

## **GUÍA DE LABORATORIO 3**

### **“Java: Paquetes útiles, interfaces y excepciones”**

## **LABORATORIO**

### **Objetivos:**

- Utilizar las Clases más principales que vienen en los paquetes java.lang y java.util.

### **Equipos, Materiales, Programas y Recursos:**

- PC con Sistema Operativo con soporte a Java.
- IDE para Java

### **Introducción:**

En la presente sesión se detalla las clases String y StringBuffer. Se detallan las clases de los paquetes java.lang y java.util. Finalmente, se ve la implementación de Clases abstractas y finales.

### **Seguridad:**

- Ubicar maletines y/o mochilas en el gabinete al final de aula de laboratorio.
- No ingresar con líquidos ni comida al aula de laboratorio.
- Al culminar la sesión de laboratorio, apagar correctamente la computadora y el monitor.

### **Preparación:**

Durante el desarrollo de los temas de clase se tendrán ejercicios explicativos en cada uno de los puntos, ello le dará a la sesión una interacción de la teoría y la parte práctica, ya que en todo el momento el alumno podrá comprobar en su propia PC, todos los ítems del manual.

### **Procedimiento y Resultados:**

**Ejercicio: Centros de Distribución**

Una empresa de venta de productos está formada por varios centros de distribución en toda la ciudad de Lima y maneja la información siguiente:

- Los centros de distribución pueden ser farmacias y boticas.
- De las boticas y farmacias se conoce el código, RUC, razón social, dirección y teléfonos.
- Las boticas están abiertas las 24 horas del día a diferencia de las farmacias que tienen horas de apertura y cierre.
- Los productos que vende la empresa son artículos de cuidado personal y medicamentos.
- De los medicamentos y artículos de cuidado personal se conoce el código, descripción, unidad de medida y precio.
- De los medicamentos se conoce adicionalmente la posología y contraindicaciones.
- Solo se tiene definido un descuento para los artículos de cuidado personal.
- Las boticas pueden vender cualquier tipo de producto, las farmacias solo medicamentos.
- Los productos pueden ser vendidos en varios centros de distribución.
- Cada centro de distribución está ubicado en un distrito de la capital.
- Por cada medicamento comprado en una farmacia el cliente debe presentar una receta médica que muestre la fecha de emisión, fecha de vencimiento, cantidad recetada del medicamento y nombre del médico que la recetó.
- De cada cliente se conoce el número de dni, nombre, teléfono, dirección y distrito de residencia.
- De cada médico que elabora una receta se conoce el número de licencia, nombre y año de graduado.

**EJERCICIO toString() y equals()**

```
package laboratorio3;

public class Socio {

    private int codigo;
    private String nombre;
    private String direccion;

    // Generar SET y GET

    public Socio(int c, String n, String d) {
        this.codigo = c;
        this.nombre = n;
        this.direccion = d;
    }

    public String toString(){
        String cadena = "Codigo : " + this.codigo;
        cadena += "nombre: " + this.nombre;
        cadena += "direccion: " + this.direccion;
        return cadena;
    }

    public boolean equals(Object o){
        Socio s = (Socio)o;
        if(this.codigo == s.codigo &&
            this.nombre.equals(s.nombre) &&
            this.direccion.equals(s.direccion) ){
            return true;
        } else {
            return false;
        }
    }
}
```

```
package laboratorio3;

public class Test {

    public static void main(String[] arg){

        Socio s1 = new Socio(302, "David", "Tecsups");
        Socio s2= new Socio(302, "David", "Tecsups");

        if (s1 == s2 ) {
            System.out.println("Comprobando referencia");
        }

        if (s1.equals(s2)){
            System.out.println("Comprobando el valor");
        }

    }
}
```



**EJERCICIO CONCATENANDO STRING**

```
package laboratorio3;

public class StringTest1 {

    public static void main(String[] args) {

        String nombre = "James";
        String apellido = "Gosling";

        if(nombre.equals("James")) {
            System.out.println(nombre+" "+apellido);
        }

        String completo = nombre.concat(" "+apellido);

        if(nombre.equals("James")) {
            System.out.println(completo);
        }
    }
}
```

