Object Oriented Programming

COMP6699001 / Jude Joseph Lamug Martinez

BINUS
UNIVERSITY
INTERNATIONAL

Final Project

By : Christopher Owen / 2502019180 / L2AC

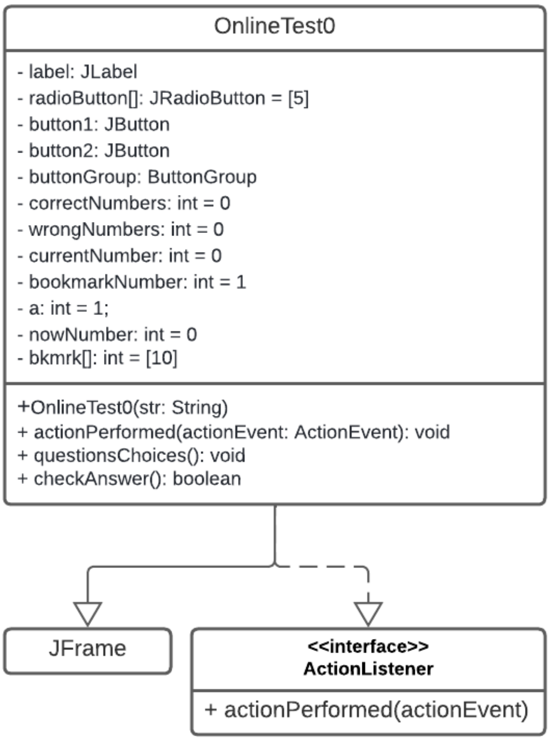
Table of Contents

Project Report: Java Test

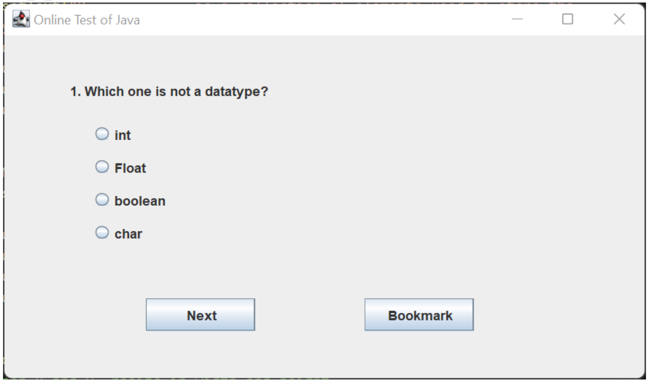
I. Program Description

Java test is a simple test application using Java programming language for teachers or lecturers in giving tests for students. The project uses JFrame interface alongside ActionListener for the API. This simple test application can be implemented further online using other various programming languages, such as MySQL. It can also be locally copied to other devices using pure Java. This application is very user-friendly to simply run tests and get a score based on their correct and incorrect answers. It also allows the user to bookmark questions to be answered later.

II. Class Diagram


OnlineTest0
- label: JLabel
- radioButton[]: JRadioButton = [5]
- button1: JButton
- button2: JButton
- buttonGroup: ButtonGroup
- correctNumbers: int = 0
- wrongNumbers: int = 0
- currentNumber: int = 0
- bookmarkNumber: int = 1
- a: int = 1;
- nowNumber: int = 0
- bkmrk[]: int = [10]
+OnlineTest0(str: String)
+ actionPerformed(actionEvent: ActionEvent): void
+ questionsChoices(): void
+ checkAnswer(): boolean
JFrame
<<interface>>
ActionListener
+ actionPerformed(actionEvent)

III. Application Flow



1. Which one is not a datatype?

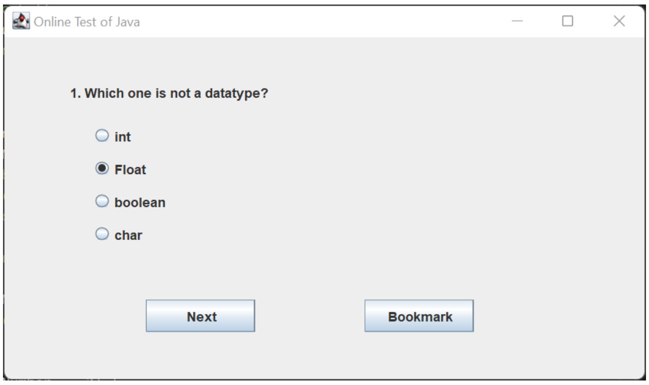
○ int

○ Float

○ boolean

○ char

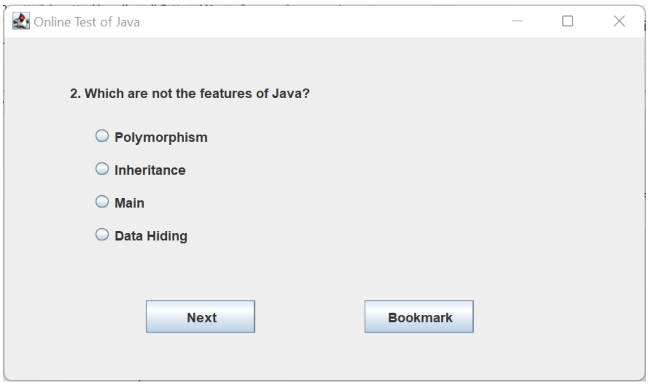
Program run, first appearance, first question.



1. Which one is not a datatype?

○ int
● Float
○ boolean
○ char

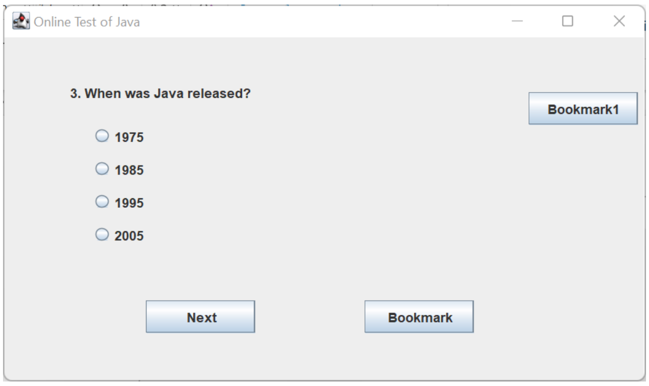
A choice has been selected.



