

*TEST: Respuesta correcta +3 puntos, incorrecta -1 punto, en blanco +0*

*Sólo hay una respuesta correcta*

*Es obligatorio presentar documento de identificación*

**Apellidos y Nombre:**

**D.N.I :**

1. Considerando las restricciones impuestas por el sistema visual del ser humano es aconsejable:
  - (a) No utilizar códigos redundantes (formas, por ejemplo) para evitar la sobrecarga de información.
  - (b) No superar el tamaño de letra de 12 puntos para evitar sobrecarga visual
  - (c) Utilizar siempre las mayúsculas en todos los mensajes para poder leer los textos con mayor claridad
  - (d) Más de una respuesta de las anteriores es cierta
  - (e) Ninguna de las respuestas anteriores es cierta
2. Acerca de los menús:
  - (a) Contextuales o pop-up no llevan nemónicos
  - (b) Contextuales o pop-up tienen que poseer opciones que estén además disponibles en algún otro lugar visible de la pantalla
  - (c) Contribuyen a aliviar las limitaciones de la memoria a corto plazo
  - (d) Si todos los ítems de un menú no están actualmente disponibles hay que establecer el título de menú como no disponible
  - (e) Más de una respuesta de las anteriores es cierta

```
1. URL hsURL = HelpSet.findHelpSet(null, "Ayuda/ayuda.hs");  
2. Helpset hs = new HelpSet(null,hsURL);  
3. HelpBroker hb = hs.createHelpBroker();
```

3. Dado el código anterior, indica cuál es la línea a incluir si se desea que al pulsar "F1" se habilite la ayuda, siendo "boton1" el botón para la obtención de la ayuda en la interfaz e "introduccion" el alias del archivo html que deseamos cargar cuando se muestre la ayuda:
  - (a) hb.enableHelpKey(boton1,"introduccion",hs);
  - (b) hb.enableHelpOnButton(boton1,"F1", "introduccion",hs);
  - (c) hb.enableHelpKey(getRootPane(),"introduccion",hs);
  - (d) hb.enableHelpOnButton(getRootPane(),"introduccion",hs);
  - (e) hb.enableHelpKey(getRootPane(),"F1", "introduccion",hs);
4. Respecto a los tooltips, es cierto que:
  - (a) Si la aplicación los proporciona es recomendable que puedan ser desactivados por el usuario
  - (b) Son recomendables en los botones por defecto y de cancelación que tengan texto y gráficos
  - (c) Son recomendables en los botones de una barra de herramientas que tengan texto
  - (d) Más de una respuesta es cierta
  - (e) Ninguna de las respuestas anteriores es cierta
5. Acerca de los nemónicos es cierto que:
  - (a) Facilitan la recuperación de la información almacenada en la memoria a largo plazo
  - (b) Facilitan la recuperación de la información almacenada en la memoria a corto plazo
  - (c) Facilitan el acceso por teclado
  - (d) Se establecen eligiendo la primera letra libre del texto asociado al componente.
  - (e) No es recomendable utilizar nemónicos debido a la sobrecarga de memoria que éstos producen
6. Un grupo de cajas de chequeo (CheckBoxes):
  - (a) Pueden ser sustituidas por un JList de selección simple
  - (b) Pueden ser sustituidas por un JComboBox editable de selección múltiple
  - (c) Pueden ser sustituidas por un JList de selección múltiple
  - (d) Pueden ser sustituidas por un JComboBox no editable de selección múltiple
  - (e) Pueden ser sustituidas por un conjunto de radio botones incorporados a un JButtonGroup

```

1. import java.util.*;
2. class Ejemplo{
3.     public static void main ( String[] args){
4.         Locale localizacion = Locale.getDefault();
5.         ResourceBundle datos = ResourceBundle.getBundle("Datos",localizacion);
6.         System.out.println(datos.getString("dato1"));
7.         System.out.println(datos.getString("dato2"));
8.     }
9. }

```

7. Supongamos que en el ejemplo anterior se quiere que los textos se muestren en español (España-ES), inglés (Estados Unidos-US) y francés (Francia-FR). Tendríamos que:

- (a) Crear un fichero Datos\_ES\_US\_FR.properties
- (b) Crear tres ficheros: Datos\_es\_ES.properties, Datos\_en\_US.properties y Datos\_fr\_FR.properties
- (c) Crear un fichero Datos\_es\_en\_fr.properties
- (d) Crear un fichero para la localización obtenida de la máquina en la que se ejecute la aplicación (Datos\_locale.properties) y otro para los demás idiomas (Datos\_noLocale.properties)
- (e) Ninguna de las respuestas anteriores es cierta

