



CIFP Pau Casesnoves

Joan Miró, 22 07300 Inca

Tel. 971 881710 11

secretaria@paucasesnovescifp.cat

PROVA D' AVALUACIÓ

Curs	----	Grup	IFC32D	Data	--/--	Qualificació	
Mòdul	DESARROLLO DE INTERFACES						
Alumne/a							

		Count	Puntos
Preguntas multiple choice...			
- Acertadas:			
- Falladas:			
- No contestadas:			
Preguntas a desarrollar:			

Instrucciones para el Examen Parcial de SPDVI

Funcionamiento de la prueba:

- La prueba consiste en desarrollar por escrito y en papel lo que se pide en el enunciado.
- Todas las preguntas valen 1 punto excepto la pregunta x que vale a puntos y la pregunta y que vale b puntos. Por lo tanto el examen se calificará sobre n puntos por lo que hay que obtener un mínimo de p puntos para aprobar.
- La calificación final de la prueba se escalará para expresar la nota con un valor numérico entre 0 y 10, con dos decimales.
- Las preguntas de selección múltiple mal contestadas restan 0.25 puntos.
- En las preguntas de selección múltiple sólo hay una respuesta válida.
- La duración de la prueba es de 2 horas.
- Se deben entregar todas las hojas utilizadas.
- Se tienen que firmar todas las hojas del examen por delante, junto al logo de Pau Casesnoves.

Recursos que pueden llevar y consultar durante la prueba:

- No se puede consultar ningún recurso ni material.

Traer:

- DNI
- Bolígrafo



CIFP Pau Casesnoves

Joan Miró, 22 07300 Inca

Tel. 971 881710 11

secretaria@paucasesnovescifp.cat



CIFP Pau Casesnoves

Joan Miró, 22 07300 Inca

Tel. 971 881710 11

secretaria@paucasesnovescifp.cat

1. En una aplicació con interfaz gráfica de usuario para gestionar las reservas de viviendas de turismo vacacional, los establecimientos pueden tener servicios como Wifi, Piscina, Parking, Aire Acondicionado, etc. ¿Qué controles swing son los más adecuados para incluir en un formulario para crear un nuevo establecimiento y representar si un alojamiento incluye esos servicios o no?

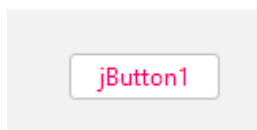
- a) JCheckBox
- b) JButton
- c) JRadioButton
- d) JTextField

2. Las imágenes estáticas que aparecen en la interfaz de una aplicación como iconos y logos, ¿en qué carpeta del proyecto maven es conveniente incluirlas para evitar problemas en NetBeans al ejecutar el proyecto?

- a) src/main/java/resources
- b) src/main/resources
- c) src/images
- d) Ninguna de las opciones anteriores es correcta

3. ¿Qué propiedad hay que modificar para cambiar el color del texto que se muestra en los JButton ?

- a) foreground
- b) text
- c) font
- d) color



4. ¿Qué componente swing podemos utilizar en una interfaz gráfica como lista desplegable?

- a) JComboBox
- b) JTextField
- c) JList
- d) JScrollPane

5. La propiedad 'enabled' común a la mayoría de controles de la biblioteca swing:

- a) Permite mostrar u ocultar el control en la GUI.
- b) Es de tipo String.
- c) Hará que el control no pueda recibir el foco de la GUI en caso de estar deshabilitada.
- d) Ninguna de las anteriores es correcta.



6. Si hemos declarado e inicializado un diálogo con la línea '***JDialog myDialog = new JDialog(this, true);***', ¿qué orden deberemos ejecutar para mostrar el diálogo en modo modal?

- a) myDialog.showDialog(true);
- b) myDialog.setVisible(true);
- c) myDialog.showModal("OK", "CANCEL");
- d) Ninguna de las anteriores es correcta.

7. Una consistencia adecuada de la interfaz gráfica de usuario contribuye a:

- a) Recuperación de información, en caso de error.
- b) Fácil mantenimiento.
- c) Rápido aprendizaje.
- d) Economizar en el desarrollo de la aplicación.

8. Se desea añadir a un componente que hereda de JPanel la capacidad de detectar cuando el ratón se ha movido dentro del área del componente. ¿Qué tipo de listener se debe añadir como argumento de la instrucción this.addMouseListener()?

- a) MouseAdapter
- b) MouseListener
- c) ActionListener
- d) mouseMoved

9. ¿Qué condiciones debe cumplir una clase para considerarse un JavaBean?

