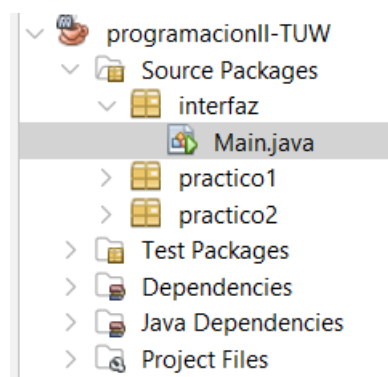


## PRÁCTICO N.º 2 – Lenguaje JAVA

Usando NetBeans, cree un nuevo paquete dentro del proyecto usado para el Práctico 1 llamado practico2. Dentro de este paquete colocará todas las clases de los ejercicios de este práctico. Renombre el paquete de su proyecto donde tenía las clases del Práctico 1 y nómbrelo practico1. De esta manera, su proyecto ahora debería contener dos paquetes con nombres practico1 y practico2. Cree un paquete adicional llamado interfaz, el cual contendrá la clase java desde donde comenzará la ejecución del sistema (clase con método main) y el menú que permitirá seleccionar los ejercicios a ejecutar de cada práctico. Finalizado esto, la estructura del proyecto debería ser similar a la imagen siguiente:



### Ejercicio 1:

Defina la clase Persona con las siguientes características:

El estado de las instancias de la clase estará representado por los siguientes atributos:

- Nombre
- Edad
- Domicilio
- Estado Civil

El comportamiento de las instancias de la clase está determinado por los siguientes métodos:

- Constructores que permitan:
  - Crear instancias inicializando los atributos con valores recibidos por parámetros
  - Crear instancias sin inicializar nada.
- Observadores para todos los atributos.
- Generadores para todos los atributos.

Agregue este ejercicio al menú de ejecución del proyecto NetBeans permitiendo crear dos instancias de la clase Persona, completarlas con sus datos y luego imprimirlos por pantalla. Use los métodos de la clase para esto.

### Ejercicio 2:

Defina la clase CuentaCorriente con las siguientes características:

El estado de las instancias de la clase estará representado por los siguientes atributos:

- Número de Cuenta Corriente.

- Nombre del Cliente.
- Cantidad de Dinero Depositado por el Cliente.
- Cantidad de Dinero que Debe Pagar el Cliente.

El comportamiento de las instancias de la clase está determinado por las siguientes operaciones:

- Un constructor
- Observadores para cada atributo.
- Generadores para cada atributo.
- Método `calcularSaldo()`: Informa al usuario el monto a favor o en contra que tiene el cliente.

Agregue este ejercicio al menú de ejecución del proyecto NetBeans permitiendo crear dos instancias de la clase `CuentaCorriente`, completarlas con sus datos y luego imprimir por pantalla el nombre del cliente, el número de cuenta corriente y el saldo que posee. Use los métodos de la clase para esto.

**Ejercicio 3:**

Crear una clase `Cliente` y otra `Producto`, defina los atributos necesarios y los métodos para acceder y setear estos (Set y Get).

**Ejercicio 4:**

Desarrolle una clase `Compra`, que sea la responsable de administrar una compra de varios productos por un cliente en una fecha dada.

Defina los métodos:

```
agregarProducto(int IDProducto, int cantidad);  
agregarProducto(int IDProducto); // valor por default para cantidad = 1
```

**Ejercicio 5:**

Defina una clase que permita administrar las distintas compras que se van realizando en un kiosco. Implemente la solución utilizando:

- a) Arrays
- b) ArrayLists

¿Qué diferencias encuentra en ambas implementaciones?

**Ejercicio 6:**

Dadas las clases definidas en los ejercicios 4 y 5, defina una clase que permita a un cliente, del cual se deberá pedir sus datos, realizar la compra de uno o más productos. Además, se deben poder responder a los siguientes métodos:

- `totalMonto()`: Devuelve el monto total de todas las compras y clientes.
- `totalMonto(Cliente unCliente)`: Devuelve el monto de las compras de un cliente dado.
- `totalMonto(Compra unaCompra)`: Devuelve el monto de una compra dada.