2. Which are not the features of Java?

○ Polymorphism
○ Inheritance
○ Main
○ Data Hiding

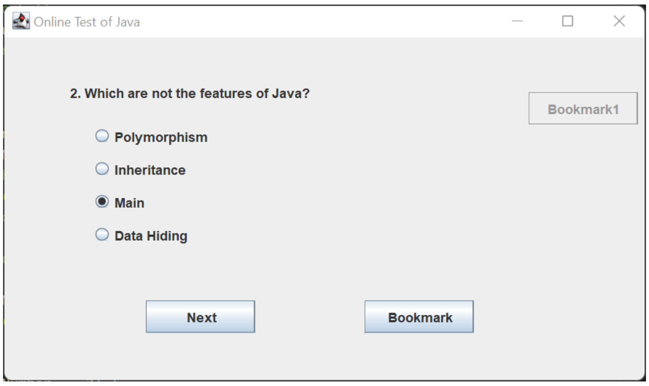
Clicked next from the first question onto the next question.



3. When was Java released?

○ 1975
○ 1985
○ 1995
○ 2005

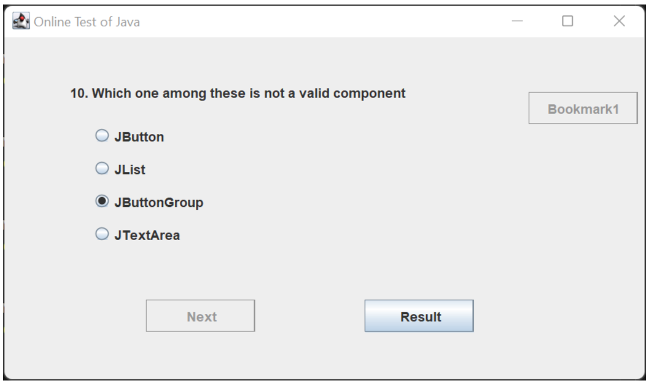
Second question is bookmarked, Bookmark1 is added, onto the third question.



2. Which are not the features of Java?

○ Polymorphism
○ Inheritance
● Main
○ Data Hiding

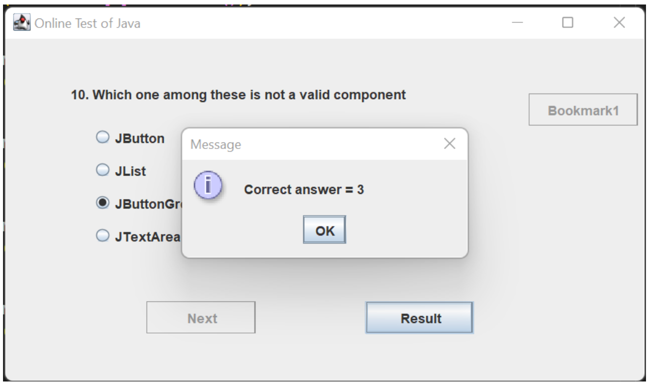
Bookmark1 is selected, onto the question contain in the Bookmark1.



10. Which one among these is not a valid component

○ JButton
○ JList
● JButtonGroup
○ JTextArea

After finishing the test until question 10, the only available button is Result.

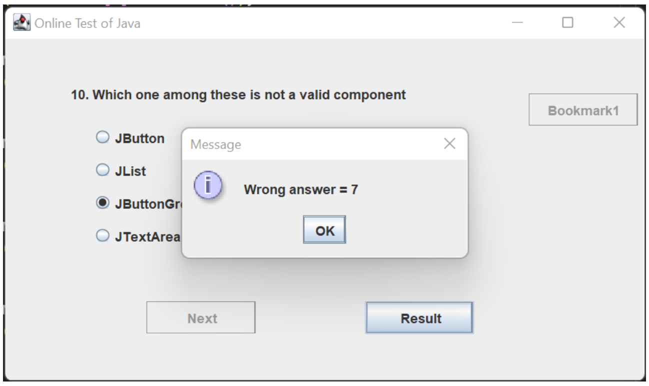


10. Which one among these is not a valid component

○ JButton
○ JList
● JButtonGr
○ JTextArea

Correct answer = 3

When Result is pressed, a message will pop up stating the correct answers.

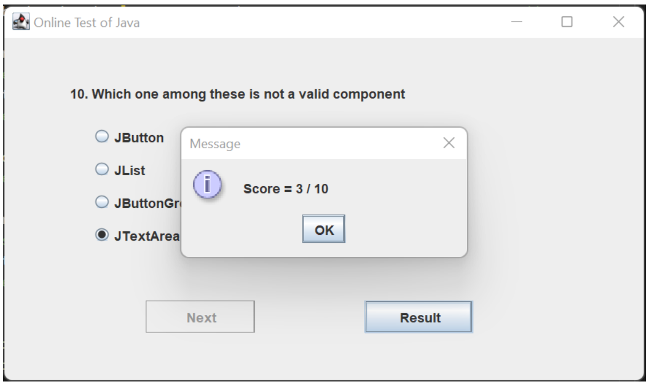


10. Which one among these is not a valid component

○ JButton
○ JList
● JButtonGr
○ JTextArea

Wrong answer = 7

After pressing OK, the next message will appear stating the wrong answers



10. Which one among these is not a valid component

○ JButton
○ JList
○ JButtonGr
● JTextArea

Score = 3 / 10

Finally after pressing OK, the last message will pop up stating the score (correct answers / total numbers).

IV. Lessons that Have Been Learned

In this project, I have learned new Java libraries that can be very useful in making an application based on pure Java, such as java.awt, java.awt.event, and javax.swing. Along with their classes, I learned new implementations of Java libraries that make the application more interactive. With this project, I get to learn, understand, and improve myself on Java and OOP.

V. Project Technical Description

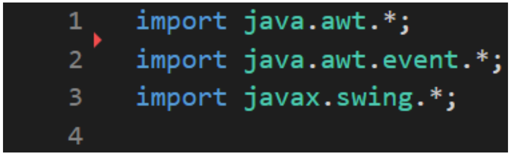
JFrame API

JFrame is a class imported from java.awt and the extension of java.awt.Frame. It has the constructors and methods in making an API based on pure Java. In this project, I use this API for the simplicity and showcasing Java libraries in making a useful application using only Java.

