

Librería

El objetivo de este control es continuar la familiarización con la definición de TADs específicos apoyados en TADs de propósito general.

1) El problema

Se desea implementar un TAD que permita realizar la gestión básica de una tienda de libros virtual. Dicho TAD soportará las siguientes operaciones:

- `crear`: $\rightarrow L$. Crea una librería L con un catálogo de libros vacío (constructora)
- `an_libro`: $L \times l \times n \rightarrow L$. Añade n ejemplares del libro l a la librería. Si el libro no está, lo da de alta con el número indicado de ejemplares. Si el libro está, suma el número indicado de ejemplares a los ya existentes.
- `comprar`: $L \times l \rightarrow L$. Registra la compra de un ejemplar del libro l en la librería, restando 1 al número de ejemplares de l disponibles. Son situaciones de error: (i) que el libro l no esté dado de alta en el catálogo, (ii) que el número de ejemplares disponibles sea 0.
- `esta_libro`: $L \times l \rightarrow \text{bool}$. *Cierto* si l está dado de alta en el catálogo de la librería, *false* en otro caso.
- `elim_libro`: $L \times l \rightarrow L$. Elimina el libro l de la librería. Si el libro no estaba dado de alta, la operación no tiene ningún efecto.
- `num_ejemplares`: $L \times l \rightarrow |N$. Devuelve el número de ejemplares disponibles para el libro l . Es un error si l no está dado de alta.
- `num_libros`: $L \rightarrow |N$. Devuelve el número de libros distintos dados de alta en la librería (**nota:** el valor devuelto es el número de libros distintos que contiene el catálogo de la librería, independientemente del número de ejemplares disponibles de cada libro)

2) Trabajo a realizar

Debe construirse un programa que lea órdenes de operación sobre la librería, una por línea, y ejecute dichas órdenes. Las órdenes posibles, junto con las respuestas dadas por el programa, son las siguientes:

- `an_libro l n`: Añade n ejemplares del libro l a la librería. Una vez hecho esto, imprime una línea con el número total de libros disponible en la librería.
- `comprar l`: Registra una compra de l . Una vez registrada la compra, imprime una línea con el número total de ejemplares de l . Si no puede registrarse la compra, debido a algún error, en su lugar imprime: (i) `LIBRO_NO_EXISTE`, si el libro no está dado de alta, (ii) `NO_HAY_EJEMPLARES` si, aunque el libro sí está dado de alta, el número de ejemplares disponibles es 0.
- `comprar l`: Imprime, en una línea, *true* si l está dado de alta, o *false* en otro caso.
- `elim_libro l`: Elimina l del catálogo, e imprime, en una línea, el número de libros que continúan estando registrados en el mismo.
- `num_ejemplares l`: Imprime el número de ejemplares de l . Si l no está dado de alta, imprime `LIBRO_NO_EXISTE` en su lugar.

Ejemplo de entrada / salida:

Entrada	Salida
<code>an_libro el_quijote 4</code>	1
<code>comprar el_quijote</code>	3
<code>an_libro la_colmena 1</code>	2
<code>comprar la_colmena</code>	0
<code>comprar la_colmena</code>	NO_HAY_EJEMPLARES
<code>num_ejemplares el_quijote</code>	3
<code>num_ejemplares la_colmena</code>	0
<code>esta_libro el_quijote</code>	true
<code>elim_libro el_quijote</code>	1
<code>esta_libro el_quijote</code>	false

Se proporciona el archivo `main.cpp` en el que se implementa la lógica de entrada/salida necesaria. El código proporcionado no debe modificarse.

Se proporciona, así mismo, el archivo `Libreria.h`, con la definición de las operaciones de TAD. Dicho archivo debe completarse eligiendo una representación apropiada para gestionar la información de la librería.

Las operaciones del TAD deben implementarse, en el archivo `Libreria.cpp`.

Se proporciona, por último, la clase `Libro`, que representan los libros que se registran en el catálogo. Si fuera necesario, deberá añadirse a `Libro.h` los métodos necesarios para que esta clase *funcione* con la representación elegida (dichos métodos se implementarán en `Libro.cpp`).