package quiz.application;

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

public class Quiz extends JFrame implements ActionListener{

String questions[][] = new String[10][5];

String answers[][] = new String[10][2];

String useranswers[][] = new String[10][1];

JLabel Qno, question;

JRadioButton opt1, opt2, opt3, opt4;

ButtonGroup groupoptions;

JButton next, submit, lifeline;

public static int timer = 15;

public static int ans\_given =0;

public static int count = 0;

public static int score = 0;

String name;

Quiz(String name){

this.name = name;

setBounds(50, 0, 1440, 850);

getContentPane().setBackground(Color.WHITE);

setLayout(null);

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/quiz.jpg"));

JLabel image = new JLabel(i1);

image.setBounds(0, 0, 1440, 390);

add(image);

Qno = new JLabel();

Qno.setBounds(100, 450, 50, 30);

Qno.setFont(new Font("Tahoma", Font.PLAIN, 24));

add(Qno);

question = new JLabel();

question.setBounds(150, 450, 900, 30);

question.setFont(new Font("Tahoma", Font.PLAIN, 24));

add(question);

questions[0][0] = "Which is used to find and fix bugs in the Java programs.?";

questions[0][1] = "JVM";

questions[0][2] = "JDB";

questions[0][3] = "JDK";

questions[0][4] = "JRE";

questions[1][0] = "What is the return type of the hashCode() method in the Object class?";

questions[1][1] = "int";

questions[1][2] = "Object";

questions[1][3] = "long";

questions[1][4] = "void";

questions[2][0] = "Which package contains the Random class?";

questions[2][1] = "java.util package";

questions[2][2] = "java.lang package";

questions[2][3] = "java.awt package";

questions[2][4] = "java.io package";

questions[3][0] = "An interface with no fields or methods is known as?";

questions[3][1] = "Runnable Interface";

questions[3][2] = "Abstract Interface";

questions[3][3] = "Marker Interface";

questions[3][4] = "CharSequence Interface";

questions[4][0] = "In which memory a String is stored, when we create a string using new operator?";

questions[4][1] = "Stack";

questions[4][2] = "String memory";

questions[4][3] = "Random storage space";

questions[4][4] = "Heap memory";

questions[5][0] = "Which of the following is a marker interface?";

questions[5][1] = "Runnable interface";

questions[5][2] = "Remote interface";

questions[5][3] = "Readable interface";

questions[5][4] = "Result interface";

questions[6][0] = "Which keyword is used for accessing the features of a package?";

questions[6][1] = "import";

questions[6][2] = "package";

questions[6][3] = "extends";

questions[6][4] = "export";

questions[7][0] = "In java, jar stands for?";

questions[7][1] = "Java Archive Runner";

questions[7][2] = "Java Archive";

questions[7][3] = "Java Application Resource";

questions[7][4] = "Java Application Runner";

questions[8][0] = "Which of the following is a mutable class in java?";

questions[8][1] = "java.lang.StringBuilder";

questions[8][2] = "java.lang.Short";

questions[8][3] = "java.lang.Byte";

questions[8][4] = "java.lang.String";

questions[9][0] = "Which of the following option leads to the portability and security of Java?";

questions[9][1] = "Bytecode is executed by JVM";

questions[9][2] = "The applet makes the Java code secure and portable";

questions[9][3] = "Use of exception handling";

questions[9][4] = "Dynamic binding between objects";

answers[0][1] = "JDB";

answers[1][1] = "int";

answers[2][1] = "java.util package";

answers[3][1] = "Marker Interface";

answers[4][1] = "Heap memory";

answers[5][1] = "Remote interface";

answers[6][1] = "import";

answers[7][1] = "Java Archive";

answers[8][1] = "java.lang.StringBuilder";

answers[9][1] = "Bytecode is executed by JVM";

opt1 = new JRadioButton();

opt1.setBounds(170,520,700,30);

opt1.setBackground(Color.WHITE);

opt1.setFont(new Font("Dialog", Font.PLAIN, 20));

add(opt1);

opt2 = new JRadioButton();

opt2.setBounds(170,560,700,30);

opt2.setBackground(Color.WHITE);

opt2.setFont(new Font("Dialog", Font.PLAIN, 20));

add(opt2);

opt3 = new JRadioButton();

opt3.setBounds(170,600,700,30);

opt3.setBackground(Color.WHITE);

opt3.setFont(new Font("Dialog", Font.PLAIN, 20));

add(opt3);

opt4 = new JRadioButton();

opt4.setBounds(170,640,700,30);

opt4.setBackground(Color.WHITE);

