1 to 10 as shown in **Figure 3**. The image of animals is then would be set and displayed in the JLabel. The size of the picture will be set based on the size of the JLabel.

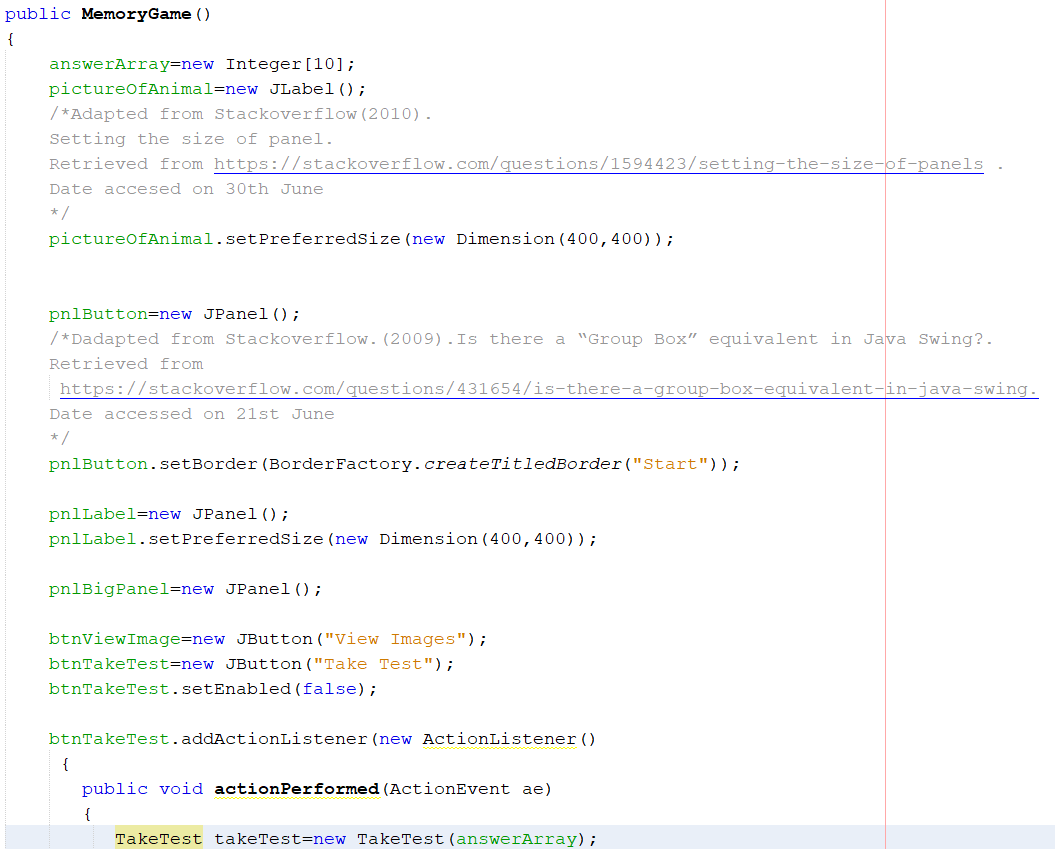


**Figure 2**: The MemoryGame class structure



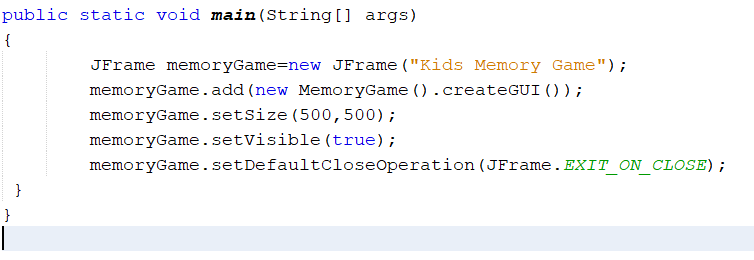
**Figure 3:** ButtonViewImage class

The constructor for MemoryGame as shown in the following **Figure 4** creates new objects for JLabel, three JPanel and two JButton. The JLabel named pictureOfAnimal is used to display picture images. The three JPanel is used for the two buttons and the JLabel for the picture of animal.



