NVISION

Java miniproject

Ms. Nidhee Kamble (ID no. 171071037)  
Ms. Vidhi Rambhia (ID no. 171071015) **Project duration:** 5th November 2018 – 12th November 2018

Contents

[Abstract](#_Toc325634774) 1

[Software Used](#_Toc325634775) 2

[Software and Code](#_Toc325634776) 3

[Use and Demo](#_Toc325634777) 5

[Future Scope](#_Toc325634778) 9

[Bug Reports and Challenges](#_Toc325634779) 10

[Bibliography](#_Toc325634779) 11

# Abstract

### NVision is a student-centric application that assists the user in keeping a track of, and staying updated with their activities – academic and extra-curricular. A To-Do List and a Quiz may be utilized as seen fit to suit this purpose.

1. To Do List: An organizer wherein the user can enter the tasks to be completed that day. Upon completion, these can be checked and are automatically removed from the list. New tasks may be added. The application may be run for as long as desired. For increased efficiency, multiple lists can be created under different headings.
2. Quiz: A session with questions based on user’s academic subjects. These are multiple choice questions which help the user revise.

## MOTIVATION

As the developers of the application, we find its motivation rooted in our own needs: it is difficult for students to organize and schedule their day because of increased academic studies. *NVision* was conceived to give them a one-stop solution, and thus help them tackle this problem effectively and efficiently.

Software Used

## Source code

* Backend: Java (JDK v1.8)
* Frontend: Java (Swing)
* Version control: GitHub

## EXECUTION

* IDE: Eclipse (v2018-09)
* Command Prompt on Windows 10

# Software and Code

## GITHUB REPOSITORY

The project can be found at: <https://github.com/VidhiRambhia/JAVA_project.git>

The main directory consists of:

* **.git** (folder)
* **.metadata** (folder)
* **.recommenders (**folder)
* **Final\_Project** (folder) – Main implementation:

1. src (folder) – Has all the images, .java, and .class files
2. bin (folder) – Contains post-compilation elements
3. NVision – Documentation of the project

* **BackUp** (folder with individual folders ToDo and OnlineTest)
* **Code of Conduct** (MD file) – Lays out the GitHub etiquette concerning the project
* **README** (MD file) – An overview of the GitHub repository

The application can be run by running the file FirstPage.java in either Command Prompt or an IDE (preferably the same as was used while designing).

## EXISTING FEATURES

### The application, as of on 13th November 2018, works with the following facets fully implemented:

* **Main screen**  
  Introduces the application. Offers the user to either take a quiz or list down tasks for the day.
* **An Interactive Quiz Session**There are four subjects, that have a set questionnaire:

1. Java
2. DLD
3. Data Structures
4. EVS

The questions may be bookmarked for reference. Such questions may be revisited once till the quiz is submitted.

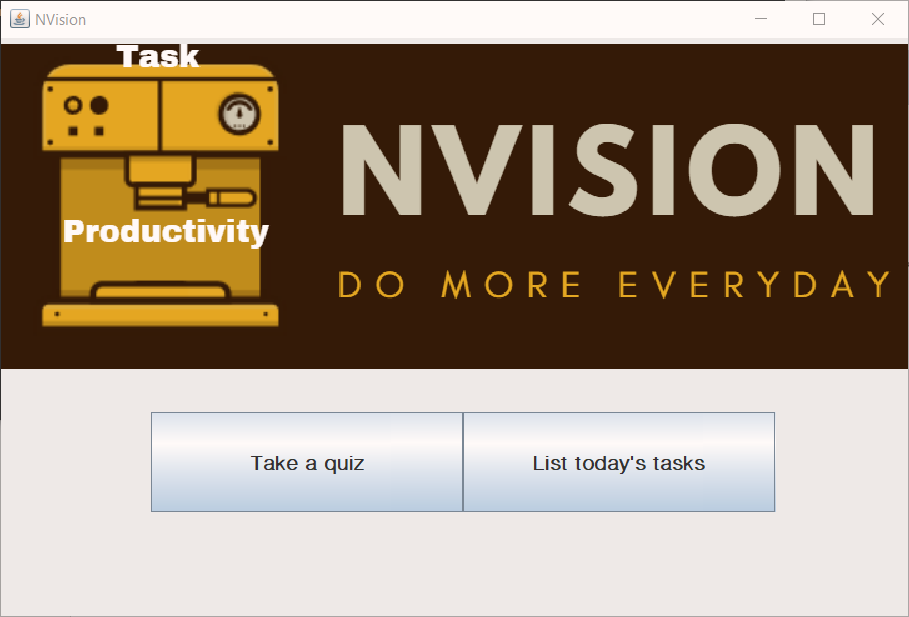
At the end of the session, the user gets to know his final score. Thus helping him stay up-to-date with his academic curriculum.

* **To-Do List**Three tasks can be listed. Once a task is marked as done, it is cleared and a new task may be added in its place. The cycle continues as long as the user operates on this keeping the frame open.

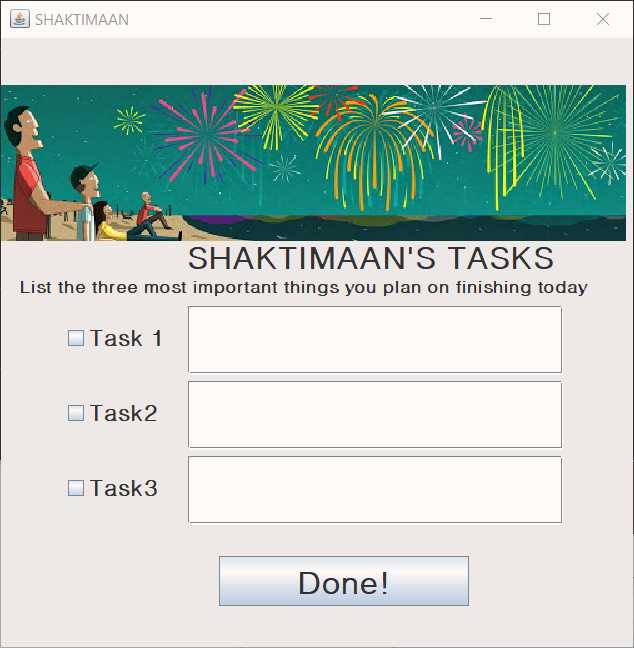
Frames for To-Do Lists for multiple users can be generated with each run. All such frames function independently of each other.

