Mentoría de Software

**Nuevo…!** Accede a los videos de ejercicios ya desarrollados utilizando [éste link](http://www.mentoriadesoftware.com/ejercitarios-para-practicar-en-java)

**Ejercitario JAVA – Programación Orientada a Objetos**

**Clases, Objetos, Atributos, Métodos**

**Ejercicio 1**

1.a Teniendo en cuenta el siguiente diagrama UML, implementar la siguiente clase en Java:



1.b. En el método main, instanciar la clase **Computador** en un objeto llamado **compu**, luego, asignar valores a los atributos de la instancia compu y seguidamente imprimir los atributos de la clase.

**Complemento**

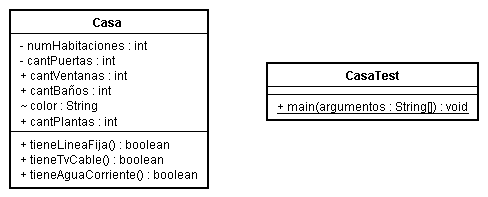
1. Por convención, el nombre de una clase siempre empieza con mayúsculas.
2. El nombre de un atributo siempre empieza con minúsculas. Caso sea utilizado una palabra compleja, a partir de la segunda palabra cada letra inicial debe empezar con mayúsculas, ejemplo: motorGrande, motorEnMarcha, velocidadAlVacio. Esta misma convención se utilizan para los **métodos** y las **variables**.
3. En el diagrama UML los atributos o métodos subrayados representan atributos o métodos estáticos (static)

**Responde:**

1. Qué se puede representar con la Programación Orientada a Objetos?
2. En UML, que representa el carácter ~ que se antepone a los atributos o a los métodos.? Y que significado tiene específicamente?
3. En UML, que significa el método o atributo subrayado? Explica claramente su significado
4. En un Diagrama de Clase, que hace el carácter + que se antepone a los atributos y/o métodos?
5. Son representados la implementación de código o el valor de los atributos en un díagrama UML?

**Ejercicio 2**

2.a Implementar el siguiente diagrama de clases:



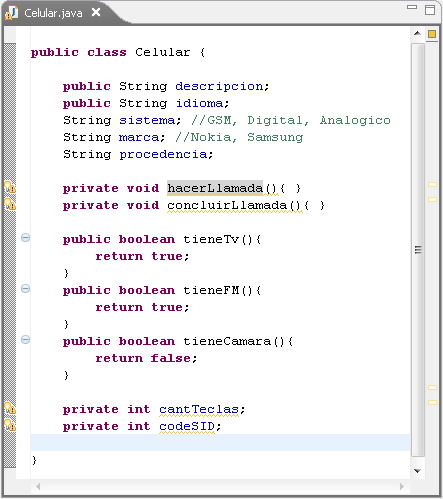
2.b. En el método *main* de la clase *CasaTest* instanciar *Casa* en un objeto, luego atribuir valores a sus atributos y seguidamente imprimir la clase.

**Ejercicio 3**

3. Desarrollar la implementación en Java de una Clase basada en cualquier objeto de la vida real con la cual estás familiarizado, creando al menos 5 atributos y 5 métodos relacionados al mismo. Variar los tipos de datos a utilizarse para los atributos y métodos

**Ejercicio 4**

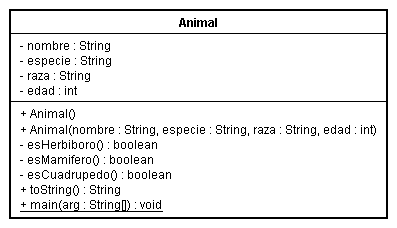
4 Representa gráficamente en un diagrama de clase, la siguiente implementación:



**Constructor, Getter and Setter, toString**

**Ejercicio 1**

1.a Implementar la siguiente clase en Java:



1.b. Implementar correctamente los constructores, el constructor por defecto y el constructor por parámetro.

1.c. Crear los métodos getter y setter para los atributos privados de la clase.

1.d. Sobreescribir el método toString() para imprimir la clase.

1.e. Todos los métodos que retornan booleanos deberán retornar true.

1.f. En el método main, instanciar la clase Animal en tres objetos diferentes, *perro*, *gato* y *leon*, luego, cargar valores a sus atributos y seguidamente imprimir los tres objetos utilizando el método toString(). Utilizar el constructor por defecto para instanciar *gato* y *leon*.

