**Prueba de recuperación 3º Evaluación Programación. Colecciones. Junio 2022:**

Una empresa de mensajería quiere crear una aplicación que le permita hacer un seguimiento de los envíos que se realizan recogiendo información sobre los paquetes, su estado y fecha de entrega, los mensajeros responsables de estos y la satisfacción de sus clientes.

Para ello, hemos un elaborado un diagrama de clases y parte de la aplicación en Java y necesitamos tu ayuda para construir varias funcionalidades que nos han pedido:

* **mostrarEnviosPorEstadoEntrega**: recibe el estado de los envíos (uno de los siguientes estados: ENTREGADO, EN\_REPARTO, SIN\_ASIGNAR, DEVUELTO) y muestra información en el siguiente formato ordenada por nombre de empleado:

| Envíos en reparto:   * Envío con id **A3567** asignado a **Manolo Pérez** y fecha prevista de entrega **15-01-2022** compuesto por **2** bultos. * Envío con id {id} asignado a {mensajero} y fecha prevista de entrega {fecha} compuesto por {número bultos} bultos. |
| --- |

* **mostrarEnviosPorFechaPrevistaEntrega**: muestra todos los envíos existentes agrupados por fecha de entrega e id creciente (sugerencia: ordena la colección por fecha e id, o bien, genera un mapa donde las fechas de los envíos sean las claves):

| Fecha prevista de entrega 01/12/2022:   * Envío con id A1123. * Envío con id A3567.   Fecha prevista de entrega 05/12/2022:   * Envío con id B1456 * Envío con id C8547 |
| --- |

* **addEnvio:** añade un envío a la colección ***envios*** después del último envío ya registrado para esa fecha de entrega. Si no existiera ningún envío para esa fecha se añade en el lugar que le corresponde por fecha.
* **asignarPaqueteAMensajero:** esta función ayuda a que todos los mensajeros tengan la misma carga de trabajo, por lo que calcula cuál es el mensajero que menos envíos ha realizado hasta el momento actual, es decir, obtiene de todos los mensajeros aquel que ha entregado menos envíos, y le asigna una orden de entrega, añadiendo éste a la lista de envíos.

Un paquete solo se puede asignar a un mensajero que esté trabajando en la empresa, es decir, aquellos cuya fecha de alta es anterior o igual a la fecha prevista de entrega del envío y que aún estén trabajando (fecha de baja nula).

La fecha prevista de entrega será igual a la fecha de entrada del paquete + 5 días.

El envío se insertará en la colección justo después del último envío existente para esa misma fecha.

* **valorarEntrega:** recibe el código de un envío y asigna una puntuación entre 0 y 10 a la valoración de la entrega. Si el envío no existe se lanzará una EnvioException.
* **obtenerMejorEmpleado:** devuelve aquel empleado cuyas entregas tienen la mayor puntuación media (suma de la valoración de todos sus envíos/total de envíos del empleado).
* **obtenerPrimerEnvioParaEmpleado**: devuelve cuál ha sido el primer envío ENTREGADO por un empleado. Devuelve una excepción si no existe tal empleado.

⇒ Puedes añadir los métodos que consideres necesarios.

⇒ Puedes utilizar la clase MainApp con los datos de muestra y el menú.

