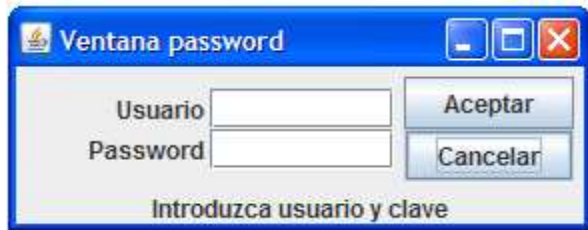


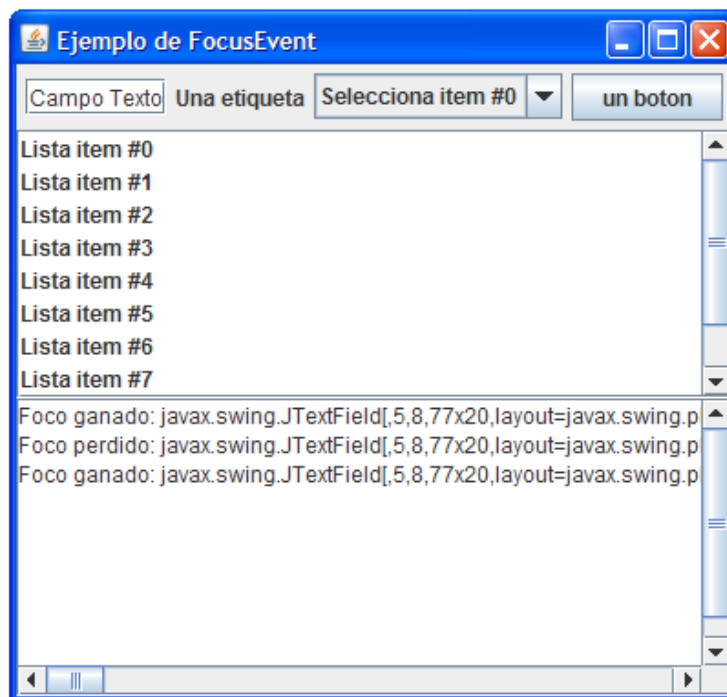
Práctica 12: Programación de eventos

1. Diseña un interfaz gráfica como la de la figura, de forma que cuando se pulse sobre el botón SI, aparezca el mensaje “Si pulsado”, y cuando se pulse el botón NO, aparezca el mensaje “No pulsado”. Inicialmente, aparecerá un mensaje en el que ponga pulsaciones.

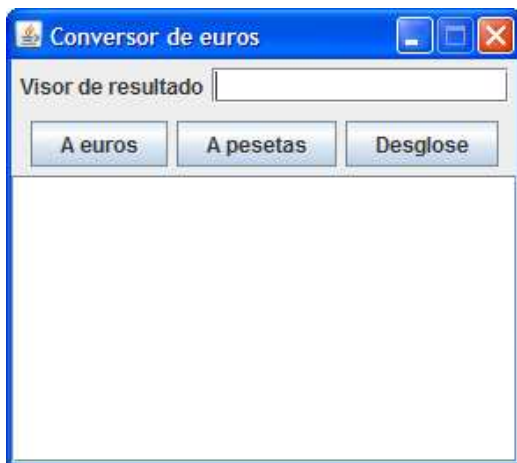


2. Diseñar el siguiente GUI interactivo, de forma que cuando el usuario pulse el botón Aceptar, si el nombre de usuario y su contraseña son correctas, aparezca el mensaje “Clave correcta” y si no el mensaje “Clave incorrecta”. Al pulsar el botón Cancelar, deberá borrarse lo que se hubieses escrito en los campos Usuario y Password.

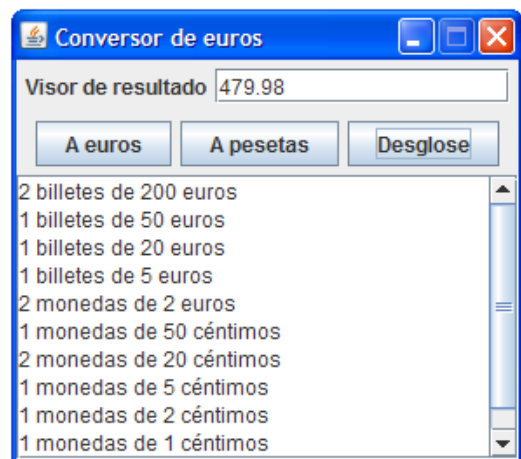
3. Diseñar la siguiente interfaz gráfica, de forma que inicialmente, el área de texto de la figura vayan apareciendo, los distintos componentes de la ventana que va ganando o perdiendo el “focus”.