**Complemento**

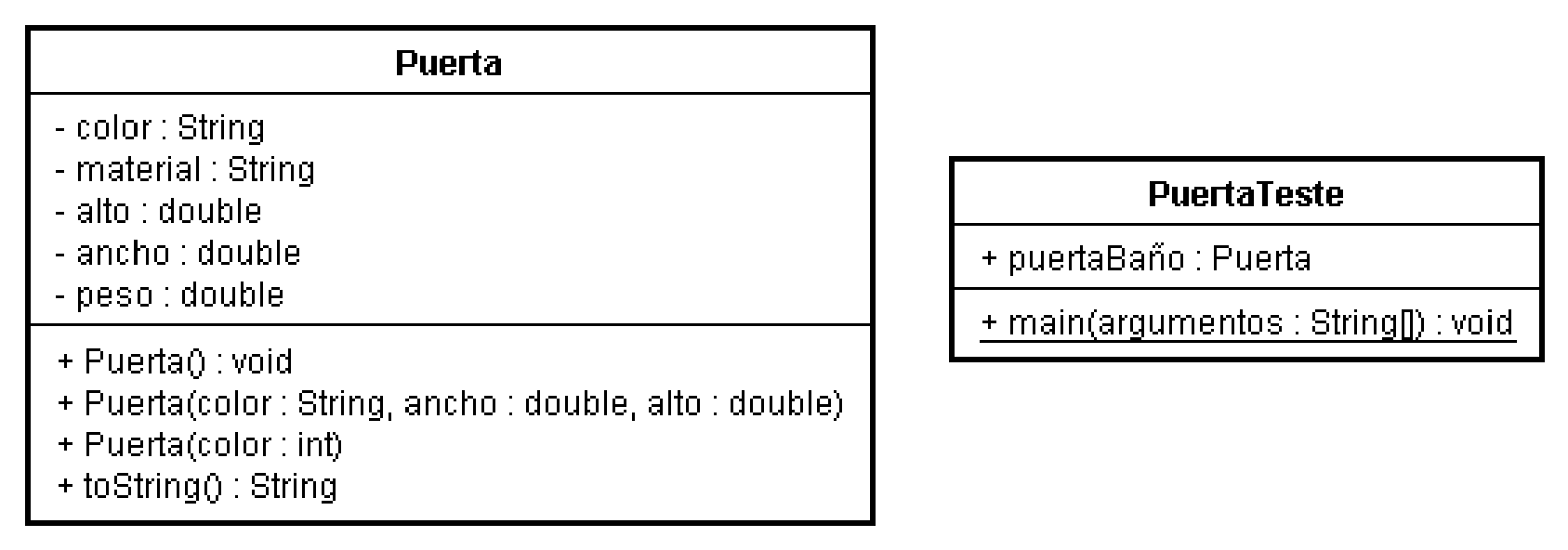
1. Los métodos getter y setter no son necesarios representar en el diagrama de clases UML, ya que se subentiende que si un atributo es *private* tiene su método de asignación(*set*) y de lectura (*get*)
2. Los métodos get y set siguen el estandar *JavaBean* para creación de objetos.

**Responde:**

1. En el diagrama de la clase Animal *(Ejercicio 1)* que diferencia a los constructores de los demás métodos? Menciona por lo menos 3 itens.
2. Qué entiendes por constructor? Qué diferencia existe entre constructor por parámetro y constructor por defecto? Cuántos constructores puede tener una clase? Que específicamente se realiza en un constructor?
3. Que hace el método toString() y para que es utilizado?
4. Que significado tiene la palabra reservada *this* y en que casos puede ser utilizado?

**Ejercicio 2**

2.a Implementar el siguiente diagrama de clases:



2.b. Implementar los métodos constructores, teniendo en cuenta que los valores por defecto para color es marron, para material es madera, para alto es 210,00 y para ancho es 80,00.

2.c. En el método *main* de la clase *PuertaTeste* instanciar *Puerta* en el atributo puertaBaño a través del constructor por defecto, luego en las variable *puertaDelantera* a través del 1er. constructor por parámetro y por último instanciar también en la variable *puertaTrasera* a través del 2do. constructor por parámetro. *(No asignar valores manualmente a los atributos, solo por el constructor)*. Por último imprimir el atributo y las variables instanciadas.