# Use and Demo

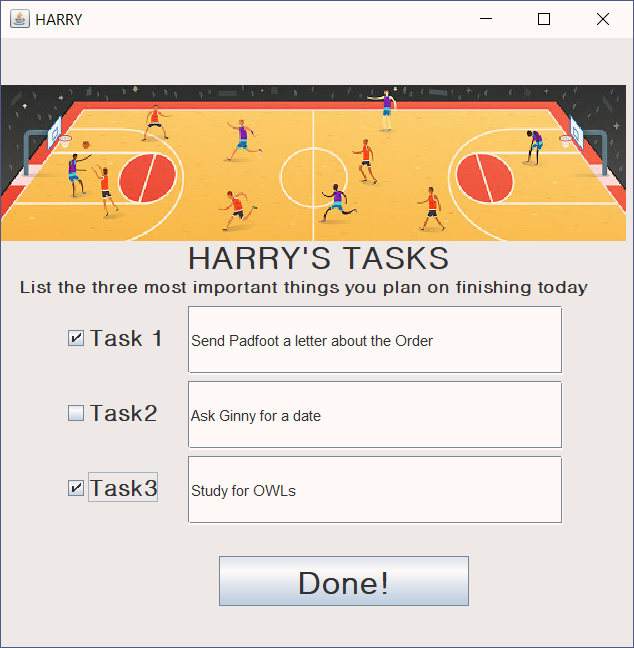
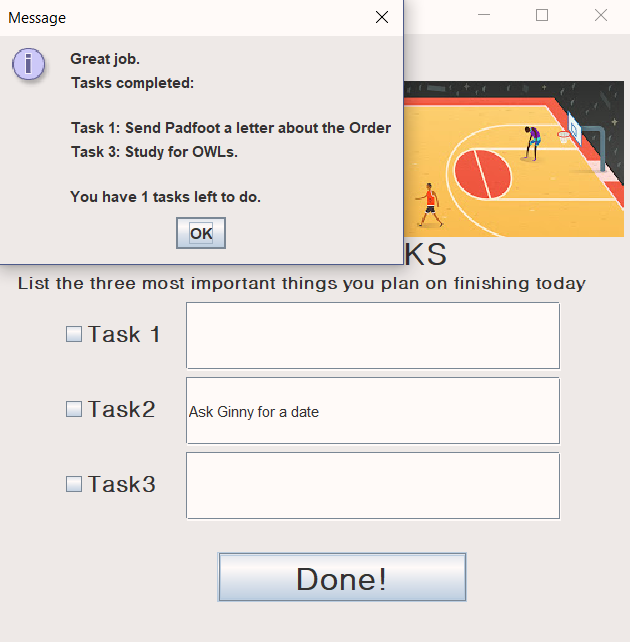
The application opens on the Navigator Page. ‘Take a quiz’ and ‘List today’s tasks’, on clicking, redirect to the quiz menu and To-Do List respectively.



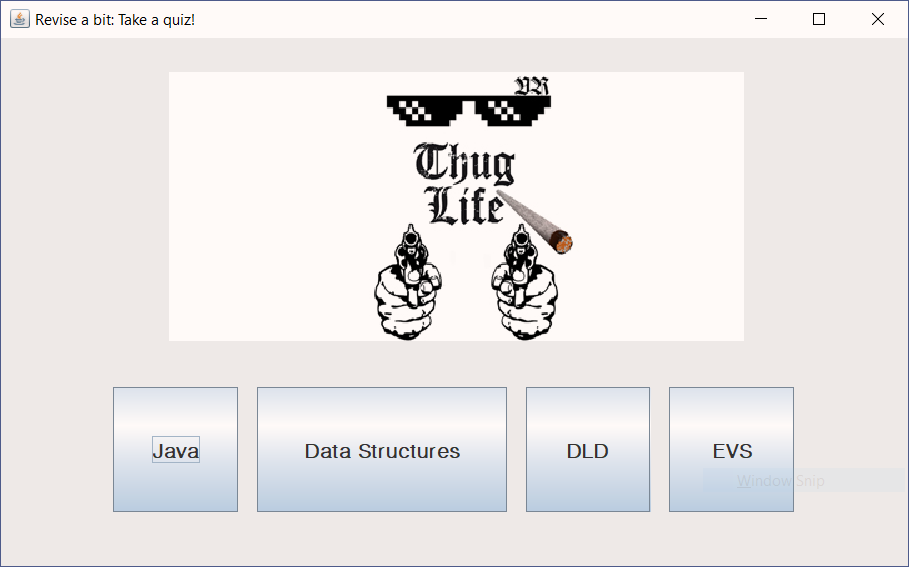
On selecting the second option, a prompt for entering a name is displayed. The name so entered appears as the header on the list of the new To-Do List. An empty To-Do List looks like this:



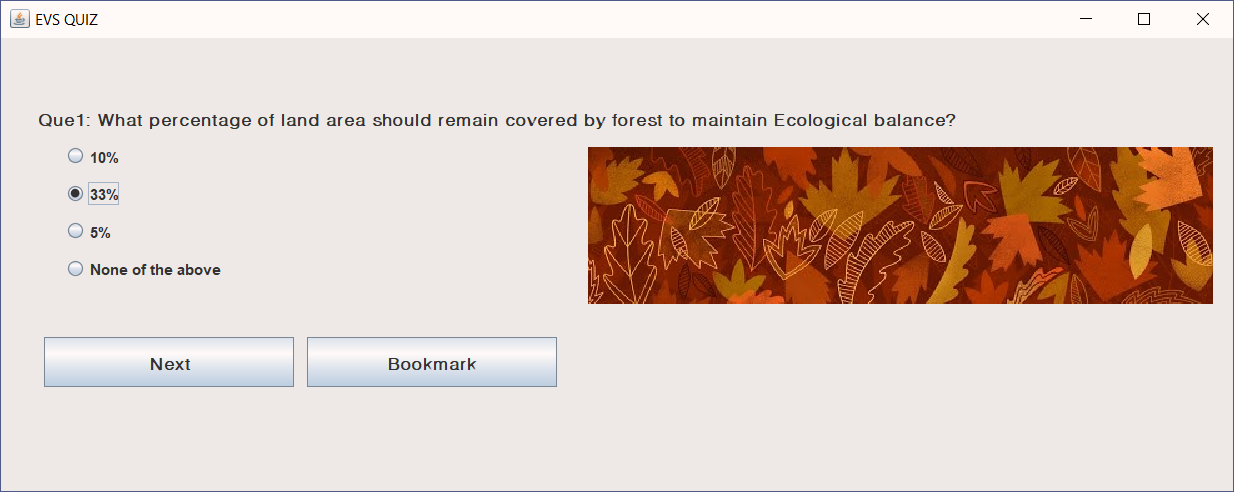
While running in an IDE or even using Windows cmd or Linux Terminal, multiple lists can be generated. The tasks in the list are tracked according to their completion independently for different lists.

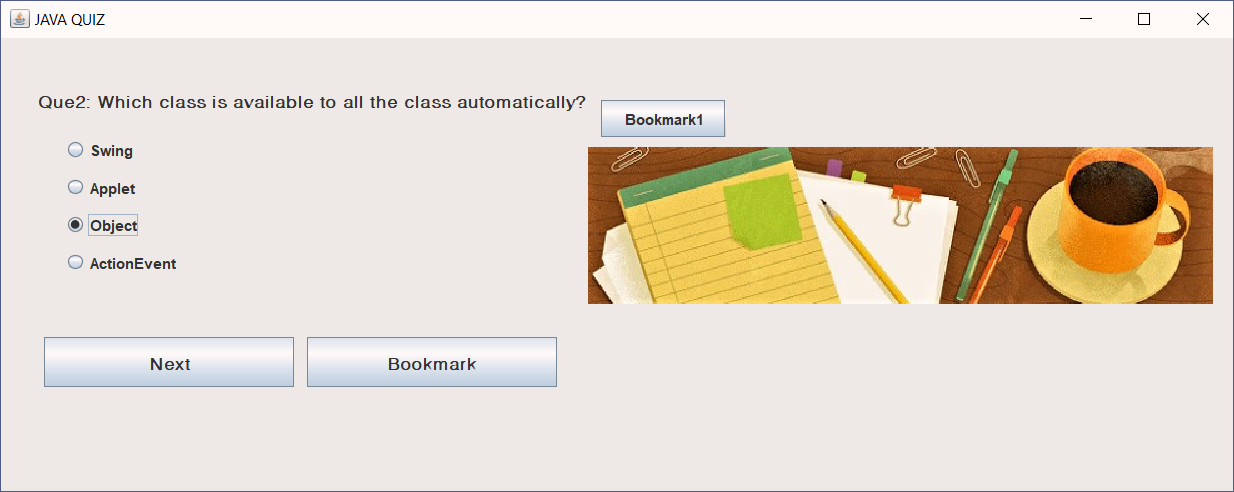
A task one marked as done, is cleared. A reminder is displayed on clicking on ‘Done!’. New asks may be added.

On selecting the option for the quiz from the Navigation Page, the following frame appears:  
  


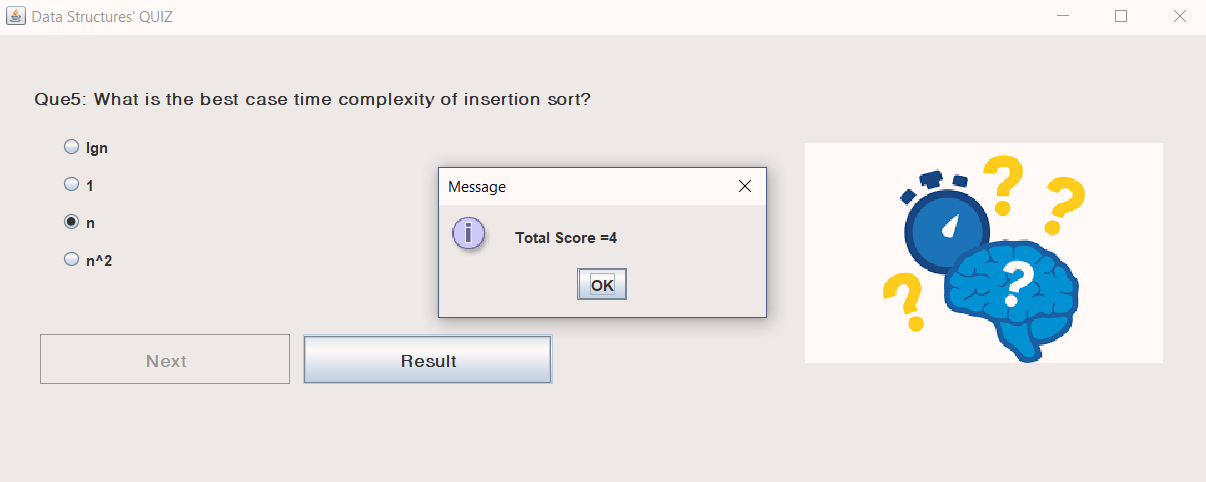
Questions are displayed according to the subject selected:



The bookmarked questions may be visited once again by clicking on the ‘Bookmark[number]’ button. The button appears only if a question has been previously labelled accordingly by the user:



After reaching the final question, the result is displayed in a message prompt that is flashed on clicking on ‘Result’:



# Source Code

// To-do List

import javax.swing.\*; // JFrame and components

import java.awt.\*; // for Font class(es)

import java.awt.event.\*; // for actionListeners

import java.util.Random; // for image idices picked randomly

import java.util.Scanner;

import java.io.\*;

public class ToDo extends JFrame implements ActionListener

{

// all Swing variables are (implicitly) declared static/global since they are just indicators for actual data they point to

// (components remain the same throughout the program)

JLabel l, l2;

JCheckBox cb1,cb2,cb3;

JButton b;

JTextField tf1, tf2, tf3;

ImageIcon i[] = new ImageIcon[16];

JLabel il[] = new JLabel[16];

int more, n=0;

Random rand = new Random();

static String name="";

Scanner s = new Scanner(System.in);

ToDo(String name) // constructor

{

// image arsenal

i[0] = new ImageIcon(getClass().getResource("birthday.jpg"));

il[0] = new JLabel(i[0]);

il[0].setBounds(0,0,500, 200);

i[1] = new ImageIcon(getClass().getResource("books.jpg"));

il[1] = new JLabel(i[1]);

il[1].setBounds(0,0,500, 200);

i[2] = new ImageIcon(getClass().getResource("bowling.jpg"));

il[2] = new JLabel(i[2]);

il[2].setBounds(0,0,500, 200);

i[3] = new ImageIcon(getClass().getResource("breakfast.jpg"));

il[3] = new JLabel(i[3]);

il[3].setBounds(0,0,500, 200);

i[4] = new ImageIcon(getClass().getResource("coffee.jpg"));

il[4] = new JLabel(i[4]);

il[4].setBounds(0,0,500, 200);

i[5] = new ImageIcon(getClass().getResource("desk.jpg"));

il[5] = new JLabel(i[5]);

il[5].setBounds(0,0,500, 200);

i[6] = new ImageIcon(getClass().getResource("dinner.jpg"));

il[6] = new JLabel(i[6]);

il[6].setBounds(0,0,500, 200);

i[7] = new ImageIcon(getClass().getResource("fair.jpg"));

il[7] = new JLabel(i[7]);

il[7].setBounds(0,0,500, 200);

i[8] = new ImageIcon(getClass().getResource("fireworks.jpg"));

il[8] = new JLabel(i[8]);

il[8].setBounds(0,0,500, 200);

i[9] = new ImageIcon(getClass().getResource("game.jpg"));

il[9] = new JLabel(i[9]);

il[9].setBounds(0,0,500, 200);

i[10] = new ImageIcon(getClass().getResource("leaves.jpg"));

il[10] = new JLabel(i[10]);

il[10].setBounds(0,0,500, 200);

i[11] = new ImageIcon(getClass().getResource("marathon.jpg"));

il[11] = new JLabel(i[11]);

il[11].setBounds(0,0,500, 200);

i[12] = new ImageIcon(getClass().getResource("pizza.jpg"));

il[12] = new JLabel(i[12]);

il[12].setBounds(0,0,500, 200);

i[13] = new ImageIcon(getClass().getResource("pool.jpg"));

il[13] = new JLabel(i[13]);

il[13].setBounds(0,0,500, 200);

i[14] = new ImageIcon(getClass().getResource("tie.jpg"));

il[14] = new JLabel(i[14]);

il[14].setBounds(0,0,500, 200);

i[15] = new ImageIcon(getClass().getResource("traffic.jpg"));

il[15] = new JLabel(i[15]);

il[15].setBounds(0,0,500, 200);

// image arsenal ends

System.out.println("Enter your name");

name = s.nextLine();

// initializing required font styles

Font prompt = new Font("Helvlight", Font.BOLD, 25);

Font description = new Font("Helvlight", Font.BOLD, 18);

Font caption = new Font("Helvlight", Font.BOLD, 14);

// adding labels and descriptions

name = name.toUpperCase();

l=new JLabel(name+"'S TASKS");

l.setFont(prompt);

l.setBounds(150,165,300,20);

l2 = new JLabel("List the three most important things you plan on finishing today");

l2.setFont(caption);

l2.setBounds(15,188,600,20);

// three checkboxes for three tasks; a textfield for each.

// aiming to centrally align them all; adjusting dimensions accordingly

cb1=new JCheckBox("Task 1");

cb1.setFont(description);

cb1.setBounds(50,215,90,50);

tf1 = new JTextField();

tf1.setBounds(150,215,300,55);

cb2=new JCheckBox("Task2");

cb2.setFont(description);

cb2.setBounds(50,275,90,50);

tf2 = new JTextField();

tf2.setBounds(150,275,300,55);

cb3=new JCheckBox("Task3");

cb3.setFont(description);

cb3.setBounds(50,335,90,50);

tf3 = new JTextField();

tf3.setBounds(150,335,300,55);

// 'submit' button

b=new JButton("Done!");

b.setFont(prompt);

b.setBounds(175,415,200,40);

b.addActionListener(this);

// adding all components to the frame

add(l); add(l2);

n = rand.nextInt(15);

add(il[n]);

add(il[0]);

add(cb1); add(cb2); add(cb3);

add(tf1);add(tf2);add(tf3);

add(b);

// frame size. The method is called directly since the 'this' object is implicitly referred to inside the constructor

setSize(520,525);

setTitle(name);

setLayout(null);

setLocation(690,250);

setVisible(true); // frame visible hai (how stupid)

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

public void actionPerformed(ActionEvent e)

{

System.out.println(n);

String s="";

more=0;

if(cb1.isSelected())

{

s = s + "Task 1: " + tf1.getText() + "\n";

tf1.setText("");

cb1.setSelected(false);

more++;

}

if(cb2.isSelected())

{

s = s + "Task 2: " + tf2.getText() + "\n";

tf2.setText("");

cb2.setSelected(false);

more++;

}

if(cb3.isSelected())

{

s = s + "Task 3: " + tf3.getText() + ".";

tf3.setText("");

cb3.setSelected(false);

more++;