ActionListener Interface

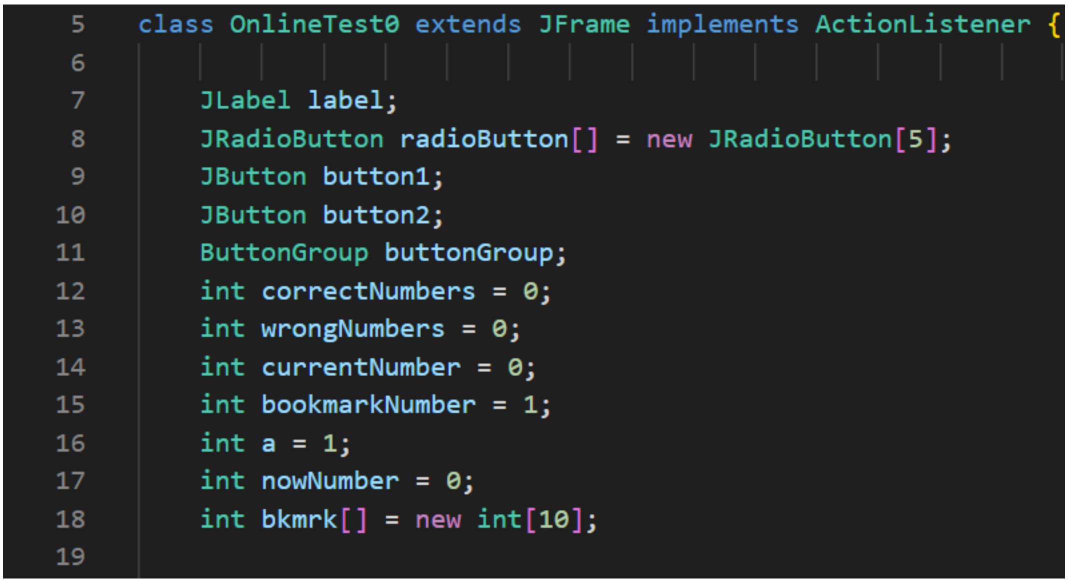
Importing from java.awt.event, Java ActionListener interface is called whenever the user click a button. Although it only has one method, which is actionPerformed, it is a very useful tool in making a Java application more interactive. With receiving ActionEvent, actions, that a button will do, will get called and run.

VI. Code Explanation



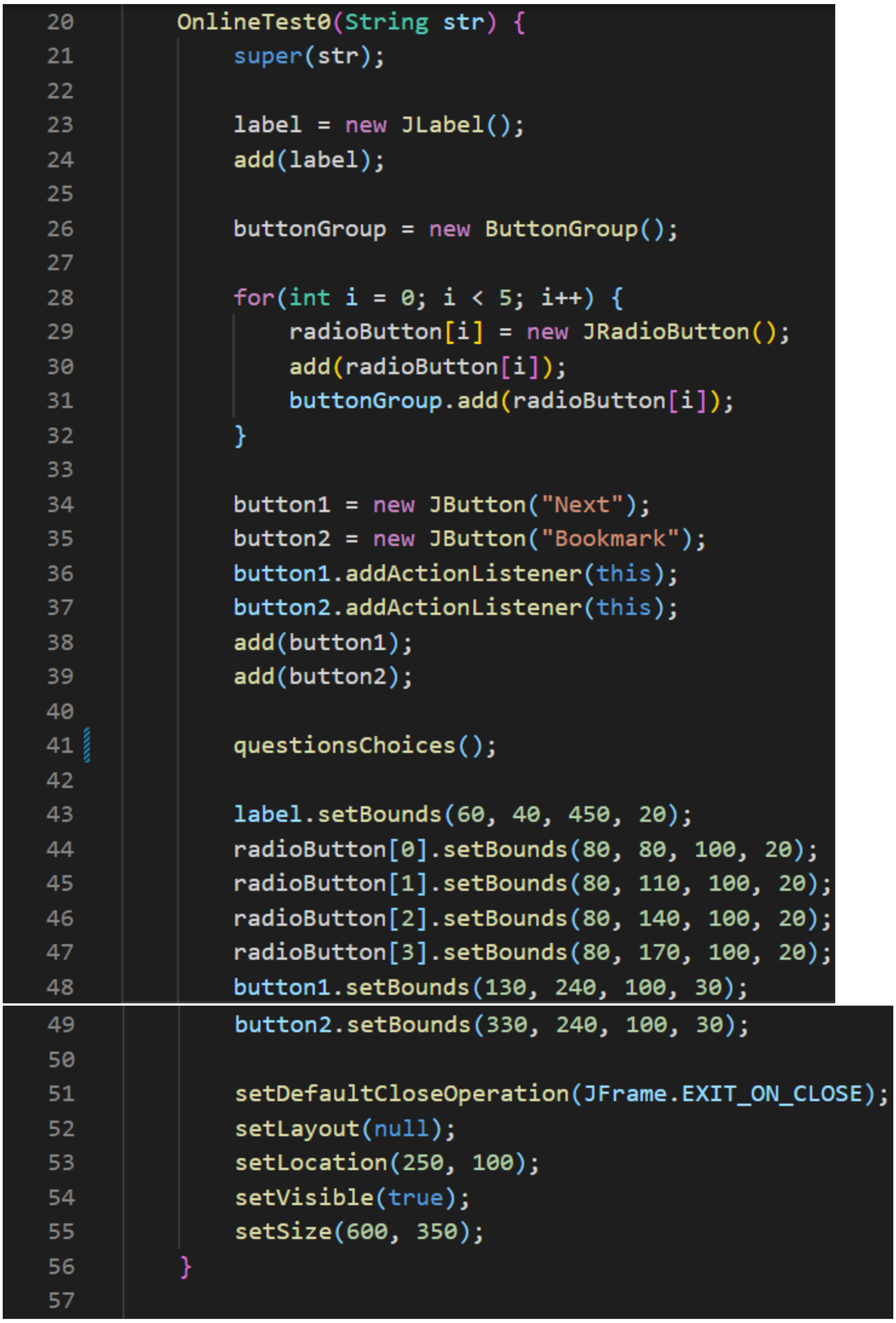
```java
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

```

Here are the Java libraries that are used in the application.
java.awt has all the classes for creating user interfaces and graphics.
java.awt.event has the extension classes from java.awt for accessing java.awt's classes to be interactive.
javax.swing provides java swing classes for interacting with the program.



