

Actividad Tema 4 – POO en JAVA (parte II)

Objetivo:

Continuar trabajando los conceptos de la POO. Inicializar objetos a partir de sus constructores.

Las modificaciones planteadas en los ejercicios 1, 2 y 6 deben realizarse sobre los archivos .java de las clases definidas en la Actividad 3.

1-A- Defina constructores para la clase Triángulo (definida en la Act. 3): el *primer constructor* debe recibir un valor para cada lado y para los colores de línea y relleno; el *segundo constructor* no debe poseer parámetros ni código (constructor nulo).

B- Realice un programa que instancie un triángulo mediante los distintos constructores.

2-A- Defina un constructor para la clase Entrenador (definida en la Act. 3) que reciba los datos necesarios (nombre, sueldo básico, cantidad de campeonatos ganados). Además defina un constructor nulo.

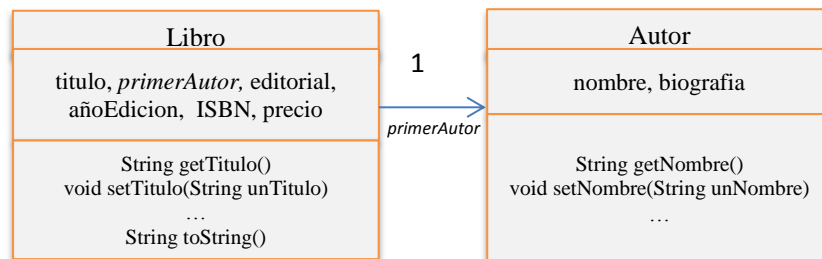
B- Realice un programa que instancie un entrenador mediante el primer constructor.

3-A- Describa todas las modificaciones que deben hacerse en la clase Libro presentada en la explicación para ahora considerar que el primer autor es un objeto (instancia de una clase Autor). Un autor se caracteriza por su nombre y biografía. El autor debe poder devolver/modificar el valor de sus atributos.

B- Describa las modificaciones que deben hacerse en el programa ppal. para crear un libro.

C- Implemente

Diagrama de clases (gráfico):



4-A- Definir una clase para representar micros. Un micro se caracteriza por su patente, destino, hora salida, una cantidad de asientos fija (20) para los que debe registrar su estado (es decir si está o no ocupado) y cantidad de asientos ocupados. El micro puede ocuparse total o parcialmente.

- Defina métodos getters/setters para patente, destino y hora de salida.
- Implemente un constructor que permita inicializar el micro creado con una patente, un destino y una hora de salida (recibidas como parámetro) y sin pasajeros.
- Implemente métodos para:
 - devolver la cantidad de asientos ocupados
 - devolver si el micro está lleno
 - validar un número de asiento recibido como parámetro
 - devolver el estado de un nro. de asiento válido recibido como parámetro
 - ocupar un nro. de asiento válido recibido como parámetro
 - liberar un nro. de asiento válido recibido como parámetro
 - devolver el nro. del primer asiento libre

B- Realice un programa ppal. que cree un micro con patente “ABC123”, destino “Mar del Plata” y hora de salida “5:00 am”. Cargue pasajeros en el micro de la siguiente manera. Leer nros. de asientos desde teclado que corresponden a pedidos de personas. La lectura finaliza cuando se ingresa el nro. de asiento -1 o cuando se llenó el micro. Para cada nro. de asiento leído debe: validar el nro; en caso que esté libre, ocuparlo e informar a la persona el éxito de la operación; en caso que esté ocupado informar a la persona la situación y mostrar el nro. del primer asiento libre. Al finalizar, informe la cantidad de asientos ocupados del micro.

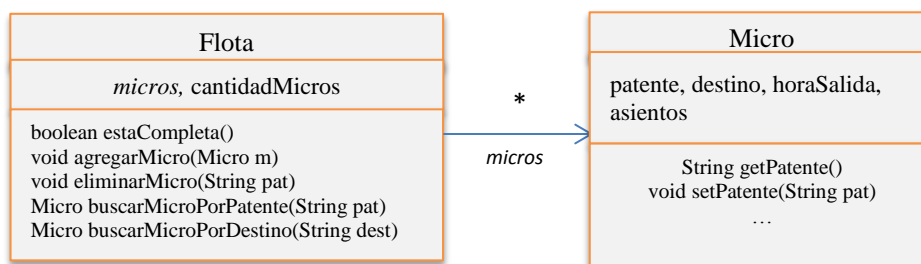
Adicionales

5-A- Definir una clase para representar flotas de micros. Una flota se caracteriza por conocer a los micros que la componen (a lo sumo 15).

- i. Defina un constructor para crear una flota vacía (sin micros)

Implemente métodos para:

- ii. devolver si la flota está completa (tiene 15 micros) o no.
- iii. agregar a la flota un micro recibido como parámetro.
- iv. eliminar de la flota el micro con patente igual a una recibida como parámetro.
- v. buscar en la flota un micro con patente igual a una recibida como parámetro y retornarlo (en caso de no existir dicho micro, retornar null).
- vi. buscar en la flota un micro con destino igual a uno recibido como parámetro y retornarlo (en caso de no existir dicho micro, retornar null).



B- Genere un programa que cree una flota vacía. Cargue micros (sin pasajeros) a la flota con información leída desde teclado (hasta que se ingresa la patente “ZZZ000” o hasta completar la flota). Luego lea una patente, busque si existe un micro con dicha patente y en ese caso elimínelo de la flota. Lea un destino e informe la patente del micro que va a dicho destino y su hora de salida.

6-A- Defina constructores para la clase Círculo (definida en la Act. 3): el *primer constructor* debe recibir un valor para el radio y para los colores de línea y relleno; el *segundo constructor* no debe poseer parámetros ni código (constructor nulo).

B- Realice un programa que instancie un círculo mediante los distintos constructores.