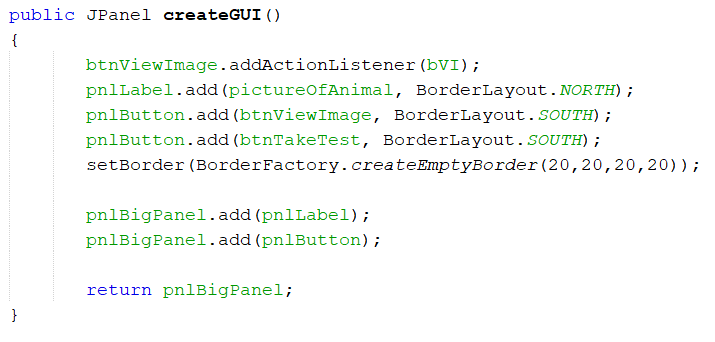
**Figure 4** Default constructor

The main() method as shown in **Figure 5** is a static method that would be executed first during the project’s execution. The the main() method, it creates the MemoryGame object named memoryGame with the Frame title - Kids Memory Game. The size of the frame was set to 500x500 and it was made displayed by calling the setvisible() method.



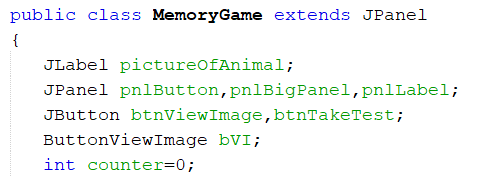
**Figure 5** The main() method

The createGUI method as shown in **Figure 6**  is called by the main method when creating the object for MemoryGame. In this method, it adds the objects created in the default constructor to the respective JPanels.



**Figure 6** createGUI() method

The MemoryGame class as shown in **Figure 7** is the main class. It inherits from the JPanel class. It consist five objects which is pictureOfAnimal, pnlButton, pnlBigPanel, pnlLabel, btnViewImage, btnTakeTest and bvi. It has a primitive data declared as counter.



**Figure 7** MemoryGame class

**1.2 TakeTest class**

**Figure 8** is the methods in the TakeTest There is a default constructor and two methods namely createGUI() and displayAnimalName(). Besides the the methods, it also has two inner class named ButtonCheckScore and ButtonShowAnswer.

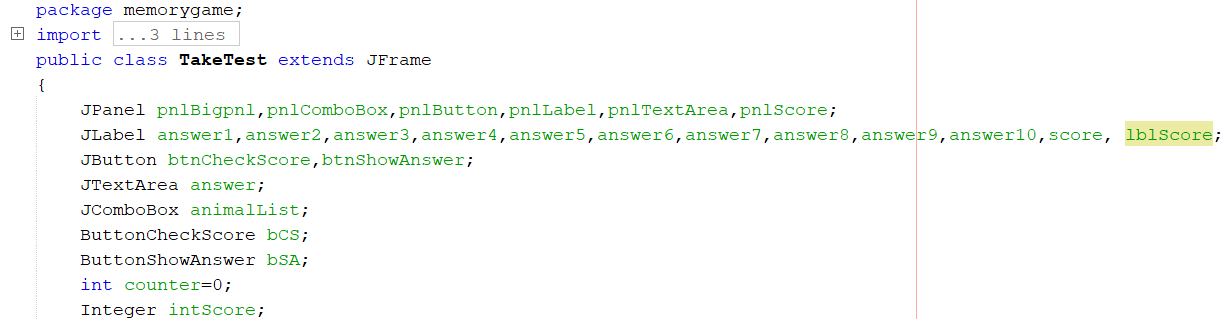
The default constructor creates the following object which is six panel for the ComboBox,button, label, textarea, score and the big panel to store each object’s panel. The label consist of ten answers and the score to keep the marks and a score for label purposes. Besides, there are two buttons for score checking and answer shoring. Moreover there are textarea for the answer to display and a combo box for player to select the answer.

The createGUI method is called by the main method when creating the object for MemoryGame. In this method, it adds the objects created in the default constructor to the respective JPanels.

The displayAnimalName() is a method to assign each value to the answer label.