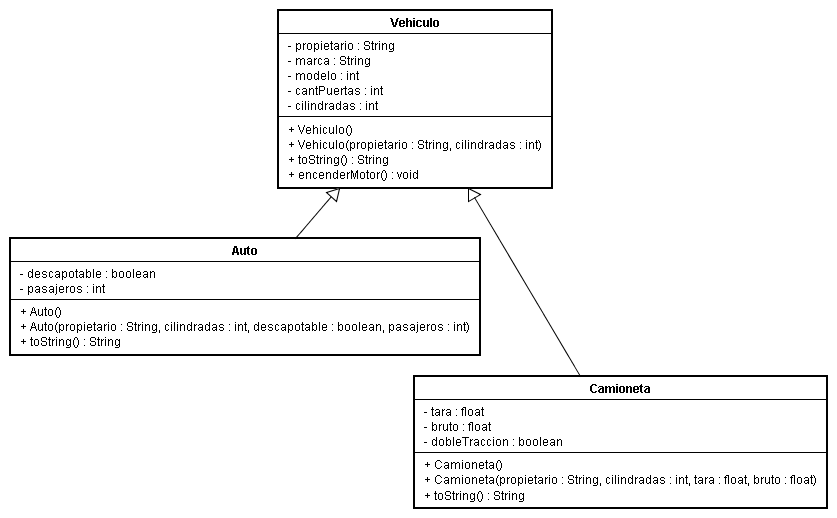
**Ejercicio 3**

3. Selecciona 3 objetos de la vida real y representala gráficamente mediante el lenguaje UML, indicando en ella el método constructor por defecto, los métodos constructores por parámetro, el método toString(). Seguidamente implementar las clases en el Eclipse. Cada clase deberá contener como mínimo 3 atributos y 3 métodos, además del toString() y sus constructores.

**Sección C - Herencia, Polimorfismo, Abstracción e Interfase**

**Ejercicio 1**

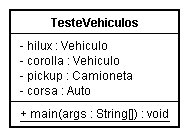
1.a Implementar el siguiente diagrama de clases en el eclipse:



1.b. Implementar correctamente los constructores en la clase base y teniendo en cuenta la sobrecarga a ser implementada en las subclases. Los valores por defecto para los atributos son vacío para propietario, vacío para marca, 2009 para modelo, 4 para cantidad de puertas, y 1800 para cilindradas. Para Auto son falso para descapotable y 5 para pasasjeros, Para Camioneta son cero para tara, bruto; y falso para 4x4.

1.c. Sobreescribir y sobrecargar correctamente el método toString() para las 3 clases.

1.d. Implementar la siguiente clase.

.

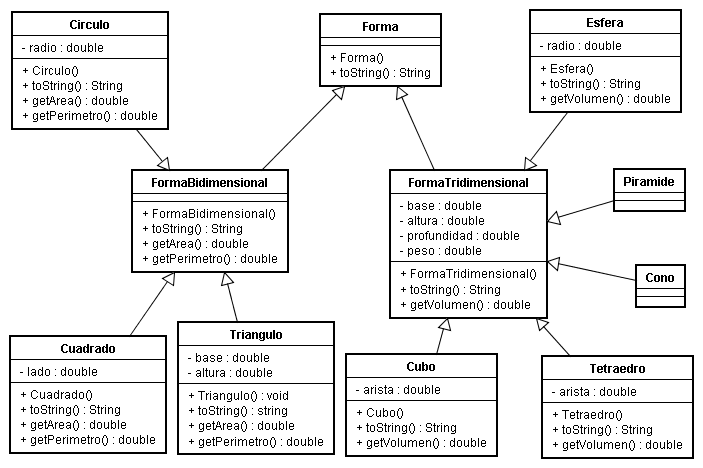
1.e. En el método main de la clase *TesteVehiculos*, cargar valores en los atributos de la clase y luego imprimir cada atributo. Utilizar el método toString para imprimir y utilizar los conceptos de up-casting y down-casting para el atributo hilux y corolla.

**Responde las siguientes cuestiones:**

1. Que entiendes por up-casting y down-casting?.
2. Puede un objeto *Auto* instanciarse desde una clase *Vehiculo*? Puede hacerse lo contrario, un *Vehiculo* instanciarse desde una clase Auto?
3. Para que se utiliza la palabra reservada super.
4. Cual es la clase base de todas las clases, en Java?
5. Que ventajas tiene una clase base normal frente a una clase abstracta?
6. Cual es la diferencia entre clase abstracta e Interfase?

