

CLASE 3-ESTUDIO EJ. JDIALOG

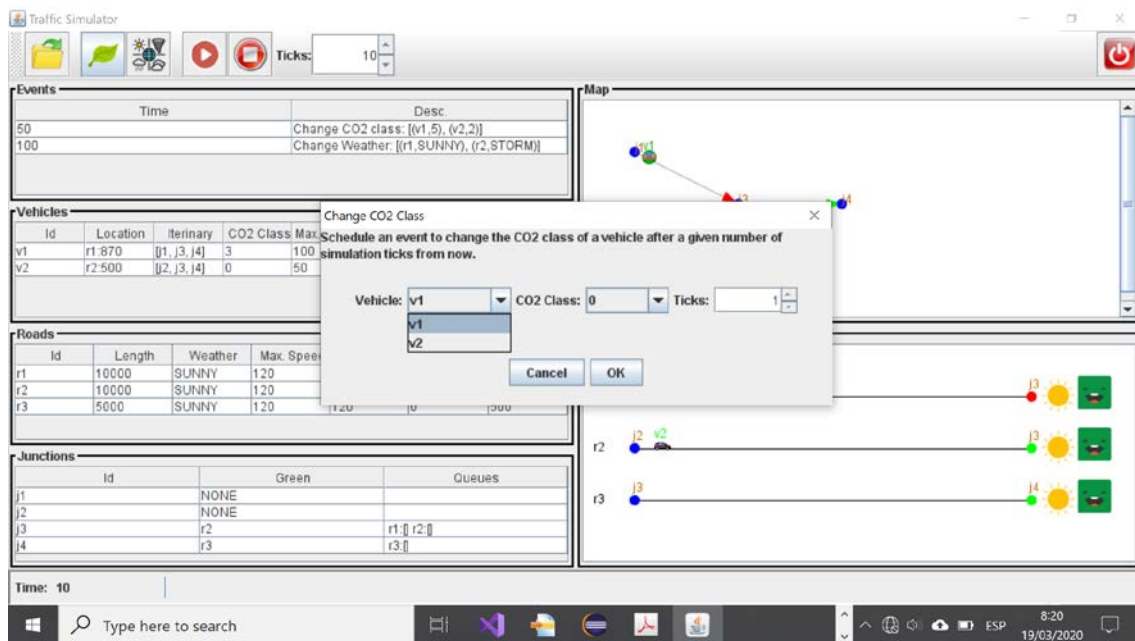
Vamos a estudiar el ejemplo: DialogWindowExample que os dejo en el CV porque os pedimos algo muy parecido en la PR2.

Bien, vamos a situarnos, hemos visto ventanas sencillas ya preparadas para informar al usuario con el método showMessageDialog de la clase JOptionPane (vimos el ejemplo JRadioButtonYJOptionPane). Hay más tipos de ventanas preparadas (que no dio tiempo a ver), las tenéis en las páginas 10-12 del tema Componentes Visuales II y de todas hay ejemplos. Todas estas ventanas heredan de JDialog.

En algunos puntos de la práctica utilizaremos estas ventanas, por ejemplo, al hacer click en el botón para salir de la aplicación utilizamos el método showOptionDialog de la clase JOptionPane, que nos mostrará una ventana preparada con opciones si/no (esto lo tenéis en el ejemplo EjemploJOptionPane)

Lo que vamos a hacer en la práctica es crear nuestro propio cuadro de diálogo, para ello vamos a extender a JDialog. Esto lo vais a hacer tras hacer clic en 2 botones, uno para añadir un evento que cambia la clase de contaminación de un vehículo y el otro para cambiar las condiciones meteorológicas de una carretera.

Me centro en unos de ellos, cambiar la clase de contaminación de un vehículo, y el otro es análogo:



Este ejemplo que os dejo en el CV crea un cuadro de diálogo parecido a los que vosotros vais a crear en a práctica. ¿Qué contiene? Pues principalmente tiene dos JComboBox, un JSpinner y dos botones, uno para aceptar lo seleccionado y otro para cancelar (estos dos botones van a controlar como salgo del cuadro de diálogo).

Me centro en el JComboBox, tenéis un ejemplo sencillo: EjemploCombo1, coge los datos de un array de String, pero esto es insuficiente para nuestra práctica (mirad primero este ejemplo para comprenderlo) porque pasándole los datos iniciales al JComboBox con un array de String no se permite ni añadir elementos al JComboBox ni modificarlos, sin embargo, en vuestra práctica 2 los vehículos que se mostrarán en ese JComboBox pueden cambiar cada vez que, interactuando con la GUI, cambiamos el fichero de entrada. Así, necesitamos utilizar un modelo por defecto que ya viene hecho en Java, la clase DefaultComboBoxModel (viene explicado en la página 7 del tema Componentes Visuales II).

Este modelo ya incluye métodos para refrescar el JComboBox tras hacer operaciones como: addElement, removeElement...Es decir, nos despreocupamos de refrescarlo.

Así que en este ejemplo que os paso (os lo doy comentado) vais a aprender a utilizar el modelo de JComboBox DefaultComboBoxModel y a crear vuestros propios cuadros de diálogo, para ello hay que extender a JDialog.

¡Pues chicos! A por el ejemplo DialogWindowExample y continúo explicando ahí....