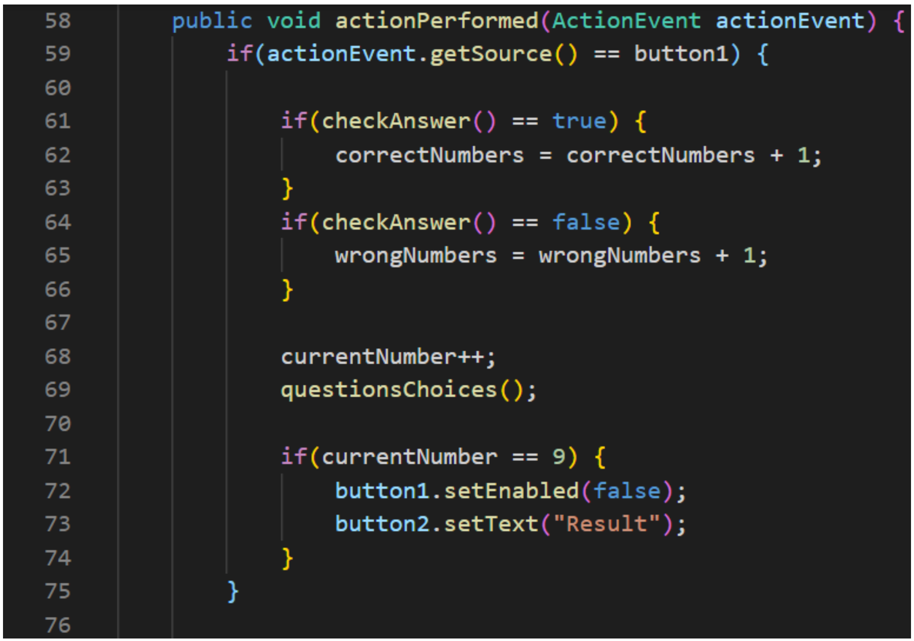
```java
class OnlineTest0 extends JFrame implements ActionListener {

    JLabel label;
    JRadioButton radioButton[] = new JRadioButton[5];
    JButton button1;
    JButton button2;
    ButtonGroup buttonGroup;
    int correctNumbers = 0;
    int wrongNumbers = 0;
    int currentNumber = 0;
    int bookmarkNumber = 1;
    int a = 1;
    int nowNumber = 0;
    int bkmrk[] = new int[10];

```

The OnlineTest0 class extends JFrame class from the java library inheriting all JFrame’s included class to be used.
Also, implements ActionListener interface so the user can interact with JFrame’s classes. Declaring JLable, JRadioButton, JButton, ButtonGroup, ints, and an array.

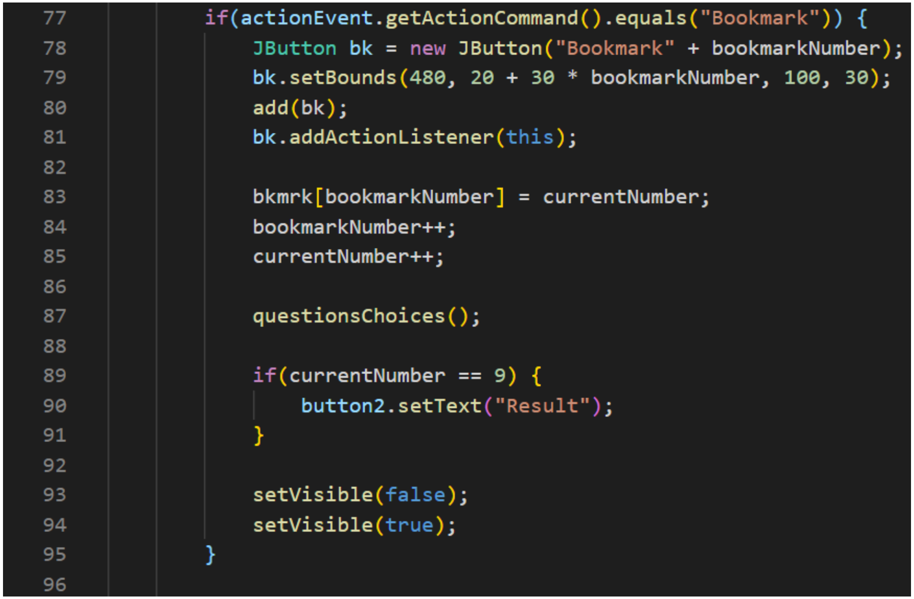


```java
    OnlineTest0(String str) {
        super(str);

        label = new JLabel();
        add(label);

        buttonGroup = new ButtonGroup();

        for(int i = 0; i < 5; i++) {
            radioButton[i] = new JRadioButton();
            add(radioButton[i]);
            buttonGroup.add(radioButton[i]);
        }

        button1 = new JButton("Next");
        button2 = new JButton("Bookmark");
        button1.addActionListener(this);
        button2.addActionListener(this);
        add(button1);
        add(button2);

        questionsChoices();

        label.setBounds(60, 40, 450, 20);
        radioButton[0].setBounds(80, 80, 100, 20);
        radioButton[1].setBounds(80, 110, 100, 20);
        radioButton[2].setBounds(80, 140, 100, 20);
        radioButton[3].setBounds(80, 170, 100, 20);
        button1.setBounds(130, 240, 100, 30);
        button2.setBounds(330, 240, 100, 30);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(null);
        setLocation(250, 100);
        setVisible(true);
        setSize(600, 350);
    }

