

Programa

111

Mil

*Guia 6: Programación Orientada a Objetos*



**Coordinadora: Adriana Brugnoli**

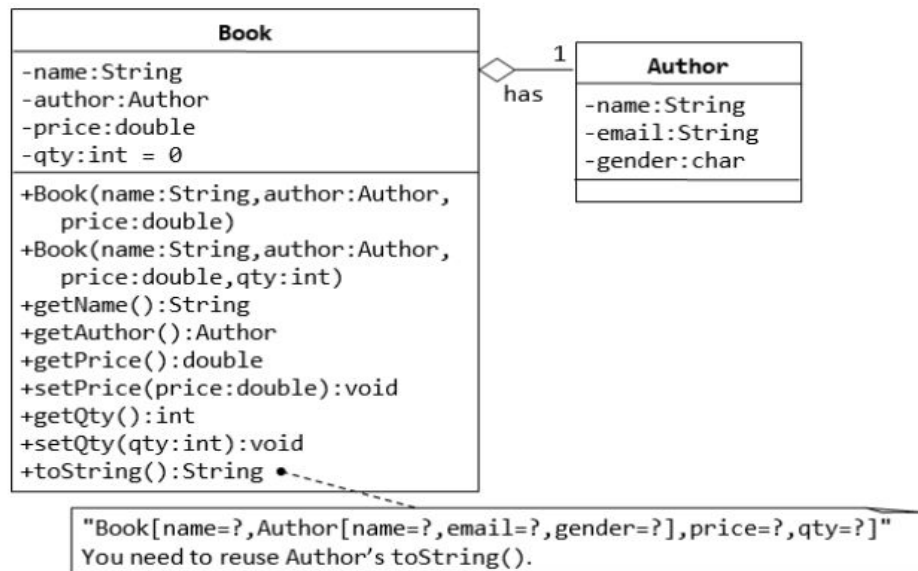
**Instructores: Leandro Maro, Matias Parra**

28/06/18

*E.E.T. N° 24 Simon de Iriondo*

## EJERCICIOS

1. Escriba la clase Book (que utiliza la clase Author escrita anteriormente). También escribe un controlador de prueba llamado TestBook para probar todos los métodos públicos en la clase Libro. Tome Nota que tiene que construir una instancia de Autor antes de poder construir una instancia de Libro.



2. Escribe una superclase llamada Shape (como se muestra en el diagrama de clases), que contenga:
  - a. Constructores.
  - b. Getters y Setters.
  - c. Un método toString() que devuelve "Una figura de color de xxx y llena/vacía".

Para las clases Círculo y Rectángulo:

- d. Los métodos getArea() y getPerimeter().
- e. Sobreescriba toString() acorde a la figura dada.

Para cuadrado:

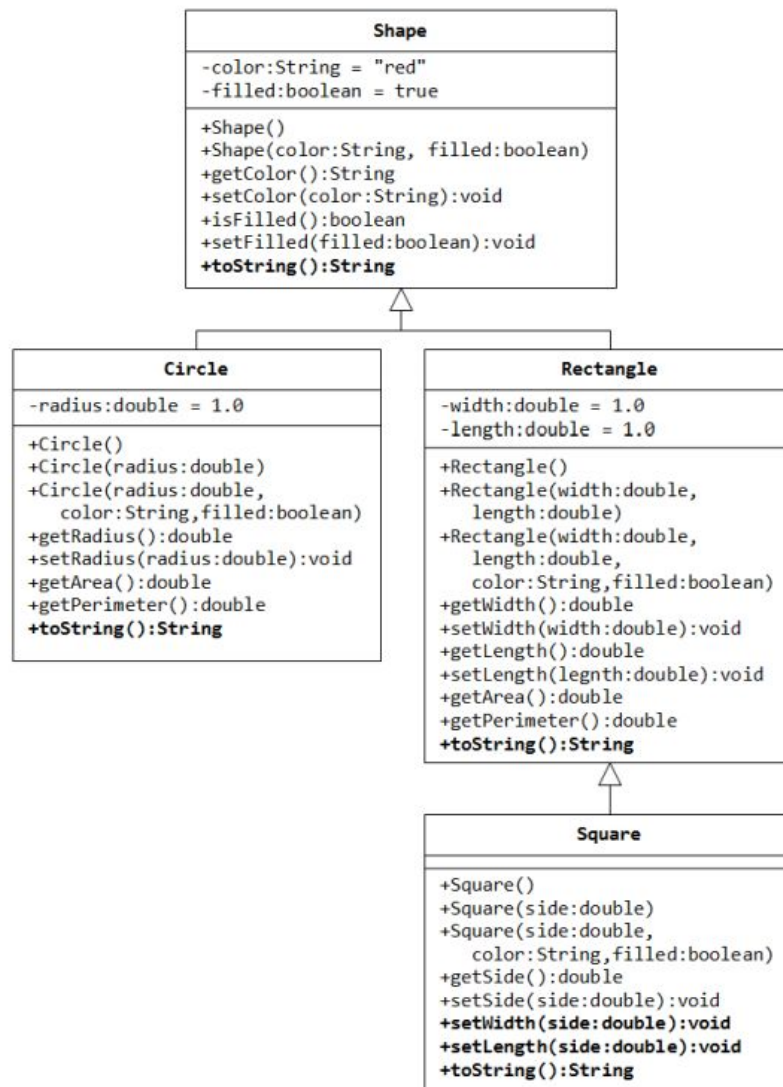
- f. Proporcione los constructores apropiados (como se muestra en el diagrama de clases). Pista:

```
public Square(double side) {

    super(side, side); // Call superclass Rectangle(double, double)

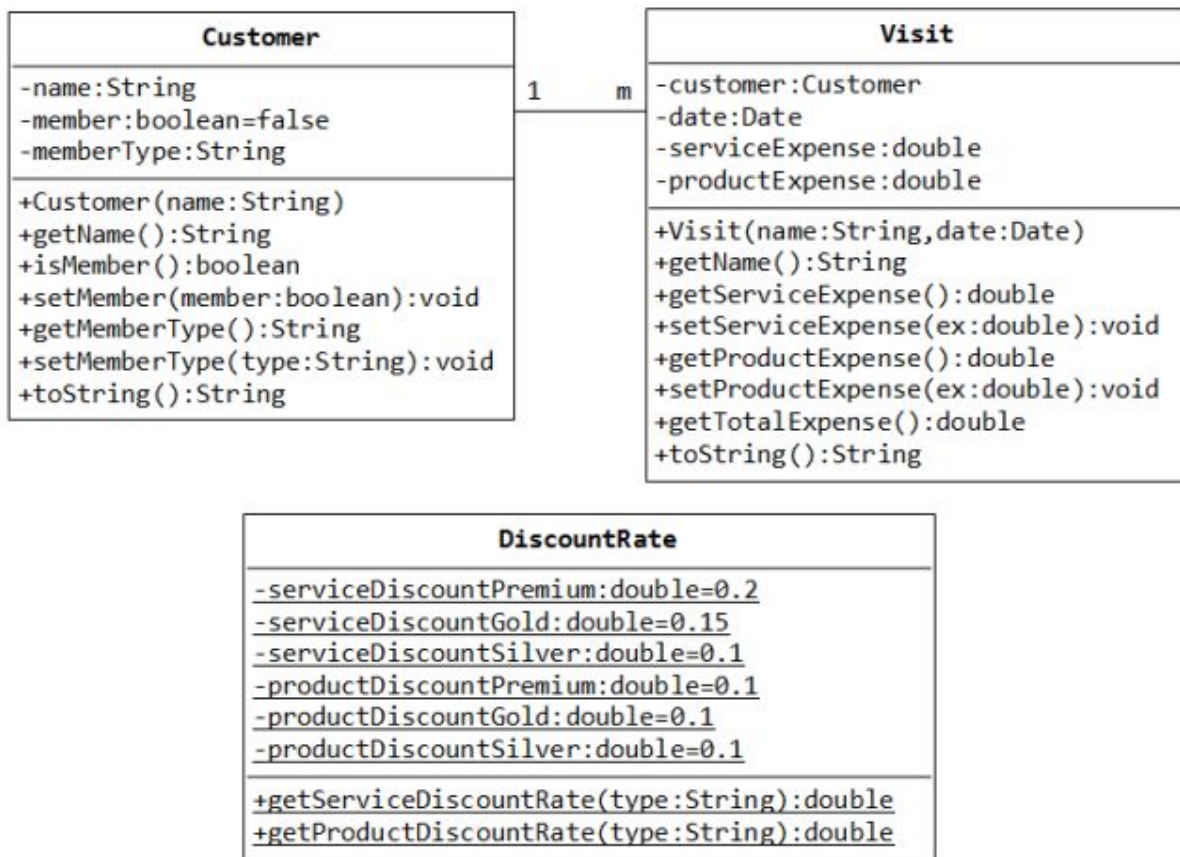
}
```

- g. ¿Necesita sobre escribir los métodos getArea() y getPerimeter()? Pruébalos.
- h. Sobre escriba el setLength () y el setWidth () para cambiar tanto el ancho como la longitud, para mantener la geometría cuadrada. Escriba programas de prueba para probar todas las clases.