opt4.setFont(new Font("Dialog", Font.PLAIN, 20));

add(opt4);

groupoptions = new ButtonGroup();

groupoptions.add(opt1);

groupoptions.add(opt2);

groupoptions.add(opt3);

groupoptions.add(opt4);

next = new JButton("Next");

next.setBounds(1100, 550, 200, 40);

next.setFont(new Font("Tahuma", Font.PLAIN, 22));

next.setBackground(new Color(30, 144, 255));

next.setForeground(Color.WHITE);

next.addActionListener(this);

add(next);

lifeline = new JButton("50 50 Lifeline");

lifeline.setBounds(1100, 630, 200, 40);

lifeline.setFont(new Font("Tahuma", Font.PLAIN, 22));

lifeline.setBackground(new Color(30, 144, 255));

lifeline.setForeground(Color.WHITE);

lifeline.addActionListener(this);

add(lifeline);

submit = new JButton("submit");

submit.setBounds(1100, 710, 200, 40);

submit.setFont(new Font("Tahuma", Font.PLAIN, 22));

submit.setBackground(new Color(30, 144, 255));

submit.setForeground(Color.WHITE);

submit.addActionListener(this);

submit.setEnabled(false);

add(submit);

start(count);

setVisible(true);

}

// public void actionPerformed(ActionEvent ae){

// }

@Override

public void actionPerformed(java.awt.event.ActionEvent ae) {

if(ae.getSource() == next){

repaint();

opt1.setEnabled(true);

opt2.setEnabled(true);

opt3.setEnabled(true);

opt4.setEnabled(true);

ans\_given = 1;

if(groupoptions.getSelection() == null){

useranswers[count][0] = "";

}else{

useranswers[count][0]=groupoptions.getSelection().getActionCommand();

}

if (count == 8){

next.setEnabled(false);

submit.setEnabled(true);

}

count++;

start(count);

}else if(ae.getSource() == lifeline){

if(count == 2 || count == 4 || count == 6 || count == 8 || count == 9 ){

opt2.setEnabled(false);

opt3.setEnabled(false);

}else{

opt1.setEnabled(false);

opt4.setEnabled(false);

}

lifeline.setEnabled(false);

}else if (ae.getSource() == submit){

ans\_given = 1;

if(groupoptions.getSelection() == null){

useranswers[count][0] = "";

}else{

useranswers[count][0]=groupoptions.getSelection().getActionCommand();

}

for(int i =0; i<useranswers.length;i++){

if(useranswers[i][0].equals(answers[i][1])){

score +=10;

}else{

score +=0;

}

}

setVisible(false);

new Score(name, score);

}

}

public void paint(Graphics g){

super.paint(g);

String time = "Time left: " + timer + " seconds"; // 15

g.setColor(Color.red);

g.setFont(new Font("Tahoma", Font.BOLD, 25));

if (timer > 0){

g.drawString(time, 1100, 500);

}else{

g.drawString("Times Up!", 1100, 500);

}

timer--;

try{

Thread.sleep(1000);

repaint();

}catch(Exception e){

e.printStackTrace();

}

if(ans\_given == 1){

ans\_given = 0;

timer = 15;

}else if(timer < 0){

timer = 15;

opt1.setEnabled(true);

opt2.setEnabled(true);

opt3.setEnabled(true);

opt4.setEnabled(true);

if (count == 8){

next.setEnabled(false);

submit.setEnabled(true);

}

if(count == 9){

if(groupoptions.getSelection() == null){

useranswers[count][0] = "";

}else{

useranswers[count][0]=groupoptions.getSelection().getActionCommand();

}

for(int i =0; i<useranswers.length;i++){

if(useranswers[i][0].equals(answers[i][1])){

score +=10;

}else{

score +=0;

}

}

setVisible(false);

new Score(name, score);

}else{

if(groupoptions.getSelection() == null){

useranswers[count][0] = "";

}else{

useranswers[count][0]=groupoptions.getSelection().getActionCommand();

}

count++;

start(count);

}

}}

public void start(int count){

Qno.setText("" + (count + 1) + ". ");

question.setText(questions[count][0]);

opt1.setText(questions[count][1]);

opt1.setActionCommand(questions[count][1]);

opt2.setText(questions[count][2]);

opt2.setActionCommand(questions[count][2]);

opt3.setText(questions[count][3]);

opt3.setActionCommand(questions[count][3]);

opt4.setText(questions[count][4]);

opt4.setActionCommand(questions[count][4]);

groupoptions.clearSelection();

}

public static void main(String[] args){

new Quiz("User");

}

}