**EJERCICIO EXTRACCIÓN STRING**

```
package laboratorio3;

public class StringTest2 {

    public static void main(String[] args) {

        String texto = "Tecsup";

        System.out.println("Número de Caracteres: "+texto.length()+"\n");

        for (int i=0; i<texto.length(); i++) {
            System.out.println(texto.substring(i,i+1)+"\n");
        }
    }
}
```

**EJERCICIO DE REEMPLAZO STRING**

```
package laboratorio3;

public class StringTest3 {
    public static void main(String[] args) {
        String texto = "java permite crear java applets, javabeans y java servlets";
        System.out.println(texto.replaceAll("java", "JAVA"));
    }
}
```

## EJERCICIO STRINGBUFFER

```
package laboratorio3;

public class StringBufferTest {

    public static void main(String[] args) {
        StringBuffer sb = new StringBuffer("Java Basico Tecsup Lima");

        sb.insert(11," en");
        sb.delete(21,sb.length());

        System.out.println(sb);
    }
}
```

## EJERCICIO NUMBERFORMAT

```
package laboratorio3;
import java.text.NumberFormat;
import java.util.Locale;

public class NumberFormatTest1 {
    public static void main(String[] args) {
        // Establecer el Locale como US para usar el punto como
        // separador decimal.
        NumberFormat nf = NumberFormat.getInstance(Locale.US);

        // Agrupar
        nf.setGroupingUsed(true);
        System.out.println(nf.format(10000000.0045451));

        // No agrupar
        nf.setGroupingUsed(false);
        System.out.println(nf.format(10000000.0045451));

        System.out.println("Con 3 enteros mínimo");
        nf.setMinimumIntegerDigits(3);
        System.out.println(nf.format(12));

        System.out.println("Con 5 enteros máximo");
        nf.setMaximumIntegerDigits(5);
        System.out.println(nf.format(123456));

        System.out.println("Con 4 enteros máximo en los decimales");
        nf.setMaximumFractionDigits(4);
        System.out.println(nf.format(123.981454));

        System.out.println("Con 2 enteros mínimo en los decimales");
        nf.setMinimumFractionDigits(2);
        System.out.println(nf.format(45.2));
    }
}
```

**EJERCICIO FORMATO MONEDA**

```
package laboratorio3;

import java.util.*;
import java.text.*;

public class NumberFormatTest2 {

    public static void main(String[] args) {

        // Formato de moneda
        NumberFormat nfe = NumberFormat.getCurrencyInstance(Locale.GERMANY);
        nfe.setGroupingUsed(false);
        NumberFormat nfd = NumberFormat.getCurrencyInstance(Locale.US);
        nfd.setGroupingUsed(false);
        Locale lp = new Locale("es", "PE");
        NumberFormat nfp = NumberFormat.getCurrencyInstance(lp);
        nfp.setGroupingUsed(false);

        System.out.println("Moneda Euro :"+nfe.format(150));
        System.out.println("Moneda Dolar :"+nfd.format(150));
        System.out.println("Moneda Soles :"+nfp.format(150));
    }
}
```



