

ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO

ESCUELA DE INGENIERIA EN SISTEMAS

TAREA No. 02

ASIGNATURA: Aplicaciones Empresariales en JAVA

DOCENTE: Ing. Paúl Paguay

SEMESTRE: Noveno

NOMBRE: Holger Morales (4116)

FECHA: Martes, 5 de Abril del 2010

1. INTRODUCCIÓN

JDK

Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en java. Puede instalarse en una computadora local o en una unidad de red.

Los programas más importantes que se incluyen son:

- **Appletviewer:** es un visor de applet para generar sus vistas previas, ya que un applet carece de método main y no se puede ejecutar con el programa java.
- **Javac:** es el compilador de JAVA.
- **java:** es el intérprete de JAVA.
- **javadoc:** genera la documentación de las clases java de un programa.

NETBEANS

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos.

INTRODUCCION AL LENGUAJE JAVA

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

Actualmente la versión de Netbeans es 6.8.

2. OBJETIVOS

- Conocer el funcionamiento de la plataforma java y su lenguaje.
- Recapitular los conocimientos de POO.
- Familiarizar con la herramienta Netbeans versión 6.5.

3. DESARROLLO

- a) **Realice un programa que permita imprimir de forma inversa una cadena de caracteres.**

CÓDIGO:

```
public void imprimirCadenaInversa(){
    Scanner lector=new Scanner(System.in);
    System.out.println("Ingrese una cadena de caracteres:");
    String cadena=lector.nextLine();
    System.out.println("La cadena de caracteres invertida es:");
    for(int i=cadena.length()-1;i>-1;i--)
        System.out.print(cadena.charAt(i));
}
```

OUTPUT:

INTRODUCCION AL LENGUAJE JAVA

```
Ingrese una cadena de caracteres:  
123  
La cadena de caracteres invertida es:  
321
```

- b) Desarrolle un programa que permita la validación de usuario y contraseña. A la cuarta vez de intento fallido, el usuario recibirá un mensaje de error.

CODIGO

```
public void ValidarUsuario(){  
    final String user="hmorales";  
    final String password="123456";  
    int contadorIntentos=0;  
    boolean sw=false;//para controlar la autentificacion  
    Scanner lector=new Scanner(System.in);  
    while(contadorIntentos<4&&! sw){  
        System.out.print("Usuario:");  
        String usuario=lector.next();  
        System.out.print("Contraseña:");  
        String contraseña=lector.next();  
        if(usuario.equalsIgnoreCase(user)&&password.equalsIgnoreCase(contraseña)){  
            sw=true;  
            System.out.println("Bienvenido, ud se autentificado como  
Administrador"  
        }  
        else  
            System.out.println("Datos incorrectos");  
            contadorIntentos++;  
        }  
        if(contadorIntentos==4){  
            System.out.println("Bloqueado, Intente mas tarde...");  
        }  
    }  
}
```

OUTPUT:

A.

INTRODUCCION AL LENGUAJE JAVA

```
Usuario:hmorales
Contraseña:123456
Bienvenido, ud se autentificado como Administrador
```

B.

```
Usuario:hmorales
Contraseña:1
Datos incorrectos
Usuario:1
Contraseña:123456
Datos incorrectos
Usuario:hm
Contraseña:hm
Datos incorrectos
Usuario:h
Contraseña:h
Datos incorrectos
Bloqueado, Intente mas tarde...
```

4. Realice la lectura de una matriz que corresponde a las edades de los estudiantes y otra que corresponde a los nombres completos. Realice la ordenación por edad, conservando la relación nombres y edades. Los datos deben ser leídos por consola (Clase Scanner).

CODIGO

```
public void OrdenarMatriz(){
    Scanner lector=new Scanner(System.in);
    System.out.println("----DATOS ESTUDIANTE---");
    System.out.println("Numero Estudiantes:");
    int numeroEstudiantes=lector.nextInt();
    byte[]matrizEdadEstudiantes=new byte[numeroEstudiantes];
    String[]matrizNombresEstudiantes=new String [numeroEstudiantes];
    //leer matriz
    for(int i=0;i<numeroEstudiantes;i++){
        System.out.print("Nombre:");
        String nombre=lector.next();
        System.out.print("Edad:");
        byte edad=lector.nextByte();
        matrizEdadEstudiantes[i]=edad;
        matrizNombresEstudiantes[i]=nombre;
    }
}
```