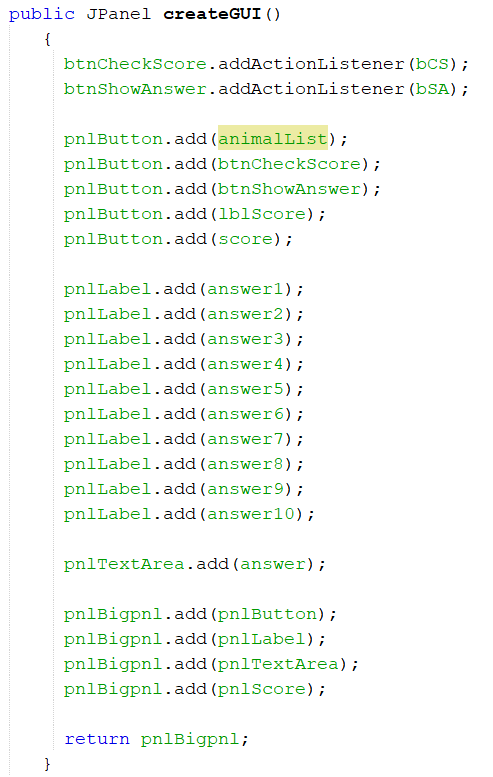
A class name called ButtonCheckScore is to allow the program to compare the right answer and what has the user assign the answer to.

A class name called ButtonShowAnswer allows player to view the answer in the textarea.



**Figure 8** The TakeTest class

The createGUI method as shown in **Figure 9**  is called by the MemoryGame class. It was called when the Take Test button was clicked in the MemoryGame class. This method adds the instiated objects such as JButton and JLabel to the respective JPanels.

****

**Figure 9** The createGUI() method

The following picture which is **Figure 10** is a default constructor for TakeTest class. The default constructor has an overriding method named actionPerformed() to detect the animal name selected. It invokes the displayAnimalName() method to display the selected animal name in the following label which is answer 1 to answer 10 JLabel. This method also creates objects for JLabel and JPanel.



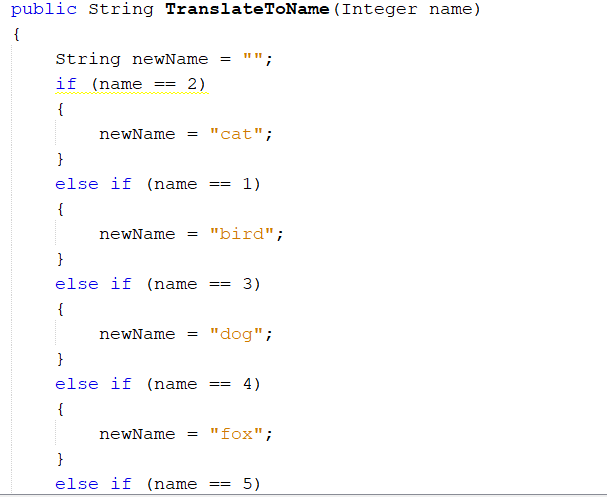
**Figure 10** TakeTest() default constructor

The displayAnimalName() method as shown in **Figure 11** is called by the TakeTest constructor. It is used to display the animal name selected by the user from the combo box to the answer label from one to ten. It implemented a switch structure to determine the index of the selected animal in the combo box.



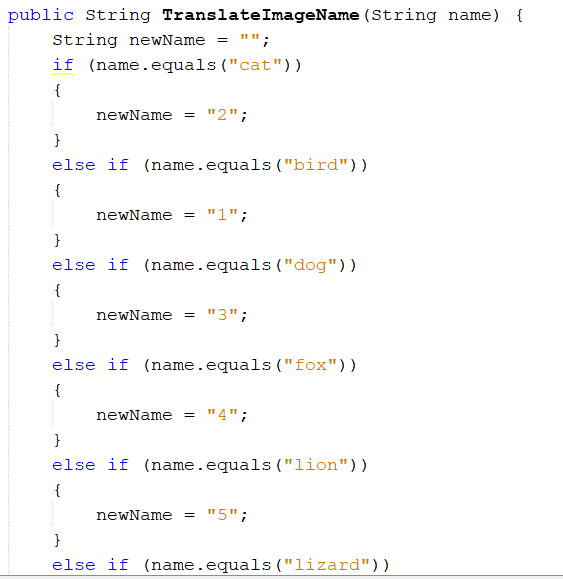
**Figure 11**  The displayAnimal() method

The TranslateToName is a method that accepts Integer datatype called name. This method has a return value of of String called newName. The purpose of this method is to convert numbers to name for each picture as shown in **Figure 12**. The converted name (from number to string) is used for display in the answer text area when the user clicks the Show Answer button