```

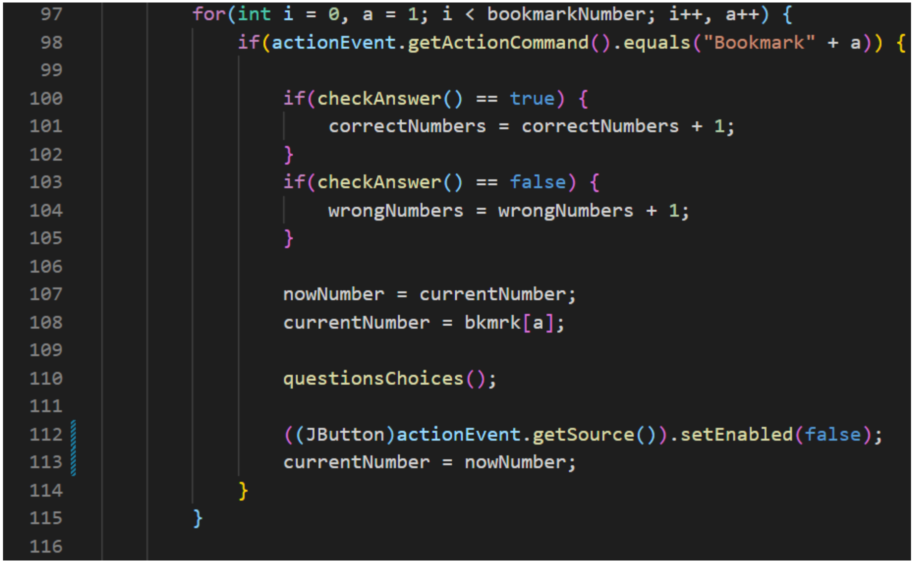
The OnlineTest0 will be the core function and will be called in the main function. This function initiates everything needed within the interface, such as the window called by super, radioButton for choices, buttons, calling the questionsChoices function, setting the location, visibility, and size of the window. The string parameter will become the title window.



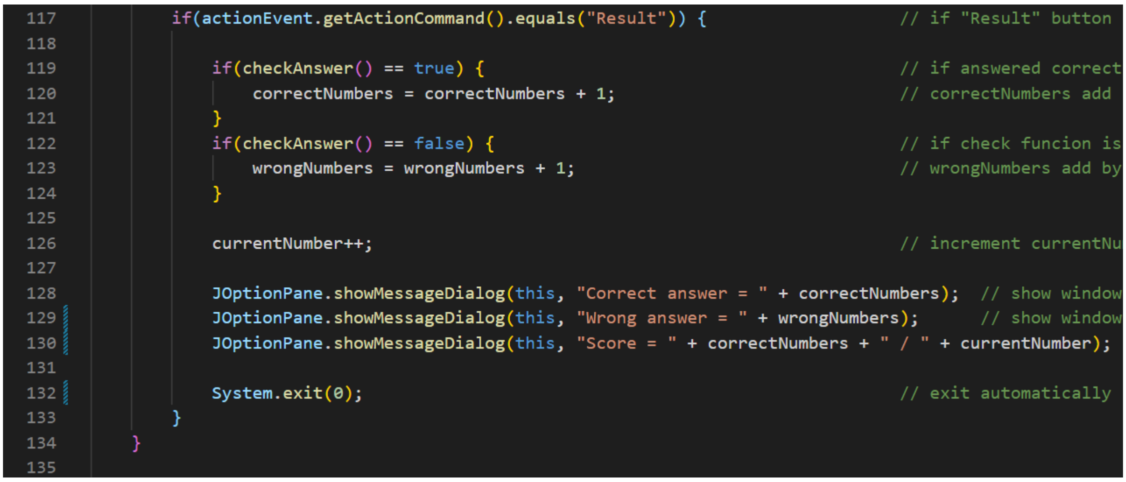
```java
    public void actionPerformed(ActionEvent actionEvent) {
        if(actionEvent.getSource() == button1) {

            if(checkAnswer() == true) {
                correctNumbers = correctNumbers + 1;
            }
            if(checkAnswer() == false) {
                wrongNumbers = wrongNumbers + 1;
            }

            currentNumber++;
            questionsChoices();

            if(currentNumber == 9) {
                button1.setEnabled(false);
                button2.setText("Result");
            }
        }

```



```java
        if(actionEvent.getActionCommand().equals("Bookmark")) {
            JButton bk = new JButton("Bookmark" + bookmarkNumber);
            bk.setBounds(480, 20 + 30 * bookmarkNumber, 100, 30);
            add(bk);
            bk.addActionListener(this);

            bkmrk[bookmarkNumber] = currentNumber;
            bookmarkNumber++;
            currentNumber++;

            questionsChoices();

            if(currentNumber == 9) {
                button2.setText("Result");
            }

            setVisible(false);
            setVisible(true);
        }

```

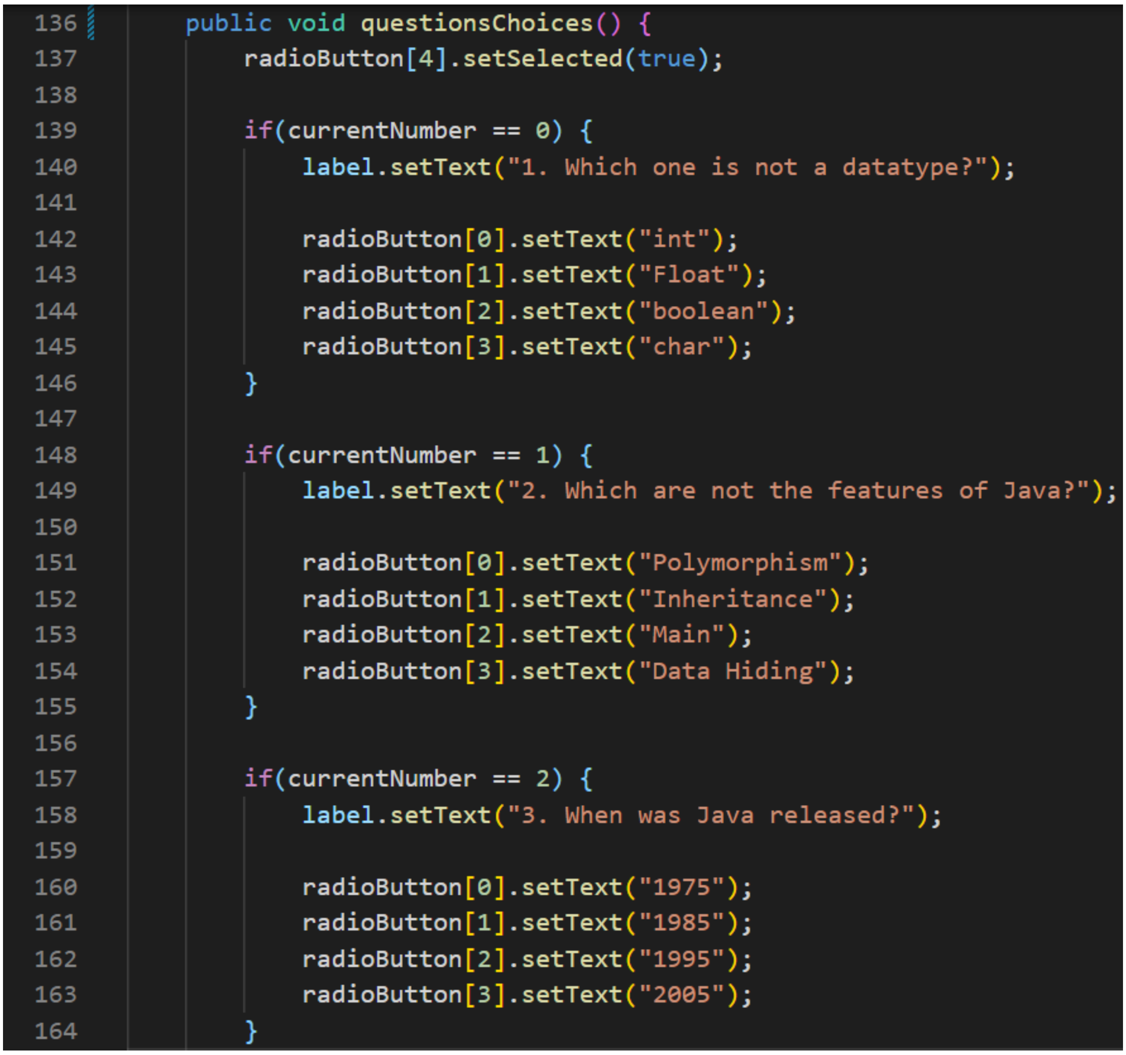


```java
        for(int i = 0, a = 1; i < bookmarkNumber; i++, a++) {
            if(actionEvent.getActionCommand().equals("Bookmark" + a)) {

                if(checkAnswer() == true) {
                    correctNumbers = correctNumbers + 1;
                }
                if(checkAnswer() == false) {
                    wrongNumbers = wrongNumbers + 1;
                }

                nowNumber = currentNumber;
                currentNumber = bkmrk[a];

                questionsChoices();

                ((JButton)actionEvent.getSource()).setEnabled(false);
                currentNumber = nowNumber;
            }
        }

