

## GUÍA PRÁCTICA N° 12

**Escuela Profesional:** Ingeniería de Sistemas.

**Ciclo y Turno:** Segundo: Noche

**Docente:** Ing. Dany Montoya Negrillo.

**Asignatura:** Metodología de la Programación.

**Semestre Académico:** 2013 - II

**Fecha:** 11 de Noviembre al 15 de Noviembre del 2013

### Sesión 12: Serialización de Objetos

**Objetivo:** Permitir al alumno, mediante la resolución de problemas utilizar la serialización de objetos con la ayuda de los ArrayList y los archivos binarios como elementos de almacenamiento a través de la lectura y escritura con los métodos necesarios en un JFrame Form en JAVA NetBeans con los controles Swing.

- I. Resolver todos los aplicativos con las opciones de actualizar, eliminar y consultar registros
1. Construya un programa que guarde la siguiente información en un archivo binario llamado "Competencia.bin" de los competidores de lanzamiento de jabalina Código, Nombre del participante, Nacionalidad, Categoría (infantil, juvenil, adulto), edad y distancia del lanzamiento. Su programa deberá mostrar los siguientes datos:
  - a) Nombre del participante más joven.
  - b) Nombre de los ganadores en cada una de las categorías.
  - c) Nacionalidad del participante que tuvo el menor lanzamiento.
  - d) Buscar, Modificar y Eliminar un registro en base al código.
2. La Oficina de Servicios Académicos de la UCV cuenta con la siguiente información: código, nombres del alumno, su foto, nota de práctica, control de lectura, nota de investigación, promedio (dato calculado), sexo, semestre (2009-II, 2010-I y 2010-II) y escuela (Sistemas, Industrial y Ambiental). La Universidad desea los siguiente datos estadísticos:
  - Número de estudiantes de cuyo promedio no pasa de la nota aprobatoria (10.5)
  - La escuela que donde se encuentra el alumno con el menor promedio.
  - Número de Estudiantes de la Escuela de Ingeniería de Escuela de Ingeniería de Sistemas en el semestre 2010-I y 2010-II aprobaron la nota de investigación.
  - Porcentaje de estudiantes que desaprobaron las notas de control de lectura y de práctica de la Escuela de Ingeniería de Ambiental en el semestre 2009-II.
3. Construir un programa que permita calcular la planilla de la Empresa Boxers S.A. que tiene N empleados. La empresa cuenta con la siguiente información: Nombres del empleado y horas trabajadas, y el pago por hora y las bonificaciones está de acuerdo a las siguientes tablas:

Puesto	Pago Hora
Capataz	55.00
Maestro	42.00
Operario	25.00

Modalidad	% de Bonificación
Estable	25%
Contratado	10%

Además al empleado se le puede hacer a lo mucho 3 descuentos por planilla, y de acuerdo a la siguiente tabla:

Descuento	% de Descuento
AFP	3.5%
SEGURO	3.0%
FONAVI	2.5%

Determinar el monto total que la empresa gastara en pagar a sus empleados y cuanto pagara según el Puesto.

4. Una empresa tiene N empleados para cada uno de los cuales registra los siguientes datos: Nombre, Sueldo base, Ventas realizadas, Estado civil, Número de Hijos y Sueldo Neto.  
El programa debe calcular para cada empleado el sueldo neto mediante la relación siguiente:

$$\text{Sueldo Neto} = \text{Sueldo Base} + \text{Comisión por Ventas} - \text{Descuento por Impuesto} - \text{Descuento por Seguro.}$$

Donde: la comisión por ventas es el 5% de las ventas realizadas

El descuento por seguro se obtiene como sigue:

- Si el empleado es soltero el descuento es de S/. 100.
- Si el empleado es casado sin hijos el descuento es de S/. 120.
- Si el empleado es casado con hijos el descuento es de S/. 50 + S/. 70 por cada hijo.

Suponga que llamamos TA a la suma del Sueldo base + comisión por ventas, entonces el Descuento por el Impuesto (DI) se obtiene como sigue:

Rango del TA	Monto de DI
0 a 1500	0
1500 a 2300	3% del TA
2301 a 3000	4% del TA
3001 a mas	6% del TA

El programa debe mostrar el monto total que la empresa debe pagar por concepto de sueldos, el monto total por comisiones de ventas, el monto total de los descuentos por impuesto y por seguro.