**Ejercicio 2**

2.a Implementar el siguiente diagrama de clases:



2.b. Implementar el método constructor y el método *toString()* de la clase Forma y de todos sus desendientes, teniendo en cuenta la sobrecarga a ser implementada en las subclases.

2.c. Codificar los métodos *getArea()* y *getPerimetro()* de todas las clases del Tipo *FormaBidimensional*, utilizando el cálculo adecuado para cada una de las clases.

2.d. Implementar el método *getVolumen()*, realizando el cálculo correspondiente del volumen de cada forma tridimensional.

2.e. Modelar la clase *Piramide* y la clase *Cono*, agregando atributos (si necesarios) y métodos de forma que también se pueda calcular el volumen de cada uno de ellos. Terminar el diagrama de clases diseñando en UML y luego implementar en el lenguaje.

[Información sobre figuras geométricas podrá encontrar en: http://www.ditutor.com](http://www.ditutor.com/)

**Ejercicio 3**

3.a Realizar el diagrama de clases UML, considerando varios objetos tomados de la vida real e indicando en ellas claramente el concepto de constructor por defecto, constructores por parámetros y método toString así como el concepto de herencia hasta un mínimo de 3 niveles de descendencia.

**Herramienta UML disponible en** http://astah.net/download

**Ejercicio 4**

Sean las siguientes clases:

**public class** Animal **{**

**public** void sonido() **{**

System.out.println(“Grunt”);

**}**

**}**

**public class** Vaca **extends** Animal **{**

**public** void sonido() **{**

System.out.println(“MUUUUUUU!”);

**}**

**}**

**public class** Perro **extends** Animal **{**

**public** void sonido() **{**

System.out.println(“GUAU GUAU!”);

**}**

**}**

**public class** Armadillo **extends** Animal **{}**

¿Cual es la salida de la ejecución del siguiente programa?

**public class** Test1Animal **{**

**public static** void main(String[] args) **{**

Animal animal = **new** Animal();

Armadillo armadillo = **new** Armadillo();

Animal perro = **new** Perro();

System.out.println(armadillo.sonido());

System.out.println(perro.sonido());

System.out.println(animal.sonido());

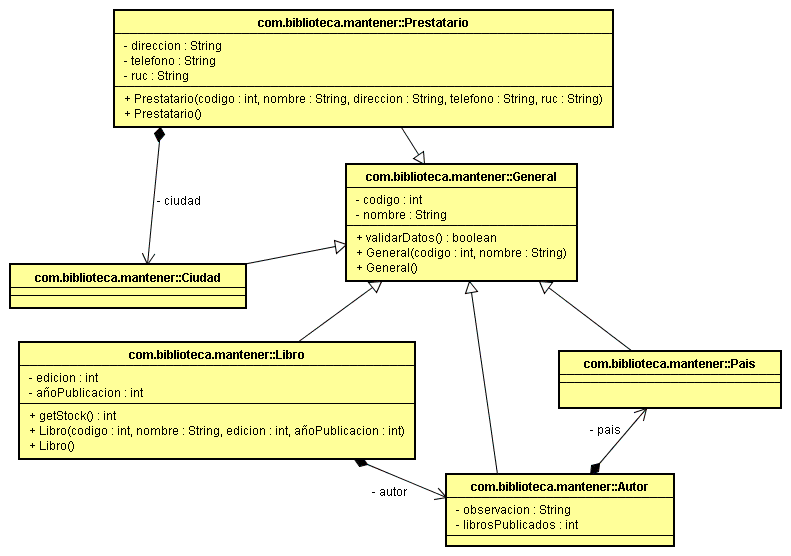
**}**

**}**

**Ejercitario General de POO**

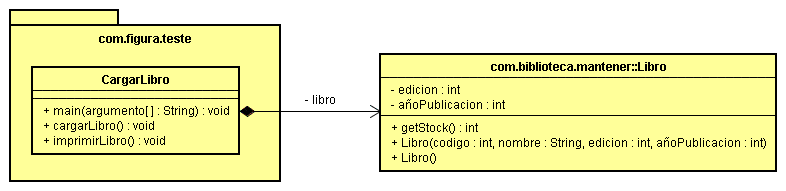
**Ejercicio**

1. Crear un nuevo proyecto JAVA en el Eclipse, llamarlo BIBLIOTECA.
2. Crear la carpeta “src” para almacenar todas las clases fuentes que serán creadas en el proyecto.
3. Implementar la siguiente representación de Diagrama de Clases.