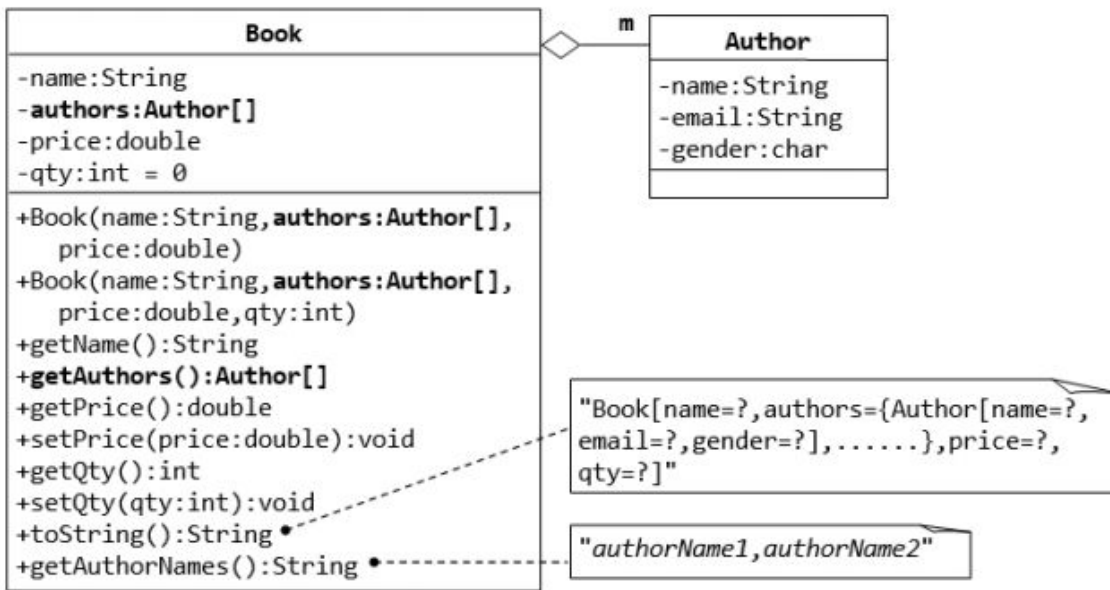


3. Se le pide que escriba un sistema de descuento para un salón de belleza, que ofrece servicios y vende productos cosméticos Ofrece 3 tipos de membresías:

Premium, Oro y Plata. Los miembros premium, oro y plata reciben un descuento del 20%, 15% y 10%, respectivamente, por todos los servicios prestados. Los clientes sin afiliación no reciben ningún descuento. Todos los miembros reciben un descuento del 10% en los productos comprados (esto podría cambiar en el futuro). Su sistema consistirá en tres clases: Cliente, Descuento y Visita, como se muestra en el diagrama de clases. Calculará la factura total si un cliente compra \$x de productos y \$y de servicios, para una visita. También escribir un programa de prueba para ejercer todas las clases.

