

## Tarea Repaso

### Enunciado

Vamos a crear un programa que almacena información sobre donaciones.

Recibiremos información sobre las donaciones que la ONG tiene previsto facturar, en el formato: `"date;document;ngo;projectCode;amount"`

Ejemplo: `"15/04/2023;12345678N;ACN;ACN8455;15.50"`

- **date:** Se refiere a la fecha en que se tiene previsto facturar el importe de la donación, en formato dd/mm/yyyy (dd: día con dos dígitos, mm: mes con dos dígitos, yyyy: año con cuatro dígitos).
- **document:** Se refiere al documento de identidad de la persona que hace la donación.
- **ngo:** Se refiere al código de la ONG que recibe la donación en formato CCC (CCC: código de 3 letras que identifica la ONG).
- **projectCode:** Se refiere al código del proyecto y tiene el formato CCCNNNN (CCC: código de la ONG, NNNN: código numérico de 4 cifras que identifica unívocamente el proyecto dentro de la ONG). Este código indica a qué proyecto o causa específica se realiza la donación, por ejemplo, para paliar el hambre en Somalia. Si la donación es genérica, no vinculada a una causa en particular, pero se destina a la ONG, entonces se usará el código de proyecto 0000 y tendremos el código CCC0000.
- **amount:** Se refiere a cantidad (en euros) aportada en la donación y que está previsto facturar, empleando siempre dos cifras para los céntimos (por ejemplo, podemos donar 10.00 o 45.99).

Crea una clase llamada `Donation`, que contenga la información de las donaciones.

Otra clase llamada `DonationData`, para guardar todos los datos de las donaciones que vayamos recibiendo ordenados **por código de proyecto**. Se puede asumir que se recibirán unas 5 o 10 donaciones.

Dentro de `DonationData`, implementa los siguientes métodos:

- donationDataInit:** se inicializa la estructura de datos donde se guardarán todas las donaciones.
- donationParse:** se recibe un string con el formato expuesto al inicio del enunciado, se parsea en una donación que se devuelve.
- donationDataAdd:** dada una donación, se insertará en nuestra estructura de donaciones manteniendo la ordenación por código de proyecto. Si ya existe una donación de la misma persona, para la misma fecha y proyecto, la donación se ignorará y no se añadirá. Recordamos que la persona se identifica con su documento de identidad y los proyectos tienen un código único que los identifica.
- donationDataGet:** dada una posición, obtiene la donación y la devuelve en formato String como está al principio del enunciado: `"15/04/2023;123..."`
- donationDataDel:** dada una fecha, documento de una persona y código de proyecto, elimina esa donación de la lista. Si no existe ninguna donación con esas características no hace nada.

Crea un main que realice las siguientes acciones:

- Añada 4 donaciones que tú quieras.
- Añada 2 donaciones más,
- Añade 1 donación ya existente.
- Elimina 2 donaciones existentes.

e) Elimina 1 donación NO existente.

Después de realizar estas acciones, ahora muestra un menú que permita realizar las siguientes operaciones sobre las donaciones:

1. Muestra la primera donación.
2. Salta a la siguiente donación.
3. Salta a la donación anterior.
4. Muestra la donación actual.
5. Elimina la donación actual.
6. Añade una nueva donación.
7. Imprimir todas las donaciones.

De las estructuras de datos vistas en la unidad de Colecciones, elige la que creas más adecuada para realizar la actividad. Razona tu elección.

## Entrega

Sube un PDF con el código copiado y preguntas respondidas.