

Primer Parcial IPOO
“Sistema para una bicicletería”

Una importante bicicletería de la región desea implementar una aplicación que le permita gestionar la información de sus clientes, de las bicis y de las ventas realizadas. Para ello se almacena información de todos sus clientes, de cada una de las bicicletas disponibles en el local y de todas las ventas realizadas.

Implementar las siguientes clases: **Empresa, Venta, Cliente y Bicicleta**

En la clase **Cliente**:

0. Se registra la siguiente información: nombre, apellido, si está o no dado de baja, el tipo y el número de documento. Si un cliente está dado de baja, no puede registrar compras desde el momento de su baja.
1. Método constructor que recibe como parámetros los valores iniciales para los atributos.
2. Los métodos de acceso de cada uno de los atributos de la clase.
3. Redefinir el método **_toString** para que retorne la información de los atributos de la clase.

En la clase **Bicicleta**:

1. Se registra la siguiente información: código, costo, año fabricación, descripción, porcentaje, incremento anual, activa (atributo que va a contener un valor true, si la bici está disponible para la venta y false en caso contrario).
2. Método constructor que recibe como parámetros los valores iniciales para los atributos definidos en la clase.
3. Los métodos de acceso de cada uno de los atributos de la clase.
4. Redefinir el método **toString** para que retorne la información de los atributos de la clase.
5. Implementar el método **darPrecioVenta** el cual calcula el valor por el cual puede ser vendida una bici. Si la bici no se encuentra disponible para la venta retorna un valor < 0. Si la bici está disponible para la venta, el método realiza el siguiente cálculo:

$$\$_{venta} = \$_{compra} + \$_{compra} * (anio * por_inc_anual)$$

donde $\$_{compra}$: es el costo de la bici.

$anio$: cantidad de años transcurridos desde que se fabricó la bici.

por_inc_anual : porcentaje de incremento anual de la bici.

En la clase **Venta**:

1. Se registra la siguiente información: número, fecha, referencia al cliente, referencia a una colección de bicicletas y el precio final.
2. Método constructor que recibe como parámetros cada uno de los valores a ser asignados a cada atributo de la clase.
3. Los métodos de acceso de cada uno de los atributos de la clase.
4. Redefinir el método **_toString** para que retorne la información de los atributos de la clase.
5. Implementar el método **incorporarBicicleta(\$objBici)** que recibe por parámetro un objeto bicicleta y lo incorpora a la colección de bicicletas de la venta, siempre y cuando sea posible la venta. El método cada vez que incorpora una *bicicleta* a la venta, debe actualizar la variable instancia **precio final** de la venta. Utilizar el método que calcula el precio de venta de la bici donde crea necesario.

En la clase **Empresa**:

1. Se registra la siguiente información: denominación, dirección, la colección de clientes, colección de bicicletas y la colección de ventas realizadas.
2. Método constructor que recibe como parámetros los valores iniciales para los atributos de la clase.
3. Los métodos de acceso para cada una de las variables instancias de la clase.
4. Redefinir el método **_toString** para que retorne la información de los atributos de la clase.

**FACULTAD DE INFORMATICA
CÁTEDRA INTRODUCCIÓN POO**

5. Implementar el método **retornarBici(\$codigoBici)** que recorre la colección de bicis de la Empresa y retorna la referencia al objeto bicicleta cuyo código coincide con el recibido por parámetro.
6. Implementar el método **registrarVenta(\$colCodigosBici, \$objCliente)** método que recibe por parámetro una colección de códigos de bicicletas, la cual es recorrida, y por cada elemento de la colección se busca el objeto bicicleta correspondiente al código y se incorpora a la colección de bicis de la instancia Venta que debe ser creada. Recordar que no todos los clientes ni todas las bicis, están disponibles para registrar una venta en un momento determinado.
El método debe setear los variables instancias de venta que corresponda y retornar el importe final de la venta.
7. Implementar el método **retornarVentasXCliente(\$tipo,\$numDoc)** que recibe por parámetro el tipo y número de documento de un Cliente y retorna una colección con las ventas realizadas al cliente.

Implementar un script **TestEmpresa** en la cual:

1. Cree 2 instancias de la clase Cliente: \$objCliente1, \$objCliente2.
2. Cree 3 objetos Bicicletas con la información visualizada en la tabla: código, costo, año fabricación, descripción, porcentaje incremento anual, activo.

código	costo	anio_fabricacion	Descripcion	porc_increment	activo
11	89500	2022	Fire Bird Plegable Cuadro Acero	85%	true
12	310000	2021	Bicicleta Trek Marlin 5 Rodado 29 Talle L	70%	true
13	10000	2023	Bicicleta Topmega Fixie Streeter R28 Acero 1vel Azul Osc T54	55%	false

4. Se crea un objeto **Empresa** con la siguiente información: **denominación** = "Alta Gama", **dirección** = "Av Argentina 123", **colección de bicicletas** = [\$objBici1, \$objBici2, \$objBici3], **colección de clientes** = [\$objCliente1, \$objCliente2], **la colección de ventas realizadas** = [].
5. Invocar al método **registrarVenta(\$colCodigosBici, \$objCliente)** de la Clase Empresa donde el \$objCliente es una referencia a la clase Cliente almacenada en la variable \$objCliente2 (creada en el punto 1) y la colección de códigos de bicicletas es la siguiente [11,12,13]. Visualizar el resultado obtenido.
6. Invocar al método **registrarVenta(\$colCodigosBicicletas, \$objCliente)** de la Clase Empresa donde el \$objCliente es una referencia a la clase Cliente almacenada en la variable \$objCliente2 (creada en el punto 1) y la colección de códigos de bicicletas es la siguiente [0]. Visualizar el resultado obtenido.
7. Invocar al método **registrarVenta(\$colCodigosBicicletas, \$objCliente)** de la Clase Empresa donde el \$objCliente es una referencia a la clase Cliente almacenada en la variable \$objCliente2 (creada en el punto 1) y la colección de códigos de bicicletas es la siguiente [2]. Visualizar el resultado obtenido.
8. Invocar al método **retornarVentasXCliente(\$tipo,\$numDoc)** donde el tipo y número de documento se corresponden con el tipo y número de documento del \$objCliente1.
9. Invocar al método **retornarVentasXCliente(\$tipo,\$numDoc)** donde el tipo y número de documento se corresponden con el tipo y número de documento del \$objCliente2.
10. Realizar un echo de la variable Empresa creada en 2.