Trabajo Práctico
Lavadero Industrial
(entrega final)

Estructuras de Datos
Tecnicatura en Programación Informática
Universidad Nacional de Quilmes

1. Introducción

El objetivo de este trabajo es utilizar las estructuras de datos vistas en la materia para representar la información utilizada por un lavadero industrial. El lavadero en cuestión se encarga básicamente de lavar prendas y sábanas con una modalidad particular, las prendas son propiedad del lavadero. Es decir que el lavadero presta prendas y sábanas que luego entran en el circuito de lavado. Realizaremos una implementación en el lenguaje `C++` de lo mencionado.

Habiendo implementado a los clientes del lavadero y el TAD Map que almacenará a los mismos, procederemos a implementar otras estructuras que completarán este dominio.

2. Lavadero

Modelaremos esta vez el lavadero en su totalidad. Un lavadero es simplemente un Map que relaciona CUITs con clientes, y además agregaremos un conjunto de planes, en los que sabremos para cada plan qué clientes posee actualmente.

2.1. Clientes

Los clientes actualizan su estructura. Ahora contarán también con una cantidad de prendas asignadas. Esta cantidad podemos incrementarla o decrementarla, y eso influirá en el plan que paga el cliente. Para el cliente el plan es simplemente un nombre, es decir, un simple string que indica en qué plan se encuentra.

El código del cliente ya se encuentra implementado, por lo que sólo resta ser usuario del mismo.

2.2. Planes

Existen varios tipos de planes. Un cliente al ingresar al lavadero entra con un plan llamado ïnicial", y a medida que incrementa su número de prendas, su plan se irá actualizando de forma automática. Existe una tabla, implementada como una función, que indica para cierta cantidad de prendas un plan.

La parte de planes cuenta con una estructura de datos propia, que debemos implementar. Dicha estructura es un Set implementado directamente como un array de tipo Plan. A su vez, un Plan será un nombre y una lista de clientes (sólo los CUITs) que indicarán qué clientes poseen dicho plan. La idea es que a medida que el lavadero aumenta la cantidad de prendas de un cliente, esta estructura se actualiza de forma automática, pasando un cliente de un plan al siguiente. Una vez alcanzado el plan máximo, ya nunca más será actualizado.

3. Fecha de entrega de entrega

La fecha de entrega es el lunes 12/12, a las 23:59. Cualquier entrega fuera de término significará la desaprobación automática del TP.

4. Forma de entrega

1. La entrega deberá realizarse vía mail a la dirección tpi-doc-ed@listas.unq.edu.ar
2. El asunto del correo debe tener la forma [entrega-lavadero][Juan Perez][C1] siendo "Juan Perez" su nombre y "C1" su comisión. Cualquier mail que no coincida con este formato no será tenido en cuenta. El formato del asunto es MUY importante.
3. El archivo a entregar será en formato zip, con el nombre del alumno en cuestión. Ejemplo: "JuanPerez.zip". Debe zipearse la carpeta que incluye todo el proyecto, aunque borrando primero las carpetas "binz .ºbj", que impedirán enviar por mail el trabajo. Debe respetarse este formato, o se volverá a pedir la subida del TP hasta respetarlo.

5. Pautas de evaluación

5.1. Se pide

1. Definir implementaciones eficientes utilizando exactamente la representación y las subtareas propuestas. NO debe modificarse ninguna parte del código ya implementado, y deben completarse solo las partes que indican çompletar".
2. Dar los órdenes de complejidad en peor caso y justificarlos.
3. Definir, en caso de existir, los invariantes de estas estructuras.

5.2. Se penaliza

1. Utilizar una representación distinta a la dada.
2. Modificar código dado.
3. No indicar propósitos, precondiciones, órdenes de complejidad e invariantes de representación.
4. Entrega fuera de término.
5. Incumplimientos de los propósitos de cada función.
6. Violar barreras de abstracción.
7. Incoherencia en las subtareas definidas.
8. No respetar la forma de entrega establecida.
9. Que el código no pueda compilar.
10. Que el código introduzca memory leaks.
11. Que el código introduzca bucles infinitos.
12. Utilizar conceptos que exceden el alcance de la materia (recursos de C++ que no se han visto).