Explicación del Código del Programa

Lo primero que he desarrollado es la clase “paquete”, donde almacenar la información propia de cada uno y poder trabajar con objetos para cada paquete. En el constructor, he implementado un método que calcula el coste de envío de cada uno en función de la información del enunciado. Así, cada vez que se registre un paquete automáticamente se calculara el coste en función de su peso. Además, esa clase tiene implementados los getters y los setters del atributo numero de seguimiento que es el que se utiliza para el método ObtenerPaqueteporN() de la clase principal. Por último, he desarrollado el método toString para imprimir la información de los objetos.

En la clase Principal, he desarrollado un ArrayList estático para almacenar todos los objetos “paquetes” que se registren y un Scanner estático para poder utilizarlo en todos los métodos de esta clase.

En el método main hay un menú para poder elegir las funcionales que quieras en función del número que introduzcas por consola. Todos los ingresos de números están desarrollados para que solo puedas introducir números y que sean mayor que 0. Gestionando los errores y los valores introducidos con los métodos gestionErrores() y numeroIntroducidoCorrecto(). El primero no permite que se metan parámetros erróneos como por ejemplo letras o palabras y el segundo, comprueba que el numero introducido sea mayor de 0.

**Funcionalidad 1 - Registrar un paquete**

El método registrarPaquete() pide que introduzcas por consola los datos necesarios para registrar un paquete. Estos son: su numero de seguimiento, su dirección y su peso. El método comprueba que el numero de seguimiento sea único y no se haya asignado a ningún otro paquete y el peso tiene que ser mayor de 0. Si todo es correcto se crea un objeto de la clase Paquete y se añade al ArrayList.

**Funcionalidad 2 - Listar todos los paquetes**

Este método comprueba que se haya registrado al menos un paquete y si es así, se recorre el ArrayList donde están almacenado e imprime toda su información. De no ser así, muestra un mensaje por consola de que no existen paquetes registrados.

**Funcionalidad 3 - Buscar un paquete**

Este método se utiliza para obtener la información de un paquete concreto, que se busca por su numero de seguimiento. Para ello se pide que se introduzca el numero solo si ya se ha registrado algún paquete antes. Una vez tienes el dato correcto, se llama a otro método que compara ese número con el de cada uno de los paquetes que estén añadidos en el ArrayList. Si hay alguno igual, devuelve ese objeto que se imprime por pantalla. Si no coincidiera con ninguno de la lista se devuelve un mensaje de aviso por pantalla.

**Funcionalidad 4 - Eliminar un paquete**

Este método cumple la misma operativa que el anterior. Lo que varia es que si existe un paquete registrado, con el mismo numero de seguimiento que se ha introducido este se eliminaría del ArrayList.