8. En los diálogos:

- (a) El botón por defecto y el de cancelación no llevan nemónico
- (b) No es obligatorio incorporar el botón por defecto ni el de cancelación
- (c) El botón de Ayuda, si lo hay, se sitúa antes del de Aceptar
- (d) Más de una respuesta es cierta
- (e) Ninguna de las respuestas anteriores es cierta

9. A incrementar la usabilidad de una aplicación contribuye:

- (a) La capacidad del usuario de recordar lo que había en pantallas anteriores (memoria a corto plazo) cuando utiliza una aplicación
- (b) El hecho de que la notificación de cambios producidos en el sistema sea eventual
- (c) Innovar con asiduidad en las técnicas de interacción para incrementar a capacidad de atención del usuario
- (d) Evitar la utilización de metáforas del mundo real
- (e) Ninguna de las anteriores es cierta

10. Partimos de una ayuda ya construida con JavaHelp y en la que hay tabla de contenidos e índice (no hay vista para la búsqueda). Ahora es necesario añadir un nuevo archivo html a la ayuda (que es el que se visualizará por defecto al desplegar la ayuda). ¿Qué es necesario hacer para que dicho archivo se vea

reflejado en las dos vistas  ?

- (a) Modificar el fichero ayuda.hs y ejecutar el jhindexer
- (b) Modificar el fichero toc, el fichero índice y ejecutar el hviewer
- (c) Modificar el fichero map, el fichero toc y el fichero índice
- (d) La vista de la búsqueda se proporciona por defecto y no es posible eliminarla en javahelp 2.0
- (e) Ninguna de las respuestas anteriores es cierta

11. Respecto a los layouts es cierto que:

- (a) Por defecto, *JFrame* y *JDialog* incorporan *FlowLayout* y *JPanel*, *BorderLayout*.
- (b) Un *JPanel* que incorpore un *GridBagLayout* se divide en un número de celdas de idéntico tamaño.
- (c) Un ejemplo de uso del *FlowLayout* es el contenedor *JToolBar*
- (d) Si un contenedor incorpora un *BorderLayout*, los componentes situados en dicho contenedor pueden modificar su posición en el contenedor mediante la propiedad *constraint*
- (e) Más de una de las respuestas anteriores es cierta

12. En JavaHelp el fichero helpset (ayuda.hs):

- (a) Incorpora los términos indexados para que se pueda proporcionar búsqueda en el sistema de ayuda
- (b) Es el fichero que es referenciado explícitamente por la aplicación
- (c) Describe el contenido y la distribución de la tabla de contenidos
- (d) Asocia identificadores o alias a los ficheros HTML
- (e) Más de una de las respuestas anteriores es cierta

13. Respecto a los componentes para texto, es cierto que:

- (a) Para asociar un nemónico a un *JTextField* debe modificarse la propiedad *displayedMnemonic* del mismo
- (b) Una etiqueta ha de estar habilitada (*enabled* a *true*) cuando el componente con el que está relacionada esté deshabilitado (*enabled* a *false*)
- (c) Para sustituir el texto actual de un área de texto el método más apropiado es el *append*
- (d) Un *PasswordField* proporciona todas las capacidades de edición de un *TextField* incluidas las operaciones de cortar y copiar.
- (e) Ninguna respuesta es cierta

```
1. public class VentanaPrincipal extends JFrame{
2.     private JTextArea areal = null;
3.     private ProcesaTecla pT = null ;
4.     ...
5.     class ProcesaTecla implements KeyAdapter {
6.         public void keyTyped(KeyEvent e){
7.             pintar(e);
8.         }
9.     } // fin clase ProcesaTecla
10.    private void initialize(){
11.        pT = new ProcesaTecla();
12.        ...
13.    } // Fin del initialize
14.    private void getAreal{
15.        ...
16.        areal.addKeyListener(pT);
17.    ...}
```

14. Dado el código anterior y considerando que los métodos del interface *KeyListener* son más de dos:

- (a) *pT* representa el objeto fuente del evento y *areal* representa el objeto receptor del evento
- (b) La instrucción de la línea 16 es incorrecta ya que debería ser *areal.addKeyAdapter(pT)*;
- (c) La instrucción de la línea 5 es incorrecta ya que se debería utilizar *extends* en lugar de *implements*
- (d) La clase *ProcesaTecla* es incorrecta ya que se deberían haber implementado todos los métodos del interface *MouseAdapter*
- (e) Más de una respuesta es correcta

```
public void pintar (KeyEvent e){
1.    JTextArea area = e.getSource();
2.    area.setBackground(Color.red);
}
```

