

Ejercicios de Colecciones - Herencia – Polimorfismo - Composición

1. Dada la siguiente **jerarquía** de clases:

Clase base: **Figura** (abstracta)

Atributos: no posee atributos.

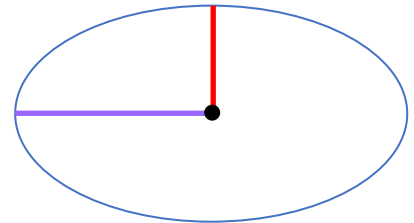
Métodos abstractos:

- getArea: double,
- getPerímetro: double,
- getInfoAtributos: String,
- getTipoFigura: String

Subclase: **Elipse** (es una **Figura**)

Atributos: **Semieje mayor**, **Semieje menor**.

Métodos: getSemiejeMayor, getSemiejeMenor.



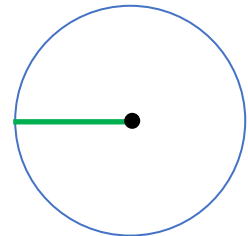
Subclase: **Círculo** (es una **Elipse**)

Atributos: no agrega atributos.

Métodos: getRadio.

NOTA: El Círculo es un caso especial de Elipse, donde

semieje mayor = semieje menor = **radio**



Subclase: **Rectángulo** (es una **Figura**)

Atributos: **base**, **altura**.

Métodos: getBase, getAltura.



Subclase: **Cuadrado** (es un **Rectángulo**)

Atributos: no agrega atributos.

Métodos: getLado.

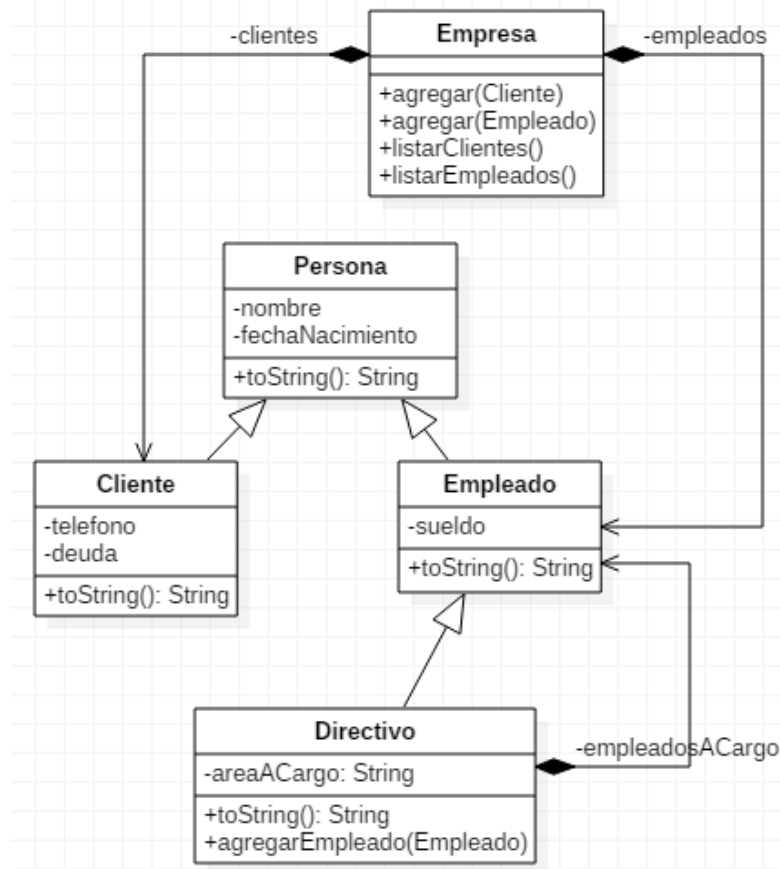


NOTA: El **Cuadrado** es un caso especial de **Rectángulo**, donde

base = altura = **lado**

- Dibuja el Diagrama de Clases de las clases descriptas
- Implementa las clases con sus atributos, métodos y constructores
- Instancia figuras de distinta clase en un arreglo (al menos 5). Recorre el arreglo y muestra para cada elemento, el **tipo de figura**, sus **atributos**, el **área** y el **perímetro**.

2. Una **Empresa** necesita tener registro de sus **Clientes** y **Empleados**. Ambos se caracterizan por su nombre y su fecha de nacimiento. Los empleados tienen número de legajo y sueldo. Para aquellos empleados que son **Directivos**, se registra cuál es el área a su cargo y la lista de empleados a los que dirigen. De cada cliente se necesita conocer su teléfono de contacto y el total que adeuda.