}

JOptionPane.showMessageDialog(this,"Great job.\nTasks completed:\n\n"+s+"\n\nYou have "+(3-more)+" tasks left to do.");

}

public static void main(String[] args)

{

new ToDo(name);

}

}

/\*Online DLD Paper Test\*/

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class DLD extends JFrame implements ActionListener

{

ImageIcon image1;

JLabel l,l2;

JRadioButton jb[]=new JRadioButton[5];

JButton b1,b2;

ButtonGroup bg;

int count=0,current=0,x=1,y=1,now=0;

int m[]=new int[10];

Font caption = new Font("Helvlight", Font.BOLD,14);

DLD(String s)

{

super(s);

image1 = new ImageIcon(getClass().getResource("quiz6.jpg"));

l=new JLabel();

l2=new JLabel(image1);

add(l2);

add(l);

bg=new ButtonGroup();

for(int i=0;i<5;i++)

{

jb[i]=new JRadioButton();

jb[i].setFont(caption);

add(jb[i]);

bg.add(jb[i]);

}

b1=new JButton("Next");

b2=new JButton("Bookmark");

b1.addActionListener(this);

b2.addActionListener(this);

add(b1);add(b2);

set();

b1.setFont(caption);

b2.setFont(caption);

l.setFont(caption);

l.setBounds(30,40,750,50);

l2.setBounds(550, 0, 445, 450);

jb[0].setBounds(50,80,400,30);

jb[1].setBounds(50,110,400,30);

jb[2].setBounds(50,140,400,30);

jb[3].setBounds(50,170,400,30);

b1.setBounds(35,240,200,40);

b2.setBounds(245,240,200,40);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(null);

setLocation(460,280);

setVisible(true);

setSize(985,500);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==b1)

{

if(check())

count=count+1;

current++;

set();

if(current==9)

{

b1.setEnabled(false);

b2.setText("Result");

}

}

if(e.getActionCommand().equals("Bookmark"))

{

JButton bk=new JButton("Bookmark"+x);

bk.setBounds(480,20+30\*x,100,30);

add(bk);

bk.addActionListener(this);

m[x]=current;

x++;

current++;

set();

if(current==9)

b2.setText("Result");

setVisible(false);

setVisible(true);

}

for(int i=0,y=1;i<x;i++,y++)

{

if(e.getActionCommand().equals("Bookmark"+y))

{

if(check())

count=count+1;

now=current;

current=m[y];

set();

((JButton)e.getSource()).setEnabled(false);

current=now;

}

}

if(e.getActionCommand().equals("Result"))

{

if(check())

count=count+1;

current++;

//System.out.println("correct ans="+count);

JOptionPane.showMessageDialog(this," Total Score ="+count);

System.exit(0);

}

}

void set()

{

jb[4].setSelected(true);

if(current==0)

{

l.setText("Que1: Which is a read only memory?");

jb[0].setText("ROM");jb[1].setText("EPROM");jb[2].setText("RAM");jb[3].setText("Registers");

}

if(current==1)

{

l.setText("Que2: How many states does a decimal counter have?");

jb[0].setText("5");jb[1].setText("10");jb[2].setText("15");jb[3].setText("20");

}

if(current==2)

{

l.setText("Que3:How many BCD counters does a three decade counter have ?");

jb[0].setText("2");jb[1].setText("3");jb[2].setText("4");jb[3].setText("5");

}

if(current==3)

{

l.setText("Que4: Simplest Registers consist only of");

jb[0].setText("counter");jb[1].setText("EPROM");jb[2].setText("latch");jb[3].setText("flip-flop");

}

if(current==4)

{

l.setText("Que5: Ripple counters are also called");

jb[0].setText("SSI counters");jb[1].setText("asynchronous counters");jb[2].setText("synchronous counters");jb[3].setText("VLSI counters");

}

if(current==5)

{

l.setText("Que6: First operator precedence for evaluating Boolean expressions is?");

jb[0].setText("paranthesis");jb[1].setText("AND");jb[2].setText("OR");jb[3].setText("NOT");

}

if(current==6)

{

l.setText("Que7: What is x.x equal to?");

jb[0].setText("x");jb[1].setText("1");jb[2].setText("0");

jb[3].setText("x'");

}

if(current==7)

{

l.setText("Que8: Which of these represents AND operation?");

jb[0].setText("+");jb[1].setText("/");jb[2].setText(".");

jb[3].setText("\*");

}

if(current==8)

{

l.setText("Que9: Which of these represents OR operation?");

jb[0].setText("+");jb[1].setText("/");jb[2].setText(".");jb[3].setText("\*");

}

if(current==9)

{

l.setText("Que10: What is x+x'?");

jb[0].setText("0");jb[1].setText("1");jb[2].setText("x");

jb[3].setText("x'");

}

l.setBounds(30,40,450,20);

for(int i=0,j=0;i<=90;i+=30,j++)

jb[j].setBounds(50,80+i,200,20);

}

boolean check()

{

if(current==0)

return(jb[1].isSelected());

if(current==1)

return(jb[2].isSelected());

if(current==2)

return(jb[3].isSelected());

if(current==3)

return(jb[0].isSelected());

if(current==4)

return(jb[0].isSelected());

if(current==5)

return(jb[2].isSelected());

if(current==6)

return(jb[1].isSelected());

if(current==7)

return(jb[3].isSelected());

if(current==8)

return(jb[1].isSelected());

if(current==9)

return(jb[2].isSelected());

return false;

}

public static void main(String s[])

{

new DLD("Test your DLD skills");

}

}

/\*Online Java Paper Test\*/

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class JAVA extends JFrame implements ActionListener

{

ImageIcon image1;

JLabel l,l2;

JRadioButton jb[]=new JRadioButton[5];

JButton b1,b2;

ButtonGroup bg;

int count=0,current=0,x=1,y=1,now=0;

int m[]=new int[10];

Font caption = new Font("Helvlight", Font.BOLD,14);

JAVA(String s)