- a) Implementar la interfaz Serializable y tener un constructor sin parámetros.
- b) Las del apartado a y además proporcionar de un editor de propiedades.
- c) Las del apartado b y además el BeanInfo correspondiente.
- d) Ninguna de las anteriores es correcta.

10. ¿Cuál es la salida de la siguiente aplicación?

```
public class MyActionListener implements ActionListener {  
  
    @Override  
    public void actionPerformed(ActionEvent evt) {  
        System.out.println("zxc");  
    }  
}
```



CIFP Pau Casesnoves

Joan Miró, 22 07300 Inca

Tel. 971 881710 11

secretaria@paucasesnovescifp.cat

```
public class Main extends javax.swing.JFrame {

    public Main() {
        initComponents();
        setSize(400, 300);
        ActionListener myX = new MyActionListener();
        Timer t = new Timer(400, myX);
        t.start();
    }

    @SuppressWarnings("unchecked")
    Generated Code

    /**...3 lines */
    public static void main(String args[]) {...31 lines }

    // Variables declaration - do not modify
    // End of variables declaration
}
```

- a) Muestra por la consola estándar el mensaje "zxc" cada 0.4 segundos.
- b) No muestra ningún mensaje en la consola.
- c) La aplicación no llega a compilar.
- d) La aplicación lanza una 'Java Runtime Listener Exception' cuando ejecuta la línea '*Timer t = new Timer(400, myX);*'.

11. Se ha creado un JavaBean 'ImageShowerJB' y se ha añadido a la Palette de controles swing del Designer de NetBeans. El código de control se muestra a continuación. Indica cuál de las siguientes opciones es la correcta.



```
public class ImageShowerJB extends JPanel implements Serializable {
    ArrayList<File> images = new ArrayList<>();
    int showingImageIndex = 0;

    public ImageShowerJB() {
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        File imageToShow = images.get(index:showingImageIndex);
        if (imageToShow != null && imageToShow.exists()) {
            try {
                BufferedImage image = ImageIO.read(input:imageToShow);
                g.drawImage(img:image, x:0, y:0,
                    width:this.getWidth(), height:this.getHeight(), observer:null);
            } catch (IOException ex) {
                Logger.getLogger(name:ImageShowerJB.class.getName())
                    .log(level:Level.SEVERE, msg:null, thrown:ex);
            }
        }
    }
}
```

- a) Las propiedades 'images' y 'showingImageIndex' no se mostrarán en el panel de Properties del Designer de NetBeans.
- b) El componente debería haber implementado la interfaz BeanInfo para poderse haber añadido a la 'Palette' del Designer de NetBeans.
- c) Al hacer click en el botón '...' del panel de Properties del componente  aparecerá un JFileChooser que nos permitirá seleccionar los ficheros de imagen a mostrar.
- d) El componente emitirá un evento 'OnPaintComponent' cada vez que sea redimensionado.

12. Se desea programar un listener para dotar a un componente de la posibilidad de emitir un evento el cual transfiera información a sus listeners (por ejemplo los datos de un fichero de imagen) a través del objeto **event**. ¿De qué clase debe heredar **ETVImageChangedEvent**?

```
public interface ETVImageChangedListener extends EventListener {
    void etvImageChanged(ETVImageChangedEvent event);
}
```

- a) ActionListener
- b) EventObject
- c) EventListener
- d) PropertyChangeSupport



