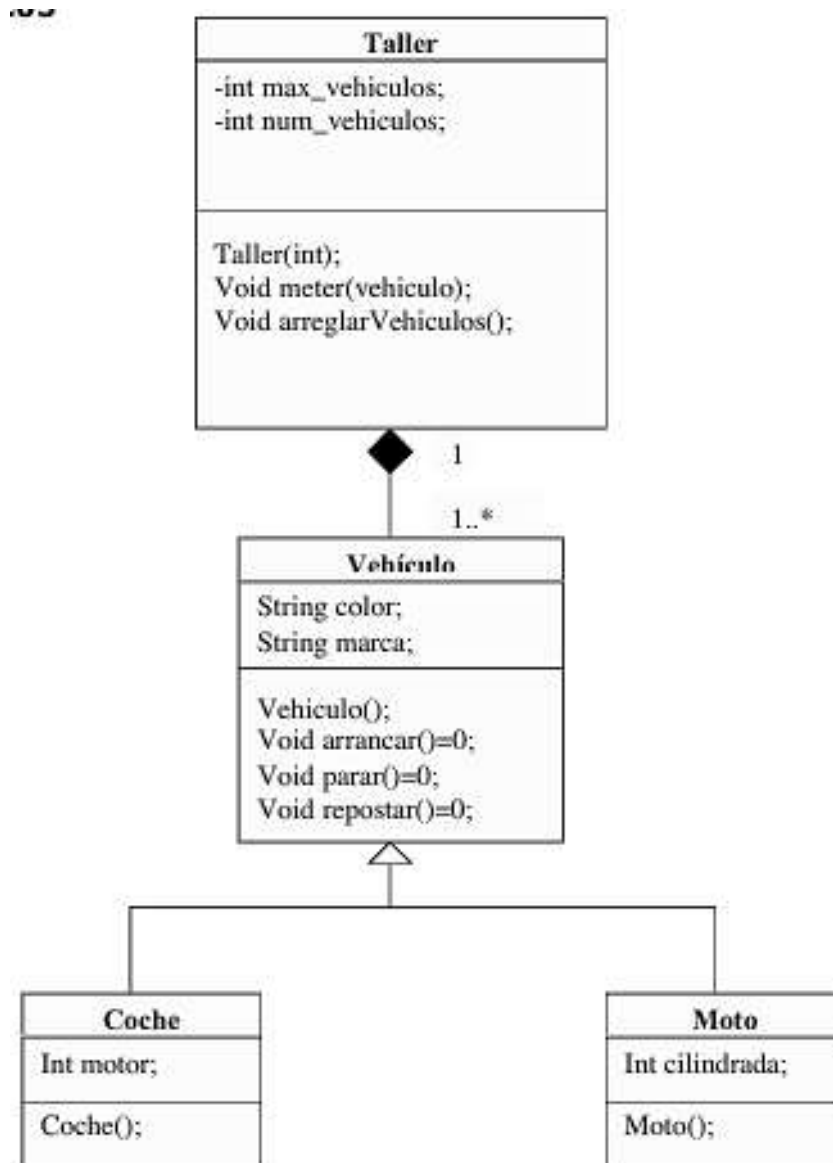


Módulo Java

Ejercicios de Refuerzo

Ejercicio 1

Escribe el código Java que corresponde al siguiente diagrama de clases.



Ejercicio 2

Define una función genérica llamada 'Intercambio' que permita intercambiar el valor de dos objetos del mismo tipo. Indica qué restricciones debe cumplir el tipo de los objetos para poder usar la función Intercambio con ellos.

Ejercicio 3

Define una plantilla **resta()** que permita restar dos objetos de cualquier tipo primitivo incluyendo las cadenas, indicando que requisitos deben cumplir dichos objetos para que la plantilla funcione correctamente. La resta de cadenas consiste en que se eliminan las letras comunes de la primera y que aparecen en la segunda.

Ejercicio 4

En el siguiente programa en Java, se define una clase genérica 'Vector' que permite crear vectores de objetos de cualquier tipo, añadir elementos al vector, obtener el elemento en una determinada posición y obtener el número de elementos que contiene el vector y su capacidad máxima. De este tipo de vector debe crear dos: el primero, se crea con tamaño fijo y el segundo, de dinámico (cuando llega a su límite crecen en proporción de 2 veces cuando es menor o igual a 1000 elementos y de 1.5 cuando es superior)

Esta clase vector no usa ninguna de las clases Collections. El vector debe ser gestionado con sus propios algoritmos.

Pueden producirse ciertos errores en tiempo de ejecución que preferiríamos poder controlar mediante excepciones. Se pide:

- a) Cuando se intente añadir un objeto a un vector que ya está lleno, debe lanzarse una excepción que indique la capacidad máxima del vector y que ésta ha sido superada.
- b) Cuando se intente acceder a una posición que no contiene ningún objeto previamente añadido con el método 'addElement', se debe producir una excepción que, al ser capturada, permita conocer la posición a la que se intentó acceder y el número máximo de posiciones válidas.
- c) En el programa principal se debe capturar cualquier otra excepción imprevista, y emitir un mensaje de error genérico.

Nota: Para los puntos a y b, es aconsejable que crees tus propias clases de excepciones. Los objetos que se lanzan como excepción deben contener la información sobre el acceso incorrecto. Además, debes añadir tanto el código que lanza las excepciones (en la clase 'Vector') como el código que las captura (en el programa principal).

Ejercicio 5

Diseña e implemente una aplicación en Java de inventario y consulta de libros de una pequeña biblioteca.

En la biblioteca existe una lista de Libros ordenada por título. Esta lista permite: agregar el libro, conocer la cantidad de libros que hay, consultar los libros, la cantidad de copias de cada libro (cada libro es independiente). También se puede dar de baja a los libros por pérdida, daño o obsolescencia.

Diseña el diagrama UML de la aplicación y construya el programa que implemente todas las operaciones que se indican.