INTRODUCCION AL LENGUAJE JAVA

```
//ordenar matriz ascendente
byte edadAux=0;
String nombreAux="";
for(int i=0;i<numeroEstudiantes-1;i++){
    for(int j=i;j<numeroEstudiantes;j++){
        if(matrizEdadEstudiantes[i]>matrizEdadEstudiantes[j]){
            edadAux=matrizEdadEstudiantes[i];
            matrizEdadEstudiantes[i]=matrizEdadEstudiantes[j];
            matrizEdadEstudiantes[j]=edadAux;
            nombreAux=matrizNombresEstudiantes[i];
            matrizNombresEstudiantes[i]=matrizNombresEstudiantes[j];
            matrizNombresEstudiantes[j]=nombreAux;
        }
    }
}
//imprimir
System.out.println("-----ESTUDIANTES-----");
for(int i=0;i<numeroEstudiantes;i++){
    System.out.println("Nombre:      "+matrizNombresEstudiantes[i]+"   Edad:
"+matrizEdadEstudiantes[i]);
}
}
```

OUTPUT

```
-----DATOS ESTUDIANTE-----
Numero Estudiantes:
3
Nombre: HOLGER
Edad: 18
Nombre: NEPTALI
Edad: 15
Nombre: JOSE
Edad: 10
-----ESTUDIANTES-----
Nombre: JOSE Edad: 10
Nombre: NEPTALI Edad: 15
Nombre: HOLGER Edad: 18
```

5. Una clase llamada Estudiante posee tres propiedades: Nombre, Apellido y Edad. Crear una matriz de 10 objetos, cargue las propiedades y recorra la matriz mostrando todas sus propiedades.

CÓDIGO:

INTRODUCCION AL LENGUAJE JAVA

```
public class Estudiante {
    private String nombre;
    private String apellido;
    private byte edad;

    public Estudiante(){
        nombre="";
        apellido="";
        edad=-1;
    }

    public void leerEstudiante(){
        Scanner oscan=new Scanner(System.in);
        System.out.println("Ingrese el nombre del estudiante:");
        nombre=oscan.nextLine();
        System.out.println("Ingrese el apellido del estudiante:");
        apellido=oscan.next();
        System.out.println("Ingrese la edad del estudiante:");
        edad=oscan.nextByte();
    }

    public void mostrarEstudiante(){
        System.out.println("El nombre del estudiante es: "+nombre);
        System.out.println("El apellido del estudiante es: "+apellido);
        System.out.println("La edad del estudiante es: "+edad);
    }
    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public String getApellido() {
        return apellido;
    }
    public void setApellido(String apellido) {
```

INTRODUCCION AL LENGUAJE JAVA

```
        this.apellido = apellido;
    }
    public byte getEdad() {
        return edad;
    }
    public void setEdad(byte edad) {
        this.edad = edad;
    }
}
```

OUTPUT:

```
-----
6
Ingrese el nombre del estudiante:
Ingrese el apellido del estudiante:
7
7
Ingrese la edad del estudiante:
Ingrese el nombre del estudiante:
7
8
Ingrese el apellido del estudiante:
Ingrese la edad del estudiante:
8
8
Ingrese el nombre del estudiante:
9
Ingrese el apellido del estudiante:
9
Ingrese la edad del estudiante:
9
Ingrese el nombre del estudiante:
10
Ingrese el apellido del estudiante:
10
Ingrese la edad del estudiante:
10
El nombre del estudiante es: 1
El apellido del estudiante es: 1
La edad del estudiante es: 1
El nombre del estudiante es: 2
El apellido del estudiante es: 2
La edad del estudiante es: 2
El nombre del estudiante es: 3
El apellido del estudiante es: 3
La edad del estudiante es: 3
El nombre del estudiante es: 4
El apellido del estudiante es: 4
La edad del estudiante es: 4
El nombre del estudiante es: 5
```

INTRODUCCION AL LENGUAJE JAVA

6. Basado en la clase anterior Estudiante, implemente un método llamado verificarIgual que imprima si dos objetos son iguales. Esta condición se da si las tres propiedades de dos objetos son iguales.