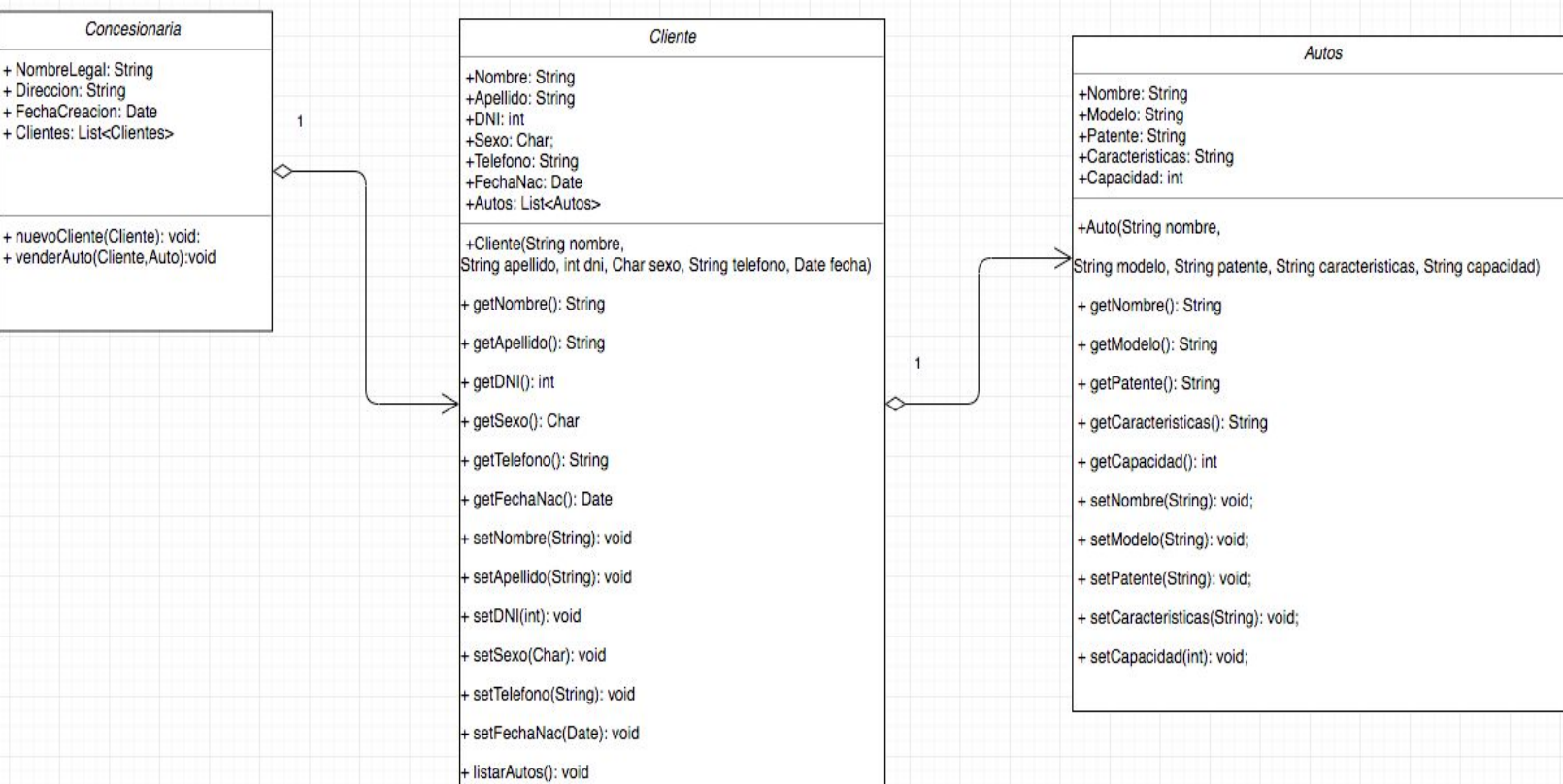


- Ahora que estamos más avanzados en el tema le vamos a agregar complejidad a las clases Autor y Libro.



5. Realice un diagrama UML DE una clase Matrix. Después de realizar el diagrama debe programa correspondiente en JAVA. Orientación a Objetos La clase debe contener y poder mostrar los siguientes datos:
  - a. Nombre de la matriz.
  - b. Dimensión mxn (Filas y columnas).
  - c. Valores de los elementos. También debe soportar las siguientes operaciones con matrices:
    - i. Carga Manual y carga aleatoria de los elementos componentes.
    - ii. Suma de matrices.
    - iii. Opuesto de una matriz
6. Se pide:
  - a. Escribe una clase cuyas instancias representan una única carta de juego de una baraja de cartas. Cada carta tiene dos propiedades distintivas: rango y palo.
  - b. Escribe una clase cuyas instancias representan una baraja completa de cartas.
  - c. Escribe un pequeño programa para probar tus clases de baraja y cartas. El programa puede ser tan simple como crear un mazo de cartas y mostrar sus cartas.
7. Dado el siguiente diagrama de clases





Se le pide que lleve diseño el sistema para una concesionaria de autos “Acapulco Autos” creada en 1973 con dirección: Remedios de Escalada de San Martín 1700, Gerli, Buenos Aires.

Nuestro cliente desea contar con las funcionalidades de agregar nuevos clientes y vender autos.

A su vez se desea saber de cada cliente, la cantidad y los datos de los autos que posee.

Consideraciones:

- La concesionaria no puede tener instancias
- Los clientes por defecto no tienen autos
- La concesionaria por defecto no tiene clientes
- Las clases deben tener `toString()`

Utilice una clase main para agregar 5 clientes. Considere que el primer y el tercer cliente solamente compran autos. Se desea saber las características de los mismos y la cantidad

que compraron.

8. Dado el escenario anterior agregue las siguientes funcionalidades
  - a. Además de el método ya existente (agregar alumno) cree un método para eliminar un cliente dado su número de documento. En caso de eliminarlo, debe mostrar un mensaje de “eliminación satisfactoria”, de no encontrarlo debe mostrar por consola que no existe ese cliente.
  - b. Crear la funcionalidad de la concesionaria de poder comprar autos (No deben pertenecer a ningún cliente)
  - c. Modificar el método vender auto ya que al vender el auto, no pertenece más a la concesionaria. Se desea tener un registro también de los autos vendidos.
  - d. Se desea conocer también cuáles fueron los autos vendidos por la concesionaria
  - e. También existe la posibilidad de que un auto de la concesionaria deje de funcionar y debe sacarse de los autos a la venta.

Utilice una clase main para agregar 15 autos a la concesionaria. Considere que el 2 y 4 cliente de los agregados previamente compren un auto cada uno. Luego considere que auto número 8 tiene una falla mecánica y se saca de la venta y que el cliente número 3 deja de ser cliente. Luego listar todos los autos vendidos por la concesionaria así como los que tienen en stock actualmente.