4. Diseña una Interfaz Gráfica de Usuario como la que aparece en la siguiente figura A. Prográmala de forma que cuando se pulse el botón **A euros** aparezca en el visor de resultados la cantidad equivalente a una que se halla teclado anteriormente pero convertida en euros. Igualmente deberá ocurrir cuando se pulse el botón **A pesetas**, transformando esta vez la cantidad en pesetas. Por último, cuando se pulse el botón **Desglose**, se deberá mostrar por pantalla el desglose de billetes y monedas óptimo (mínimo número de unidades monetarias) de la cantidad que aparece en el visor, tal y como se muestra en la figura B.



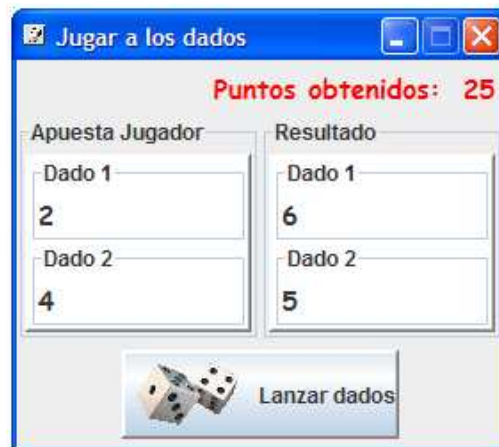
(A)



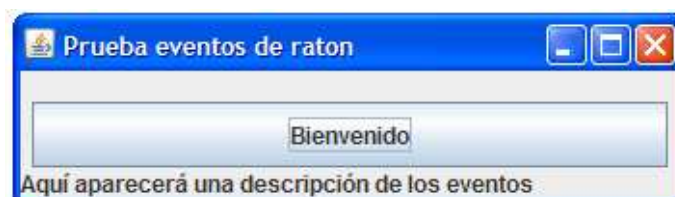
(B)

5. Hacer un programa en **Java** para jugar con el ordenador a acertar el resultado del lanzamiento de dos dados. La interfaz del juego deberá ser similar a la que se muestra en la figura 1. El juego consiste en lo siguiente:

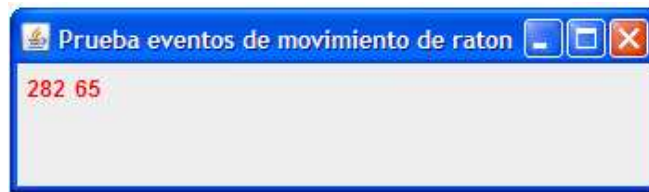
- ✓ El jugador dará el posible resultado de cada uno de los dados, y el ordenador internamente generará el número que saldrá en cada dado cada vez que se pulse el botón pulsar.
- ✓ Si el resultado dado por el jugador coincide con el del ordenador el jugador sumará 25 puntos a su acumulador, si solo coincide uno de los dados sumará 10 puntos, y si no coincide ninguno restará 5 puntos.
- ✓ A modo de ejemplo se muestra en la siguiente imagen, el aspecto que podría tener el interfaz tras haber realizado un determinado número de lanzamientos.



6. Diseña un interfaz gráfica con una apariencia similar a la de la figura que pruebe los 5 tipos de eventos de la interfaz *MouseListener*, de forma que en la etiqueta aparezca una breve descripción de lo que ha ocurrido, en los casos de presionar el botón del ratón y de soltarlo, se indicará también en la etiqueta la posición en que ha ocurrido la acción. Asimismo cuando el puntero del ratón se coloque sobre el botón, el color de fondo del mismo se pondrá rojo y cuando salga del botón, el color de fondo del mismo se pondrá amarillo.

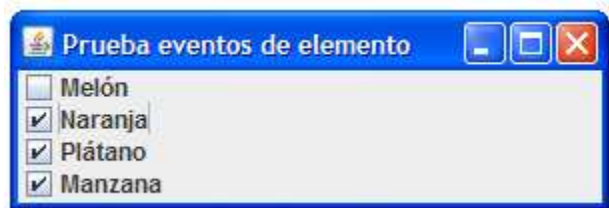


7. Hacer una interfaz gráfica que muestre una ventana como la de la figura, en la que se irán mostrando las posiciones X,Y que va tomando el puntero del ratón según el usuario lo va moviendo por la pantalla. El color del texto de las etiquetas que muestran las posiciones será de color azul si tenemos pulsado un botón del ratón y rojo si no lo tenemos. Nota: utilizar la interfaz *MouseMotionListener*.



