

Siglo XXXIII. Transporte aereo

La compañía de viajes espaciales *Viaje seguro* tiene una serie de naves espaciales con las que transporta mercancías y viajeros. Cada nave está equipada con distintos módulos (motores, cabinas, escudos, láseres etc.), dependiendo de su finalidad. Cuando las naves están en el hangar se procede a realizar las tareas de mantenimiento. Estas tareas pueden añadir nuevos módulos o reparar los ya existentes. Para conocer si un módulo se debe reparar, cada módulo lleva incorporado el nivel de funcionalidad en que se encuentra. Los módulos se incorporan a la nave con un cierto nivel de funcionalidad y según se van usando se pueden estropear, lo que supone una pérdida de funcionalidad. Cuando se realiza una operación de mantenimiento sobre un módulo que ya está en la nave se incrementa su nivel de funcionalidad. No hay límite superior a la funcionalidad de un módulo.

La aplicación hará uso de un tipo `id_nave` y otro `id_modulo` de tipo `std::string`.

Las operaciones son las siguientes:

- **alta(nave)**: añade una nueva nave espacial (vacía) al sistema con el identificador **nave**. Si la nave ya está en el sistema se lanza una excepción con mensaje **Nave repetida**.
- **mantenimiento(nave,modulo,f)**: Dado el identificador de una nave, un nombre de módulo y un nivel de funcionalidad (entero > 1) añade el módulo correspondiente a esa nave con el nivel indicado. Si la nave ya tenía ese módulo se suma el nuevo nivel al anterior (esto permite reparar módulos de naves). Si la nave no está en el sistema se lanza una excepción con mensaje **Nave desconocida**.
- **estropear(nave, modulo)**: dado el identificador de una nave, y un nombre de módulo, resta uno al nivel de ese módulo en esa nave (asumiendo que tuviese un nivel positivo). Si el identificador de la nave no existe o el módulo no pertenece a la nave se lanza una excepción con mensaje **Nave desconocida** o **El modulo no pertenece a la nave**.
- **defectuosas()**: Devuelve una lista de identificadores de naves que tienen uno o más módulos completamente estropeados (nivel de funcionalidad cero) en el orden en que fueron perdiendo alguna funcionalidad. Si ya habían perdido una funcionalidad y pierden otra mantienen el orden que tuviesen. La operación debe tener coste del orden del número de naves que tienen módulos completamente estropeados.
- **modulos(nave)**: dado el identificador de una nave, devuelve una lista con los nombres de los módulos que tiene equipados (tengan el nivel que tengan), ordenada alfabéticamente.

Requisitos de implementación.

Seleccionar un tipo de datos adecuado para representar la información. En la cabecera de cada función debe indicarse el coste de la misma.

Los métodos del TAD no deben mostrar nada por pantalla. El manejo de la entrada y salida de datos se realizará en funciones externas al TAD.

Debe justificarse el tipo representante elegido y dar el coste de todas las operaciones

Entrada

La entrada consta de una serie de casos de prueba. Cada caso está formado por una serie de líneas, en las que se muestran las operaciones a llevar a cabo, una por cada línea: el nombre de la operación seguido de sus argumentos. La palabra FIN en una línea indica el final de cada caso.

Salida

Para cada caso de prueba se escribirán los datos que se piden. Las operaciones que generan datos de salida son:

- **defectuosas**, que debe escribir **Naves defectuosas** : seguido de los identificadores de las naves defectuosas separados por un carácter blanco.

- **modulos**, que debe escribir **Modulos de la nave** *identificador de la nave* : seguido de los módulos equipados en esa nave.

Cada caso termina con una línea con tres guiones (---).

Si alguna operación produce una excepción se mostrará el mensaje de la excepción como resultado de la operación.

Entrada de ejemplo

```
alta nave1
mantenimiento nave1 motor 2
mantenimiento cabina 1
alta nave2
mantenimiento laser 1
modulos nave1
modulos nave2
estropear nave1 motor
estropear nave2 laser
defectuosas
estropear nave1 motor
mantenimiento nave2 laser 2
defectuosas
alta nave1
FIN
```

Salida de ejemplo

```
Modulos de la nave nave1 : cabina motor
Modulos de la nave nave2 : laser
Naves defectuosas : nave2
Naves defectuosas : nave1
ERROR: Nave repetida
---
```

Autor: Facultad Informática (UCM)