

Object-Oriented Programming in Java

Quiz 2

Jan 24, 2001

Problem 1: Multiple Choice (Write letter corresponding to your choice in space provided.)

Fill in your answer in the space provided.

Question 1: In Java, Color, JButton, and Exception are all what?

- A. widgets B. frames
- C. events D. objects

Your Answer:

Question 2: Menus, buttons, and text fields are all what?

- A. handlers B. streams
- C. events D. widgets

Your Answer:

Question 3: Which of the following classes is not commonly used in File Processing?

- A. OutputStream B. StringTokenizer
- C. ActionAdapter D. PrintWriter

Your Answer:

Question 4: Which of the following classes must be extended to be used?

- A. AbstractAction B. InputStream
- C. Math D. StringTokenizer

Your Answer:

Question 5: Which of the following method types cannot return an error code?

- A. accessor B. private
- C. constructor D. synchronized

Your Answer:

Question 6: Which of the following keywords is NOT used in exception handling?

- A. try B. catch
- C. throw D. final

Your Answer:

Question 7: Which of the following would most likely be used in a program that download from the URL `ftp://104.200.34.0:8080/SameGame.jar`?

- A. `new Socket(8080, "104.200.34.0")`
- B. `new ServerSocket(8080)`
- C. `new BufferedReader("104.200.34.0", 8080)`
- D. `new Socket("104.200.34.0", 8080)`

Your Answer:

Question 8: In Program 1 (page 3 of quiz), the line `super.paintComponent()` does what?

- A. Calls the `paintComponent()` of `JPanel`
- B. Calls the `paintComponent()` of `MyFrame`
- C. Calls the `paintComponent()` of `contentpane`
- D. Draws the window nicer than usual

Your Answer:

Question 9: In Program 1, what color is the rectangle after the button is clicked?

- A. red
- B. blue
- C. green
- D. black

Your Answer:

Question 10: In Program 1, what color is the rectangle initially?

- A. red
- B. blue
- C. green
- D. black

Your Answer:

Question 11: In Program 1, which of following MUST be an inner class as implemented in the code below? (In other words, which if the following would not work if moved verbatim outside of `MyPanel`?)

- A. `ButtonHandler`
- B. `MouseHandler`
- C. Both A. and B.
- D. Neither A. nor B.

Your Answer:

Question 12: In Program 1, what is `Color.red`?

- A. a public static variable
- B. a public instance variable
- C. a public static method
- D. a public instance method

Your Answer:

Question 13: In Program 1, what is `paintComponent` (in `MyPanel`)?

- A. an inner class
- B. a public static method
- C. an event listener
- D. a public instance method

Your Answer:

Question 14: Can event-based programming be implemented without multi-threading?

- A. Yes
- B. No
- C.
- D.

Your Answer:

Program 1

```
class MyFrame extends JFrame{
    public MyFrame(){
        setTitle("WidgetTest");
        setSize(300,300);        // size in pixels
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class MyPanel extends JPanel{
    Color current = Color.red;
    JButton button = new JButton("Button");

    class ButtonHandler implements ActionListener{
        public void actionPerformed(ActionEvent e){
            current = Color.green;
            System.out.println("Mike");
            repaint();
        }
    }

    class MouseHandler extends MouseAdapter{
        public void mouseClicked(MouseEvent ev){
            if(ev.getX() <= 100) System.out.println("Rusty");
            if(ev.getY() >= 100) current = Color.blue;
            System.out.println("Dimitri");
            repaint();
        }
    }

    MyPanel(){
        button.addActionListener(new ButtonHandler());
        addMouseListener(new MouseHandler());
        add(button);
        System.out.println("Alan");
    }

    public void paintComponent(Graphics g){
        super.paintComponent(g);
        Graphics2D g2 = (Graphics2D)g;
        g.setColor(current);
        g2.fill(new Rectangle2D.Float(0,100,100,100));
    }
}

public class WidgetTest{
    public static void main(String[] args){
        MyFrame myframe = new MyFrame();
        Container contentpane = myframe.getContentPane();
        contentpane.add(new MyPanel());
        myframe.show();
    }
}
```

Program 2

```
import java.io.*;

class MyException extends Exception{
    int value;
    MyException(int val){
        value = val;
    }

    public int getValue(){
        return(value);
    }
}

class TestClass2{

    public static void foo(int x) throws MyException,IOException{
        if(x < 0){
            System.out.println("X too low");
            throw(new MyException(x));
        }
        if(x > 100){
            System.out.println("X too high");
            throw(new IOException("Mike"));
        }
        System.out.println("X OK");
    }

    // Test main, convert argument to int can call test.
    public static void main(String[] args){
        int x = Integer.parseInt(args[0]); // convert first argument to int
        //
        try{
            foo(x);
            System.out.println("Mike");
        }
        catch(IOException e){
            System.out.println("Rusty");
        }
        catch(MyException e){
            System.out.println("Dimitri");
            System.out.println(e.getValue());
        }
    }
}
```

Problem 2: Program Analysis: Short Answer

Question 1: What does Program 1 print when it starts up?

Answer:

Question 2: What does Program 1 print when the mouse is clicked inside the filled rectangle?

Answer:

Question 3: What color is the filled rectangle drawn by Program 1 after the mouse is clicked inside it?

Answer:

Question 4: If the line `button.addActionListener(new ButtonHandler());` is deleted from Program 1 (and the program is recompiled) what will it print when the button is clicked?

Answer:

Question 5: What does Program 2 (see previous page) print out when called as `java TestClass2 200?`

Answer:

Question 6: What does Program 2 print out when called as `java TestClass2 200?`

Answer:

Question 7: What does Program 2 print out when called as `java TestClass2 -200?`

Answer:

The next questions refer to the Account class in an imaginary banking application shown below. When answering the questions, consider the following scenario. There are two Account instances, `savings` and `checking`. There are several threads sharing these instance, some threads transferring from `savings` to `checking`, some transferring from `checking` to `savings`.

```
class Account{
    int amount = 1000; // always start with 1000

    public void deposit(int x){
        amount = amount + x;
    }
    public void withdraw(int x){
        amount = amount - x;
    }

    public int balance(){
        return(amount);
    }
    public void transfer(int x, Account dest){
        withdraw(x); // withdraw x from this account
        dest.deposit(x); // deposit in transfer account
    }
}
```

Question 8: Is the class implementation as written thread-safe? Meaning, can any number of threads call any sequence of methods on shared instances and produce a deterministic predictable result regardless of scheduling policy.

Answer:

Question 9: If the `deposit()` and `withdraw` methods are `synchronized`, is the class implementation thread-safe?

Answer:

Question 10: If the `transfer()` method **ONLY** is `synchronized`, is the class implementation thread-safe?

Answer:

Question 11: If **ALL** methods in the class are `synchronized` will this class work properly? If not, why not?

Answer: