

Nos contratan de la agencia de recaudación de la provincia de Buenos Aires para hacer un sistema para el cálculo del impuesto que deben pagar los contribuyentes.

El sistema ofrece la siguiente funcionalidad:

Dar de alta un contribuyente: Se provee nombre, dni, email y localidad. El sistema da de alta al contribuyente y lo retorna. El contribuyente no tiene ningún bien a su nombre.

Dar de alta un inmueble: se provee el número de partida, el valor del lote, el valor de la edificación, y el contribuyente (propietario). El sistema da de alta el inmueble y lo retorna.

Dar de alta un automotor: se provee patente, marca, modelo, fecha de fabricación, valor y el contribuyente (propietario). El sistema da de alta al automotor y lo retorna.

Dar de alta una embarcación: se provee patente, nombre, fecha de fabricación, valor y el contribuyente (propietario). El sistema da de alta la embarcación y la retorna.

Calcular el impuesto que debe pagar un contribuyente: dado un contribuyente, se debe calcular cuánto debe pagar de impuestos, según la siguiente especificación:

- Por cada inmueble que posea, el contribuyente debe pagar un 1% del valor del mismo, que se calcula sumando el valor del lote y el valor de la edificación.
- Por cada uno de los otros bienes, se debe pagar un porcentaje del valor de los mismos, en función de su fecha de fabricación.
 - En caso de superar los 10 años, no deben pagar nada.
 - En caso contrario, el porcentaje para un automotor es el 5% mientras que para una embarcación ese porcentaje varía según el valor de la misma. Si éste es menor a 1 millón, es el 10%, caso contrario, el 15%.

Contribuyentes que más pagan de una localidad: dada una localidad y un número N, se debe retornar los N contribuyentes de la localidad recibida que más deben pagar por sus bienes.

Su tarea es diseñar y programar en Java lo que sea necesario para ofrecer la funcionalidad antes descrita. Se espera que entregue lo siguiente:

1. Diseño de su solución en un diagrama de clases UML.
2. Implementación en Java de la funcionalidad requerida.

3. Implemente los tests necesarios para la funcionalidad de calcular el impuesto, justificando su elección en base a valores de borde y particiones equivalentes. Considere contribuyentes que solamente pueden tener embarcaciones.

Notas:

- Para calcular los años entre dos fechas puede utilizar la siguiente expresión

`ChronoUnit.YEARS.between(fecha1, fecha2);`

Donde fecha1 es anterior a fecha2. La expresión retorna la cantidad de años entre ambas fechas

- Implemente todos los constructores que considere necesarios.
- Puede implementar un getter y un setter, y asumir la existencia del resto.