{

super(s);

image1 = new ImageIcon(getClass().getResource("desk.jpg"));

l=new JLabel();

l2=new JLabel(image1);

add(l2);

add(l);

bg=new ButtonGroup();

for(int i=0;i<5;i++)

{

jb[i]=new JRadioButton();

add(jb[i]);

bg.add(jb[i]);

}

b1=new JButton("Next");

b2=new JButton("Bookmark");

b1.addActionListener(this);

b2.addActionListener(this);

add(b1);add(b2);

set();

b1.setFont(caption);

b2.setFont(caption);

l.setFont(caption);

l.setBounds(30,40,750,50);

l2.setBounds(470, 0, 500, 300);

jb[0].setBounds(50,80,400,30);

jb[1].setBounds(50,110,400,30);

jb[2].setBounds(50,140,400,30);

jb[3].setBounds(50,170,400,30);

b1.setBounds(35,240,200,40);

b2.setBounds(245,240,200,40);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(null);

setLocation(460,280);

setVisible(true);

setSize(1000,400);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==b1)

{

if(check())

count=count+1;

current++;

set();

if(current==9)

{

b1.setEnabled(false);

b2.setText("Result");

}

}

if(e.getActionCommand().equals("Bookmark"))

{

JButton bk=new JButton("Bookmark"+x);

bk.setBounds(480,20+30\*x,100,30);

add(bk);

bk.addActionListener(this);

m[x]=current;

x++;

current++;

set();

if(current==9)

b2.setText("Result");

setVisible(false);

setVisible(true);

}

for(int i=0,y=1;i<x;i++,y++)

{

if(e.getActionCommand().equals("Bookmark"+y))

{

if(check())

count=count+1;

now=current;

current=m[y];

set();

((JButton)e.getSource()).setEnabled(false);

current=now;

}

}

if(e.getActionCommand().equals("Result"))

{

if(check())

count=count+1;

current++;

//System.out.println("correct ans="+count);

JOptionPane.showMessageDialog(this," Total Score ="+count);

System.exit(0);

}

}

void set()

{

jb[4].setSelected(true);

if(current==0)

{

l.setText("Que1: Which one among these is not a primitive datatype?");

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

}

if(current==1)

{

l.setText("Que2: Which class is available to all the class automatically?");

jb[0].setText("Swing");jb[1].setText("Applet");jb[2].setText("Object");jb[3].setText("ActionEvent");

}

if(current==2)

{

l.setText("Que3: Which package is directly available to our class without importing it?");

jb[0].setText("swing");jb[1].setText("applet");jb[2].setText("net");jb[3].setText("lang");

}

if(current==3)

{

l.setText("Que4: String class is defined in which package?");

jb[0].setText("lang");jb[1].setText("Swing");jb[2].setText("Applet");jb[3].setText("awt");

}

if(current==4)

{

l.setText("Que5: Which institute is best for java coaching?");

jb[0].setText("ChitraSahu.com");jb[1].setText("Aptech");jb[2].setText("SSS IT");jb[3].setText("jtek");

}

if(current==5)

{

l.setText("Que6: Which one among these is not a keyword?");

jb[0].setText("class");jb[1].setText("int");jb[2].setText("get");jb[3].setText("if");

}

if(current==6)

{

l.setText("Que7: Which one among these is not a class? ");

jb[0].setText("Swing");jb[1].setText("Actionperformed");jb[2].setText("ActionEvent");

jb[3].setText("Button");

}

if(current==7)

{

l.setText("Que8: which one among these is not a function of Object class?");

jb[0].setText("toString");jb[1].setText("finalize");jb[2].setText("equals");

jb[3].setText("getDocumentBase");

}

if(current==8)

{

l.setText("Que9: which function is not present in Applet class?");

jb[0].setText("init");jb[1].setText("main");jb[2].setText("start");jb[3].setText("destroy");

}

if(current==9)

{

l.setText("Que10: Which one among these is not a valid component?");

jb[0].setText("JButton");jb[1].setText("JList");jb[2].setText("JButtonGroup");

jb[3].setText("JTextArea");

}

l.setBounds(30,40,450,20);

for(int i=0,j=0;i<=90;i+=30,j++)

jb[j].setBounds(50,80+i,200,20);

}

boolean check()

{

if(current==0)

return(jb[1].isSelected());

if(current==1)

return(jb[2].isSelected());

if(current==2)

return(jb[3].isSelected());

if(current==3)

return(jb[0].isSelected());

if(current==4)

return(jb[0].isSelected());

if(current==5)

return(jb[2].isSelected());

if(current==6)

return(jb[1].isSelected());

if(current==7)

return(jb[3].isSelected());

if(current==8)

return(jb[1].isSelected());

if(current==9)

return(jb[2].isSelected());

return false;

}

public static void main(String s[])

{

new JAVA("Test your JAVA skills!");

}

}

/\*Online DS Paper Test\*/

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class DS extends JFrame implements ActionListener

{

ImageIcon image1;

JLabel l,l2;

JRadioButton jb[]=new JRadioButton[5];

JButton b1,b2;

ButtonGroup bg;

int count=0,current=0,x=1,y=1,now=0;

int m[]=new int[10];

Font caption = new Font("Helvlight", Font.BOLD,14);

DS(String s)

{

super(s);

image1 = new ImageIcon(getClass().getResource("quiz3.png"));

l=new JLabel();

l2=new JLabel(image1);

add(l2);

add(l);

bg=new ButtonGroup();

for(int i=0;i<5;i++)

{

jb[i]=new JRadioButton();

add(jb[i]);

bg.add(jb[i]);

}

b1=new JButton("Next");

b2=new JButton("Bookmark");

b1.addActionListener(this);

b2.addActionListener(this);

add(b1);add(b2);

set();

b1.setFont(caption);

b2.setFont(caption);

l.setFont(caption);

l.setBounds(30,40,750,50);

l2.setBounds(540, 0, 500, 350);

jb[0].setBounds(50,80,400,30);

jb[1].setBounds(50,110,400,30);

jb[2].setBounds(50,140,400,30);

jb[3].setBounds(50,170,400,30);

b1.setBounds(35,240,200,40);

b2.setBounds(245,240,200,40);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(null);

setLocation(460,280);

setVisible(true);

setSize(985,400);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==b1)

