================================= TAD TESTDATA =============================== #Fecha: 17 de mayo de 2024

#Modificación:

#Autores: Piazza J., Ibañez F., Lucas M.

==============================================================================

cargarDatos(listadoTareas, listadoEmpleados, cola):

"""

Carga los datos de 15 empleados con sus tareas, descripción de la tarea,

nombre del empleado, estado de la tarea y fecha de creación

Args:

listadoTareas: una instancia de la clase tadListado que

representa la lista de tareas (list)

listadoEmpleados: Una instancia de la clase tadListado que

representa la lista de empleados (list)

cola: Una instancia de la clase tadCola que representa la

cola de tareas (list)

Returns:

None

"""

# Ejemplo de la estructura de la lista de tareas

listaDatosTareasFicticias = [

[

"Tarea",

"Descripción",

"Nombre",

"Estado",

"Fecha de vencimiento",

]

...

]

==================================== TAD COLA ================================

# File: tadCola.py

# Autores: Piazza J., Ibañez F., Lucas M..

# Date: 11 de Mayo de 2024

# Modified: 17 de Mayo de 2024

==============================================================================

"""

Spec:

Cola:

queue of Tarea

"""

crearCola():

"""

Crea una cola vacía

Args:

None

Returns:

list: Una lista vacía que representa una cola (list)

"""

colaEsVacia(cola):

"""

Retorna True si la cola no tiene elementos

Args:

cola: La cola a verificar si está vacía o no (list)

Returns:

Valor booleano True si la cola está vacía, False en caso contrario (bool)

"""

encolar(cola, elem):

"""

Agrega un elemento al final de la cola

Args:

cola: La cola a agregar el elemento (list)

elem: El elemento a agregar a la cola (Tarea)

Returns:

None

"""

desencolar(cola):

"""

Retorna y elimina el primer elemento de la cola

Args:

cola: La cola de la que se eliminará el elemento (list)

Returns:

Tarea: devuelve el primer elemento de la cola (list)

"""

tamanioCola(cola):

"""

Retorna la cantidad de elementos de la cola

Args:

cola: La cola de la que se calculará el tamaño (list)

Returns:

La cantidad de elementos en la cola (int)

"""

copiarCola(cola1, cola2):

"""

Copia los elementos de cola2 en cola1

Args:

cola1: La cola destino donde se copiaran los elementos (list)

cola2: La cola origen cuyos elementos serán copiados (list)

Returns:

None

"""

==================================== TAD EMPLEADOS ========================

# File: tadEmpleados.py

# Autores: Piazza J., Ibañez F., Lucas M..

# Date:

# Modified: 17 de Mayo de 2024

===========================================================================

"""

Spec:

Listado:

List of string

"""

crearEmpleados():

"""

Retorna un listado de empleados vacío

Args:

None

Returns:

empleados: una lista vacía (list)

"""

agregarEmpleado(empleados, e):

"""

Agrega un empleado al listado de empleados

Args:

empleados: Listado a la cual agregarle el empleado (list)

e: Empleado a ser agregado (str)

Returns:

None

"""

eliminarEmpleado(empleados, e):

"""

Elimina un empleado del listado de empleados

Args:

empleados: Listado del cual eliminar el empleado (list)

e: Empleado a ser eliminado (str)

Returns:

None

"""

recuperarEmpleado(empleados, i):

"""

Retorna el empleado en la posición y (el índice empieza en 0)

Args:

empleados: Lista de empleados a recuperar datos (List)

i: Entero en cuya posición se encuentra el empleado deseado (int)

Returns:

empleados[i]: Nombre del empleado en la posición i (str)

"""

tamanio(empleados):

"""

Retorna la cantidad de empleados en el listado

Args:

empleados: Lista de empleados a la cual leer su longitud (list)

Returns:

longitud de la lista (int)

"""

==================================== TAD LISTADO ===========================

# File: tadListado.py

# Autores: Piazza J., Ibañez F., Lucas M..

# Date:

# Modified: 17 de Mayo de 2024

===========================================================================

"""

Spec:

Listado:

List of Tareas

"""

crearListado():

"""

Retorna un listado de tareas vacío

Args:

None

Returns:

- listado: una lista vacía (list)

"""

agregarTarea(listado, t):

"""

Agrega una tarea al listado de tareas

Args:

listado: Listado al cual agregarle la tarea (list)

t: Tarea a ser agregada (list)

Returns:

None

"""

eliminarTarea(listado, t):

"""

Elimina una tarea del listado de tareas

Args:

listado: Listado del cual eliminar la tarea (list)

t: Tarea a ser eliminada (list)

Returns:

None

"""

recuperarTarea(listado, i):

"""

Retorna la tarea en la posición i (el índice empieza en 0)

Args:

listado: Listado del cual recuperar la tarea (list)

i: Entero que indica la posición del listado (int)

Returns:

Listado[i]: retorna la tarea en la posición i (str)

"""

tamanio(listado):

"""

Retorna la cantidad de tareas en el listado

Args:

listado: Listado a calcular su longitud (list)

Returns:

- longitud del listado (int)

"""

==================================== TAD TAREA =============================

# File: tadTareapy

# Autores: Piazza J., Ibañez F., Lucas M..

# Date:

# Modified: 17 de Mayo de 2024

===========================================================================

# Spec:

# tarea:

# nombre: string

# descripcion: string

# asignado: string (nombre de empleado)

# estado: string o tadEstado (pendiente, en progreso, completada)

# fecha de vencimiento: datetime

def crearTarea():

"""

Retorna una tarea vacía.