```

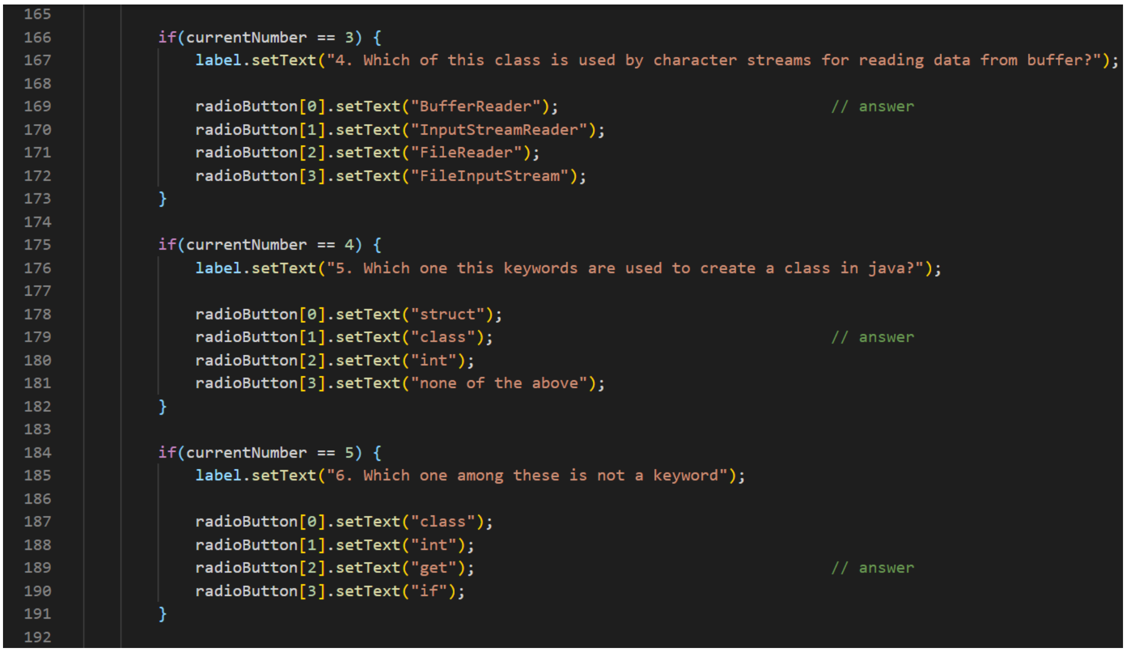


```java
        if(actionEvent.getActionCommand().equals("Result")) {                   // if "Result" button

            if(checkAnswer() == true) {                                         // if answered correct
                correctNumbers = correctNumbers + 1;                            // correctNumbers add
            }
            if(checkAnswer() == false) {                                        // if check funcion is
                wrongNumbers = wrongNumbers + 1;                                // wrongNumbers add by
            }

            currentNumber++;                                                    // increment currentNu

            JOptionPane.showMessageDialog(this, "Correct answer = " + correctNumbers);  // show window
            JOptionPane.showMessageDialog(this, "Wrong answer = " + wrongNumbers);      // show window
            JOptionPane.showMessageDialog(this, "Score = " + correctNumbers + " / " + currentNumber);

            System.exit(0);                                                     // exit automatically
        }
    }