{

if(check())

count=count+1;

current++;

set();

if(current==4)

{

b1.setEnabled(false);

b2.setText("Result");

}

}

if(e.getActionCommand().equals("Bookmark"))

{

JButton bk=new JButton("Bookmark"+x);

bk.setBounds(480,20+30\*x,100,30);

add(bk);

bk.addActionListener(this);

m[x]=current;

x++;

current++;

set();

if(current==4)

b2.setText("Result");

setVisible(false);

setVisible(true);

}

for(int i=0,y=1;i<x;i++,y++)

{

if(e.getActionCommand().equals("Bookmark"+y))

{

if(check())

count=count+1;

now=current;

current=m[y];

set();

((JButton)e.getSource()).setEnabled(false);

current=now;

}

}

if(e.getActionCommand().equals("Result"))

{

if(check())

count=count+1;

current++;

//System.out.println("correct ans="+count);

JOptionPane.showMessageDialog(this," Total Score ="+count);

System.exit(0);

}

}

void set()

{

jb[4].setSelected(true);

if(current==0)

{

l.setText("Que1: What is the time complexity of bubble sort?");

jb[0].setText("n");jb[1].setText("n^2");jb[2].setText("nlgn");jb[3].setText("None ");

}

if(current==1)

{

l.setText("Que2: What is the space complexity of bubble sort?");

jb[0].setText("n");jb[1].setText("n^2");jb[2].setText("1");jb[3].setText("None ");

}

if(current==2)

{

l.setText("Que3: Which data structure works on the basis of FIFO?");

jb[0].setText("queue");jb[1].setText("stack");jb[2].setText("binary tree");jb[3].setText("None ");

}

if(current==3)

{

l.setText("Que4: Which data structure works on the basis of LIFO?");

jb[0].setText("queue");jb[1].setText("stack");jb[2].setText("binary tree");jb[3].setText("None ");

}

if(current==4)

{

l.setText("Que5: What is the best case time complexity of insertion sort?");

jb[0].setText("lgn");jb[1].setText("1");jb[2].setText("n");jb[3].setText("n^2");

}

l.setBounds(30,40,450,20);

for(int i=0,j=0;i<=90;i+=30,j++)

jb[j].setBounds(50,80+i,200,20);

}

boolean check()

{

if(current==0)

return(jb[1].isSelected());

if(current==1)

return(jb[2].isSelected());

if(current==2)

return(jb[0].isSelected());

if(current==3)

return(jb[1].isSelected());

if(current==4)

return(jb[2].isSelected());

return false;

}

public static void main(String s[])

{

new DS("Test your Data Structure skills");

}

}

/\*Online EVS Paper Test\*/

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class EVS extends JFrame implements ActionListener

{

ImageIcon image1;

JLabel l,l2;

JRadioButton jb[]=new JRadioButton[5];

JButton b1,b2;

ButtonGroup bg;

int count=0,current=0,x=1,y=1,now=0;

int m[]=new int[10];

Font caption = new Font("Helvlight", Font.BOLD,14);

EVS(String s)

{

super(s);

image1 = new ImageIcon(getClass().getResource("leaves.jpg"));

l=new JLabel();

l2=new JLabel(image1);

add(l2);

add(l);

bg=new ButtonGroup();

for(int i=0;i<5;i++)

{

jb[i]=new JRadioButton();

add(jb[i]);

bg.add(jb[i]);

}

b1=new JButton("Next");

b2=new JButton("Bookmark");

b1.addActionListener(this);

b2.addActionListener(this);

add(b1);add(b2);

set();

b1.setFont(caption);

b2.setFont(caption);

l.setFont(caption);

l.setBounds(30,40,750,50);

l2.setBounds(470, 0, 500, 300);

jb[0].setBounds(50,80,400,30);

jb[1].setBounds(50,110,400,30);

jb[2].setBounds(50,140,400,30);

jb[3].setBounds(50,170,400,30);

b1.setBounds(35,240,200,40);

b2.setBounds(245,240,200,40);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(null);

setLocation(460,280);

setVisible(true);

setSize(1000,400);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==b1)

{

if(check())

count=count+1;

current++;

set();

if(current==4)

{

b1.setEnabled(false);

b2.setText("Result");

}

}

if(e.getActionCommand().equals("Bookmark"))

{

JButton bk=new JButton("Bookmark"+x);

bk.setBounds(480,20+30\*x,100,30);

add(bk);

bk.addActionListener(this);

m[x]=current;

x++;

current++;

set();

if(current==4)

b2.setText("Result");

setVisible(false);

setVisible(true);

}

for(int i=0,y=1;i<x;i++,y++)

{

if(e.getActionCommand().equals("Bookmark"+y))

{

if(check())

count=count+1;

now=current;

current=m[y];

set();

((JButton)e.getSource()).setEnabled(false);

current=now;

}

}

if(e.getActionCommand().equals("Result"))

{

if(check())

count=count+1;

current++;

//System.out.println("correct ans="+count);

JOptionPane.showMessageDialog(this," Total Score ="+count);

System.exit(0);

}

}

void set()

{

jb[4].setSelected(true);

if(current==0)

{

l.setText("Que1: What percentage of land area should remain covered by forest to maintain Ecological balance?");

jb[0].setText("10%");jb[1].setText("33%");jb[2].setText("5%");jb[3].setText("None of the above");

}

if(current==1)

{

l.setText("Que2: What is the full form of BOD?");

jb[0].setText("Biological Ozone Demand");jb[1].setText("Biological Oxygen Demand");jb[2].setText("Bonded Oxygen Demand");jb[3].setText("None of the above");

}

if(current==2)

