**SEMANA 14**

**GUIA DE LABORATORIO 14**

**MENÚ y Arreglo de Objetos Polimórficos**

#### Ejercicio 1:

#### Nuevo proyecto de nombre P14E01. Nuevo paquete de nombre p14e01.

#### Copie el contenido del paquete p13e01 de la clase anterior.

#### Diseñe una clase administradora de nombre ArregloFiguras que permita la administración de objetos de tipo Cuadrado y/o Circulo a la vez, utilizando un objeto de la clase ArrayList.

public class Arreglo**Figuras**{

protected ArrayList <**Figura**> a;

public **ArregloFiguras**(){

a = new ArrayList<**Figura**>();

}

// métodos de administración

public void agrega(Figura f){

a.add(f);

}

public Figura getFigura(int i){

Return a.get(i);

}

public int getN(){ return a.size();}

public void elimina(int i){

a.remove(i);

}

//... complete métodos adicionales

**}**

#### Diseñe una clase de interfaz de nombre PanelPrincipal donde cree un objeto de la clase ArregloFiguras, como atributo de la clase.

protected **ArregloFiguras** af = new **ArregloFiguras**();

**Programación del botón Nuevo Circulo:{**

Figura a = new Circulo(leeRadio());

lista(a);

af.agrega(a); // agrega un objeto Circulo

**}**

**Programación del botón Nuevo Cuadrado:{**

**Figura** b = new Cuadrado(leeLado());

lista(b);

af.agrega(b); // agrega un objeto Cuadrado

**}**

public void lista(**Figura** f){

imprime(f.**info**());

}

**Programación del botón Lista:{**

for(int i=0; i<af.getN(); i++){

Figura f=af.getFigura(i);

if (f **instanceof** Circulo)

imprime(“Circulo: “+f.info());

else

imprime(“Cuadrado: “+f.info());

}

**}**

#### La palabra reservada instanceof permite identificar el tipo personalizado de un objeto genérico, es decir, la clase a la que pertenece. Esto permite darle un nombre diferente antes de mostrar su información que es un método común.

#### Ejercicio 2

#### Nuevo proyecto de nombre P14E02. Nuevo paquete de nombre p14e02.

#### Copie el contenido del paquete p14e01 en éste paquete creado.

#### Aplique herencia en el ejemplo1 y desarrolle la clase ArchivoFiguras que permita grabar la información de las figuras en un archivo de texto.

**public class ArchivoFiguras extends ArregloFiguras{**

protected String nombre;

public ArchivoFiguras(String nombre){

this.nombre=nombre;

lee();

}

public void graba(){

try{

FileWriter fw = new FileWriter(nombre);

PrintWriter pw = new FilePrinter(fw);

for(int i=0; i<getN(); i++){

Figura f=getFigura(i);

if(f **instanceof** Circulo)

pw.println("1/"+

**((Circulo)(f))**.getRadio() );

else

pw.println("2/"+

**((Cuadrado)(f))**.getLado() );

}

pw.close();

}catch(Exception ex){ }

}

public void lee(){

try{

FileReader fr = new FileReader(nombre);

BufferedReader br = new BufferedReader(fr);

String linea = br.readLine();

while(linea!=null){

StringTokenizer st =new StringTokenizer(linea,"/");

int tipo=Integer.parseInt(st.nextToken());

**if(tipo==1){ // circulo**

double radio=Double.parseDouble(st.nextToken());

Figura a=new Circulo(radio);

agrega(a);

**}else{// cuadrado**

double lado=Double.parseDouble(st.nextToken());

Figura b=new Cuadrado(lado);

agrega(b);

}

linea=br.readLine();

}

br.close();

}catch(Exception ex){}

}

}

#### En un nuevo proyecto, haga las modificaciones necesarias en la clase PanelPrincipal para utilizar la clase ArchivoFiguras en lugar de la clase ArregloFiguras y programe la acción de los botones, así:

protected **ArchivoFiguras** af = new **ArchivoFiguras**();

**Programación del botón Nuevo Circulo:{**

Figura a = new Circulo(leeRadio());

lista(a);

af.agrega(a); // agrega un objeto Circulo

**af.graba(); // graba la información en el archivo de texto**

**}**

**Programación del botón Nuevo Cuadrado:{**

**Figura** b = new Cuadrado(leeLado());

lista(b);

af.agrega(b); // agrega un objeto Cuadrado

**af.graba(); // graba la información en el archivo de texto**

**}**

#### Ejercicio 3:

#### Nuevo proyecto de nombre P14E03. Nuevo paquete de nombre p14e03.

#### Copie el contenido del paquete p13e03 en éste paquete creado.

#### Luego, diseñe una clase administradora de nombre ArregloEmpleados que tenga como atributo un objeto de la clase ArrayList . Considere los métodos de administración necesarios para guardar objetos polimórficos de cualquiera de las clases hijas.