```

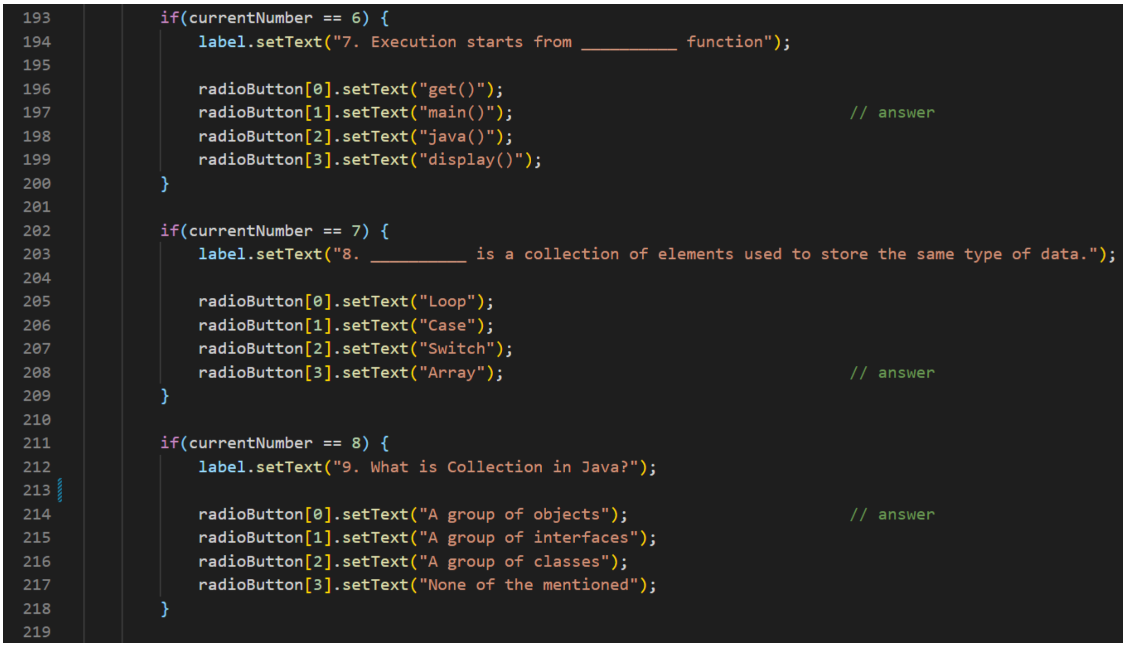
The actionPerformed function, which is implemented from actionListener, will decide the flow of the application, especially when using the buttons. Whenever the user presses a button, it will immediately direct to this function and run which condition the user chooses. The actions of the buttons will be decided based of the names of the buttons. Also, it can enable or disable buttons.



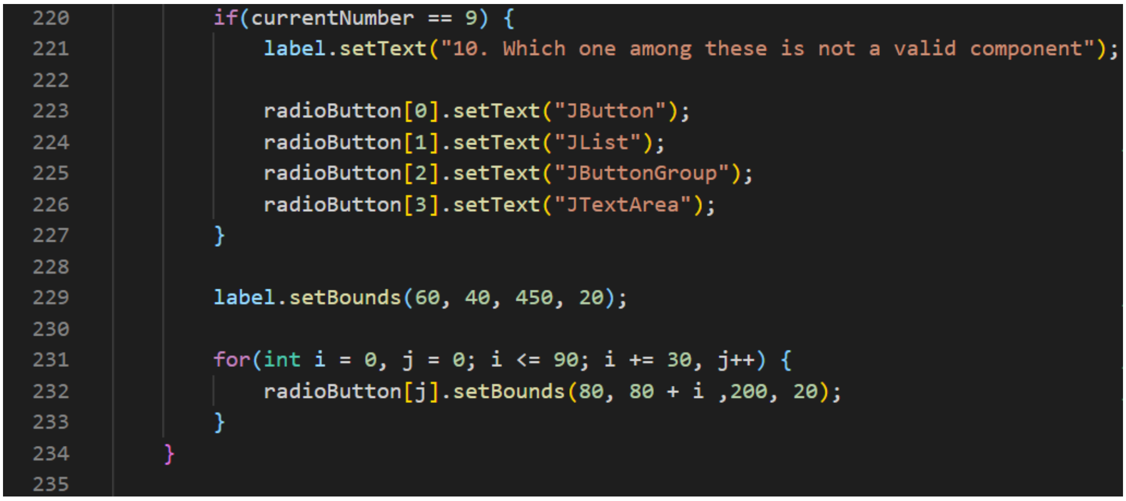
```java
    public void questionsChoices() {
        radioButton[4].setSelected(true);

        if(currentNumber == 0) {
            label.setText("1. Which one is not a datatype?");

            radioButton[0].setText("int");
            radioButton[1].setText("Float");
            radioButton[2].setText("boolean");
            radioButton[3].setText("char");
        }

        if(currentNumber == 1) {
            label.setText("2. Which are not the features of Java?");

            radioButton[0].setText("Polymorphism");
            radioButton[1].setText("Inheritance");
            radioButton[2].setText("Main");
            radioButton[3].setText("Data Hiding");
        }

        if(currentNumber == 2) {
            label.setText("3. When was Java released?");

            radioButton[0].setText("1975");
            radioButton[1].setText("1985");
            radioButton[2].setText("1995");
            radioButton[3].setText("2005");
        }
```



```java

        if(currentNumber == 3) {
            label.setText("4. Which of this class is used by character streams for reading data from buffer?");

            radioButton[0].setText("BufferReader");                          // answer
            radioButton[1].setText("InputStreamReader");
            radioButton[2].setText("FileReader");
            radioButton[3].setText("FileInputStream");
        }

        if(currentNumber == 4) {
            label.setText("5. Which one this keywords are used to create a class in java?");

            radioButton[0].setText("struct");
            radioButton[1].setText("class");                                 // answer
            radioButton[2].setText("int");
            radioButton[3].setText("none of the above");
        }

        if(currentNumber == 5) {
            label.setText("6. Which one among these is not a keyword");

            radioButton[0].setText("class");
            radioButton[1].setText("int");
            radioButton[2].setText("get");                                   // answer
            radioButton[3].setText("if");
        }