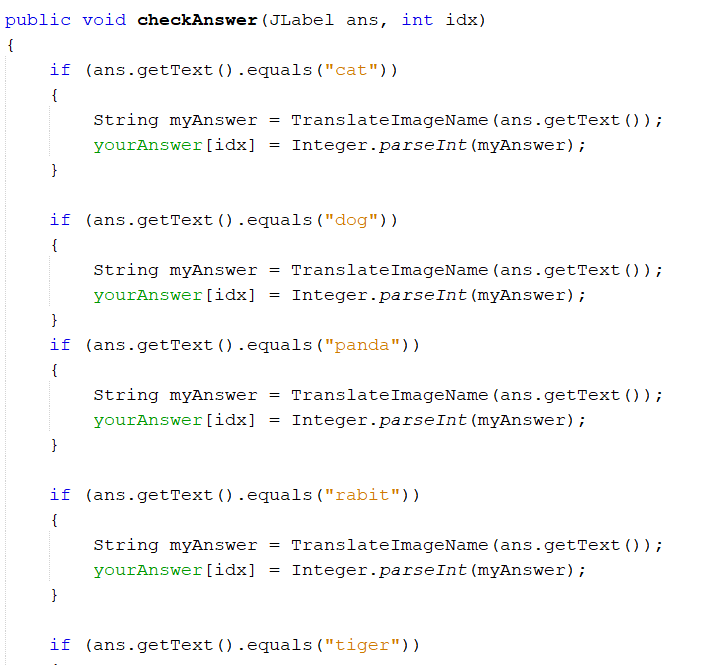
**Figure 12**

The TranslateImageName is a method that accepts String datatype called name. This method has a return value of String called newName. The purpose of this method is to convert from name to numbers. The converted name (from string to number) is stored in myAnswer array. The myAnswer array is used for checking the correct or actual answer by comparing the items in myAnswer with the answerArray.



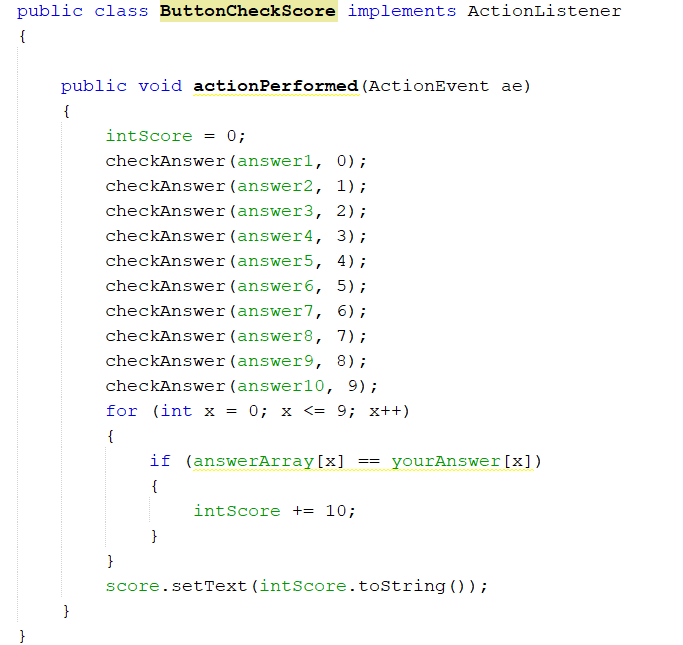
**Figure 13**

The checkAnswer is a method has accept two arguments which is ans and idx. The purpose of this method is to check what has the user selected for the answer. It will convert from string to integer. Later on it will store in an array.



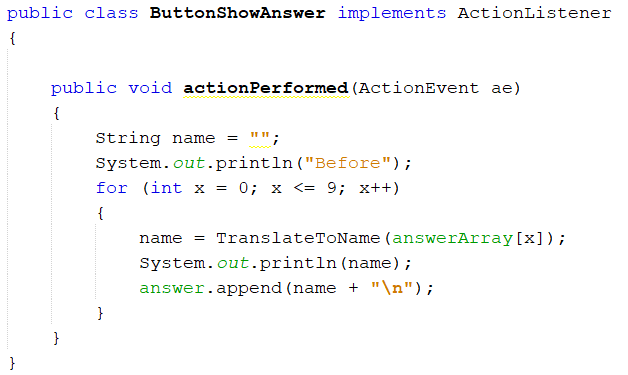
**Figure 14**

In the following **Figure 15**, there is an inner class named ButtonCheckScore. In this class, it has an overriding method named actionPerformed(). There is a selection structure to determine the answers in the JLabel from answer 1 to answer 10. The intScore is declared as Integer wrapper class to sum up the total scores from the user. The intScore is then converted to String and displayed in the JLabel Scorel.