**FORMATO DE FECHAS**

```
package laboratorio3;

import java.util.Calendar;
import java.util.GregorianCalendar;

public class Fechas1 {

    public static void main(String args[]){

        GregorianCalendar ahora = new GregorianCalendar();

        // Crear una fecha
        //Calendar ahora = new GregorianCalendar(2007,2,2);

        System.out.println("Fecha : " + ahora.getTime());
        System.out.println("Año: " + ahora.get(Calendar.YEAR));
        System.out.println("Mes: " + ahora.get(Calendar.MONTH));
        System.out.println("Día del mes: " + ahora.get(Calendar.DAY_OF_MONTH));
        //
        System.out.println("Hora: " + ahora.get(Calendar.HOUR));
        System.out.println("Minuto: " + ahora.get(Calendar.MINUTE));
        System.out.println("Segundo: " + ahora.get(Calendar.SECOND));
        System.out.println("Milisegundo: " + ahora.get(Calendar.MILLISECOND));
        //
        System.out.println("Primer día de la semana : " + ahora.getFirstDayOfWeek());
        System.out.println("Semana          del          mes:          "          +
ahora.get(Calendar.WEEK_OF_MONTH));
        System.out.println("Semana          del          año:          "          +
ahora.get(Calendar.WEEK_OF_YEAR));
        System.out.println("Día del año: " + ahora.get(Calendar.DAY_OF_YEAR));
    }
}
```

```
package laboratorio3;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;

public class Fechas2 {

    public static void main(String[] args) {

        Date now = new Date();
        System.out.println(now.getTime());

        DateFormat df = DateFormat.getDateInstance();
        DateFormat df1 = DateFormat.getDateInstance(DateFormat.SHORT);
        DateFormat df2 = DateFormat.getDateInstance(DateFormat.MEDIUM);
        DateFormat df3 = DateFormat.getDateInstance(DateFormat.LONG);
        DateFormat df4 = DateFormat.getDateInstance(DateFormat.FULL);
        SimpleDateFormat df5 = new SimpleDateFormat("EEE, d MMM yyyy HH:mm:ss
Z");
    }
}
```

```
//DateFormat df4 = DateFormat.getDateInstance(DateFormat.FULL, Locale.US);

String s = df.format(now);
String s1 = df1.format(now);
String s2 = df2.format(now);
String s3 = df3.format(now);
String s4 = df4.format(now);
String s5 = df5.format(now);

System.out.println("(Default) Hoy es " + s);
System.out.println("(SHORT) Hoy es " + s1);
System.out.println("(MEDIUM) Hoy es " + s2);
System.out.println("(LONG) Hoy es " + s3);
System.out.println("(FULL) Hoy es " + s4);
System.out.println("(CUSTOM) Hoy es " + s5);
}
}
```

```
package laboratorio3;

import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;

public class Fechas3 {

    public static void main(String args[]){

        Calendar c1 = new GregorianCalendar(2011,9,1);

        Date d1 = c1.getTime();

        Calendar c2 = new GregorianCalendar(2011,9,3);
        Date d2 = c2.getTime();

        if (d1.after(d2)) {
            System.out.println("d1 es después que d2");
        }
        if (d1.before(d2)) {
            System.out.println("d1 es antes que d2");
        }

    }
}
```

## INTERFASES

```
package laboratorio4;

public interface Capitalizable {
    public String cambiaMayuscula();
    public String cambiaMinuscula();
}
```

```
package laboratorio4;

public class Oracion implements Capitalizable {

    String texto;

    Oracion(String n){
        texto=n;
    }

    public String agregaPalabra(String t) {
        texto=texto.concat(" "+t);
        return texto;
    }

    //metodos de Interfase
    public String cambiaMayuscula(){
        return texto.toUpperCase();
    }

    public String cambiaMinuscula(){
        return texto.toLowerCase();
    }
}
```

```
package laboratorio4;

public class Palabra implements Capitalizable{

    String texto;

    Palabra(String n){
        texto=n.replaceAll(" ","");
    }

    //metodos de Interfase
    public String cambiaMayuscula(){
        return texto.toUpperCase();
    }

    public String cambiaMinuscula(){
        return texto.toLowerCase();
    }
}
```

```
package laboratorio4;

public class Uso_interfase {

    public static void main(String[] args) {

        Capitalizable[] obj1 = new Capitalizable[3];

        obj1[0] = new Palabra("Programacion en Java");
        obj1[1] = new Palabra("Tecsup");
        obj1[2] = new Oracion("Curso Java Basico - Tecsup");

        for (int i=0; i<obj1.length;i++){
            System.out.println( "Texto " + i + ": " + obj1[i].cambiaMayuscula());
        }
    }
}
```

