

Descripción de Algoritmos

- 1) Inicializar un contador en 0; si la posición (valor recibido como parametro) es mayor a -1 entonces; si la posición es igual a 0 entonces se guarda lo que haya en la lista en una variable temporal; a la primer posición de la lista se le agrega el nuevo nodo con el valor (parametro) y a lo que le sigue al primer nodo se le asigna lo que había en la lista, por ultimo retornar True (Tener en cuenta que la variable de posición en la función estará sin inicializar); sino, si posición es mayor a 0 se inicializan dos variables que empiezan en el primer elemento de la lista, mientras una de esas variables (current) tenga un siguiente: se debe mover una posición y el contador aumentar una unidad, verificar si el contador tiene el mismo valor que la posición para que el siguiente de la otra variable que no se ha movido (before) sea el nuevo nodo y lo siguiente de lo siguiente de esa variable (before) sea igual a lo siguiente del current, saliendo de esa condición mover el before una posición para salir del while.

def pop(self, pos):
② función que recibe como parametro la posición de la lista del nodo a eliminar.

Se verifica si existe un primer elemento en la lista sino existe se retorna False;

Se verifica si el parametro es mayor a -1 entonces se verifica si el parametro es maigual a 0 entonces se le asigna el valor del primer elemento de la lista a una variable (current) y luego se asigna como primer (valor) elemento de la lista al que le sigue a esa variable (current.next);

Sino (significa que el parametro es mayor a 0) se inicializa un contador en 1, se inicializa una variable en el primer elemento de la lista y una segunda variable en el elemento que le sigue de la primer variable.

mientras exista la segunda variable (last):
si el contador es igual a la posición el siguiente de la primer variable (prev.next) sera sobreescrito por el siguiente de la segunda variable (last.next), saliendo de la condición se iguala la primer variable (prev) con la segunda (last), a la segunda se le mueve una posición y el contador se aumenta una unidad y ya con eso se puede salir del while (mientras).

def search (self, pos = 0):

③ El parametro posición se inicializa en 0
Se verifica si el parametro es cero entonces
retorna el valor del nodo del 1er primer
elemento de la lista.

Sino se inicializa un contador en 1 y
dos variables, una en el primer elemento de
la lista y el otro en el siguiente elemento de
la primer variable, mientras exista esa segunda
variable (last): si el contador es igual a la posición
se retorna el valor del nodo de la segunda variable
saliendo de esa condición que mover la segunda
variable una posición y aumentar el contador.

def openPlainText (self)

④ Se abrirá un archivo (open) en modo lectura ("r")
el cual se leera cada fila ("readlines") por separado
dado que cada fila representa los valores de los
parametros de la clase product (name, cost, coin, desc)
para por linea crear los nodos al abrir la
pantalla principal.

⑤ Se crea un arreglo y haciendo usos de el
ciclo for se limitaran el espacio que pueda
abarcas cada fila luego concatenar cada
elemento de el arreglo.

⑥ Haciendo uso de el metodo "clear" en cada
uno de los textbox se limpia el contenido de
los mismos (función al final de cada evento)