**OBS:** Para crear los paquetes, tener en cuenta que en el diagrama aparece el nombre calificado completo o namespace de la clase.

1. Crear los métodos SETTER y GETTER de los atributos que están con el modificador “private”
2. Implementar correctamente los métodos “Constructor” de cada clase.
3. Crear la clase CargarLibro dentro del paquete indicado.



Obs: Los atributos del objeto libro, deberán introducirse uno a uno por teclado, esto deberá implementarse en el método cargarLibro(). Luego, la impresión de todos los atributos del libro, deberá realizarse en el método imprimirLibro(). Realizar la llamada del método cargarLibro() e imprimirLibro() desde el método main()

1. De acuerdo al diagrama de Clases, responde las siguientes preguntas:

a. En que paquete se encuentra la clase Libro.

b. Cuántos atributos y cuántos métodos posee la clase Pais?

c. Cuántos atributos posee la clase Prestatario?

d. La clase Ciudad y Pais, tienen o no tienen método constructores?

e. Según tu opinión personal, por que crees que todas las clases heredan de la clase

General?

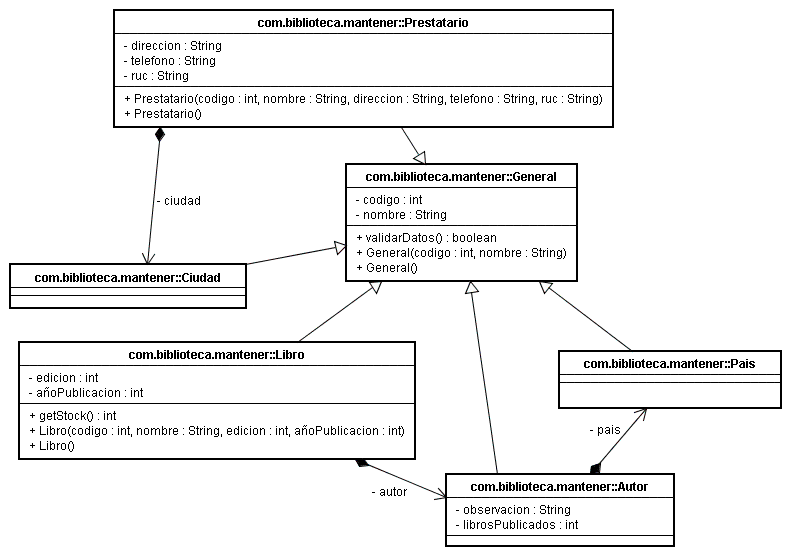
1. Señala en el diagrama de clases los siguientes conceptos, utilizar diagrama de abajo:

a. Constructor por defecto de las clases.

b. Constructor con parámetros.

b. Herencia entre clases.

c. Lugar donde existen atributos de Composición.

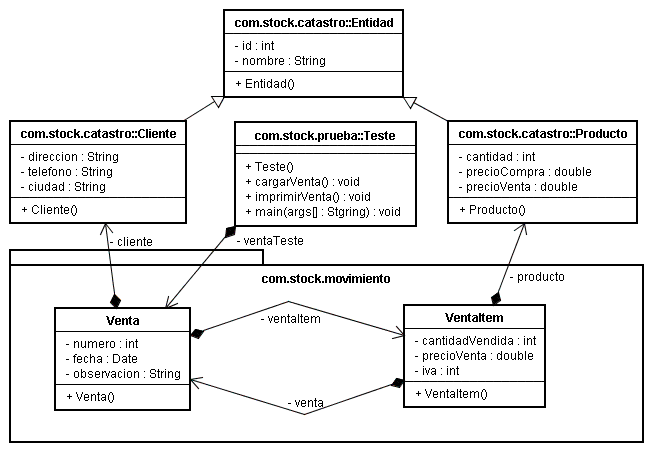


**Ejercitario General POO II**

**Ejercicios**

**REALIZA LOS SIGUIENTES EJERCICIOS**

1. Crear un nuevo proyecto JAVA en el Eclipse, llamarlo ROPERIA.
2. Crear la carpeta “source” para almacenar todas las clases fuentes que serán creadas en el proyecto. Implementar los métodos set() y get().



1. Realizar la carga del objeto Teste.venta y de todos los objetos subinternos por Teclado.
2. Imprimir todos los atributos del objeto Teste.venta utilizando el método toString()