- Crear por código algunas instancias de Clientes, de Empleados y de Directivos y agregarlos a una Lista.
- Recorrer la lista y mostrar en dos grandes grupos y **utilizando Polimorfismo**, toda la información posible de las Personas. Ejemplo:

CLIENTES

Nombre: Carlos F.Nac:14/01/1990
 Telefono: 996991992
 Deuda acumulada: 900.0

Nombre: Juan F.Nac:22/12/1995
 Telefono: 225447411
 Deuda acumulada: 0.0

EMPLEADOS

Nombre: Fernando F.Nac:12/11/1986
 Sueldo: 11111.0

Nombre: Andrea F.Nac:12/11/1955
 Sueldo: 22222.0

Nombre: Mascherano F.Nac:30/08/1960
 Sueldo: 33333.0

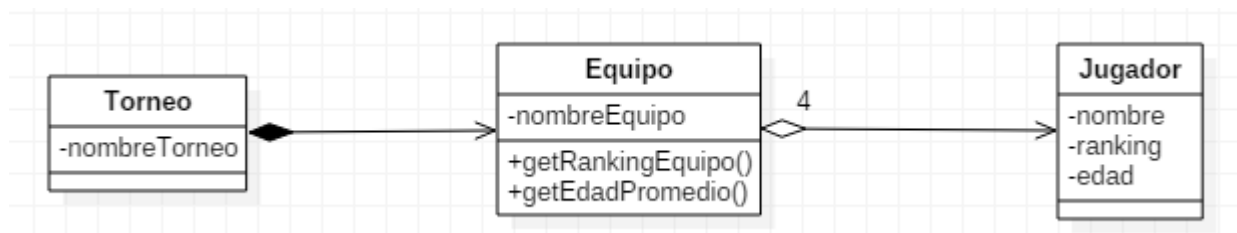
Area: Desarrollo DIRECTIVO Empleados a cargo: Fernando; Andrea

Nombre: Mario F.Nac:01/11/1986
 Sueldo: 44444.0

Nombre: Clara F.Nac:12/12/1975
 Sueldo: 55555.0

Area: GerenciaDIRECTIVO Empleados a cargo: Mario

3. Se desea tener información de los equipos que juegan un torneo de tenis. Cada equipo tiene un nombre y se compone de 4 jugadores. De los jugadores, además de su nombre, se debe conocer su ranking y su edad.



La información del torneo, los equipos y jugadores se obtiene leyéndola de un archivo .TXT que contiene la siguiente información:

```

NOMBRE_DEL_TORNEO
NOMBRE_EQUIPO; DATOS_JUGADOR1; DATOS_JUGADOR2; DATOS_JUGADOR3; DATOS_JUGADOR4
...

```

La estructura del archivo contiene:

- En la primera línea el nombre del torneo
- En las restantes líneas hasta el final del archivo, los datos de cada equipo
 - Una línea por equipo
 - Los datos del equipo están delimitados por *punto y coma (;)*, siendo el primer campo el nombre del equipo y, los cuatro campos restantes, los datos de cada jugador.
 - Los datos de cada jugador están a su vez delimitados por *coma (,)*, donde el primer dato es el nombre del jugador, el segundo, el ranking y el tercero, la edad:

NOMBRE_JUGADOR, RANKING, EDAD

Archivo TorneoTenis.txt:

```

Copa Davis
Argentina; Juan, 10, 25; Pablo, 180, 33; Pedro, 54, 28; Aldo, 74, 22
Francia; Jean, 3, 28; Paul, 12, 24; Pierre, 61, 21; Gerard, 99, 34
Italia; Gianni, 9, 29; Giuseppe, 19, 24; Luigi, 43, 26; Mario, 154, 33
EEUU; John, 17, 26; Sean, 78, 23; Joe, 39, 22; Ronald, 8, 27

```

La aplicación deberá:

- Listar la información de los equipos y de sus integrantes
- Listar los equipos ordenándolos por ranking ascendente (los mejores se muestran primero), mostrando para cada uno, su ranking y la edad promedio de sus jugadores. El ranking del Equipo se obtiene con la sumatoria de los rankings de los jugadores
- ¿Qué cambio implementarías en el diagrama de clases para saber a qué equipo pertenece el Jugador? Implementa listar los jugadores ordenados por ranking, incluyendo el equipo al que pertenece cada uno:

```

Jean (3º) - Francia
Ronald (8º) - EEUU
Gianni (9º) - Italia
Juan (10º) - Argentina
...

```

4. Implementa “Herencia Múltiple” en la siguiente jerarquía de clases, utilizando interfaces:

Clase: ATLETA

Atributos: nombre, categoría.

Métodos: Correr, CargarDatos, VerDatos.

Interface: NADADOR

Atributos: sin atributos.

Métodos: Nadar.

Interface: CICLISTA

Atributos: sin atributos

Métodos: Andar.

