|  |
| --- |
| import java.awt.\*; |
|  | import java.awt.event.\*; |
|  | import javax.swing.\*; |
|  | import javax.swing.event.\*; |
|  | import java.util.\*; |
|  | class Quiz extends JFrame implements ActionListener{ |
|  | JPanel panel; |
|  | JPanel panelresult; |
|  | JRadioButton choice1; |
|  | JRadioButton choice2; |
|  | JRadioButton choice3; |
|  | JRadioButton choice4; |
|  | ButtonGroup bg; |
|  | JLabel lblmess; |
|  | JButton btnext; |
|  | String qpa [][]; |
|  | String qca[][]; |
|  | int qaid; |
|  | HashMap<Integer, String> map; |
|  | Quiz() |
|  | { |
|  | initializedata(); |
|  | setTitle("Quiz Program"); |
|  | setDefaultCloseOperation(EXIT\_ON\_CLOSE); |
|  | setSize(430,350); |
|  | setLocation(300,100); |
|  | setResizable(true); |
|  | Container cont=getContentPane(); |
|  | cont.setLayout(null); |
|  | cont.setBackground(Color.GRAY); |
|  | bg=new ButtonGroup(); |
|  | choice1=new JRadioButton("Choice1",true); |
|  | choice2=new JRadioButton("Choice2",false); |
|  | choice3=new JRadioButton("Choice3",false); |
|  | choice4=new JRadioButton("Choice4",false); |
|  | bg.add(choice1); |
|  | bg.add(choice2); |
|  | bg.add(choice3); |
|  | bg.add(choice4); |
|  | lblmess=new JLabel("Choose a correct anwswer"); |
|  | lblmess.setForeground(Color.BLUE); |
|  | lblmess.setFont(new Font("Arial", Font.BOLD, 11)); |
|  | btnext=new JButton("Next"); |
|  | btnext.setForeground(Color.GREEN); |
|  | btnext.addActionListener(this); |
|  | panel=new JPanel(); |
|  | panel.setBackground(Color.LIGHT\_GRAY); |
|  | panel.setLocation(10,10); |
|  | panel.setSize(400,300); |
|  | panel.setLayout(new GridLayout(6,2)); |
|  | panel.add(lblmess); |
|  | panel.add(choice1); |
|  | panel.add(choice2); |
|  | panel.add(choice3); |
|  | panel.add(choice4); |
|  | panel.add(btnext); |
|  | cont.add(panel); |
|  | setVisible(true); |
|  | qaid=0; |
|  | readqa(qaid); |
|  | } |
|  | public void actionPerformed(ActionEvent e) |
|  | { |
|  | if(btnext.getText().equals("Next")) |
|  | { |
|  | if(qaid<9) |
|  | { |
|  | map.put(qaid,getSelection()); |
|  | qaid++; |
|  | readqa(qaid); |
|  | } |
|  | else |
|  | { |
|  | map.put(qaid,getSelection()); |
|  | btnext.setText("Show answers"); |
|  | } |
|  | } |
|  | else if(btnext.getText().equals("Show answers")) |
|  | new Report(); |
|  | } |
|  | public void initializedata(){ |
|  | //qpa stores pairs of question and its possible answers |
|  | qpa=new String[10][5]; |
|  | int r;  for(int i=0; i < 10 ; i++){  r= (int)(Math.random()\*100); |
|  | qpa[0][0]="How to run Java program on the command prompt?"; |
|  | if(r==1)  {  qpa[0][1]="javac JavaProgram"; |
|  | qpa[0][2]="java JavaProgram"; |
|  | qpa[0][3]="javac JavaProgram.java"; |
|  | qpa[0][4]="No one"; |
|  | }  if (r==2){ |
|  | qpa[1][0]="What is the use of the println method?"; |
|  | qpa[1][1]="It is used to print text on the screen."; |
|  | qpa[1][2]="It is used to print text on the screen with the line break."; |
|  | qpa[1][3]="It is used to read text from keyboard."; |
|  | qpa[1][4]="It is used to read text from a file."; |
|  | }  // so on |
|  | qpa[2][0]="How to read a character from the keyboard?"; |
|  | qpa[2][1]="char c=System.read()"; |
|  | qpa[2][2]="char c=System.in.read()"; |
|  | qpa[2][3]="char c=(char)System.read()"; |
|  | qpa[2][4]="char c=(char)System.in.read()"; |
|  |  |
|  | qpa[3][0]="Which one is a single-line comment?"; |
|  | qpa[3][1]="/..."; |
|  | qpa[3][2]="//..."; |
|  | qpa[3][3]="/\*..."; |
|  | qpa[3][4]="/\*...\*/"; |
|  |  |
|  | qpa[4][0]="How do you declare an integer variable x?"; |
|  | qpa[4][1]="int x"; |
|  | qpa[4][2]="x as Integer"; |
|  | qpa[4][3]="Int[] x"; |
|  | qpa[4][4]="No one is correct."; |
|  |  |
|  | qpa[5][0]="How do you convert a string of number to a number?"; |
|  | qpa[5][1]="int num=Integer.parseInt(str\_num)"; |
|  | qpa[5][2]="int num=str\_num.toInteger()"; |
|  | qpa[5][3]="int num=(int)str\_num"; |
|  | qpa[5][4]="int num=(Integer)str\_num"; |
|  |  |
|  | qpa[6][0]="What is the value of x? int x=3>>2"; |
|  | qpa[6][1]="1"; |
|  | qpa[6][2]="0"; |
|  | qpa[6][3]="3"; |
|  | qpa[6][4]="-3"; |
|  |  |
|  | qpa[7][0]="How to do exit a loop?"; |
|  | qpa[7][1]="Using exit"; |
|  | qpa[7][2]="Using break"; |
|  | qpa[7][3]="Using continue"; |
|  | qpa[7][4]="Using terminate"; |
|  |  |
|  | qpa[8][0]="What is the correct way to allocate one-dimensional array?"; |
|  | qpa[8][1]="int[size] arr=new int[]"; |
|  | qpa[8][2]="int arr[size]=new int[]"; |
|  | qpa[8][3]="int[size] arr=new int[size]"; |
|  | qpa[8][4]="int[] arr=new int[size]"; |
|  |  |
|  | qpa[9][0]="What is the correct way to allocate two-dimensional array?"; |
|  | qpa[9][1]="int[size][] arr=new int[][]"; |
|  | qpa[9][2]="int arr=new int[rows][cols]"; |
|  | qpa[9][3]="int arr[rows][]=new int[rows][cols]"; |
|  | qpa[9][4]="int[][] arr=new int[rows][cols]"; |
|  |  |
|  |  |
|  | //qca stores pairs of question and its correct answer |
|  | qca=new String[10][2]; |
|  |  |
|  | qca[0][0]="How to run Java program on the command prompt?"; |
|  | qca[0][1]="java JavaProgram"; |
|  |  |
|  | qca[1][0]="What is the use of the println method?"; |
|  | qca[1][1]="It is used to print text on the screen with the line break."; |
|  |  |
|  | qca[2][0]="How to read a character from the keyboard?"; |
|  | qca[2][1]="char c=(char)System.in.read()"; |
|  |  |
|  | qca[3][0]="Which one is a single-line comment?"; |
|  | qca[3][1]="//..."; |
|  |  |
|  |  |
|  | qca[4][0]="How do you declare an integer variable x?"; |
|  | qca[4][1]="int x"; |
|  |  |
|  | qca[5][0]="How do you convert a string of number to a number?"; |
|  | qca[5][1]="int num=Integer.parseInt(str\_num)"; |
|  |  |
|  | qca[6][0]="What is the value of x? int x=3>>2"; |
|  | qca[6][1]="0"; |
|  |  |
|  | qca[7][0]="How to do exit a loop?"; |
|  | qca[7][1]="Using break"; |
|  |  |
|  | qca[8][0]="What is the correct way to allocate one-dimensional array?"; |
|  | qca[8][1]="int[] arr=new int[size]"; |
|  |  |
|  | qca[9][0]="What is the correct way to allocate two-dimensional array?"; |
|  | qca[9][1]="int[][] arr=new int[rows][cols]"; |
|  |  |
|  | } |
|  | //create a map object to store pairs of question and selected answer |
|  | map=new HashMap<Integer, String>(); |
|  |  |
|  | } |
|  | public String getSelection(){ |
|  | String selectedChoice=null; |
|  | Enumeration<AbstractButton> buttons=bg.getElements(); |
|  | while(buttons.hasMoreElements()) |
|  | { |
|  | JRadioButton temp=(JRadioButton)buttons.nextElement(); |
|  | if(temp.isSelected()) |
|  | { |
|  | selectedChoice=temp.getText(); |
|  | } |
|  | } |
|  | return(selectedChoice); |
|  | } |
|  | public void readqa(int qid){ |
|  | lblmess.setText(" "+qpa[qid][0]); |
|  | choice1.setText(qpa[qid][1]); |
|  | choice2.setText(qpa[qid][2]); |
|  | choice3.setText(qpa[qid][3]); |
|  | choice4.setText(qpa[qid][4]); |
|  | choice1.setSelected(true); |
|  | } |
|  | public void reset(){ |
|  | qaid=0; |
|  | map.clear(); |
|  | readqa(qaid); |
|  | btnext.setText("Next"); |
|  | } |
|  | public int calCorrectAnswer(){ |
|  | int qnum=10; |
|  | int count=0; |
|  | for(int qid=0;qid<qnum;qid++) |
|  | if(qca[qid][1].equals(map.get(qid))) count++; |
|  | return count; |
|  | } |
|  | public class Report extends JFrame{ |
|  | Report(){ |
|  | setTitle("Answers"); |
|  | setSize(850,550); |
|  | setBackground(Color.WHITE); |
|  | addWindowListener(new WindowAdapter(){ |
|  | public void windowClosing(WindowEvent e){ |
|  | dispose(); |
|  | reset(); |
|  | } |
|  | }); |
|  | Draw d=new Draw(); |
|  | add(d); |
|  | setVisible(true); |
|  | } |
|  | class Draw extends Canvas{ |
|  | public void paint(Graphics g){ |
|  | int qnum=10; |
|  | int x=10; |
|  | int y=20; |
|  | for(int i=0;i<qnum;i++){ |
|  | //print the 1st column |
|  | g.setFont(new Font("Arial",Font.BOLD,12)); |
|  | g.drawString(i+1+". "+qca[i][0], x,y); |
|  | y+=30; |
|  | g.setFont(new Font("Arial",Font.PLAIN,12)); |
|  | g.drawString(" Correct answer:"+qca[i][1], x,y); |
|  | y+=30; |
|  | g.drawString(" Your answer:"+map.get(i), x,y); |
|  | y+=30; |
|  | //print the 2nd column |
|  | if(y>400) |
|  | {y=20; |
|  | x=450; |
|  | } |
|  |  |
|  | } |
|  | //Show number of correct answers |
|  | int numc=calCorrectAnswer(); |
|  | g.setColor(Color.BLUE); |
|  | g.setFont(new Font("Arial",Font.BOLD,14)); |
|  | g.drawString("Number of correct answers:"+numc,300,500); |
|  | } |
|  | } |
|  | } |
|  | } |
|  | public class QuizProgram{ |
|  | public static void main(String args[]){ |
|  | new Quiz(); |
|  |  |
|  | } |
|  | } |