**Figure 15**  ButtonCheckScore class

In the following **Figure 16**, there is an inner class named ButtonShowAnswer. In this class, it has an overriding method named actionPerformed(). When the player has completed the test, player can click a button of show answer.



**Figure 16**  ButtonShowAnswer class

**2. Additional Java Concepts and Java API**

In the MemoryGame application, the following additional Java concepts were applied.

**2.1 Random Numbers**

The following Random Numbers have been to shuffle questions. The java.util.Random is a java library to allow you to use it. While Random is a class and rand is an object as shown in **Figure 17**.

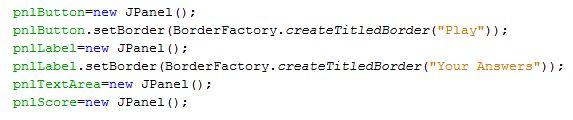




**Figure 17** Use of Random Numbers

**2.2 Use of BorderFactory**

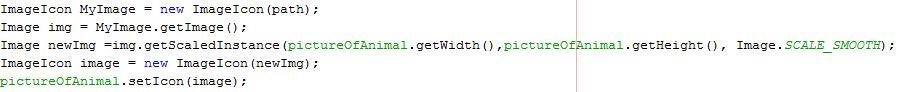
The use of BorderFactory to set the title in the JPanel. It sets the Title and Border on the JPanel. It appears like a group box to group the relevant objects together as shown in **Figure 18**.



**Figure 18** Use of BorderFactory

**2.3 Use of ImageIcon**

The use of ImageIcon enables the MemoryGame application to include some pictures in the project. The Image icon is displayed in a JLabel and the image fits nicely to the JLabel based on the pre-set size of the JLabel as shown in **Figure 19**.



**Figure 19** ImageIcon

**3. References**

Stackoverflow.(2010).[How to change the size of the font of a JLabel to take the maximum size. Retrieved from https://stackoverflow.com/questions/2715118/how-to-change-the-size-of-the-font-of-a-jlabel-to-take-the-maximum-size](https://stackoverflow.com/questions/2715118/how-to-change-the-size-of-the-font-of-a-jlabel-to-take-the-maximum-size). Date accessed on 21st June

Stackoverflow.(2012).[Java ComboBox Different Value to Name. Retrieved from](https://stackoverflow.com/questions/10387932/java-combobox-different-value-to-name)

<https://stackoverflow.com/questions/10387932/java-combobox-different-value-to-name>. Date accessed on 21st June

Stackoverflow.(2014).[Adding items to a JComboBox. Retrieved from](https://stackoverflow.com/questions/17887927/adding-items-to-a-jcombobox)

<https://stackoverflow.com/questions/17887927/adding-items-to-a-jcombobox>. Date accessed on 21st June

JavaForums.(2007). Use Multiple Form. Retrieved from

<https://www.java-forums.org/java-applets/2549-use-multiple-forms.html>.Date accessed on 21st June

Stackoverflow.(2009).[Is there a “Group Box” equivalent in Java Swing?. Retrieved from](https://stackoverflow.com/questions/431654/is-there-a-group-box-equivalent-in-java-swing)

<https://stackoverflow.com/questions/431654/is-there-a-group-box-equivalent-in-java-swing>. Date accessed on 21st June

Stackoverflow.(2011). Getting random numbers in Java. Retrieved from <https://stackoverflow.com/questions/5887709/getting-random-numbers-in-java>. Date accessed on 30th June

Oracle(2017).How to use Border.Retrieved from <https://docs.oracle.com/javase/tutorial/uiswing/components/border.html>. Date accessed on 30th June

Oracle(2017).How to use Panel. Retrieved from <https://docs.oracle.com/javase/tutorial/uiswing/components/panel.html>. Date accessed on 30th June

Stackoverflow(2010). Setting the size of panel. Retrieved from <https://stackoverflow.com/questions/1594423/setting-the-size-of-panels> .Date accesed on 30th June