```

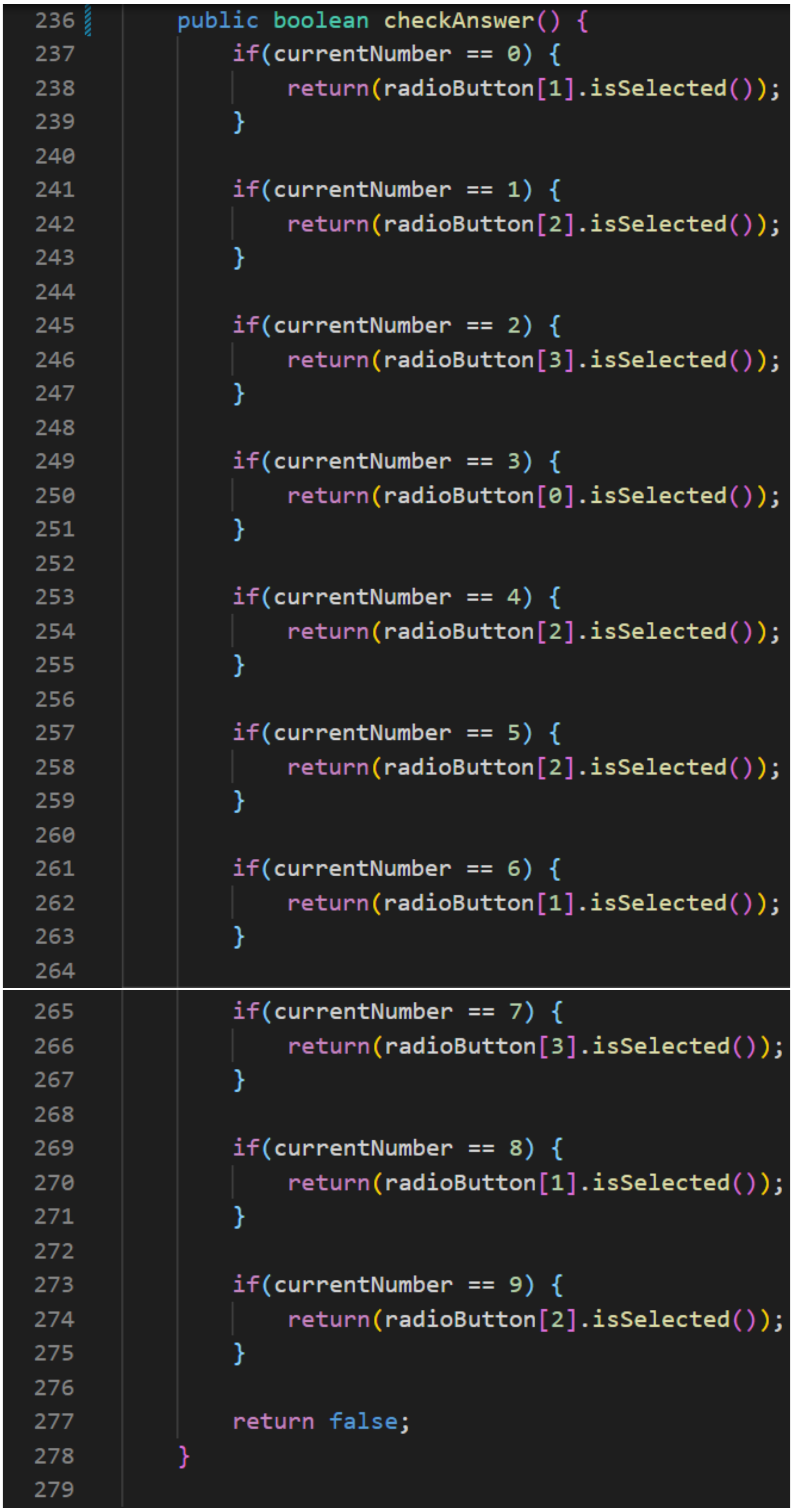


```java
        if(currentNumber == 6) {
            label.setText("7. Execution starts from __________ function");

            radioButton[0].setText("get()");
            radioButton[1].setText("main()");                                  // answer
            radioButton[2].setText("java()");
            radioButton[3].setText("display()");
        }

        if(currentNumber == 7) {
            label.setText("8. __________ is a collection of elements used to store the same type of data.");

            radioButton[0].setText("Loop");
            radioButton[1].setText("Case");
            radioButton[2].setText("Switch");
            radioButton[3].setText("Array");                                   // answer
        }

        if(currentNumber == 8) {
            label.setText("9. What is Collection in Java?");

            radioButton[0].setText("A group of objects");                      // answer
            radioButton[1].setText("A group of interfaces");
            radioButton[2].setText("A group of classes");
            radioButton[3].setText("None of the mentioned");
        }

```



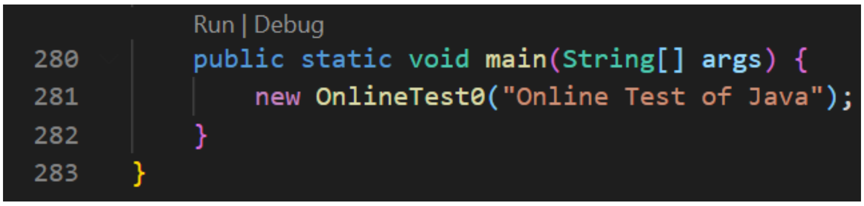
```java
        if(currentNumber == 9) {
            label.setText("10. Which one among these is not a valid component");

            radioButton[0].setText("JButton");
            radioButton[1].setText("JList");
            radioButton[2].setText("JButtonGroup");
            radioButton[3].setText("JTextArea");
        }

        label.setBounds(60, 40, 450, 20);

        for(int i = 0, j = 0; i <= 90; i += 30, j++) {
            radioButton[j].setBounds(80, 80 + i ,200, 20);
        }
    }

```

questionsChoices function contains all the questions, choices, and their locations in the interface using JLabel and radioButton.



```
    public boolean checkAnswer() {
        if(currentNumber == 0) {
            return(radioButton[1].isSelected());
        }

        if(currentNumber == 1) {
            return(radioButton[2].isSelected());
        }

        if(currentNumber == 2) {
            return(radioButton[3].isSelected());
        }

        if(currentNumber == 3) {
            return(radioButton[0].isSelected());
        }

        if(currentNumber == 4) {
            return(radioButton[2].isSelected());
        }

        if(currentNumber == 5) {
            return(radioButton[2].isSelected());
        }

        if(currentNumber == 6) {
            return(radioButton[1].isSelected());
        }

        if(currentNumber == 7) {
            return(radioButton[3].isSelected());
        }

        if(currentNumber == 8) {
            return(radioButton[1].isSelected());
        }

        if(currentNumber == 9) {
            return(radioButton[2].isSelected());
        }

        return false;
    }

```

checkAnswer function will check every number if the radioButton is chosen. If the radioButton is selected, it will return true and known as correct number. If the user chooses other radioButtons, then it will return false and known as wrong number.



```
    public static void main(String[] args) {
        new OnlineTest0("Online Test of Java");
    }
}
```

The main function will call the OnlineTest0 containing string for the title of the window.

VII. Project Link

https://github.com/Own20/GitHub/tree/main/Semester%202/Object%20Oriented%20Programming%20COMP6699001/Final%20Project

References

https://www.javatpoint.com/online-exam-project-in-java-swing-without-database

https://stackoverflow.com/