

Programación 2

Lenguaje Java - Manejo de Excepciones (*3era parte*) Típos de Clases

Profesor: Eduardo Godoy.

eduardo.gl@gmail.com

Escuela de Ingeniería Civil Informática.
Universidad de Valparaíso.

23 de octubre de 2017

Contenido

1 Acceso a Datos

2 Java Swing: Data Table

Manejo de archivos - Archivo Secuencial

- Forma básica de almacenar registros en un archivos.
- Sus registros son de similares en exstructura y tamaño.
- Se ordenan de forma secuencial en base a un valor de campo o atributo, en nuestro caso un ID.

Manejo de archivos

- Archivo Origen o Base: Encargado de almacenar los registros de forma persistente. Además gestiona y mantiene su estado en el tiempo.
- Archivo Auxiliar: Encargado de dar soporte a las operaciones básicas sobre el archivo origen, permitiendo hacer cambios de estado. Debido a que se emplea como medida de respaldo.
- Las operaciones básicas son los métodos: crear, actualizar y eliminar.

Manejo de archivos - crear()

- Se recorre archivo de inicio a fin.
- Se procede a la copia de registros desde archivo origen a archivo auxiliar
- Por cada linea leida se guarda su id siendo este sobre escrito por el valor id de la siguiente linea procesada.
- Cada Registro Leído se almacena en el archivo axiliar.
- Se continua con la inserción del nuevo registro al final del archivo auxiliar con el ultimo id obtenido
- incrementado en 1, con esto se obtienen claves únicas para cada registro.
- Se finaliza con el cipiado del archivo auxiliar al orchivo base.

Manejo de archivos - actualizar()

- Se recorre archivo de inicio a fin.
- Se procede a la copia de registros desde archivo origen a archivo auxiliar
- Por cada linea leida se comparar su id con el id pasado por parámetro.
- De ser iguales se inserta en el archivo auxiliar el registro actualizado que viene como pa´rámetro,
- dejando el registro leido desde origen sin ser escrito en auxiliar.
- De lo contrario se siguen escribiendo los registros en el archivo auxiliar sin ser afectados por cambios.
- Se finaliza con el cipiado del archivo auxiliar al orchivo base.

Manejo de archivos - eliminar()

- Se recorre archivo de inicio a fin.
- Se procede a la copia de registros desde archivo origen a archivo auxiliar
- Por cada linea leida desde origen se compara su id con el id pasado por parámetro.
- De ser iguales ese registros se marca como nulo.
- Cada registro Leído se almacena en el archivo axiliar. excepto el que ha sido marcado como nulo.

Manejo de archivos - grabar()

- Genera una copia del archivo origen con un nombre de respaldo.
- Toma el archivo origen y borra su contenido.
- Escribe la cabecera del archivo en origen.
- Recorre el archivo auxiliar leyendo cada registro.
- Para cada registro leído desde auxiliar es escrito en origen.

Concepto de mantenedor

- Interfaz de Usuario que permite administrar una base de datos mediante las operaciones de agregar, modificar y eliminar.
- Utilizados para gestionar los registros almacenados en una base de datos, ya sea, una tabla de datos, un archivo u otra fuente de almacenamiento.

Concepto de mantenedor - Java

- Componente Java Swing cuya responsabilidad es la tabulación de datos.
- Java provee de varios componentes swing para poder implementar de forma correcta mantenedores de fuentes de datos.
- En general para esto se utiliza el componente JTable.
- Se compone de Filas y Columnas.
- Es posible seleccionar filas y extraer su valor mediante eventos de selección con periféricos. Ejemplo: Mouse.

Tips - Ocultamiento de Información - Ejemplo.