Finalmente, diseñe una clase de nombre **PanelPrincipal** con la interface necesaria para crear objetos de diferente tipo de empleado y administrarlos en un objeto de la clase **ArregloEmpleados**.

#### Ejercicio 4:

#### Nuevo proyecto de nombre P14E04. Nuevo paquete de nombre p14e04.

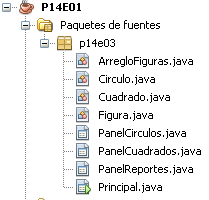
#### Copie el contenido del paquete p14e03 en éste paquete creado.

#### Aplique herencia en el ejemplo2 y desarrolle la clase ArchivoEmpleados que permita grabar la información de los empleados en un archivo de texto y en un nuevo proyecto, haga las modificaciones necesarias en la clase PanelPrincipal para utilizar la clase ArchivoEmpleados en lugar de la clase ArregloEmpleados y programe la acción de los botones correspondientes.

**MENUS Y PANELES**

#### Ejercicio 5:

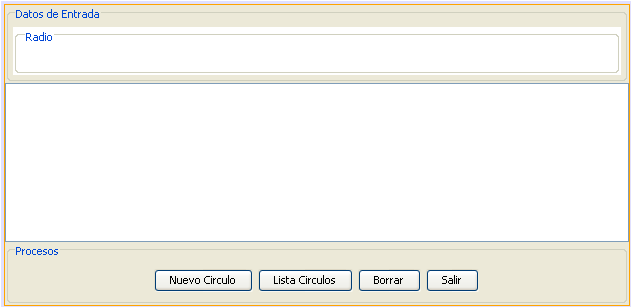
Nuevo proyecto de nombre **P14E05**, nuevo paquete de nombre **p14e05**. Abrimos el proyecto de la semana anterior y copiamos el contenido del paquete p14e02 en el nuevo paquete p14e05.



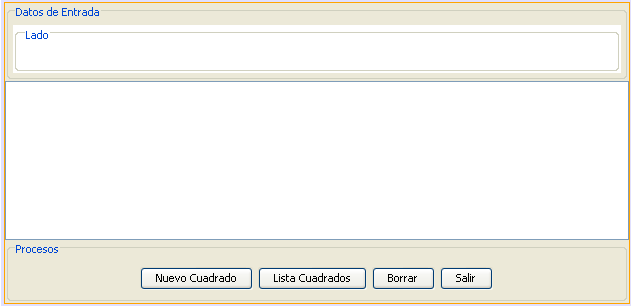
Cambiamos de nombre al paquete **PanelPrincipal** por el de **PanelCirculos**

Copiamos el **PanelCirculos** en el mismo paquete cambiándole de nombre por **PanelCuadrados**

Modificamos el diseño del **PanelCirculos** para que quede asi:



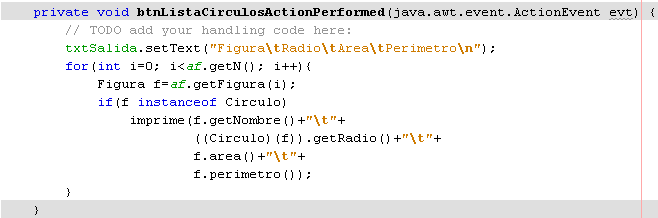
Modificamos el diseño del **PanelCuadrados** para que quede asi:



Entramos a la ficha **Fuente** del **PanelCirculos** y agregamos la palabra reservada **static** al objeto ArregloFiguras porque su información será **compartida** en otros paneles.



