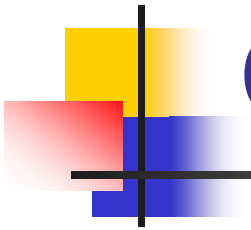
GUI para Acceder a una BD desde Java

Ingeniería del Software II
Curso 2008/2009

Sergio Ilarri Artigas
silarri@unizar.es



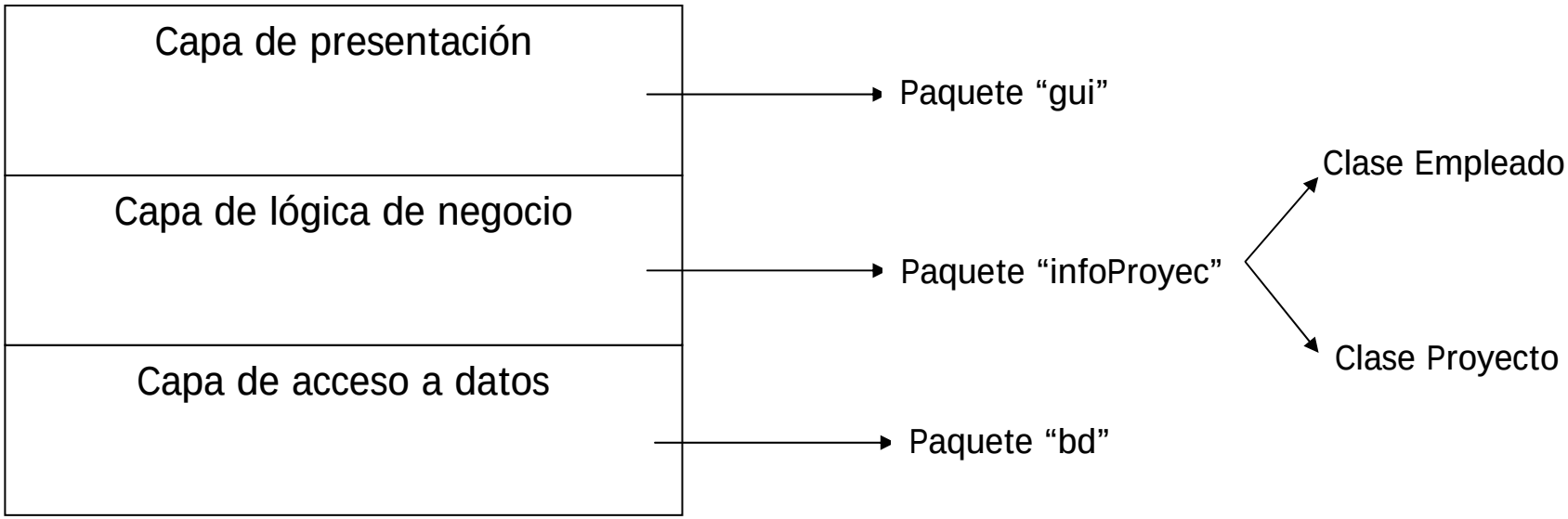
Objetivo (I)

- GUI para gestionar información de empleados y proyectos
  - Empleados: nombre, apellidos, año de nacimiento, NIF
  - Proyectos: título, fecha de inicio, fecha de fin, descripción
  - Información de quién participa en qué proyectos

Objetivo (II)

- Funcionalidades:
  1. Introducir datos de empleados
  2. Introducir datos de proyectos
  3. Modificar datos de empleados y proyectos
  4. Asociar empleados a proyectos
  5. Obtener los empleados de un proyecto
  6. Obtener la lista de proyectos de un empleado
  7. Listar todos los proyectos
  8. Listar todos los empleados

Estructura de la Solución



¿Tiene esta estructura alguna ventaja?

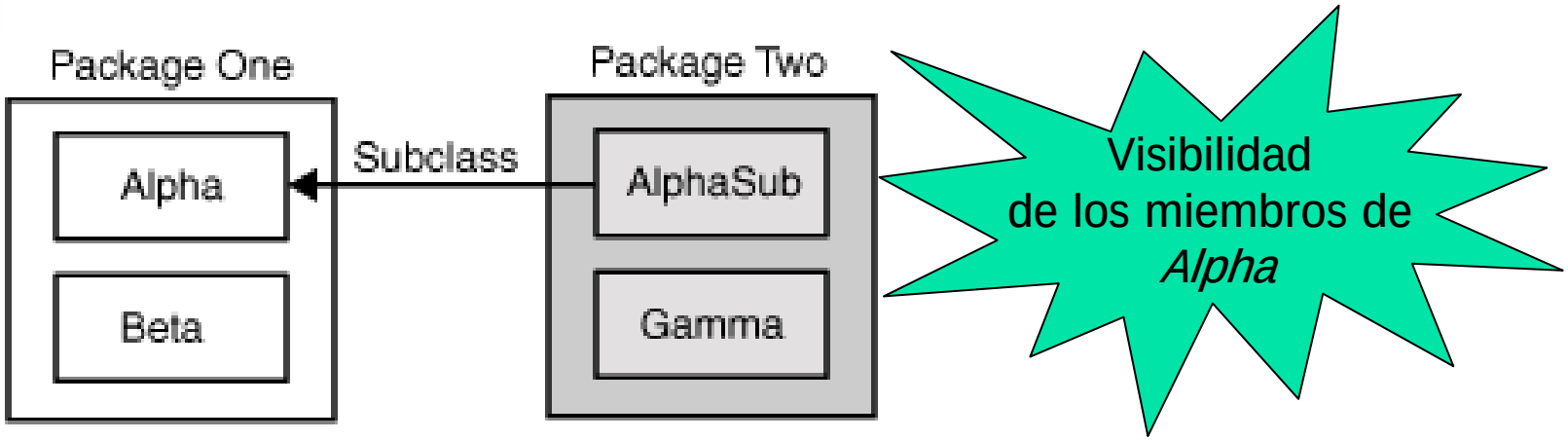
Un poco de teoría...

No recuerdo esto de los paquetes...

- Espacio de agrupamiento de clases e interfaces relacionados:
  - Estructuración
  - Protección de acceso
  - Espacio de nombres

http://java.sun.com/docs/books/tutorial/java/package/packages.html

Recuerda: Control de Acceso



Modificador	Alpha	AlphaSub	Beta	Gamma
private	X			
protected	X	X	X	
public	X	X	X	X
(sin nada)	X		X	

http://java.sun.com/docs/books/tutorial/java/javaOO/accesscontrol.html

Uso y Definición de Paquetes

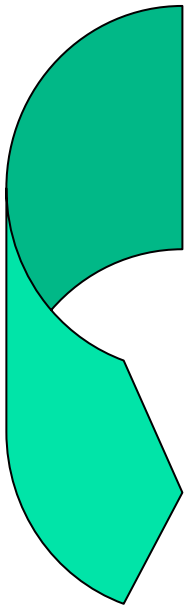
- ¿Cómo usarlos?:
  - 1) Importar el elemento:

    import graphics.Circle;
  - 2) Importar el paquete completo:

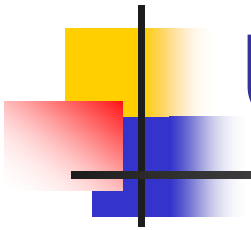
    import graphics.*;
  - 3) Indicar el paquete cuando se necesita:

    graphics.Circle c = new graphics.Circle(...);

java.lang:
***import* no necesario**



Circle c = new Circle(...);

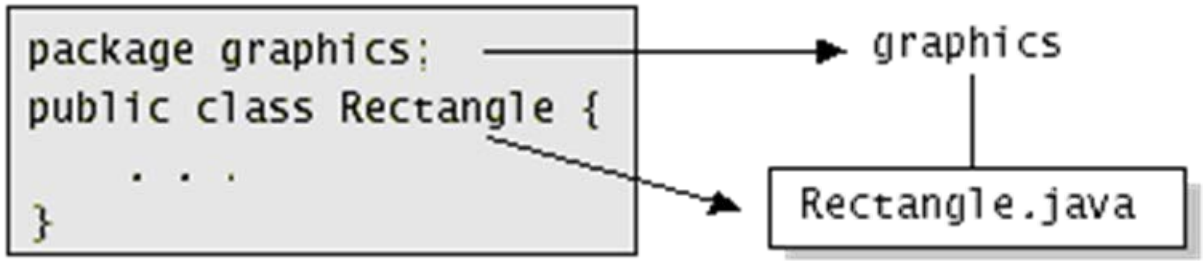


Uso y Definición de Paquetes

- ¿Cómo definirlos?:
  - 1) La primera línea del fuente puede ser:
    - `package nombrePaquete;`
  - 2) Si no, estará en el paquete por defecto

Estructura de Directorios (I)

JERARQUÍA DE FICHEROS



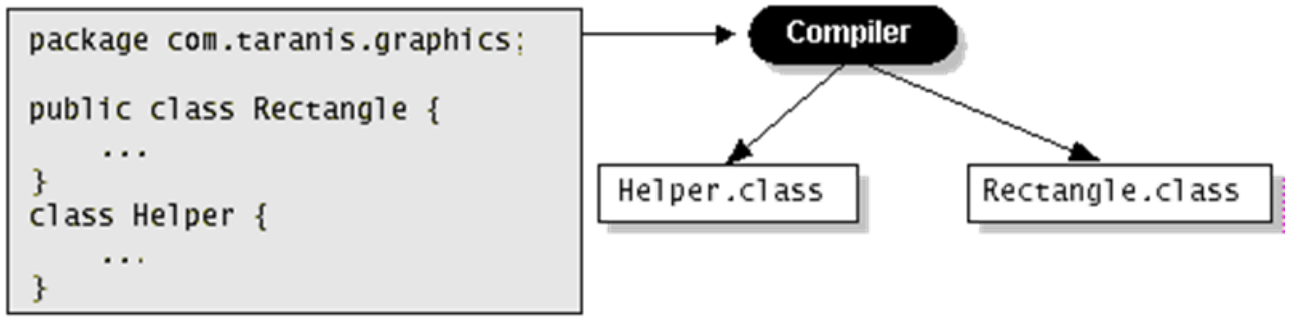
Nombre clase
graphics.Rectangle

Path al fichero
graphics/Rectangle.java

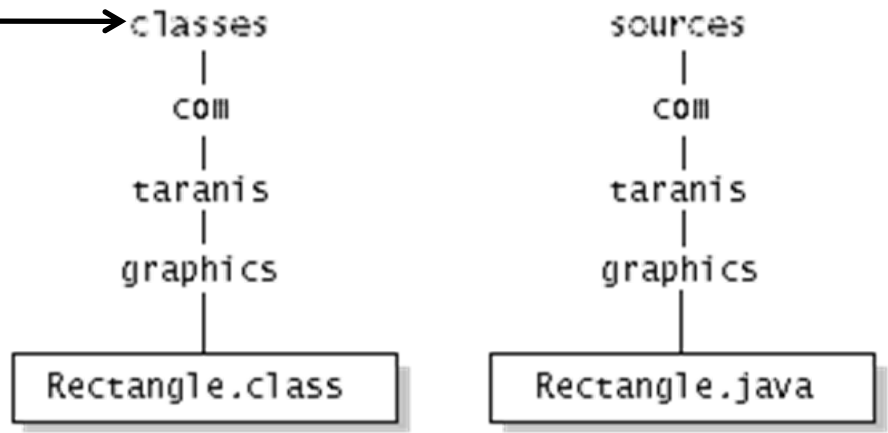
(Aunque esto no está en la especificación, lo hacen las implementaciones)

Estructura de Directorios (II)

COMPILACIÓN



Classpath (*path* al .class?)

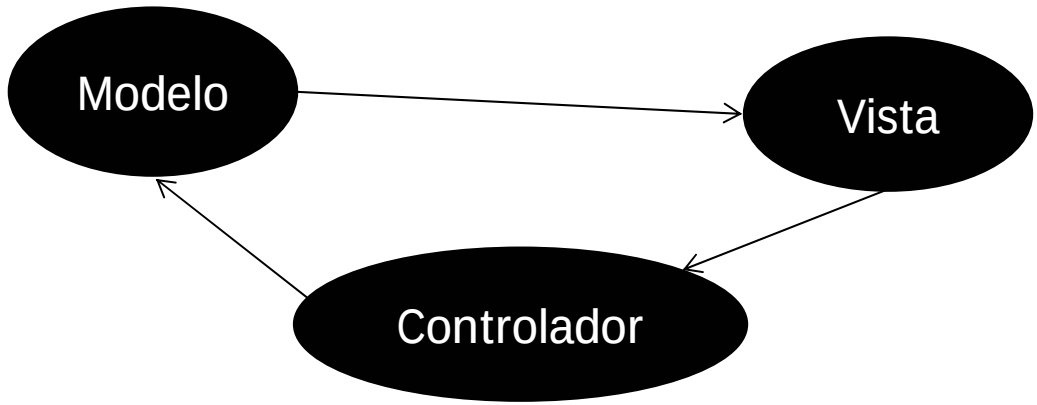


Consejo: un directorio para almacenar las clases y otro para los fuentes

Quizá necesitéis utilizar “modelos” de *Jtable*...

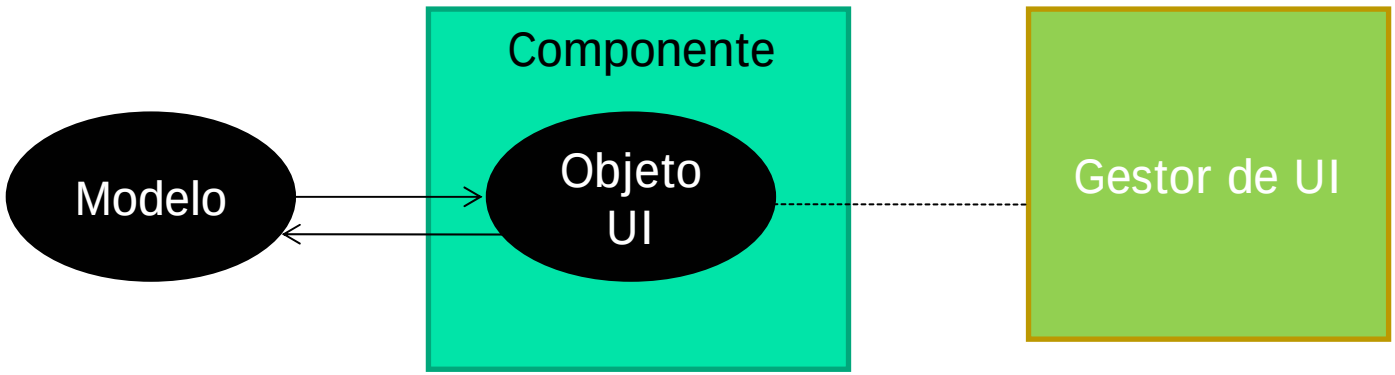
Recuerda: Arquitectura MVC

- *Model-View-Controller*
- División de una aplicación visual en 3 partes:
  - Modelo: representa los datos de la aplicación
  - Vista: es la representación visual de los datos
  - Controlador: recibe entradas del usuario a través de la vista y actualiza el modelo en consecuencia



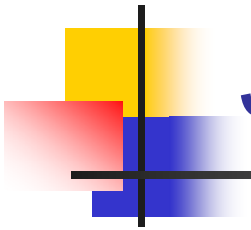
Recuerda: Arquitectura MVC

- El primer prototipo de Swing seguía ese estilo
- Pero luego se determinó que la vista y el controlador requerían un fuerte acoplamiento
- *Separable model architecture* (*quasi-MVC*)
  - *UI Object* = *UI delegate* = *delegate object*
    - Vista + controlador



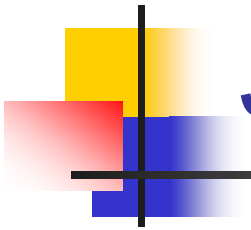
http://java.sun.com/products/jfc/tsc/articles/architecture/