Args:

None

Returns:

Una lista que representa una tarea nueva con los siguientes elementos (list):

- Nombre de la tarea (str)

- Descripción de la tarea (str)

- Asignado a (str)

- Estado de la tarea (str)

- Fecha de vencimiento (str)

"""

cargarTarea(tarea, nombre, descripcion, asignado, estado, vencimiento):

"""

Carga los valores de una tarea

Args:

tarea: tarea a cargar (list)

nombre: nombre de la tarea (str)

descripcion: descripción de la tarea (str)

asignado: nombre del empleado a quien se le asigna la tarea (str)

estado: estado de la tarea, valores válidos: pendiente, en progreso, completado (str)

vencimiento: fecha de vencimiento de la tarea, formato ISO 8601 (YYYY-MM-DD) (str)

Returns:

None

"""

# La tarea debe cargarse de la siguiente manera:

tarea[0] = nombre

tarea[1] = descripción

tarea[2] = asignado

tarea[3] = estado

tarea[4] = date.fromisoformat(vencimiento)

verNombre(tarea):

"""

Retorna el nombre de la tarea

Args:

tarea: Lista con los datos de la tarea (list)

Returns:

- nombre de la tarea (str)

"""

verDescripcion(tarea):

"""

Retorna la descripción de la tarea

Args:

tarea: Lista con los datos de la tarea (list)

Returns:

- descripción de la tarea (str)

"""

verAsignado(tarea):

"""

Retorna el nombre del empleado que tiene asignada la tarea

Args:

tarea: Lista con lo datos de la tarea (list)

Returns:

- a quien está asignada la tarea (str)

"""

verEstado(tarea):

"""

Retorna el estado de la tarea

Args:

tarea: Lista con los datos de la tarea (list)

Returns:

- Estado de la tarea: Pendiente, En Progreso, Completado (str)

"""

verVencimiento(tarea):

"""

Retorna la fecha de vencimiento de la tarea en formato ISO 8061

Args:

tarea: Lista con los datos de la tarea (list)

Returns:

- Vencimiento de la tarea (datetime)

"""

modNombre(tarea, nombre):

"""

Modifica el nombre de la tarea

Args:

tarea: Lista con los datos de la tarea (list)

nombre: nuevo nombre de la tarea (str)

Returns:

None

"""

modDescripcion(tarea, descripcion):

"""

Modifica la descripción de la tarea

Args:

tarea: Lista con los datos de la tarea (list)

descripcion: Nueva descripción de la tarea (str)

Returns:

None

"""

modAsignado(tarea, asignado):

"""

Modifica el empleado que tiene asignada la tarea

Args:

tarea: Lista con los datos de la tarea (list)

asignado: Nuevo empleado asignado (str)

Returns:

None

"""

modEstado(tarea, estado):

"""

Modifica el estado de la tarea

Args:

tarea(Tarea): Lista con los datos de la tarea

estado(str): Nuevo estado de la tarea

Returns:

None

"""

tarea[3] = estado

modVencimiento(tarea, vencimiento):

"""

Modifica la fecha de vencimiento de la tarea

Args:

tarea: Lista con los datos de la tarea (list)

vencimiento: Nueva fecha de vencimiento de la tarea (datetime)

Returns:

None

"""

asignarTarea(tareaDestino, tareaFuente):

"""

Asigna el valor de la tarea fuente a la tarea destino

Args:

tareaDestino: Tarea a la cual se asignan los cambios (list)

tareaFuente: Tarea de la cual se leen los datos para luego modificar la tarea destino (list)

Returns:

None

"""

sonIguales(tarea1, tarea2):

"""

Retorna True si tarea1 y tarea2 son exactamente iguales, de lo contrario retorna False

Args:

tarea1: Una lista que representa una tarea, con los siguientes elementos (list):

- Nombre de la tarea (str)

- Descripción de la tarea (str)

- Asignado a (str)

- Estado de la tarea (str)

- Fecha de vencimiento (str)

tarea2: Otra lista que representa una tarea, con los mismos elementos que tarea1 (list)

Returns:

True si las dos tareas son exactamente iguales, False en caso contrario. (bool)

"""