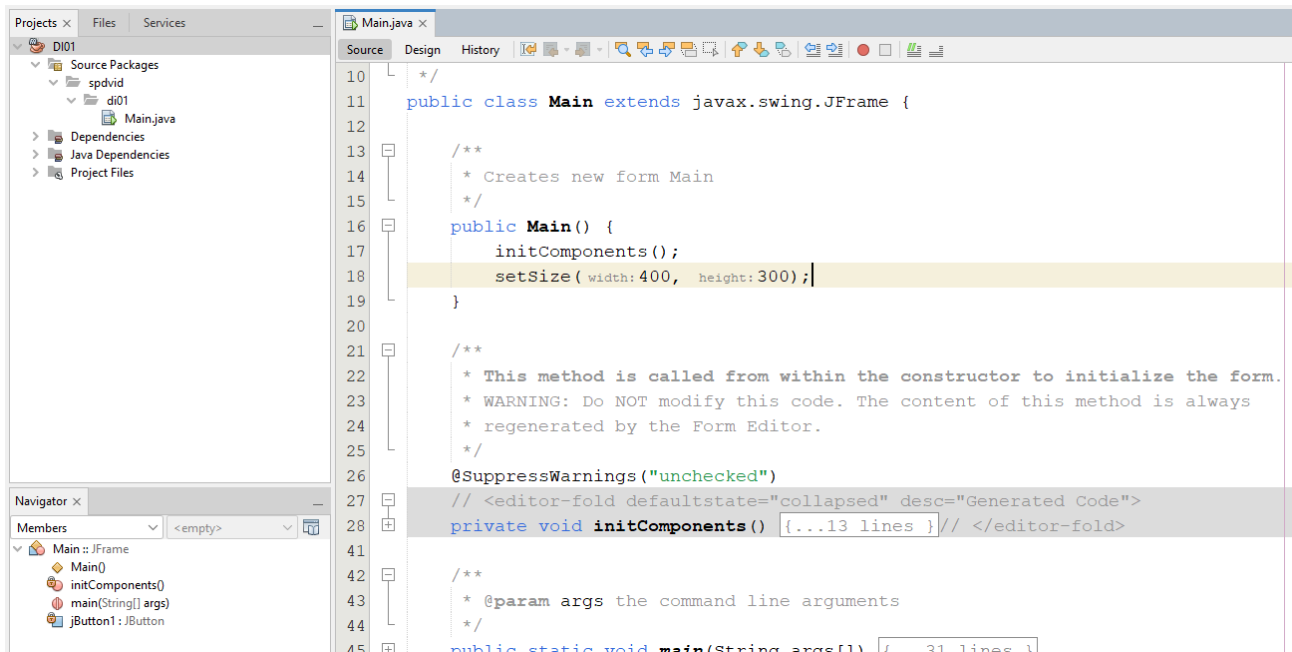
CIFP Pau Casesnoves

Joan Miró, 22 07300 Inca

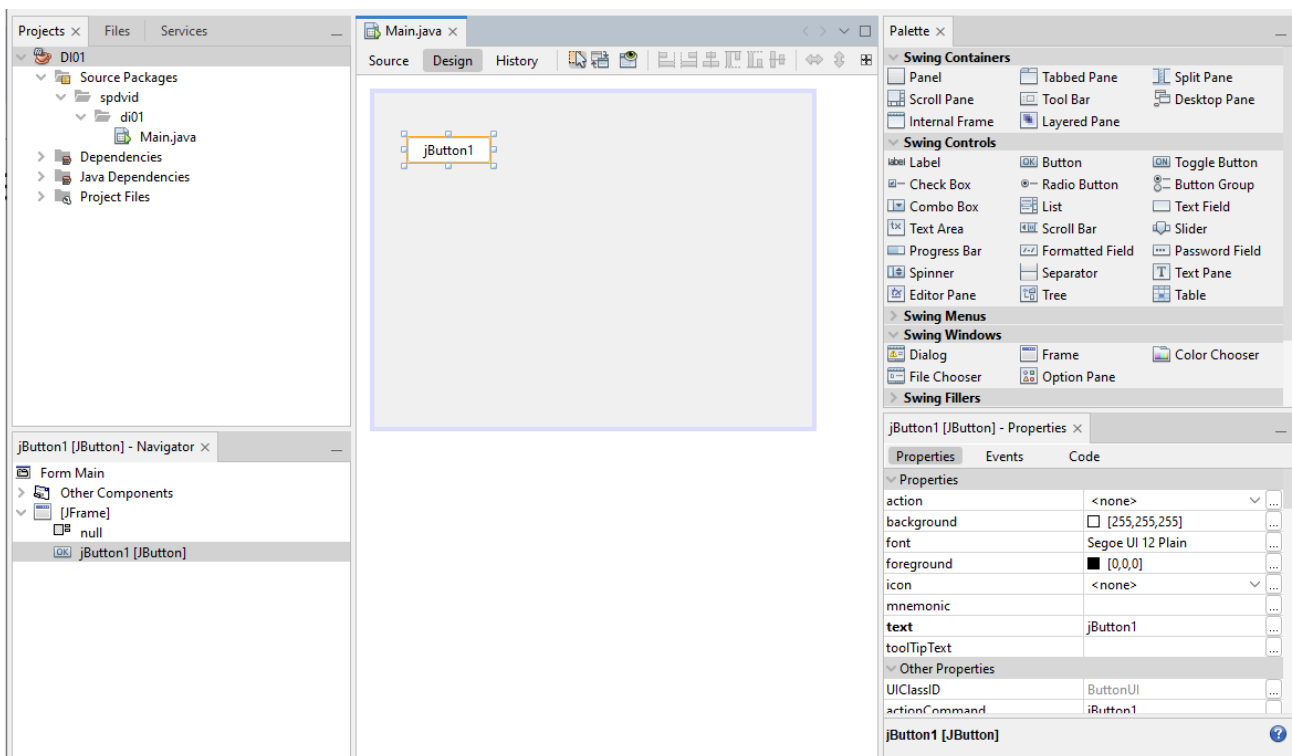
Tel. 971 881710 11

secretaria@paucasesnovescifp.cat

13. Se está trabajando en NetBeans con un proyecto que contiene el JFrame 'Main' y cuyo layout se ha establecido a null. Si se arrastra un JButton desde la Palette de controles swing a la posición (10,30) aproximadamente del JFrame y el tamaño por defecto asignado a los JButtons es de 75 x 23, escribe el código añadido por NetBeans al método initComponents del fichero Main.java. ¿Escribe NetBeans alguna línea de código más en la clase Main?



```
10  */
11  public class Main extends javax.swing.JFrame {
12
13      /**
14       * Creates new form Main
15       */
16      public Main() {
17          initComponents();
18          setSize( width: 400, height: 300);
19      }
20
21      /**
22       * This method is called from within the constructor to initialize the form.
23       * WARNING: Do NOT modify this code. The content of this method is always
24       * regenerated by the Form Editor.
25       */
26      @SuppressWarnings("unchecked")
27      // <editor-fold defaultstate="collapsed" desc="Generated Code">
28      private void initComponents() { ...13 lines } // </editor-fold>
29
30      /**
31       * @param args the command line arguments
32       */
33      public static void main(String args[]) { ...31 lines }
```



The Design view shows a JFrame with a JButton placed on it. The Palette on the right lists various Swing components under categories like Swing Containers, Swing Controls, Swing Menus, Swing Windows, and Swing Filters. The Properties window at the bottom right shows the properties of the selected JButton, including action, background, font, foreground, icon, mnemonic, text, toolTipText, and other properties.

