

Leé con cuidado el enunciado y por lo menos dos veces para resolver lo pedido. Pensá bien la estrategia de resolución antes de comenzar el desarrollo de lo que te solicitan. El objetivo de este examen es evaluar la correcta aplicación de los conceptos y técnicas vistos hasta el momento:

- Correcta implementación de constructores.
- Correcta definición de clases y asignación adecuada de sus responsabilidades.
- Encapsulamiento, ocultamiento de información y uso de getters y setters sólo cuando corresponda.
- Modularización reutilizable y mantenible con uso de métodos con correcta parametrización.
- Correcta aplicación de miembros de instancia y de clase.
- Correcta aplicación de herencia y polimorfismo.
- Correcta aplicación conceptual de las relaciones entre clases.
- Correcta aplicación de TADs vistas en clase
- Importación y exportación de proyectos desde Eclipse.

Antes de comenzar [descargá el proyecto adjunto](#), importalo desde Eclipse y renombralo con el siguiente formato:

TP1-2025C1-P2-SEDE-CURSO-APELLIDO-NOMBRE

Ejemplos:

TP1-2025C1-P2-SEDE-CURSO-Cohen-Carlos

TP1-2025C1-P2-SEDE-CURSO-Ramirez-Octavio

Este proyecto deberá ser exportado al entregar con este nombre. Tené en cuenta que esto también forma parte del examen y se califica.

NOTA IMPORTANTE:

- No te quedes con dudas, preguntá lo que haga falta.
- Al entregar el examen asegurate de finalizar la entrega. Y de ser necesario, porque recibiste algún error, confirmá la recepción con tu profesor.
- No se aceptarán exámenes por fuera del aula virtual (por email por ejemplo).

Se invalidarán automáticamente los exámenes:

- Donde no esté entregado un archivo con extensión .zip que contenga el proyecto en JAVA generado desde Eclipse.
- Donde se detecte el uso de IAs para la resolución.
- Que utilicen estructuras de datos no explicadas en la materia.
- Que utilicen las estructuras de control no contempladas por las buenas prácticas enseñadas durante el cuatrimestre.
- Donde se detecte cualquier actividad grupal y/o no lícita en la resolución del ejercicio (serán sancionados todos los participantes).

- Cuando el alumno no tenga la cámara prendida y/o su cara no sea claramente visible por el docente.

Enunciado

Una empresa de sushi delivery ofrece combos prediseñados que los clientes pueden pedir a domicilio. La empresa gestiona una flota de repartidores, cada uno con su propia cola de pedidos. Cada pedido puede contener múltiples combos, y el sistema permite calcular el importe total según la dificultad de preparación, tipo de combo, promociones y agregados.

El negocio busca automatizar el proceso de registro de repartidores, carga de pedidos y cálculo de precios, priorizando que los pedidos se repartan equitativamente entre los repartidores disponibles.

Sistema ya implementado

El sistema incluye las siguientes clases:

- **Combo** (abstracta): representa un combo de sushi, con nombre, número, cantidad de piezas y dificultad de preparación. Tiene dos subtipos:
 - **ComboEspecial**: puede tener un descuento si es “solo roll”.
 - **ComboPremium**: puede tener un recargo si incluye salsas.
- **ComboPedido**: representa la cantidad pedida de un determinado combo en un pedido.
- **Pedido**: mantiene una pila de **ComboPedido** y permite calcular el importe total del pedido.
- **Repartidor**: mantiene una cola de pedidos. Permite buscar pedidos por número.
- **Comercio**: se encarga de registrar repartidores, definir los combos disponibles, asignar pedidos y realizar operaciones generales.
- **RepartidoresPorID**: una lista ordenada por ID que permite buscar repartidores fácilmente.
- **Test**: clase con un ejemplo de ejecución. Incluye métodos incompletos o marcados con **TODO** que deben ser desarrollados.

Objetivo

A partir del código entregado, completá la implementación de las funcionalidades pendientes. En algunos casos, deberás modificar métodos existentes y en otros, agregar lógica o nuevos métodos auxiliares para cumplir con los requisitos indicados.

Se espera que utilices estructuras auxiliares y excepciones donde sea necesario para mejorar la claridad, legibilidad y robustez del código.

Funcionalidades a implementar

- Registro de repartidores
El método `darElPresente(Repartidor)` de la clase **Comercio** debe registrar al repartidor solo si no fue previamente ingresado, validando por ID. En caso de que ya exista, se debe informar por consola.

- **Asignación de pedidos**
El método `hacerPedido` debe ubicar al repartidor con menos pedidos asignados (o el primero que esté libre) y crear y asignar a éste el nuevo pedido, que debe contener el combo y la cantidad recibidos como parámetro.
Este método debe retornar el número de pedido generado.
- **Agregado de combos a pedido existente**
El método `agregarAlPedido` debe permitir agregar más combos a un pedido ya existente. Si el combo ya estaba presente en el pedido, se deben acumular las cantidades en lugar de agregarlo nuevamente como un ítem nuevo.
- **Informar el precio total de un pedido**
El método `informarPrecioPedido` debe mostrar por consola el precio total del pedido correspondiente al número recibido.

Excepciones

Se espera que uses excepciones donde se considere necesario, por ejemplo:

- Al intentar agregar un combo a un pedido que no existe.
- Al intentar acceder a un número de combo inválido.
- Al no haber repartidores disponibles.

El manejo de errores debe hacerse de forma clara y debe mantener la ejecución del programa.

Escenario de prueba

La clase `Test` incluye un escenario de ejemplo donde:

- Se dan de alta a los repartidores que se presentaron a trabajar.
- Se realizan pedidos y se les agregan combos.
- Se imprime el importe total de los pedidos.

Tu solución debe ser compatible con ese código y permitir que se ejecute correctamente una vez completadas las tareas indicadas.

Ejemplo de salida esperada

```
Ya existe un repartidor con id R002
1 'Especial Temaki-Sashimi-Inari' agregado al pedido 1.
1 'Especial Temaki-Sashimi-Inari' agregado al pedido 1.
Importe del pedido 1: 48000.0
1 'Especial Maki-Nigiri' agregado al pedido 2.
1 'Especial Maki-Sashimi-Nigiri' agregado al pedido 2.
1 'Especial Temaki-Sashimi-Inari' agregado al pedido 2.
1 'Supermaki Con Salsas' agregado al pedido 2.
1 'Supermaki Sin Salsas' agregado al pedido 2.
1 'Variedad Con Salsas' agregado al pedido 2.
1 'Variedad Sin Salsas' agregado al pedido 2.
Importe del pedido 2: 111719.99998390675
```

La entrega del examen debe completarse antes de cumplido el plazo establecido y debes asegurarte de terminar el intento y completar la entrega con “Enviar todo y Terminar”.