**Leé el enunciado que aparece a continuación y resolvé los puntos solicitados. Te aconsejamos que lo leas completo y por lo menos dos veces.**

**Al terminar el ejercicio juntá los archivos Nassi-Shneiderman de NS+ (.nsplus,  ) con ambos archivos de UMLetino y adjuntalos al examen en un archivo .zip cuyo nombre será SEDE\_CURSO\_APELLIDO\_NOMBRE.zip (reemplazando cada parte por lo que corresponda, por ejemplo YA\_FPA\_PEREZ\_JUAN.zip).**

**Asegurate de entregar los archivos correctos y de cerrar el examen. La no entrega de alguno de los archivos invalida el examen.**

**Criterios de Corrección**

**Para considerar aprobado el examen, el mismo debe resolver lo pedido y aplicar los siguientes conceptos de la programación orientada a objetos:**

* **Detección de clases, atributos, métodos y relaciones (asociativas y de uso).**
* **Modularización reutilizable y mantenible usando métodos con correcta parametrización.**
* **Asignación de responsabilidades a cada clase y correcto encapsulamiento.**
* **Manejo del concepto de instancia y de la interacción entre objetos.**
* **Manipulación de listas de objetos (ArrayList) y su uso en ciclos condicionales y for-each.**
* **Manejo de diagramas Nassi-Schneiderman y UML de clases.**

**Enunciado**

**Sistema Electoral**

El **Poder Judicial de la Nación (PJN)** nos pidió desarrollar el sistema para verificar la inscripción de las personas en el Padrón Electoral Nacional.

El Padrón tiene todas las personas registradas para votar, de las cuales se sabe su Nombre, Apellido, DNI, género (Masculino, Femenino u Otro), Fecha de Nacimiento y Domicilio (Calle, Código Postal y Provincia), y también, tiene las escuelas habilitadas para votar, de las cuales se sabe su Nombre y Domicilio (Calle, Código Postal y Provincia).

La escuela no conoce a las personas que se pueden presentar a votar.

***Importante: hay una sola escuela habilitada por cada código postal.***

**Se Pide**

* Realizar el **Diagrama de Clases UML**, completando las clases, atributos, métodos y relaciones necesarias.
* Desarrollar el método ***agregarVotantes(…)*** que recibe una colección de personas y solo las agrega como votantes si hay una escuela existente asociada a su mismo código postal. Al terminar, el método deberá devolver la colección actualizada; en esta deberán quedar sólo aquellas personas que no se pudieron agregar como votantes.
* Desarrollar el método ***obtenerInforme()*** que recibe un nombre de Escuela y devuelve (no muestra por consola) la siguiente información relacionada a todas las personas habilitadas para votar en la escuela cuyo nombre coincida con el enviado como parámetro:
  + Nombre de la escuela: este dato quedará repetido en la lista por cada persona encontrada que vote en la misma escuela.
  + DNI de la persona.
  + Apellido y Nombre de la persona.

Ejemplo de posible informe:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Escuelas ORT | Escuelas ORT | Escuelas ORT | Escuelas ORT | Escuelas ORT | Escuelas ORT |
| 12345678 | 87654321 | 12341234 | 11122233 | 3322211 | 65498712 |
| Juan Perez | Mengana Torres | Ricardo Gomez | Andrea Bergara | Roberto Bolaños | Carlos Tripero |

**Nota**: Para ambos métodos requeridos desarrollar los métodos derivados que surjan y consideres necesarios.

Tamaño máximo de archivo: 100MB, número máximo de archivos: 1