Subclase: TRIATLETA (es ATLETA, NADADOR Y CICLISTA)

Atributos: no agrega atributos.

Métodos: no agrega métodos.

Clase: ANIMAL

Atributos: especie, grupo.

Métodos: Cargar, Ver.

Subclase: DELFIN (es ANIMAL y NADADOR)

Atributos: no agrega atributos.

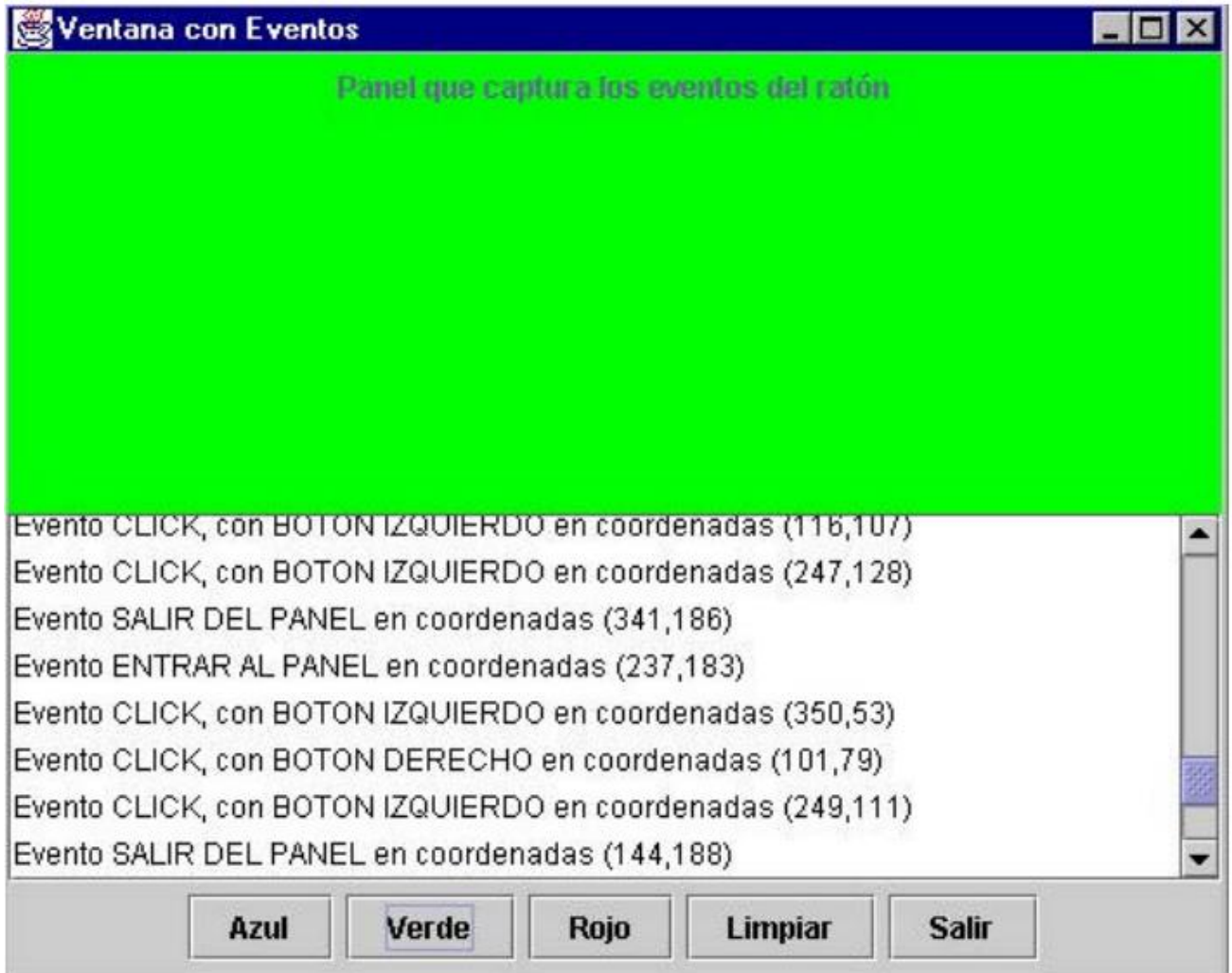
Métodos: no agrega métodos.