## EXCEPCIONES

### EJEMPLO DE EXCEPCION

```
package laboratorio4;

public class TestExcepcion {

    public static void main(String args[]){
        int i = 0, u = 0, k;
        k = i / u;
    }
}
```

### MANEJANDO EL ARITHMETICEXCEPTION

```
package laboratorio4;

public class TestExcepcion {

    public static void main(String args[]){
        int i = 0, u = 0, k;
        try {
            k = i / u;
        } catch(java.lang.ArithmeticException e){
            System.out.println("Hubo un error: " + e.toString());
        }
    }
}
```

### TEST DE EXCEPCIONES

```
package laboratorio4;

public class TestExcepcion2 {

    public static void main(String args[]){
        int x = (int)(Math.random() * 5);
        int y = (int)(Math.random() * 10);
        int [] z = new int[5];
        try {
            System.out.println("y/x gives " + (y/x));
            System.out.println("y is " + y + " z[y] is " + z[y]);
        } catch (ArithmeticException e) {
            System.out.println("Arithmetic problem " + e);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Subscript problem " + e);
        } finally {
            System.out.println("Terminando la ejecución del método");
        }
        System.out.println("Continúa la ejecución del método....");
    }
}
```

**EJERCICIO DE EXCEPCION**

```
package laboratorio4;

public class Manejo_errores {

    public static void main(String[] args) {

        int[] n=null; //= {11,22,33,44,55,66,77,88,99};

        try {
            for (int i=0; i<11;i++) {
                System.out.println( "Valor " + n[i]);
            }
            //ArrayIndexOutOfBoundsException
            //(indice de arreglo fuera de rango)
            //NullPointerException
            //(valor nulo)
        } catch (Exception e) {
            System.err.println(e);
            e.printStackTrace();
        }
    }
}
```

**EJERCICIO FINALLY**

```
package laboratorio4;

public class Manejo_errores {
    public static void main(String[] args) {
        int[] n = {11,22,33,44,55,66,77,88,99};
        try {
            for (int i=0; i<n.length;i++) {
                System.out.println( "Valor " + n[i]);
            }
        } catch (Exception e) {
            System.err.println(e);
            e.printStackTrace();
        } finally {
            System.out.println( "Aplicacion Terminada ");
        }
    }
}
```

## DEFINIENDO EXCEPCIONES PERSONALIZADAS

### **SaldoInsuficienteException.java**

```
package laboratorio4.excepciones;

public class SaldoInsuficienteException extends Exception {

    public double saldo, cuantoDesea;

    public SaldoInsuficienteException(double saldo, double cuantoDesea) {
        super("En su cuenta tiene " + saldo + " soles. No puede prestarse " +
        cuantoDesea + " soles.");
    }

    this.saldo = saldo;
    this.cuantoDesea = cuantoDesea;
}
```

### **Cuenta.java**

```
package laboratorio4.excepciones;

public class Cuenta {

    private long idCuenta;

    public Cuenta(long idCuenta) {
        this.idCuenta = idCuenta;
    }

    public double retirarDeCajero(double cuantoDesea) throws
    SaldoInsuficienteException {
        double saldoActual = 5000;
        if(saldoActual < cuantoDesea){
            throw new SaldoInsuficienteException(saldoActual, cuantoDesea);
        } else {
            return saldoActual - cuantoDesea;
        }
    }
}
```

### **ProgramaCuentas.java**

```
package laboratorio4.excepciones;

public class ProgramaCuentas {

    public static void main(String args[]){
        try {
            Cuenta c = new Cuenta(001);
            double nuevoSaldo = c.retirarDeCajero(7000);
            System.out.println("OPERACION REALIZADA, su saldo es: " + nuevoSaldo);
        } catch (SaldoInsuficienteException e){
            System.err.println(e.toString());
        }
    }
}
```

```
}
```



**Conclusiones:**

En la presente sesión, se implementó las jerarquías de clases usando la herencia. Además, se detalló a la clase String y StringBuffer. Luego se detallaron las Clases principales de los paquetes java.lang y java.util. Finalmente, se vieron los detalles de la implementación de las clases abstractas.