

Curso / Kurtsoa	Fecha / Data	Nivel / Maila	Eval. / Ebal.
2018/2019	03/06/2019	1	Final 1
Kodea / Código	U.Didak /Unid didác	Tipo / Mota	Calificación/Kalifikazioa
PROG	Todas	Todos	

Crea un nuevo **proyecto de Java** de nombre **Java** si todavía no existe. Crea dentro un paquete con el nombre **examenFinal1** y guarda dentro de él las clases de este examen. El examen consta de **varias partes**. Cada parte tiene su puntuación y es necesario que esté **completamente correcta** para obtener la puntuación. En caso de no estar totalmente correcta su puntuación será cero.

ExamenFinal1Primos

Crea la clase Java de nombre ExamenFinal1Primos que tiene la siguiente interfaz gráfica



En la zona central se coloca una JList de nombre **lstPrimos** que permite la **selección múltiple** y que contiene elementos de la clase **Integer**. Los datos de la lista se guardan en un **DefaultListModel** de nombre **dlnmPrimos** que guarda elementos de la clase **Integer**. Además, la lista cuenta con un JScrollPane de nombre **scrollPanePrimos** que permite que se vean los elementos de la lista.

Parte 0. (0,5 Ptos.)

En la zona inferior se coloca un JPanel de nombre **panelEstado** que tiene una JLabel de nombre **lblPrimos** con el texto "Estado Primos: ", otra JLabel de nombre **lblEstadoPrimos** que inicialmente tiene el texto "**Datos Sin Modificar**" pero que cuando los datos de la lista son modificados cambia al valor a "**Datos Modificados**".

Parte 1. (2 Ptos.)

lblNumero. JLabel con el texto "Numero".

txtNumero. JTextField con el texto "0". Al pulsar enter hará lo mismo que btnInsertar. Al coger el foco se seleccionará todo el texto y al perder el foco el texto dejará de estar seleccionado. (0,25 Ptos.)

btnInsertar. JButton con el texto "Insertar". Al hacer clic sobre él comprobará si el valor de txtNumero es primo o no. Si es primo añadirá ese valor a la lista que le corresponda, pero sólo si ese valor todavía no está en la lista. (0,5 Ptos.)

btnBorrar. JButton con el texto "Borrar". Al hacer clic sobre él se borrarán todos los elementos seleccionados en la lista, si es que hay alguno. (0,5 Ptos.)

btnLimpiar. JButton con el texto "Limpiar". Al hacer clic sobre él se borrarán todos los elementos de la lista. (0,25 Ptos.)

Parte 2. (1,5 Ptos.)

Al hacer clic sobre la X para **cerrar la ventana** de la aplicación se comprobará si los **datos** han sido **modificados**. Si han sido modificados se llama a un método de nombre **guardarDatos** que es el que se encarga de grabar los datos de los objetos de la clase Integer que hay en la lista en el fichero **primos.ser** usando serialización.

Parte 3. (1 Ptos.)

Al **iniciar la aplicación** se cargan los datos de los **objetos** de la clase **Integer** que hay en el fichero **primos.ser** usando **serialización** en el modelo de datos de la lista de primos usando un método de nombre **cargarDatos**.

Parte 4. (5 Ptos.)

Crea en MySQL una base de datos de nombre **bdfinal1** con cotejamiento **utf8_spanish_ci**. Crea una tabla de nombre **vehiculos** con los campos de texto **matricula**, **marca**, **modelo**. El valor del campo matricula es la clave primaria y no puede estar repetido.

Configura **Eclipse** para que permita **conexiones a bases de datos MySQL**.

Crea clase Java **ExamenFinal1BDVehiculos** que tiene un campo de texto por cada campo de la tabla vehiculos de la base de datos bdfinal1 que son matricula, marca, modelo.

Al iniciar la aplicación se cargan los datos de la tabla vehiculos de la base de datos bdfinal1 en un ArrayList.

La interfaz gráfica de la clase ExamenFinal1BDVehiculos es la siguiente



btnPrimero. JButton que al hacer clic sobre él permite ir al primer registro de datos (si es que hay). (0,25 Ptos.)

btnAnterior. JButton que al hacer clic sobre él permite ir al registro anterior (si es que hay). (0,25 Ptos.)

btnSiguiente. JButton que al hacer clic sobre él permite ir al registro siguiente (si es que hay). (0,25 Ptos.)

btnUltimo. JButton que al hacer clic sobre él permite ir al último registro (si es que hay). (0,25 Ptos.)

btnInsertar. JButton que al hacer clic sobre él añade un nuevo registro al ArrayList, si todavía no ha sido añadido. (0,5 Ptos.)

btnBorrar. JButton que al hacer clic sobre él borra el registro actual del ArrayList, si hay registros. (0,25 Ptos.)

btnActualizar. JButton que al hacer clic sobre él modifica los datos del registro actual del ArrayList, si hay registros. (0,25 Ptos.)

btnSalir. JButton que al hacer clic sobre él guarda los datos en la base de datos, sólo si han sido modificados. Para ello, borra todos los registros de la tabla vehículos y añade uno a uno todos los registros que hay en el ArrayList. (1,5 Ptos.)

lblRegistros. JLabel que si en la lista no hay registros muestra el texto "No hay registros" y, si hay registros, muestra el texto "Vehiculo Registro: " + registroActual + " de "+numeroRegistros. La etiqueta lblRegistros se tiene que actualizar cada vez que se produzcan cambios ya sea porque se va a otro registro o porque se inserta o se borra un registro. Para facilitar el proceso se usará un método de nombre **actualizarCampos**. (0,75 Ptos.)

Criterios de Corrección

Para tener la puntuación correspondiente a cada parte es necesario que todo lo que se pide en esa parte sea correcto. Si no es así la puntuación de esa parte será de 0 salvo que existan subpartes.

Si existen subpartes se valorará cada subparte, siendo necesario que todo lo que se pide en esa subparte sea correcto para tener la puntuación asignada. Si no es así la puntuación de esa subparte será de 0.

Al finalizar el tiempo se recoge el examen y se guarda el paquete con las clases dentro de una carpeta de nombre Apellido, Nombre donde Apellido hace referencia tu primer apellido y nombre a tu nombre. Por ejemplo, De Miguel, Txema.

No se permitirá ningún tipo de comunicación durante el examen. Los cables de red permanecerán desconectados, las conexiones inalámbricas (si las hubiera) se deshabilitarán y los móviles deberán estar apagados. Si un alumno incumple alguna de estas normas entregará su examen y tendrá un 0 como nota.