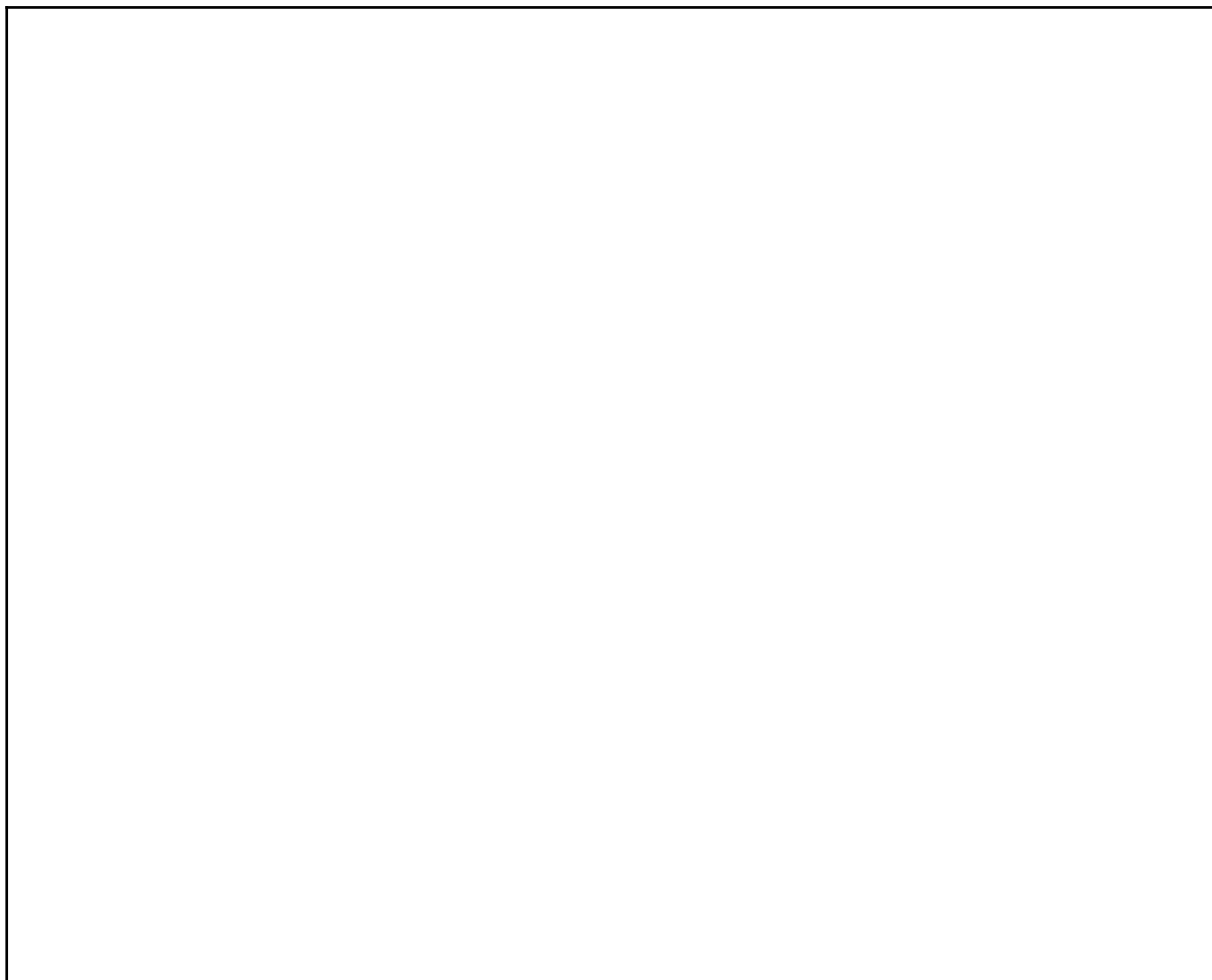


CIFP Pau Casesnoves

Joan Miró, 22 07300 Inca

Tel. 971 881710 11

secretaria@paucasesnovescifp.cat





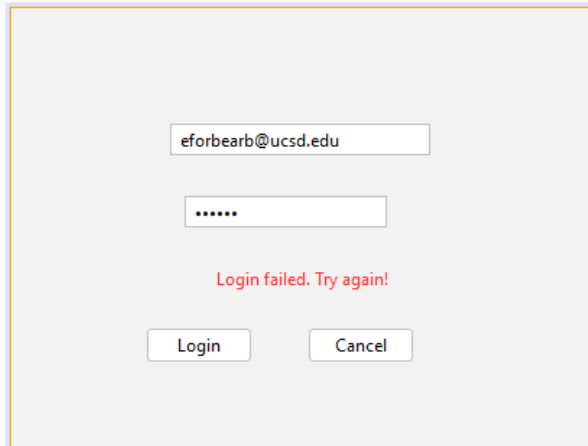
CIFP Pau Casesnoves

Joan Miró, 22 07300 Inca

Tel. 971 881710 11

secretaria@paucasesnovescifp.cat

14. Nuestro equipo de desarrollo nos pide crear el diálogo de la figura pero nos encontramos sin ordenador y no disponemos de NetBeans por lo que decidimos escribirlo directamente en un fichero de texto en nuestro smartphone. Escribe el código necesario para crear la interfaz del dialog que se muestra.



Controles:

- ventana del diálogo modal
 - tamaño = 400x300, centrado en su ventana contenedora.
- txtEmail
 - posición = 110, 80 tamaño = 180 x 22
- txtPassword
 - posición = 120, 130 tamaño = 140 x 22
- lblError
 - posición = 140, 180 tamaño = 120 x 16
 - color de texto = red
 - inicialmente oculto
- btnLogin
 - posición = 100, 220 tamaño = 72 x 23
 - Al hacer click sobre él mostrar por consola standard el mensaje "Trying to log in...".
- btnLogout
 - posición = 200, 220 tamaño = 72 x 23
 - Al hacer click sobre él se deberá cerrar el dialog.



CIFP Pau Casesnoves

Joan Miró, 22 07300 Inca

Tel. 971 881710 11

secretaria@paucasesnovescifp.cat



CIFP Pau Casesnoves

Joan Miró, 22 07300 Inca

Tel. 971 881710 11

secretaria@paucasesnovescifp.cat

15. Explica en qué consiste el concepto de “consistencia” respecto de la usabilidad de las interfaces gráficas.



CIFP Pau Casesnoves

Joan Miró, 22 07300 Inca

Tel. 971 881710 11

secretaria@paucasesnovescifp.cat

16. Da dos ejemplos de “realimentación” (feedback) en el contexto del diseño de interfaces usables.