

Uso de colecciones

Ejemplos para ver en clase

Durante la resolución de estos ejercicios iremos incorporando nuevos conceptos y técnicas. El formato está orientado a que sean los alumnos quienes diseñen la solución identificando clases, atributos y comportamientos en sucesivas lecturas de cada enunciado.

Para cada caso, se debe realizar el diagrama UML y escribir el código Java. Se deben desarrollar los métodos indicados por las funcionalidades y también cualquier otro que fuera necesario para cumplir con las mismas, en las clases en que corresponda.

1. **Mi agenda de contactos.** Queremos programar nuestra agenda de contactos. En ella agregaremos las personas que nos interesan. Los datos que tenemos de cada una son su nombre, apellido, número de teléfono, dirección de email y su domicilio (este domicilio tiene los mismos datos que el que usamos en el primer ejemplo). Además para cada contacto será necesario saber si se trata de un contacto personal o laboral. Debemos asegurarnos de que el email sea un atributo de solo lectura ya que será el que identificará únicamente a cada contacto. La agenda debe tener las siguientes funcionalidades:

- Agregar una persona verificando previamente que la misma no exista en la agenda (no se permiten duplicados). Para ello se recibirán los datos necesarios y se retornará verdadero o falso como resultado.
- Modificar el domicilio de una persona (recibiendo el identificador y su nuevo domicilio). Retorna verdadero o falso como resultado.
- Remover una persona (recibiendo su identificador. Retornará la persona borrada).
- Listar todas las personas por pantalla.
- Obtener la última persona agregada. Si la lista no está vacía retornará la persona, en caso contrario retornará nulo.
- Listar todas las personas que viven en un código postal determinado por pantalla.
- Obtener contactos laborales. Devuelve una colección que contenga solo los contactos laborales. Retorna una lista.

Además se debe crear una clase PruebaAgenda para comprobar el funcionamiento de las funcionalidades descriptas.

2. **El encargado** de un edificio está trabajando mal. Los vecinos del consorcio deciden hacer una encuesta para saber si despiden al encargado o no. Del edificio sabemos su dirección, la cantidad de departamentos y cada uno de los departamentos que lo componen. De cada departamento se tiene su número de unidad (desde 1 hasta la cantidad de dptos. del edificio), el dni y el nombre del propietario y su voto (ENBLANCO, POSITIVO o NEGATIVO).

Se debe poder agregar cada departamento con sus datos, verificando que el mismo no haya sido ingresado antes. Cuando un departamento se crea, su voto está con valor nulo. La acción de agregar debe retornar verdadero o falso según haya podido completarse o no.

Actualizar el voto: el voto de una unidad podrá actualizarse una sola vez, ingresando la unidad y el voto. Debe verificarse que dicha unidad exista. Si se pudo actualizar se retorna verdadero.

Se pide listar los propietarios que votaron por despedir al encargado y cuál fue el porcentaje de las unidades que participaron en la encuesta. Se emite por pantalla.

Se debe probar lo pedido desde una clase pruebaConsorcio.

3. Una **escuela primaria** tiene todos los grados del ciclo (actualmente siete pero podrían ser más). A su vez cada uno de ellos tiene una cantidad de divisiones que contiene distintas cantidades de alumnos. No todos los niveles tienen la misma cantidad de divisiones. Se necesita mostrar los datos de todos sus alumnos indicando el grado y la división a la que pertenecen. ¿Qué funcionalidades serían necesarias?

Para poder probar el programa se necesitará crear los grados que tiene el colegio, para cada grado sus divisiones y para cada una de ellas sus alumnos (20 para cada una, con un nombre que diga “alumno “+numero). Se deben hacer métodos tales como crearTodosLosGrados() y no crearUnGrado(),

El diseño del listado para mostrar por pantalla es el siguiente:

Escuela primaria NuestroCole

Grado: primero

División: A

Alvarez Carlos

Benitez Lola

.....

Grado: primero

División: B

López Antonio

Ramirez Ludmila

.....

Grado: segundo

Rolón Atilio

Ladrillazo Estela

Se debe probar lo pedido desde una clase pruebaEscuela.