Ejemplos de “Modelos” en Java



- Botones (“normales”, de radio, de selección):
  - Interface ButtonModel (javax.swing):
    - isSelected(), addActionListener(...), addChangeListener(...), addItemListener(...), ...
    - Ejemplo de clase que lo implementa:
      - DefaultButtonModel

Ejemplos de “Modelos” en Java

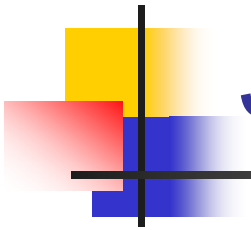


- Listas (JList):
  - Interface ListModel (javax.swing):
    - void addListDataListener(ListDataListener l)
    - void removeListDataListener(ListDataListener l)
    - Object getElementAt(int index)
    - int getSize()
    - Ejemplo de clase que lo implementa:
      - DefaultListModel

Ejemplos de “Modelos” en Java

- “Contenedores de texto”:
  - Interface Document (javax.swing.text):
    - int getLength()
    - void addDocumentListener(DocumentListener l)
    - void removeDocumentListener(DocumentListener l)
    - String getText(int offset, int length)
    - void remove(int offs, int len)
    - Ejemplo de clase que lo implementa:
      - PlainDocument, DefaultStyledDocument, HTMLDocument

Ejemplos de “Modelos” en Java



- Tablas (JTable):
  - Interface TableModel (javax.swing.table):
    - int getRowCount()
    - int getColumnCount()
    - Boolean isCellEditable(int rowIndex, int columnIndex)
    - void addTableModelListener(TableModelListener l)
    - void removeTableModelListener(TableModelListener l)
    - ...

Ejemplos de “Modelos” en Java

- Tablas (JTable):
  - Interface TableModel (javax.swing.table):
    - ...
    - Object getValueAt(int rowIndex, int columnIndex)
    - void setValueAt(Object aValue, int rowIndex, int columnIndex)
    - String getColumnName(int columnIndex)
    - Class getColumnClass(int columnIndex)
    - Ejemplo de clase que lo implementa:
      - DefaultTableModel

¿Para qué Queremos los Modelos?

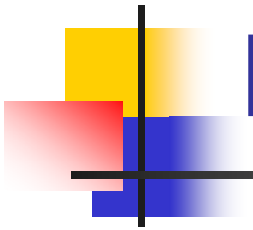
- Normalmente no necesitamos utilizar directamente los modelos. Ej.:
  - JSlider js = new JSlider();

BoundedRangeModel

js.getModel().getValue() es equivalente a:

js.getValue()

¿Para qué Queremos los Modelos?



- Para comportamientos no predefinidos, podemos definir modelos propios

- En breve veremos un ejemplo con JTable

Volvemos al ejercicio...
Interacción con el SGBD

http://java.sun.com/docs/books/tutorial/jdbc/basics/index.html

Estructura de las Tablas

- Tabla “Empleados”: nombre, apellidos, año de nacimiento, NIF
- Tabla “Proyectos”: título, fecha de inicio, fecha de fin, descripción
- Tabla “Participar”: NIF, título

¿Cuáles son las claves?

Conexión al SGBD Oracle

1) Cargar el driver JDBC-ODBC:

- `Class.forName("oracle.jdbc.driver.OracleDriver");`

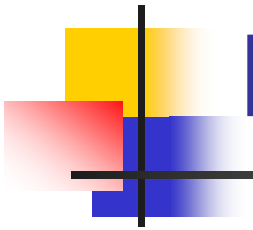
2) Conectarse a la fuente

```
String url =
  "jdbc:oracle:thin:@den.cps.unizar.es:1521:vicious";

Connection con = DriverManager.getConnection(url,
  username, password);
```

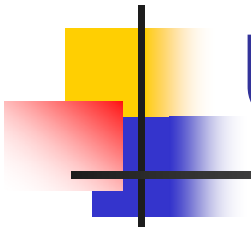
Posibles Excepciones

- Class.forName(...) puede lanzar *ClassNotFoundException*:
  - ¿Está bien escrito, letra por letra, el nombre de la clase del driver?
  - ¿Has incluido en el classpath el driver JDBC a utilizar?
    - java –cp ./CLASSES:/usr/local/pkg/instantclient_10_2/classes12.jar ...
  - ¿Estás trabajando en las máquinas del laboratorio?