{

l.setText("Que3: How much percentage of nitrogen consist in Earth's Atmosphere?");

jb[0].setText("25%");jb[1].setText("12%");jb[2].setText("92%");jb[3].setText("78%");

}

if(current==3)

{

l.setText("Que4: The disease Itai Itai is caused by?");

jb[0].setText("Manganese");jb[1].setText("Carbon");jb[2].setText("Cadmium");jb[3].setText("Ammonia");

}

if(current==4)

{

l.setText("Que5: When was the Environment Protection Act formulated?");

jb[0].setText("1980");jb[1].setText("1991");jb[2].setText("1986");jb[3].setText("1975");

}

l.setBounds(30,40,450,20);

for(int i=0,j=0;i<=90;i+=30,j++)

jb[j].setBounds(50,80+i,200,20);

}

boolean check()

{

if(current==0)

return(jb[1].isSelected());

if(current==1)

return(jb[1].isSelected());

if(current==2)

return(jb[3].isSelected());

if(current==3)

return(jb[2].isSelected());

if(current==4)

return(jb[2].isSelected());

return false;

}

public static void main(String s[])

{

new EVS("Test your EVS skills");

}

}

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

public class MainFrame extends JFrame implements ActionListener {

JButton b1,b2,b3,b4;

ImageIcon image1;

JLabel l;

int option=0;

Font caption = new Font("Helvlight", Font.BOLD, 16);

public MainFrame(String s)

{

super(s);

image1 = new ImageIcon(getClass().getResource("Thug Life.jpg"));

l=new JLabel(image1);

add(l);

b1=new JButton("Java");

b2=new JButton("Data Structures");

b3= new JButton("DLD");

b4=new JButton("EVS");

// button text initialized

b1.setFont(caption);

b2.setFont(caption);

b3.setFont(caption);

b4.setFont(caption);

// font modified

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

// actionListeners added for each

add(b1);add(b2); add(b3); add(b4); //add all buttons

l.setBounds(0,5,730,260);

b1.setBounds(90,280,100,100);

b2.setBounds(205,280,200,100);

b3.setBounds(420,280,100,100);

b4.setBounds(535,280,100,100);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(null);

setLocation(640,300);

setVisible(true);

setSize(740,460);

}

public void actionPerformed(ActionEvent e) {

if(e.getSource()==b1)

{

option=1;

new JAVA("JAVA QUIZ");

dispose();

}

if(e.getSource()==b2)

{

option=2;

new DS("Data Structures' QUIZ");

dispose();

}

if(e.getSource()==b3)

{

option=3;

new DLD("DLD QUIZ");

dispose();

}

if(e.getSource()==b4)

{

option=4;

new EVS("EVS QUIZ");

dispose();

}

}

public static void main(String[] args) {

new MainFrame("Let's quiz");

}

}

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

public class FirstPage extends JFrame implements ActionListener {

JButton b1,b2;

ImageIcon image1;

JLabel l;

int option=0;

Font caption = new Font("Helvlight", Font.BOLD, 16);

public FirstPage(String s)

{

super(s);

image1 = new ImageIcon(getClass().getResource("logo.png"));

l=new JLabel(image1);

add(l);

b1=new JButton("Take a quiz");

b1.addActionListener(this);

b1.setFont(caption);

b2=new JButton("List today's tasks");

b2.addActionListener(this);

b2.setFont(caption);

add(b1);add(b2);

l.setBounds(0,5,730,260);

b1.setBounds(120,300,250,80);

b2.setBounds(370,300,250,80);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(null);

setLocation(640,300);

setVisible(true);

setSize(740,500);

}

public void actionPerformed(ActionEvent e) {

if(e.getSource()==b1)

{

option=1;

new MainFrame("Revise a bit:\n Take a quiz!");

}

if(e.getSource()==b2)

{

option=2;

new ToDo("");

}

}

public static void main(String[] args) {

new FirstPage("NVision");

}

}

# Future Scope

## Basic modifications

* **Framework integration**: Swing shapes the core of this project. However, in today’s time, versatile frameworks and libraries have been successful in overcoming limitations of the traditional Swing. If possible, such frameworks may be incorporated in future either by integrating them with the existing module, or by replacing it.
* **Release:** Depending on the framework used, options for deploying the application on a broader platform could be explored.

## FEATURE ADDITIONS

* **Database extension (Quiz)**: At present, the questionnaire in the quiz session consists of a limited set of questions for prototype demonstration. However, in future, after deployment, a provision can be made for accepting scholarly contributions from all over the world and thus helping our users to enrich their repository of knowledge in subjects of their choice.
* **Flexibility (To Do List):** Instead of having a specific set number of tasks – as in the current version – the task list could expand dynamically along with the additions to it. The list could also have provisions of adding external files (images, documents, or hyperlinks) as tasks to it using the application.

# Bug Reports and Challenges

## Viewing experience

* *NVision* makes the best use of Swing to give its users a simple and aesthetic interface
* For this, the developers have ensured that every button that the application has, has been described aptly so as to avoid any kind of ambiguity for its users
* Appropriate messages are displayed at all times, thus keeping the user updated at all times with his quiz score, tasks completed, etc.

## avoiding errors

* **No errors**: The current version of *NVision* is totally bug-free.
* **Ant**: However, some instances of running the application on Eclipse may ask for Ant-build and may be unresponsive to those configurations as well as (new) run setups. In such a case, copying the application source contents to another folder created as a Java project in Eclipse, and re-launching Eclipse with that as the workspace is suggested as a workaround.

# Bibliography

resources referred

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | referred for | link |  |
| Stackoverflow Stackexchange | Resolution of errors | <https://stackoverflow.com/>  <https://stackexchange.com/> |  |
| java T point | Swing Basics | <https://www.javatpoint.com/java-swing> |  |
| Tutorials Point | <https://www.tutorialspoint.com/swing/> |  |

Java Mini Project, Semester 3, Year 2018-19

Subject in-charge: Professor Chitra Sahu,  
  
Veermata Jijabai Technological Institute,  
  
Matunga, Mumbai.

**NVision**