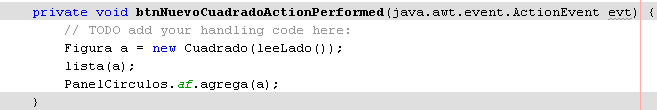
Regresamos al **diseño** y hacemos doble clic en el botón **Lista Circulos** para hacer el cambio siguiente:



Observe que hemos agregado el Radio del circulo en el título y para acceder desde el objeto genérico hemos aplicado la sintaxis del molde (down-custing):



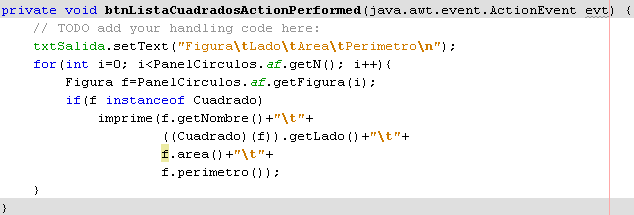
Entramos al PanelCuadrados y hacemos doble clic en el botón Nuevo Cuadrado para hacer el siguiente código:



Observe que al objeto **static** **af** se le accede directamente por la clase a la que pertenece. Así:



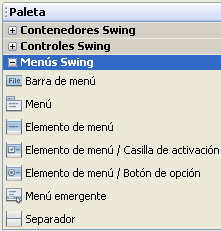
Regresamos al diseño y hacemos doble clic en el botón **Lista Cuadrados** y escribimos el siguiente código:



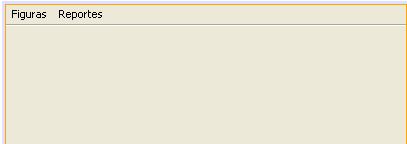
Observe que hemos agregado el lado del cuadrado en el título y para acceder desde el objeto genérico hemos aplicado la sintaxis del molde (down-custing):



Entremos al **Principal** que es el **Frame** donde vamos a colocar las opciones de menú. Observe la paleta en la parte superior derecha el acceso a los objetos menús swing:



Seleccionamos **Barra de menú** y lo colocamos en la parte **norte** del frame **Principal**. Con clic derecho editamos el texto para que quede así:



Ahora seleccionamos, de la palete, **elementos de menú** para la opción Figuras, lo colocamos y lo editamos para que quede así:



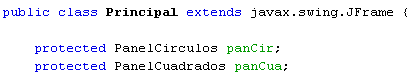
Las combinaciones de tecla se llaman **atajos** y para configurarlo solamente damos doble clic en el atajo y hacemos la configuración. Es opcional.

Igualmente, para la opción de Reportes:

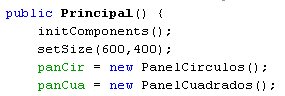


Antes de programar los elementos de menú, debemos darle nombres de variable adecuadas y con el siguiente estándar: primero el prefijo mni (menú ítem) luego el nombre conformado por toda la ruta del menú. Por ejm. **mniFigurasCirculos**, **mniFigurasCuadrados**, **mniFigurasSalir**.

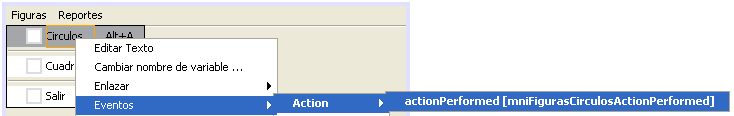
Vamos a la ficha **Fuente** del frame **Principal** para declarar los objetos para los paneles:



En el constructor lo creamos y definimos el tamaño del frame, así:



Regresamos al diseño y para programar un elemento de menú, lo seleccionamos, le damos clic derecho, seleccionamos **Eventos**, luego **Action**, finalmente **ActionPerformed**



y programamos lo siguiente:



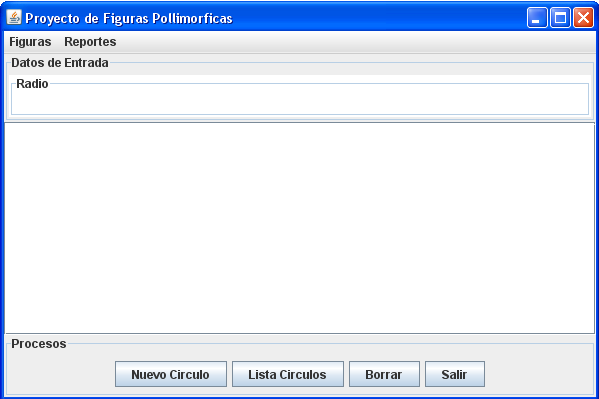
A la última línea póngale comentario porque eso lo va a completar usted.

Regrese al diseño y programe la opción de menú de Cuadrados.

Regrese al diseño y programe la opción de menú de salir, así:



Si no tiene marcas de errores, ya puede ejecutar su aplicación.



Complete todas las opciones, incluyendo la de reportes para lo cual debe desarrollar e implementar un PanelReportes, declararlo y utilizarlo en la clase Principal.