

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.

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.

3. Una vez desarrollado el componente para que las herramientas lo reconozcan y se pueda 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.

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.

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.

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