

Programación orientada a objetos

Casos de estudio modelado de clases - PYTHON

TEST 01

- 1) Una empresa de servicios guarda es una lista las tareas que debe realizar cada empleado. La estructura es la siguiente:

EmpresaLista:

- Código Empleado (Clave ordenamiento)
- Cantidad de Tareas
- Cola de Tareas

ColaTarea:

- Área Solicitante
- Descripción

Realizar procedimientos para este TDA que:

- Permita Ingresar una nueva tarea en el empleado que tenga menos tareas.
- Permita Procesar las tareas de los empleados 15 a 30, sacándolas de la cola.

Aclaración: Cuando las estructuras están ordenadas por algún campo no se deben recorrer de más.

- 2) Implemente una solución que simule el funcionamiento de una caja de un banco, a la cual llegan personas que deben esperar en una cola a ser atendidas por el cajero (simule esta atención imprimiendo: 'La persona X ha sido atendida.').
- a) Que objetos puede identificar que serán necesarios implementar? Que responsabilidades tendrá cada uno de ellos?
 - b) Realice los diagramas UML de los mismos.
 - c) Implemente las clases del diagrama.
 - d) Codifique una clase **TestCajaBanco** que posea un método de clase llamado **main()**, el cual lea y cree 5 personas, las ponga en la cola a la espera de ser atendidos y lleve a cabo la atención de las mismas.

- 3) Un supermercado organiza las cajas con las siguientes estructuras:

TDA CAJAS:

- Lista de Cajas
- Cantidad de Cajas

Lista de Cajas:

- Numero de Caja (Clave ordenamiento)
- Recaudación
- Clientes atendidos
- Clientes por atender
- Cola de Clientes:
 - Nro de Cliente
 - Monto

Realizar procedimientos para este TDA que:

- Permita procesar todos los clientes de todas las cajas. Para ello se deberá sacar todos los clientes de cada una de las cajas, actualizando el monto recaudado en cada caso.
- Permita agregar un cliente en la caja con menor cantidad de clientes por atender.

4) Se decidió implementar el **TDA_PADRÓN_2023** para las próximas votaciones. La estructura de datos del TDA es la siguiente:

Fecha de última actualización:

- TDA_FECHA (última actualización del padrón)
- Lista de provincias: TDA_LISTA_PROVINCIAS. Cada elemento tiene:
 - Provincia (clave de ordenamiento de la lista)
 - Lista de votantes: TDA_LISTA_VOTANTES. Cada elemento tiene:
 - DNI (clave de la lista)
 - Apellido y nombre
 - Código lugar de votación
 - Mesa votación

Lista de lugares de votación: TDA_LISTA_LUGARES. Cada elemento tiene:

- Código lugar de votación (clave de la lista)
- Dirección Localidad

Dado el TDA_PADRÓN_2023, desarrollar la primitiva llamada “**Consultar_Lugar_Votacion**” que recibe como parámetro una variable provincia y el DNI de un votante, y devuelve una variable booleana que indica si encontró al votante y un registro con la dirección, localidad y mesa donde votará esa persona.