



Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información

CI-2692 – Algoritmos y Estructuras II
Abril-Julio 20146

Proyecto I (10%)

1. Problema

Un grupo de físicos del CERN ha decidido retirarse de la investigación y mercadear un descubrimiento que piensa los hará increíblemente exitosos. Este grupo de físicos ha encontrado un fenómeno cuántico de portales que les permite apropiarse de “espacio volumétrico” construido en universos paralelos (multiverso), y anclarlos secuencialmente a nuestro universo mediante portales dimensionales. Dichos espacios son paralelepípedos (“*pisos tubulares*”) con capacidad de almacenar en columna un número limitado de items.

Los físicos determinan que el único negocio que podría aprovechar esta propiedad del espacio-tiempo es uno de almacenamiento de vehículos para espectáculos, por lo que fundan la compañía Parqueadero **Aleph-0**, con el eslogan “Puestos. Si no hay los fabricamos en el acto”.

Ahora, cada “*piso*” desafortunadamente tiene una sola abertura o “portal”, por lo que los vehículos se estacionan desde el “fondo” hacia el “portal” y se recuperan en sentido inverso. Dado que la capacidad de cada *piso* es limitada, sólo se puede estacionar un vehículo si existe espacio suficiente para el mismo. De no haber espacio suficiente para introducir un vehículo, se cierra el *piso* y se fabrica uno nuevo, enviando el *piso* cerrado a una cola de *pisos*.

En un momento dado se acaba el espectáculo y será necesario vaciar todos los pisos de estacionamiento, en el orden en que fueron creados.

1. Obligaciones

El énfasis en este proyecto es el **uso** reiterado de las Clases, en una jerarquía adecuada. Se deberán implementar las Clases primitivos *Pila* (de vehículos) y *Cola* (de pisos). Usar operaciones con **constructores**, **destructores**, **selectores** (*getXXX: objeto => valor*) y **mutadores** (*setXXX: objeto x valor => void*) para poder solicitar o cambiar el valor de cada atributo. Sólo se puede acceder individualmente a los diversos objetos usando las operaciones elementales de *Pila* y *Cola*.

a) Deberá crear un Clase **Vehículo**, con atributos de *longitud* en metros, *placa* alfanumérica, *modelo* alfanumérico, *año* (*anyo*) numérico y *color* (rojo, dorado, plateado, gris, azul, negro, verde, amarillo y blanco).

b) Deberá crear la Clase **Piso** como Pila de Vehículos, pero con atributos de *capacidad* (# de vehículos) y *ocupación* (# actual de vehículos \leq *capacidad*). Si esta lleno no se puede meter el vehículo en el estacionamiento, este se cierra y se envía a la Cola de Pisos cerrados y deberá crearse uno nuevo Piso vacío.

Dicha clase **Piso** tendrá a su vez como mínimo las operaciones:

- Estacionar:** *Piso x Vehículo => void*
Estaciona un vehículo en el piso, desde el fondo al portal, ajustando la *ocupación*.
- Retirar:** *Piso => Vehículo*
Retira un vehículo (si existe) del portal del piso. Si el piso queda vacío, se destruye.
- Cercano:** *Piso => Vehículo*
Devuelve el vehículo (si existe) más cercano al portal del piso.
- Cabe:** *Piso => boolean.*
Determina si existe espacio suficiente para almacenar un vehículo más.

c) Deberá crear la Clase **Estacionamiento**.

Dicha clase **Estacionamiento** tendrá a su vez como mínimo las operaciones:

- Estacionar:** *Vehículo => bool*
Estaciona un vehículo en el estacionamiento, devolviendo estacionado. Si el piso se llena, se cierra el piso, se añade a los pisos cerrados y se crea el siguiente piso.
- Retirar:** *=> Vehículo*
Retira el vehículo (que corresponda según el piso que este activo. Si el piso queda vacío, se destruye.
- Destruir:** Destruye el primer piso.
- Vaciar:** Vacía (si existe) todos los pisos de la Cola de Pisos Cerrados..

d) En la clase **Estacionamiento** proveerá un método **ProcesarLlegadas** : *nombreArchivo => void*, que leerá líneas de *eventos* de **nombreArchivo**, que deberá ser procesado secuencialmente.

El resultado de cada operación deberá escribirse en una línea en un archivo de traza, cuyo formato se proveerá.

Un evento consta de: *Código \t valor1 \t valor2 \t ... \t valorN \r* según contexto. El código puede ser

C \t nombre \r :

Crear Estacionamiento (primera instrucción siempre) y guardar la traza en el archivo *nombre*.

P \t placa \t longitud \t modelo \t anyo \t color \r :

Estacionar vehículo (crear vehículo y parquearlo)

K \r :

Vaciar Estacionamiento, cerrar el archivo de traza y finalizar

Se le proveerá de casos de prueba como diferentes archivos de eventos y los resultados esperados.

ACLARACIONES:

El constructor de Piso debe tener un parámetro capacidad, que se leerá del usuario y será único para todos los pisos. Se anexa un archivo, “andromeda.txt”, que contiene un ejemplo de llegadas, y una traza “traza.txt”.

Entrega: Lab07_XX-XXXXXX_YY-YYYYY.(rar, tgz ó zip), en el Moodle fecha < 16:30, 09/06/2016