

Programación Orientada a Objetos 2014-1

Profesora: M. Angélica Caro

(21-04-2014)

Objetivo: Uso de objetos como parámetros, uso de contenedores, lectura de múltiples datos por línea, Captura de excepciones.

Enunciado

Implemente en Java la aplicación representada por el diagrama de clases de la Figura 1.

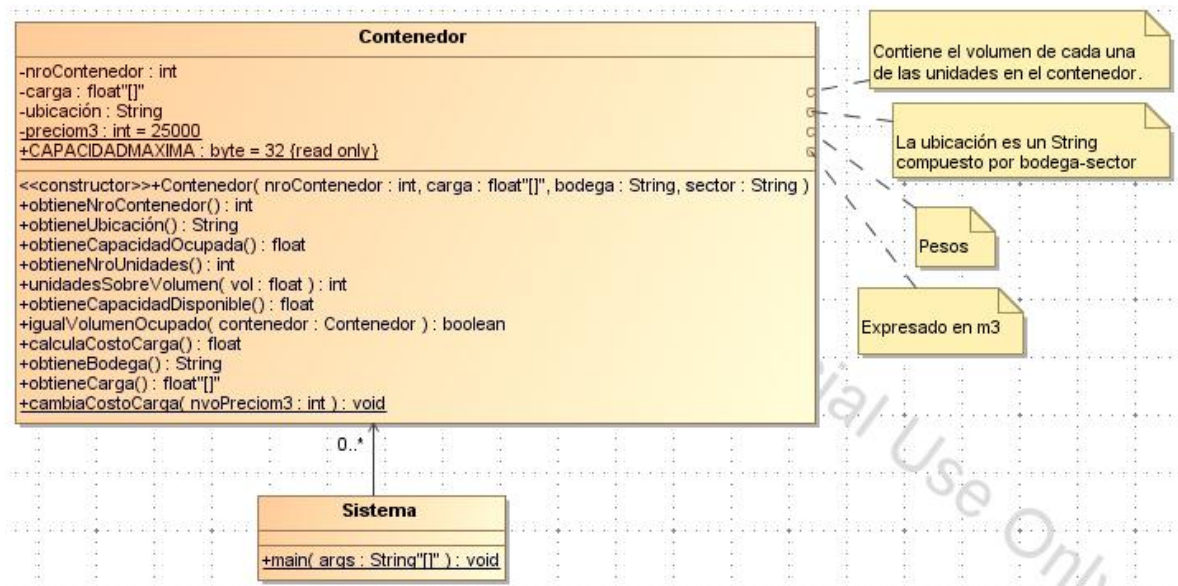


Figura 1. Diagrama de clases para un Sistema de Contenedores.

A. Clase Contenedor

Para el desarrollo de la **clase Contenedor** considere los siguientes antecedentes generales:

- Un contenedor se va llenando con distintas unidades (o paquetes) hasta que completa su capacidad máxima de almacenamiento en metros cúbicos (m3).
- Los contenedores se almacenan en bodegas antes de ser llevados a los barcos, y dentro de cada bodega en un sector. Por tanto la ubicación de un contenedor se compone del nombre de la bodega más el del sector.

Respecto de los atributos de la clase considere:

- El atributo `carga` es un arreglo cuyo tamaño dependerá del número de unidades guardadas en el contenedor. Por cada unidad se almacena su volumen en m3.
- El atributo `ubicación` contiene el nombre de la bodega y del sector, separados por un guion. Por ejemplo: "PUERTO-S3", "MONTERO-S12", etc.



Respecto del constructor y los métodos de la clase considere:

- El constructor permitirá crear un objeto contenedor, de acuerdo a los parámetros recibidos. El parámetro carga contiene los volúmenes de las unidades almacenadas en el contenedor.
- El método obtieneUbicación() retorna un String con la ubicación del contenedor usando mayúsculas.
- El método obtieneCapacidadOcupada () retorna los m3 ocupados por todas las unidades del contenedor.
- El método obtieneNroUnidades () retorna la cantidad de unidades almacenadas en el contenedor.
- El método unidadesSobreVolumen(...) retorna la cantidad de unidades que exceden el volumen indicado por parámetro.
- El método obtieneCapacidadDisponible() retorna los m3 que no están ocupados en el contenedor.
- El método igualVolumenOcupado(...) retorna verdadero si el objeto de la invocación tiene ocupado el mismo volumen total que el objeto recibido por parámetro.
- El método obtieneBodega() retorna un String con el nombre de la bodega donde se ubica el contenedor.
- El método calculaCostoCarga() retorna el total de pesos equivalentes al volumen total ocupado en el contenedor.
- El método obtieneCarga() retorna un arreglo con la carga actual del contenedor.

B. Clase Sistema (versión 1)

Respecto de la clase Sistema, considere los siguientes aspectos a considerar para su desarrollo:

1. Lea los datos de un conjunto de contenedores y, a partir de ellos, cree objetos contenedor y almacénelos en un objeto de tipo LinkedList. Los datos se ingresarán por teclado del siguiente modo:

```
nro. contenedor, bodega, sector, volumen unidad, [volumen unidad]...  
...  
nro. contenedor, bodega, sector, volumen unidad, [volumen unidad]...  
0
```

Donde:

- 0 indica el término del ingreso de los datos de contenedores.
 - La cantidad de unidades en la línea son variables, y al menos siempre viene una.
2. Despliegue los datos de todos los contenedores almacenados, de tal manera que se genere un informe como el siguiente:

CONTENEDORES

```
NroContenedor  Ubicación  Total Carga  
10 PUERTO-SECTOR1  8.6  
12 PUERTO-SECTOR3  23.600002  
15 PUERTO-SECTOR6  36.4
```



C. Clase Sistema (versión 2)

Para esta versión de la clase sistema considere la presentación de un menú al usuario con un aspecto similar al siguiente:

MENU SISTEMA CONTENDORES

```
-----  
1: Crea Contenedores  
2: Busca Contendor  
3: Genera Estadísticas  
4: Salir
```

Ingrese opción:

Donde:

1. La opción 1, lee los datos de un conjunto de contenedores y, a partir de ellos, crea los objetos contenedores y los almacena en un objeto de tipo LinkedList. Los datos se ingresarán por teclado del siguiente modo:

```
nro. contenedor, bodega, sector, volumen unidad, [volumen unidad]...  
...  
nro. contenedor, bodega, sector, volumen unidad, [volumen unidad]...  
0
```

Donde:

- 0 indica el término del ingreso de los datos de contenedores.
- La cantidad de unidades en la línea son variables, y al menos siempre viene una.

2. La opción 2, busca un contenedor dado un número leído desde teclado. Si no lo encuentra envía un mensaje apropiado. Si lo encuentra muestra la siguiente información del contenedor:

```
Contenedor : 10  
Ubicado en : PUERTO-SECTOR2  
Costo : 475000.0  
Numero Unidades : 3  
Volumen ocupado : 19.0  
Capacidad disponible: 13.0
```

3. La opción 3, despliega los datos de todos los contenedores almacenados, de tal manera que se genere un informe como el siguiente:

CONTENEDORES

```
NroContenedor  Ubicación  Total Carga  
10 PUERTO-SECTOR1  8.6  
12 PUERTO-SECTOR3  23.600002  
15 PUERTO-SECTOR6  36.4
```