

## **Trabajo Práctico N° 8:** **Módulo Objetos (Desarrollo de Clases).**

### **Ejercicio 1.**

**(a)** Definir una clase para representar triángulos. Un triángulo se caracteriza por el tamaño de sus 3 lados (*double*), el color de relleno (*String*) y el color de línea (*String*). Proveer un constructor que reciba todos los datos necesarios para iniciar el objeto. Proveer métodos para:

- Devolver/modificar el valor de cada uno de sus atributos (métodos *get* y *set*).
- Calcular el perímetro y devolverlo (método *calcularPerimetro*).
- Calcular el área y devolverla (método *calcularArea*).

NOTA: Calcular el área con la fórmula  $\text{Área} = \sqrt{s(s-a)(s-b)(s-c)}$ , donde *a*, *b* y *c* son lados y  $s = \frac{a+b+c}{2}$ . La función raíz cuadrada es *Math.sqrt(#)*.

**(b)** Realizar un programa que instancie un triángulo, le cargue información leída desde teclado e informe en consola el perímetro y el área.

## **Ejercicio 2.**

**(a)** *Modificar la clase Libro.java (carpeta tema3) para, ahora, considerar que el primer autor es un objeto instancia de la clase Autor. Implementar la clase Autor, sabiendo que se caracterizan por nombre, biografía y origen y que deben permitir devolver/modificar el valor de sus atributos y devolver una representación String formada por nombre, biografía y origen. Luego, realizar las modificaciones necesarias en la clase Libro.*

**(b)** *Modificar el programa Demo01Constructores (carpeta tema3) para instanciar los libros con su autor, considerando las modificaciones realizadas. Luego, a partir de uno de los libros instanciados, obtener e imprimir la representación del autor de ese libro.*

### **Ejercicio 3.**

**(a)** Definir una clase para representar estantes. Un estante almacena, a lo sumo, 20 libros. Implementar un constructor que permita iniciar el estante sin libros. Proveer métodos para:

- Devolver la cantidad de libros que hay almacenados.
- Devolver si el estante está lleno.
- Agregar un libro al estante.
- Devolver el libro con un título particular que se recibe.

**(b)** Realizar un programa que instancie un estante. Cargar varios libros. A partir del estante, buscar e informar el autor del libro “Mujercitas”.

**(c)** ¿Qué se modificaría en la clase definida para, ahora, permitir estantes que almacenen como máximo  $N$  libros? ¿Cómo se instanciaría el estante?

**Ejercicio 4.**

**(a)** *Un hotel posee  $N$  habitaciones. De cada habitación, se conoce costo por noche, si está ocupada y, en caso de estarlo, guarda el cliente que la reservó (nombre, DNI y edad).*

**(i)** *Generar las clases necesarias. Para cada una, proveer métodos getters/setters adecuados.*

**(ii)** *Implementar los constructores necesarios para iniciar: los clientes a partir de nombre, DNI, edad; el hotel para  $N$  habitaciones, cada una desocupada y con costo aleatorio entre 2000 y 8000.*

**(iii)** *Implementar, en las clases que corresponda, todos los métodos necesarios para:*

- *Ingresar un cliente  $C$  en la habitación número  $X$ . Asumir que  $X$  es válido (es decir, está en el rango  $1..N$ ) y que la habitación está libre.*
- *Aumentar el precio de todas las habitaciones en un monto recibido.*
- *Obtener la representación String del hotel, siguiendo el formato:  
{Habitación 1: costo, libre u ocupada, información del cliente si está ocupada},  
...  
{Habitación  $N$ : costo, libre u ocupada, información del cliente si está ocupada}.*

**(b)** *Realizar un programa que instancie un hotel, ingrese clientes en distintas habitaciones, muestre el hotel, aumente el precio de las habitaciones y vuelva a mostrar el hotel.*

*NOTAS: Reusar la clase Persona. Para cada método solicitado, pensar a qué clase debe delegar la responsabilidad de la operación.*

**Ejercicio 5.**

**(a)** Definir una clase para representar círculos. Los círculos se caracterizan por su radio (*double*), el color de relleno (*String*) y el color de línea (*String*). Proveer un constructor que reciba todos los datos necesarios para iniciar el objeto. Proveer métodos para:

- Devolver/modificar el valor de cada uno de sus atributos (métodos *get* y *set*).
- Calcular el perímetro y devolverlo (método *calcularPerimetro*).
- Calcular el área y devolverla (método *calcularArea*).

**(b)** Realizar un programa que instancie un círculo, le cargue información leída de teclado e informe en consola el perímetro y el área.

NOTA: La constante *PI* es *Math.PI*.