15. Supongamos que este es el código del método *pintar* de la pregunta anterior. Indica cual es la afirmación correcta:

- (a) Es un método correcto y se cambiará a color rojo el fondo del objeto fuente del evento
- (b) Es un método incorrecto porque faltan parámetros en el método *getSource* de la línea 1
- (c) Es un método incorrecto porque es necesario añadirle un parámetro que indique el area a pintar
- (d) Es un método incorrecto ya que en la línea 1 es necesario hacer *casting* a la clase *JTextArea*
- (e) Es un método correcto y se cambiará a color rojo el fondo del objeto receptor del evento

16. La clase receptora encargada de gestionar un *FocusEvent*, siendo dos los métodos del interface listener:

- (a) Puede ser construida implementando el interface *FocusListener*
- (b) Puede ser construida heredando de la clase adaptadora *FocusAdapter*
- (c) En este caso, al ser dos los métodos del interface listener, no puede ser construida heredando de una clase adaptadora
- (d) Puede ser construida a partir del interface *FocusAdapter*
- (e) La (a) y la (b) son ciertas

```

package cine;
...
class Aplicacion{
    public static void main(String[] args){
        ...
        Pelicula sesion1 = new Pelicula();
        sesion1.setPelicula("El Padrino");
        ...
    }
}

```

17 . Dado el código anterior y teniendo en cuenta que:

- El código correspondiente a la clase *Pelicula* se encuentra en el fichero *peliculas.jar* situado en el directorio *C:\utilidades*.
- La aplicación (*Aplicacion.class*) está en el directorio *C:\Archivos de programa\aplicaciones\cine*.

Indica cual es el valor correcto para la variable de entorno Classpath.

- (a) Classpath=c:\Archivos de programa\aplicaciones;c:\utilidades\peliculas.jar
- (b) Classpath=c:\Archivos de programa\aplicaciones\cine;c:\utilidades
- (c) Classpath=c:\Archivos de programa\aplicaciones\cine; c:\utilidades\peliculas.jar
- (d) Classpath=c:\Archivos de programa\aplicaciones\cine
- (e) Classpath=c:\Archivos de programa\aplicaciones;c:\utilidades

```

1. public class VentanaPrincipal extends JFrame {
2.     JLabel etiqueta = null;
3.     JComboBox combo = null;
4.     ...
5.     //Método get del Combo
6.     String[] bebidas =
7.         { "Agua", "Café", "Horchata", "Leche", "Limonada", "Refresco", "Té", "Zumo" };
8.     ...
9.     combo = new JComboBox(bebidas);
10.    //Método get del Panel principal
11.    etiqueta = new JLabel();
12.    etiqueta.setText("Elija bebida: ");
13.    etiqueta.setMnemonic('E');
14.    etiqueta.setLabelFor(combo);
15.    ...
16. }

```

18. Dado el fragmento de código anterior se puede afirmar que:

- (a) Considerando el código de la línea 6, no cumple todas las recomendaciones aplicables a los *JComboBox*
- (b) Dado el número de opciones a elegir, sería más conveniente utilizar un grupo de radio botones para realizar la selección.
- (c) Considerando el código de las líneas 11 y 12, no permite el acceso por teclado al *JComboBox*
- (d) Dado el número de opciones a elegir, sería más conveniente utilizar un grupo de check boxes para realizar la selección.
- (e) Ninguna de las respuestas anteriores es correcta

19. Indica cuál de las siguientes afirmaciones acerca del proceso de instalación es cierta:

- (a) Uno de los directorios que es necesario incorporar en el instalador es el *src*
- (b) Siempre se debe generar el instalador incorporando la máquina virtual de java
- (c) El hecho de modificar el *classpath* garantiza que la carga de ficheros externos tales como imágenes o ficheros de datos se realice de manera correcta una vez instalada la aplicación en el ordenador del cliente.
- (d) Más de una de las respuestas anteriores es cierta
- (e) Ninguna de las respuestas anteriores es cierta

20. Acerca de los diálogos es cierto que:

- (a) Pueden ser modales y no modales
- (b) Pueden tener botón por defecto
- (c) *JOptionPane*, *JFileChooser* y *JColorChooser* son diálogos estándar no modales
- (d) Son ciertas la (a) y la (b)
- (e) Son ciertas la (a), (b) y (c)