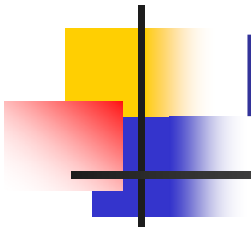
Posibles Excepciones

- DriverManager.getConnection(...) puede lanzar *SQLException*:
  - ¿Está el SGBD vivo?
    - sqlplus <USERNAME>@vicious.den.cps.unizar.es
  - ¿Has indicado correctamente el usuario y password?



Un Par de Preguntas...

- El classes12.jar, ¿tiene que estar disponible para *javac*, para *java*, o para ambos?
- ¿Cuándo se crea la instancia del driver?



Elementos Básicos del API...

- Connection:
  - Statement createStatement()
  - PreparedStatement prepareStatement(String sql)
  - void close()
- Statement:
  - ResultSet executeQuery(String sql)
  - int executeUpdate(String)
  - void close()

Elementos Básicos del API...

- PreparedStatement:
  - boolean execute()
  - ResultSet executeQuery()
  - void setInt(int parameterIndex, int x)
    - Y otros métodos setXXX(...)
- ResultSet:
  - boolean next()
  - String getString(int columnIndex)
    - Y otros métodos getXXX(...)
  - void close()

Durante la corrección, todos utilizaréis la misma BD...

Si la Tabla Ya Existe...

Ejemplo

```
private static boolean existeTabla(String nombreTab) throws SQLException{
        boolean existe;
        stmtExiste = con.createStatement();
        resultadoPeliculas = stmtExiste.executeQuery(
          "SELECT * FROM cat WHERE table_name=upper('" + nombreTab + "')");
        existe = resultadoPeliculas.next();
        stmtExiste.close();
        return(existe);
    }
```

No terminar las sentencias SQL en “;”

Si la Tabla Ya Existe...

Ejemplo

```
...
String nombreTabla = “actores”;
if (existeTabla(nombreTabla) == true)
{
   stmtActores.executeQuery("DROP TABLE " + nombreTabla);
}

 stmtActores.executeQuery("CREATE TABLE actores(id NUMBER(4) PRIMARY KEY
      NOT NULL, nombre VARCHAR(20) NOT NULL, apellidos VARCHAR(20) NOT
      NULL, nacion NUMBER(4) REFERENCES paises(id) NOT NULL, anyo NUMBER(4)
      NOT NULL, sexo VARCHAR(1) NOT NULL)");

stmtActores.close();
...
```

Si la Tabla Ya Existe...

Pero hay un posible problema: restricciones de integridad referencial definidas por otros compañeros... o por nosotros mismos si no borráis las tablas en el orden adecuado...

Ejemplo

```
DROP TABLE Proyectos cascade constraints
```

¿Cómo hacer el *mapping* entre objetos Java y tablas?

Tablas -> Objetos Java

Un ejemplo

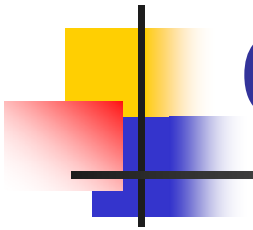
```
public static Proyecto leerProyecto()
{
   ...
   Proyecto proyecto= new Proyecto(...);
   return proyecto;
}
```

Tablas -> Objetos Java

Otro ejemplo

```
try
{
   ResultSet  result = sentencia.executeQuery("select * from PELICULAS where titulo =
     '“ + this.titulo);;
   result.next();
   this.anyo=result.getInt("anyo");
   this.duracion=result.getInt("duracion");
   this.resumen = result.getString("resumen");
}
catch(Exception e)
{
   JDialog frame = new JDialog();
   JOptionPane.showMessageDialog(frame,  e.getMessage(),"Error al obtener datos de
     la pelicula“, OptionPane.ERROR_MESSAGE);
}
```

¿Alguna clase de *Swing* a tener en cuenta para la práctica?

