Programación II - Trabajo Práctico Integrador
2do. Cuatrimestre 2023
SEGUNDA PARTE

Fecha de presentación: 19/10/2023(subido al Campus)

Fecha límite de entrega: 30/10/2023 (por el Campus)

En esta segunda parte deben entregar la implementación, el diagrama de clases actualizado, el análisis de complejidad en donde se pida, y el IREP para cada tipo de datos modelado en su solución. Para poder empezar con la segunda parte deben tener aprobado el diseño presentado en la primera parte.

Requerimientos técnicos:

i- Grupos: **El mismo grupo de la primera parte del TP.** Si hay alguna modificación debe ser aprobada por sus docentes de la comisión.

ii- Se deben utilizar donde sea conveniente las herramientas de Tecnologías Java que se vieron en la materia. Al menos una vez deben usarse:

- *Stringbuilder,* cuyo uso debe basarse en la necesidad de modificar el string.
- *Iteradores y Foreach* para recorrer las colecciones de Java

iii- Se deberá utilizar en el desarrollo del trabajo **herencia y polimorfismo,** y al menos 2 de estos conceptos: **sobreescritura, sobrecarga e interfaces**. Como también, en los casos que corresponda, se deberá implementar **clases/métodos abstractos**.

iv- En el informe (documento) se debe explicar qué conceptos se utilizaron, dónde y de qué modo.

v- **Escribir el *IREP* de la representación elegida para la implementación**. Es parte de la documentación.

Por otro lado, desde la materia se proveerá

a) Un código cliente y la explicación de sus métodos para que se utilicen como base para la implementación, **NO SE DEBE MODIFICAR**.
b) Una clase de testeo (junit). Será condición necesaria para aprobar esta parte del trabajo que tanto el código cliente como el test se ejecuten sin errores.

- Además de pasar el test de *junit* suministrado junto con el TP, en la corrección se testean los ejercicios con otro junit adicional, por lo que se recomienda que el grupo arme un conjunto propio de testeo acorde a su implementación, antes de entregar el TP. Puede entregarlo también si lo desea.

La entrega se realiza subiendo al Campus el proyecto de Java con su implementación **(Seleccionar solamente los archivos .java)** Se debe subir la documentación con los puntos pedidos previamente por escrito en un archivo Word en lo posible o PDF, junto con el proyecto.

Consideraciones importantes para la implementación y la documentación del trabajo:

La implementación de los TADs debe responder a su diseño presentado en la Primera Parte ***teniendo en cuenta las correcciones que se indicaron/indicarán a cada grupo*. Además, será condición necesaria para aprobar que se cumpla con:**

- Deberá correr satisfactoriamente con el código cliente entregado
- Deberá pasar satisfactoriamente el test junit proporcionado.
- Deberá aprovechar correctamente las estructuras de datos elegidas.
- El código deberá tener implementado el método ***toString*** del TAD principal, lo que implica que se deban implementar los toString de los TADs relacionados.
- Se deberá usar herencia, polimorfismo y abstracción.

Algunas modificaciones y aclaraciones al enunciado de la primera parte para facilitar la implementación:

- Se agrega el **DNI** del cliente como un dato más del mismo.
- No se considerará en esta parte el punto 5 de la parte 1.
- Los *utilitarios* tienen un valor extra que cobran a la empresa por cada viaje si la entrega supera los **3 paquetes** y no 10 como dice la primera parte.
- El listado de paquetes cargados se genera y se devuelve al cargar un transporte.
- Se agrega una operación **que cierre un pedido** dado su código.
    - *Los transportes se cargan con paquetes de pedidos cerrados.*
    - *La facturación total de los pedidos se hace con los pedidos cerrados.*
    - *El listado de paquetes no entregados se los toma de los pedidos cerrados.*
- Los paquetes del carrito de un pedido, una vez que esté cerrado, **no se cargan necesariamente en un mismo transporte**. Pueden estar distribuidos en diferentes transportes. O pueden estar algunos cargados y otros a la espera de cargarlos.
- En el punto 11 de la primera parte se modifica: "*Devolver un listado con los códigos de pedidos que no fueron entregados en su totalidad aún. Identificando a qué cliente corresponden*." por esta solicitud "***Devuelve los pedidos cerrados y cuyos paquetes no fueron entregados en su totalidad***."
- ***Se agrega el requerimiento de decir si la Empresa tiene dos camiones idénticos***.

Con las observaciones anteriores, se deberá implementar el TP, teniendo en cuenta las siguientes condiciones :

1) Para implementar los siguientes puntos pedidos en la primera parte:
    - 9. Dado un transporte cargado devolver el costo que debe pagar la empresa por ese viaje.
    - 10. Calcular el total de facturación de los pedidos registrados y terminados.

    Se deben responder todos en O(1).
2) La carga de un transporte devuelve un listado de los paquetes cargados creando un String con forma de listado donde cada renglón representa un paquete.

    " + [NroPedido - codPaquete] dirección"

    por ejemplo:

    " + [1002 - 101] Gutierrez 1147"

3) Se agrega

- ***Decir si hay dos transportes idénticos.***

Devuelve True si encuentra dos transportes cargados que son iguales, considerando que serán iguales si:

- *Tienen diferente patente, pero*
- *son de la misma clase y*
- *tienen carga idéntica,*

… aunque el volumen máximo sea distinto.

La carga será idéntica si tienen la misma cantidad de paquetes y los paquetes coinciden en volumen, clase y precio, además de los atributos propios de cada clase de paquete.

También se deberá entregar en el documento el siguiente análisis de la complejidad:

Explicar la complejidad y justificar por medio de Álgebra de Órdenes para el punto

4. Quitar un paquete del carrito de un pedido.

Test (JUnit):

Se habilitará el archivo de test en el Moodle, junto a este enunciado, de donde deberán descargarlo. Se avisará en cuanto esté disponible.