- a. Dibuja el diagrama de clases correspondiente a las clases dadas
- b. Implementa las clases con sus métodos.
- c. Define un arreglo de 4 atletas y muestra los datos de cada uno.
- d. Define un arreglo de 4 Nadadores y agrégale Triatletas y Delfines. Recorre el arreglo e invoca el método *nadar* de cada uno de los elementos. El método debe mostrar un mensaje que diga “<nombre>, *Nombre de la Clase* nadando”. Ejemplos: “Alberto, Triatleta nadando”; “Skeepy, Delfín nadando”

Ejercicios GUIs: Componentes visuales, eventos, streams

5. Desarrolla un programa que utilice componentes de Swing e implemente la siguiente funcionalidad:

- Detectar entrada / salida del Panel
- Detectar *clicks* en los botones de colores, dar información del botón presionado y cambiar el color del Panel de acuerdo al botón presionado.
- Detectar *click* sobre el botón de *Salir* y cerrar la aplicación



6. Al ejercicio 1, agrégale la funcionalidad necesaria para que el usuario cree figuras de distinto tipo desde una GUI.

- El usuario podrá seleccionar qué tipo de figura desea crear e ingresar los valores propios de dicha figura.
- Al confirmar la creación de la figura, controlar que los valores ingresados sean válidos. Por ejemplo, que semieje menor < semieje mayor en el caso de las elipses, que los valores ingresados sean mayores a cero, etc.:
 - Si la validación falla, mostrar una notificación con el problema detectado.
 - Si las validaciones son correctas, crear la figura, agregarla a una lista y limpiar los controles de la interfaz

- El usuario podrá mostrar la información de las figuras creadas:
 - Se deberá mostrar toda la información posible de las mismas, utilizando polimorfismo, tal como se hizo en el ejercicio 1.
 - Incluir la posibilidad de filtrar las figuras a mostrar seleccionando por tipo de figura, por área y/o por perímetro y ordenadas por distintos criterios. Si no se especifican valores en los filtros, incluir todos: Si no se seleccionan tipos de figura, listar todos los tipos; si no se especifican áreas/perímetros, no filtrar por Área/Perímetro

Listado de Figuras				
Rectángulo	Perímetro: 38.00	Área: 60.00	Base: 15	Altura: 4
Círculo	Perímetro: 31.41	Área: 78.53	Radio: 5	
Rectángulo	Perímetro: 50.00	Área: 150.00	Base: 10	Altura: 15
Rectángulo	Perímetro: 64.00	Área: 240.00	Base: 12	Altura: 20
Círculo	Perímetro: 314.15	Área: 7853.98	Radio: 50	
Cantidad de figuras: 5				

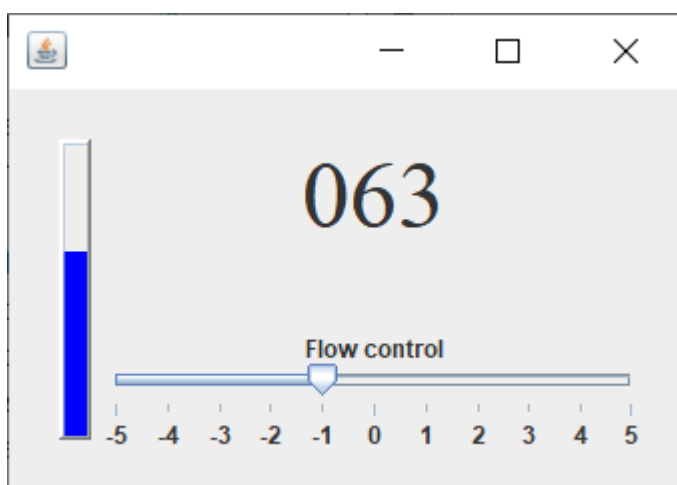
Tips:

- Para el filtrado, utiliza *streams* y *filter*. Para filtrar por clases, puedes crear un Set de Clases (Set<Class>) e invocar el método *contains* de dicho set

7. Si en el ejercicio anterior la validación de los atributos la hiciste en la Vista, modifica tu código para que la validación se haga en las clases del Modelo, mediante la utilización de métodos estáticos.

8. Lanzamiento de Eventos generados por el desarrollador

Desarrollar un **sistema para control de un tanque industrial**. El tanque tiene un sensor de nivel. Un panel de mando digital dispone de un regulador de flujo que comanda la entrada o salida de líquido. Hay 5 niveles de flujo, ya sea para llenar o desagotar el tanque. El panel dispone de una display digital que indica gráficamente y numéricamente, el nivel de líquido que contiene el tanque.



El sensor del tanque informa al controlador tres eventos diferentes:

- NIVEL: cada vez que hay una variación de su nivel, informa el valor del nivel
- TANQUE VACIO: cada vez que el tanque se vacía completamente
- TANQUE LLENO: cada vez que el tanque se completa

Si el tanque se llena, se deberá cerrar la entrada de líquido poniendo en 0 el regulador de flujo.

Se requiere utilizar el patrón de diseño MVC, independizando el modelo y la vista.

Nota: El controlador dispondrá de un Timer para simular el llenado/vaciado del tanque según lo que indique el regulador de flujo