5. Construya un programa que guarde la siguiente información de los registros de llamada: Tipo de Plan (A o B), tipo de llamada (local, celular A, celular B y celular C), horario de llamada (mañana-tarde, noche, madrugada), minutos de llamada, costo por minuto, costo por llamada (calculado de la multiplicación de minutos de llamada por costo por minuto).

Las tarifas de las llamadas se dan de la siguiente manera:

TIPO PLAN	TIPO DE LLAMADA (Costo x minuto)			
	Local	Celular A	Celular B	Celular C
Tipo A	0.20	0.40	0.45	0.70
Tipo B	0.15	0.30	0.35	0.60

Por otro lado el costo por minuto se reducirá en un 35% para llamadas locales y 25% para celulares si es que se da en el horario de la madrugada

Su programa deberá mostrar los siguientes datos:

- Número de llamadas con más de 5 minutos y menos de 20 en horario de la noche.
- Costos totales de llamada según Tipo de Plan.
- Total de minutos realizados en el turno de la noche y madrugada a teléfonos locales con costo de llamada superiores a 10 soles.
- En que horario y tipo de Plan se realizo la llamada con mayor costo de llamada
- Total de llamadas en minutos realizadas al celular C en horario de mañana-tarde.

6. Una empresa guarda la siguiente información de sus empleados: Nombres, Apellido Paterno, Apellido Materno, Área al que pertenece (Planificación, Ventas y Marketing), Tiempo de Servicio (en años), Condición (Contratado o Estable), Sueldo Base, Asignación de Movilidad, N° de Hijos, descuento de AFP (17.5% del sueldo base) y Sueldo Final.

Para la asignación movilidad es el porcentaje del sueldo base y se asigna de acuerdo al área al que pertenece y la condición del empleado tal y como se muestra en la siguiente tabla:

Condición	Área		
	Planificación	Ventas	Marketing
Contratado	7%	8%	12%
Estable	9%	12%	16%

Además por cada hijo del empleado la empresa le otorga 55 soles. El sueldo final se calcula de la siguiente fórmula:

$$\text{Sueldo Final} = \text{Sueldo Base} + \text{Asignación de Movilidad} + \text{Nº de Hijos} * 55 - 17.5\% \text{ del Sueldo Base}$$

Construir un programa que pueda proporcionar la siguiente información:

- Nombre del empleado que tenga el menor sueldo final que sea del área de Ventas que tengan entre 2 y 10 años de Servicio.
- Promedio de Salarios de los empleados contratados del área de Marketing.
- Número de empleados estables que ganan más de S/.2500 y menos de S/ 3500 con más de 5 años de servicio y con más de 3 hijos.
- Porcentaje de empleados estables del área de Marketing con sueldo menores a S/. 1200 y asignación de movilidad superior a 60 soles
- Área que tiene el empleado estable con el mayor tiempo de servicio sin hijos.

II. Analice los siguientes segmentos de código y describa la acción que realiza dentro del programa.

1. ¿Qué acción realiza este primer segmento?

Segmento de Código	Descripción
<pre>import java.io.Serializable; public class Libro implements Serializable {     private String codigo, nombre, tipo, clase;     private int anio, num_pag;     private double costo;      public Libro(String codigo, String nombre, String     tipo, String clase, int anio, int num_pag, double costo){         this.codigo=codigo;         this.nombre = nombre;         this.tipo = tipo;         this.clase = clase;         this.anio=anio;         this.num_pag=num_pag;         this.costo=costo;     } }</pre>	

2. ¿Qué acción realiza este segundo segmento?

Segmento de Código	Descripción
<pre>import java.io.Serializable; import java.util.ArrayList; public class ArregloLibros implements Serializable {     private ArrayList &lt;Libro&gt; a;     public ArregloLibros(){         a = new ArrayList();     } }</pre>	

3. ¿Qué acción realiza este tercer segmento?

Segmento de Código	Descripción
<pre>public class Arreglo_Objetos extends javax.swing.JFrame {     Libro objLibro;     ArregloLibros objArreglo;     objArreglo = new ArregloLibros();</pre>	

4. ¿Qué acción realiza este cuarto segmento?

