

Responde:

RESPONDE

1. Imagina una herramienta de desarrollo, están compuestas de un conjunto de ventanas que podemos mostrar u ocultar, cambiar su tamaño, etc., de modo que al cerrar la aplicación y volverla a abrir se mantiene la estructura que tenía al cerrarse. ¿Con qué característica de los componentes relacionas esto?

- > a) Con la persistencia.
- b) Con la introspección.
- c) Con la gestión de eventos.

La persistencia de un programa o aplicación es la característica que se relaciona con almacenar la forma de

RESPONDE

2. Indica cuál de las siguientes no es una convención requerida para un JavaBeans:

- a. Debe tener un constructor sin argumentos.
- > b. La clase Bean debe tener un método toString.
- c. Sus atributos de clase deben ser privados.
- d. Debe ser serializable.

No tiene por qué tener un método toString si no se considera necesario.

RESPONDE

3. Una vez desarrollado el componente para que las herramientas lo reconozcan y se puedan usar en las aplicaciones y en ellas programar la realización de acciones...

- > a. ...debe incluir una clase de tipo Property con sus métodos setter y getter.
- b. ...debe incluir una clase tipo ObjectProperty que envuelva al EventHandler.
- c. ...no será necesario procesar en él el evento
- d. ...se deberá procesar el ActionEvent, el único tipo de evento válido.

Para cada valor del objeto que se quiera poder modificar/comparar/observar hará falta una clase de tipo Property que lo envuelva, con su correspondientes métodos get y set.

RESPONDE

4. La introspección permite...

- > a. ...a una herramienta de desarrollo detectar los nombres de las propiedades y métodos de la interfaz de una clase.
- b. ...facilitar la implementación de los métodos de una clase.
- c. ...conocer los elementos de la interfaz de un componente mediante el uso de una clase llamada BeanInfo.

Las herramientas de desarrollo que nos facilitan información de cualquier clase en java (por ejemplo SceneBuilder o NetBeans) utilizan la introspección para esto.

RESPONDE

5. Sea cual sea la tecnología empleada para crear un componente, ¿cuál crees que es la principal característica que debe cumplir?

- > a. Que permita ver su implementación.
- b. Que ofrezca los servicios de persistencia e introspección.
- c. Que oculte su interfaz.
- d. Que no permita crear clases reutilizables

Los componentes deben de ser visibles al implementarlos para poder trabajar con ellos fácilmente.

RESPONDE

6. En un archivo jar se deben incluir todos los archivos necesarios para que el componente funcione, incluidos archivos de clase, imágenes e iconos.

-> a. Verdadero

b. Falso

El archivo jar es una librería, y para que un componente de esa librería funcione, debe de tener todos sus componentes, incluyendo todo lo del enunciado.

Requerimientos:

4.3 Eventos en los componentes:

Componente MiControl:

https://github.com/Doimian/2DAM/tree/main/Workspaces/NetBeans%20Workspace/Componentes_HermidaDamian/src/miscontroles/micontrol

Aplicación UsandoMiControl:

<https://github.com/Doimian/2DAM/tree/main/Workspaces/NetBeans%20Workspace/UsandoMiControl>

4.4 Introspección. Reflexión:

Utilizando la introspección con BeanInfo:

https://github.com/Doimian/2DAM/blob/main/Workspaces/NetBeans%20Workspace/Componentes_HermidaDamian/src/miscontroles/micontrol/MiControlBeanInfo.java

4.5 Creación de un componente FXML:

Carpeta Componentes_HermidaDamian:

https://github.com/Doimian/2DAM/tree/main/Workspaces/NetBeans%20Workspace/Componentes_HermidaDamian

Componente CampoTextoBoton:

https://github.com/Doimian/2DAM/tree/main/Workspaces/NetBeans%20Workspace/Componentes_HermidaDamian/src/miscontroles/campotextoboton

Componente Tooltip modificado:

<https://github.com/Doimian/2DAM/tree/main/Workspaces/NetBeans%20Workspace/UIControISamples/src/tooltipsample>

Componente CheckBox modificado:

<https://github.com/Doimian/2DAM/tree/main/Workspaces/NetBeans%20Workspace/UIControISamples/src/checkboxsample>

Componente TableView modificado:

<https://github.com/Doimian/2DAM/tree/main/Workspaces/NetBeans%20Workspace/UIControISamples/src/tableviewsample>

Componente CampoTextoNumerico:

https://github.com/Doimian/2DAM/tree/main/Workspaces/NetBeans%20Workspace/Componentes_HermidaDamian/src/miscontroles/campotextonumerico

4.6 Uso de componentes personalizados

Componente SelectorDeslizamiento:

https://github.com/Doimian/2DAM/tree/main/Workspaces/NetBeans%20Workspace/Componentes_HermidaDamian/src/miscontroles/selectordeslizamiento

Aplicación APPUsoSelector:

<https://github.com/Doimian/2DAM/tree/main/Workspaces/NetBeans%20Workspace/APPUsoSelector>

Aplicación AppUsoTextoBoton:

<https://github.com/Doimian/2DAM/tree/main/Workspaces/NetBeans%20Workspace/APPUsoCampoTextoBoton>

4.7 Implementación Componente intermedio

Componente Temporizador:

https://github.com/Doimian/2DAM/tree/main/Workspaces/NetBeans%20Workspace/Componentes_HermidaDamian/src/miscontroles/temporizador

Aplicación APPTemporizador:

<https://github.com/Doimian/2DAM/tree/main/Workspaces/NetBeans%20Workspace/APPTemporizador>