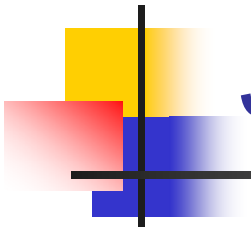


Clases de Swing interesantes

- JFrame, JDialog, JPanel, JScrollPane, JMenu, JMenuItem, JTable, JComboBox/JList, JTextField.
- Práctica 1 (manejo de eventos de menú)
- Manejadores de eventos
- Layout Managers

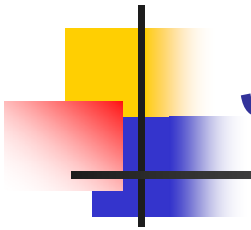
Un poco más sobre *JTable*

http://java.sun.com/docs/books/tutorial/uiswing/components/table.html



JTable

- Constructores:
  - JTable(Object[][] rowData, Object[] columnNames)
  - JTable(Vector rowData, Vector columnNames)
- Pero:
  - Celdas editables
  - Renderiza como String’s
  - Trata como String’s los argumentos



JTable y Modelos

- Modelos (interfaces):
  - TableModel
  - TableColumnModel
- Modelos (clases):
  - AbstractTableModel
    - DefaultTableModel
  - DefaultTableColumnModel

Ejemplo: Consiguiendo Celdas No Editables

Definir un modelo

```
public class ModeloTablasNoEditables extends DefaultTableModel
{
   public boolean isCellEditable (int row, int column)
   {
      return false;
   }
}
```

Establecer el modelo

```
ModeloTablasNoEditables modelo = new ModeloTablasNoEditables ();
JTable tabla = new JTable(modelo);
```

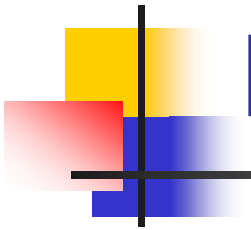
Para permitir tipos específicos, sobreescribir getColumnClass(int row, int column)

Manejo de Eventos de Tablas

```
public class SimpleTableDemo ... implements TableModelListener {
    ...
    public SimpleTableDemo() {
        ...
        table.getModel().addTableModelListener(this);
        ...
    }

    public void tableChanged(TableModelEvent e) {
        int row = e.getFirstRow();
        int column = e.getColumn();
        TableModel model = (TableModel)e.getSource();
        String columnName = model.getColumnName(column);
        Object data = model.getValueAt(row, column);

        ...// Do something with the data...
    }
    ...
}
```



Disparar Eventos

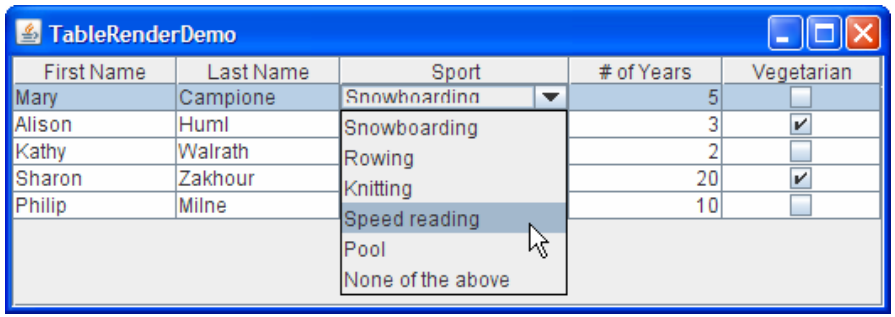
- Métodos de AbstractTableModel:
  - fireTableCellUpdated
  - fireTableRowsUpdated
  - fireTableDataChanged
  - fireTableRowsInserted.
  - fireTableRowsDeleted
  - fireTableStructureChanged

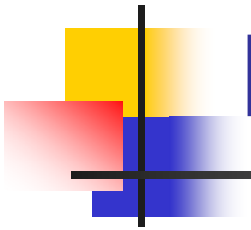
Ejemplo de Colocación de la JTable en Panel con Scroll

```
// Create the scroll pane and add the table to it.
JScrollPane scrollPane = new JScrollPane(table);

// Add the scroll pane to this panel.
add(scrollPane);
```

Y ahora veamos alguna demo...





Fin

Gracias por vuestra atención