CÓDIGO:

```
public boolean verificarIgual(String nomb,String apIl, byte edad){
    boolean respuesta=false;
    respuesta=this.nombre.equals(nombre)&&(this.apellido.equals(apIl))&&(this.edad==edad);
    return respuesta;
}

public static void main(String[] args) {
    Estudiante[] arregloEstudiante=new Estudiante[3];
    int i=0;
    for (Estudiante estudiante : arregloEstudiante) {
        estudiante=new Estudiante();
        estudiante.leerEstudiante();
        arregloEstudiante[i]=estudiante;
        i++;
    }
    Estudiante estu1;
    Estudiante estu2;

    for (int j=0;j<arregloEstudiante.length -1;j++) {
        estu1=arregloEstudiante[j];
        for(i=j+1;i<arregloEstudiante.length;i++){
            estu2=arregloEstudiante[i];

            if(estu1.verificarIgual(estu2.getNombre(), estu2.getApellido(),estu2.getEdad())){
                System.out.println();
                System.out.println("Los estudiante son iguales: ");
                System.out.println();
                estu1.mostrarEstudiante();
            }
        }
    }
}
```


INTRODUCCION AL LENGUAJE JAVA

OUTPUT:

```
run:
Ingrese el nombre del estudiante:
1
Ingrese el apellido del estudiante:
1
Ingrese la edad del estudiante:
1
Ingrese el nombre del estudiante:
2
Ingrese el apellido del estudiante:
2
Ingrese la edad del estudiante:
2
Ingrese el nombre del estudiante:
2
Ingrese el apellido del estudiante:
2
Ingrese la edad del estudiante:
2
Los estudiante son iguales:
El nombre del estudiante es: 2
El apellido del estudiante es: 2
La edad del estudiante es: 2
BUILD SUCCESSFUL (total time: 14 seconds)
```

7. Implemente un controlador de creación de instancias de una clase. Puede tomar la clase Estudiante como base.

CÓDIGO:

```
public class Controlador {
    private byte numero_instancias_max;
    private byte numero_instancias_actual;
    public Controlador(byte num_clases){
        numero_instancias_max=num_clases;
        numero_instancias_actual=1;
    };

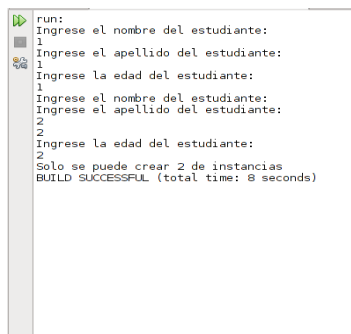
    public Estudiante instanciaEstudiante(){
        if(numero_instancias_actual<=numero_instancias_max){
            numero_instancias_actual++;
            return new Estudiante();
        }
    }
}
```

INTRODUCCION AL LENGUAJE JAVA

```
        else
            System.out.println("Solo se puede crear "+numero_instancias_max+" de
instancias");
        return null;
    }
}

public static void main(String[] args) {
    Estudiante[] arregloEstudiante=new Estudiante[4];
    Controlador objControlador=new Controlador((byte)2);
    int i=0;
    try{
        for (Estudiante estudiante : arregloEstudiante) {
            estudiante=objControlador.instanciaEstudiante();
            estudiante.leerEstudiante();
            arregloEstudiante[i]=estudiante;
            i++;
        }
    }catch(Exception e){}
}
```

OTUPUT:



```
run:
Ingrese el nombre del estudiante:
1
Ingrese el apellido del estudiante:
1
Ingrese la edad del estudiante:
1
Ingrese el nombre del estudiante:
Ingrese el apellido del estudiante:
2
2
Ingrese la edad del estudiante:
2
Solo se puede crear 2 de instancias
BUILD SUCCESSFUL (total time: 8 seconds)
```

8. RESULTADOS

Al realizar estos ejercicios se llego a obtener los siguientes resultados:

- Se logro refinar los conceptos básicos de POO como es: Herencia, polimorfismo, encapsulación; y representarlos en el lenguaje JAVA.

9. CONCLUSIONES

- El lenguaje JAVA no es 100% Orientado a objetos; debido a que todavía posee tipos de datos primitivos como el int, float, etc.
- La POO es uno de los paradigmas mas utilizados en el desarrollo de sistemas informáticos.

10. RECOMENDACIONES

- Más que una recomendación es una inquietud: Desarrollar una Aplicación Web con servicios web es aplicar el mismo paradigma de la Programación Orientada a Objetos u Orientada a servicios.