Agregue este ejercicio al menú de ejecución del proyecto NetBeans permitiendo realizar la compra y las consultas de los métodos definidos.

**Ejercicio 7:**

Un pixel es un punto con coordenadas enteras x e y, que tiene asociado un color cuyo valor se encuentra en el rango de 0 a 255. Cree una clase Pixel que permita:

- Traslarse encima de, debajo de, a la izquierda de, a la derecha de un valor real.
- Cambiar su color a otro color de un número dado (entre 0 y 255).

Agregue este ejercicio al menú de ejecución del proyecto NetBeans permitiendo crear instancias de la clase Pixel, llenarlas con valores y posibilite elegir hacia donde mover el pixel y modificar su color.

**Ejercicio 8:**

Crear una jerarquía de herencia de Roedor: *Ratón*, *Gerbil* y *Hamster*. En la clase base *Roedor*, proveer métodos que son comunes a todos los roedores, y especializar dichos métodos en las clases derivadas según corresponda. Agregue este ejercicio al menú de ejecución del proyecto NetBeans permitiendo crear un arreglo de tipo Roedor, llenarlo con diferentes tipos de roedores, y llamar a los métodos de la clase base para observar que sucede.

**Ejercicio 9:**

Un zoológico necesita mantener información acerca de los animales que se alojan en sus instalaciones. Los mismos se deben clasificar de acuerdo a alguna de las características de los animales: especie, tipo de alimentación, estructura, etc. Defina una Jerarquía que permita definir dicha clasificación.

Agregue este ejercicio al menú de ejecución del proyecto NetBeans permitiendo ingresar al menos 10 animales y luego poder hacer consultas que liste los animales según su especie, tipo de alimentación, estructura, etc. Use los métodos de la clase para esto.

**Ejercicio 10:**

Crear una jerarquía de herencia de Shape (ver Figura 1). En la clase base, proveer métodos que son comunes a todas las figuras, y especializar dichos métodos en las clases derivadas según corresponda (ver Figura 2). Agregue este ejercicio al menú de ejecución del proyecto NetBeans permitiendo crear un arreglo de tipo Shape, llenarlo con diferentes tipos de figuras, y llamar a los métodos de la clase base para observar que sucede.

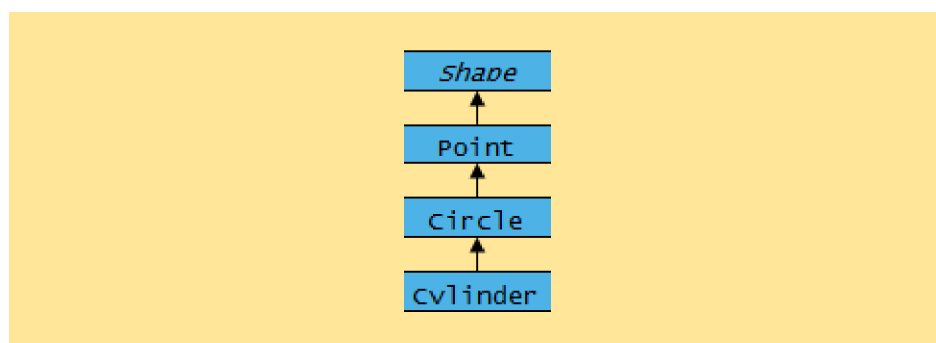


Figura 1

	<code>getArea</code>	<code>getVolume</code>	<code>getName</code>	<code>print</code>
Shape	0.0	0.0	<i>método abstracto</i>	<i>método abstracto</i>
Point	0.0	0.0	"Point"	[x,y]
Circle	$pr^2$	0.0	"Circle"	center=[x,y]; radius=r
Cylinder	$2pr^2+2prh$	$pr^2h$	"Cylinder"	center=[x,y]; radius=r; height=h

Figura 2