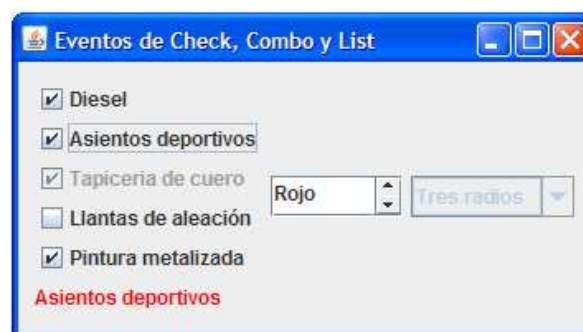
8. Diseñar una interfaz gráfica para probar el interfaz *ItemListener*, parecida a la de la figura y programarla de forma que cada vez que se vaya cambiando el estado de una de las cajas de texto, aparezca una breve descripción de lo que ha ocurrido, por ejemplo:

Componente: Naranja
Estado actual: true
Seleccionado



9. Este ejercicio prueba la forma de programar opciones y validaciones interactivas en GUI's. Para ello crearemos un interfaz, como el de la figura, con cinco cajas de verificación, una lista (tres radios, cinco radios y siete radios), una lista desplegable (rojo, azul y verde) y una etiqueta (seleccionado). La funcionalidad del método de esta interfaz será la siguiente:

- Si el elemento que ha generado el evento es “Llantas de aleación”, la lista se habilita en caso de que la caja de verificación esté seleccionada, o se deshabilita en caso de que la caja de verificación no esté seleccionada.
- La misma lógica se aplica con la caja de verificación “Pintura metalizada”, a través de la cual se controla si la lista desplegable con los colores se hace visible o no.
- La activación de la caja de verificación “Asientos Deportivos” controla el estado y la disponibilidad de la caja de verificación “Tapicería de cuero”, de esta manera, si la primera se selecciona, la segunda se selecciona automáticamente y se limita su disponibilidad.
- La etiqueta “Seleccionado” muestra en cada instante un texto relativo al elemento que ha sido seleccionado más recientemente.

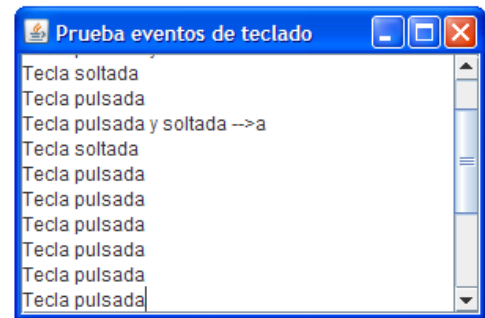


Se pide también realizar el ejercicio de la siguiente forma: todos los elementos (cajas de verificación, lista, lista desplegable y etiqueta) los añadiremos a un vector de componentes de 8 elementos. La clase que programa los eventos admite un constructor que recoge el vector de componentes con el fin de actuar sobre ellos. De esta forma los componentes no serán variables de instancia de la clase sino, variables locales al constructor que dibuja el interfaz.

Por último haremos otra versión del mismo ejercicio siguiendo el **Modelo Vista Controlador**.

10. En este ejercicio vamos a desarrollar dos ejemplos para probar el funcionamiento de los eventos de teclado.

a) Haremos un primer ejemplo que escribe un texto por pantalla cada vez que se activa uno de los tres métodos que proporciona el interfaz *KeyListener*, para ello simplemente añadiremos un Panel al contenedor principal, y a ese panel será al que le añadiremos el control (para probar el funcionamiento el panel tendrá que tener el *focus*).



b) Como segundo ejemplo diseñaremos un interfaz que contenga dos botones y una etiqueta, el control se añadirá a uno de esos dos botones, cuando el botón que tiene el control tenga el focus imprime todos los caracteres que vayamos tecleando sobre la etiqueta hasta que pulsamos el asterisco, momento en el que abandonamos la aplicación.



11. A continuación haremos un ejercicio que muestra el uso del interfaz *WindowListener*. Para ello, implementaremos los métodos del interfaz *WindowListener* para que imprima, cada uno, un texto significativo del evento. El programa de prueba creará dos ventanas.



12. Realizar una aplicación que nos permita “navegar” a través de los directorios del sistema, ofreciendo la siguiente información de cada fichero/directorio seleccionado:

- ✓ Existe permiso de lectura.
- ✓ Existe permiso de escritura.
- ✓ Es directorio.
- ✓ Está oculto.