Segmento de Código	Descripción
<pre>FileInputStream fis = new FileInputStream("Libros.bin"); ObjectInputStream in = new ObjectInputStream(fis); if (in != null){     objArreglo = (ArregloLibros)in.readObject();     in.close(); }</pre>	

5. ¿Qué acción realiza este quinto segmento?

Segmento de Código	Descripción
<pre>FileOutputStream fos = new FileOutputStream("Libros.bin"); ObjectOutputStream out = new ObjectOutputStream(fos); if (out != null){     out.writeObject(objArreglo);     out.close(); }</pre>	

6. ¿Qué acción realiza este sexto segmento?

Segmento de Código	Descripción
<pre>String cod = jtxtCodigo.getText().toUpperCase(); int p = objArreglo.busca(cod); if (p== -1) { mensaje("Codigo no existe");   limpiar_entradas(); } else{     objLibro = objArreglo.getLibro(p);     String nombre = objLibro.getNombre();     String tipo = objLibro.getTipo();     String clase= objLibro.getClase();</pre>	

7. ¿Qué acción realiza este séptimo segmento?

Segmento de Código	Descripción
<pre>objLibro = new Libro(cod,nom,ti,cla,an,np,c,por);  if (p== -1)     objArreglo.agrega(objLibro); else      objArreglo.reemplaza(p,objLibro); limpiar_entradas(); grabar(); actualizar_tabla();</pre>	

8. ¿Qué acción realiza este octavo segmento?

Segmento de Código	Descripción
<pre> consulta();  int p= objArreglo.busca(jtxtCodigo.getText().toUpperCase()); if (p!=-1) {     int r = JOptionPane.showConfirmDialog(this, "Esta seguro de eliminar a éste registro ?", "Reponder", 0);     if (r==0) {         objArreglo.elimina(p);         limpiar_entradas();         grabar();          actualizar_tabla();     }     resumen();     jtxtCodigo.requestFocus(); </pre>	

III. Relacione los códigos con su aplicación correspondiente:

import java.io.Serializable;	Permite escribir el estado de un objeto en un flujo de bytes. Además contiene referencias a otros objetos que a su vez contienen referencias a otros objetos.
public class ArregloLibros implements Serializable	Escribe un objeto obj al flujo de datos que realiza la llamada
private ArrayList <Libro> a;	Lee un objeto del flujo de datos que realiza la llamada.
a.add(nuevo);	Inserta un objeto en la lista haciendo que este se desplace hacia arriba para evitar sobrescribir algún elemento.
in.readObject();	Admite matrices dinámicas que pueden crecer o disminuir de tamaño según sea necesario a través de objetos.
out.writeObject(objArreglo);	Hace posible que se pueda utilizar métodos propios de una clase propia de JAVA que no ha sido desarrollada o implementada aún.

IV. Defina la acción que realiza el código con el archivo "Libros.bin"

- ❖ FileInputStream fis = new FileInputStream("Libros.bin");  
ObjectInputStream in = new ObjectInputStream(fis);: \_\_\_\_\_
- ❖ FileOutputStream fos = new FileOutputStream("Libros.bin");  
ObjectOutputStream out = new ObjectOutputStream(fos);: \_\_\_\_\_
- ❖ String cod = objArreglo.getLibro(i).getCodigo();: \_\_\_\_\_

- ❖ `if (objArreglo.busca(objLibro.getCodigo())!=-1)`\_\_\_\_\_
- \_\_\_\_\_
- ❖ `objArreglo.agrega(objLibro);`\_\_\_\_\_
- \_\_\_\_\_
- ❖ `public void agrega(Libro nuevo){ a.add(nuevo); }`\_\_\_\_\_
- \_\_\_\_\_
- ❖ `public void elimina(int i){ a.remove(i); }`\_\_\_\_\_
- \_\_\_\_\_
- ❖ `public int busca(String codigo){`  
`for (int i=0; i<numeroLibros(); i++){`  
`if (codigo.equalsIgnoreCase(getLibro(i).getCodigo()))`  
`return i; // retorna indice`  
`}`  
`return -1; // significa que no lo encontré`  
`}` : \_\_\_\_\_
- \_\_\_\_\_
- ❖ `import java.io.Serializable;` \_\_\_\_\_
- \_\_\_\_\_
- ❖ `public class Libro implements Serializable:`\_\_\_\_\_
- \_\_\_\_\_