Table Example

Programa Ayuda

Edad: Nivel Educacional:

Actividad: Renta Fija:

Lim Maximo de Credito: Deuda Actual:

% de Uso de Credito: N° Compras Mes Actual:

N° Compras Mes -1: N° Compras Mes -2:

N° Compras Mes -3: Estado Actual:

Cant de Atrasos de Pago: Compra:

id	edad	niveE	actividad	rentaFija	limMax	deuda	porcen	numC	numC	numC	numC	estado	canHi	compra
11	41	EDUC.	EMPR.	18090	301500	0	0.0	1	9	10	3	SIN D.	11	NO
12	33	EDUC.	DEPE.	25121	689711	689711	100.0	1	1	1	0	SIN D.	19	SI
13	53	EDUC.	DEPE.	10492	31963	31963	100.0	1	7	7	8	SIN D.	23	NO
14	66	EDUC.	DEPE.	13567	301500	0	0.0	2	10	7	9	SIN D.	20	SI
15	22	EDUC.	DEPE.	904500	923639	923639	100.0	2	8	8	3	SIN D.	12	SI
16	30	EDUC.	DEPE.	19597	301500	191103	63.4	2	8	3	1	DEUD.	6	SI
17	29	EDUC.	EMPR.	27135	715236	715236	100.0	5	0	5	7	SIN D.	12	NO
18	56	EDUC.	DEPE.	859275	617177	617177	100.0	0	4	0	3	SIN D.	14	NO
19	34	EDUC.	DEPE.	77767	301500	0	0.0	1	1	1	7	SIN D.	17	NO
20	43	EDUC.	DEPE.	18090	301500	0	0.0	10	10	0	2	SIN D.	24	NO
21	50	EDUC.	DEPE.	10492	301500	0	0.0	9	6	10	5	SIN D.	15	SI
22	42	EDUC.	DEPE.	904500	52310	52310	100.0	9	2	5	4	SIN D.	0	SI
23	40	EDUC.	DEPE.	20743	436301	436301	100.0	3	7	1	9	SIN D.	10	SI
24	55	EDUC.	DEPE.	11758	301500	76117	25.2	2	9	3	8	SIN D.	2	SI
25	55	EDUC.	DEPE.	27135	489859	489859	100.0	9	2	7	6	SIN D.	18	SI
26	47	EDUC.	DEPE.	904500	301500	0	0.0	7	8	2	9	SIN D.	25	SI
27	54	EDUC.	EMPR.	859275	301500	151	0.1	8	5	2	3	SIN D.	9	NO
28	25	ESTU	DEPE.	904500	301500	0	0.0	1	10	8	8	SIN D.	11	SI
29	66	EDUC.	DEPE.	13567	301500	207191	68.7	4	10	0	3	SIN D.	13	NO
30	30	EDUC.	DEPE.	904500	301500	83274	27.6	7	2	1	5	SIN D.	13	NO
31	42	EDUC.	DEPE.	906508	301500	0	0.0	6	3	9	4	SIN D.	15	NO
32	40	EDUC.	DEPE.	906508	301500	0	0.0	6	5	7	7	SIN D.	24	SI
33	38	EDUC.	DEPE.	712927	301500	0	0.0	9	4	6	7	DEUD.	23	NO
34	35	EDUC.	DEPE.	29034	301500	663	0.2	0	10	1	10	SIN D.	16	SI
35	49	EDUC.	DEPE.	13115	366660	366660	100.0	7	5	7	4	SIN D.	21	SI

Agregar Eliminar Actualizar

JTable: Instanciar Nueva

Ejemplo.

```
private JTable table = new DataTable;  
...  
add(this.add(new JScrollPane(table)), BorderLayout.  
    CENTER);  
...
```

JTable: Agregar Registro.

Ejemplo.

```
public void agregarFila(Persona p) {  
    DefaultTableModel model = (DefaultTableModel) table.  
        getModel();  
    model.addRow(new Object[] {  
        p.getId(), p.getEdad(), p.getNivelEduc(),  
        p.getActividad(), p.getRentaFija(),  
        p.getLimMaxCredito(), p.getDeudaActual(),  
        p.getPorcentUsoCredito(), p.  
            getNumCompraMesActual(),  
        p.getNumCompraMesActual_1(), p.  
            getNumCompraMesActual_2(),  
        p.getNumCompraMesActual_3(), p.getEstadoActual  
            (),  
        p.getCantHistAtrasosPago(), p.getCompra()  
    });  
}
```

JTable: Obtener Registro.

Ejemplo.

```
class EliminarBotonListener implements ActionListener{
    @Override
    public void actionPerformed(ActionEvent e){
        int row = table.getSelectedRow();
        String id = table.getModel().getValueAt(row, <N\ '
            umero de campo a extraer>).toString();
        ...
    }
}
```