Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2022.

PROYECTO 1

Carnet 201907608 – LUDWING ALEXANDER LÓPEZ ORTIZ

Resumen

La resolución de un proyecto puede ser muy simple si sabemos los conceptos con los cuales podemos darle solución. La manera optima en la que se resuelve el problema que plantea el proyecto 1 es utilizar listas, pero no instanciar listas que el mismo lenguaje de programación, en este caso Python, nos ofrecen, si no que crear nuestros propios métodos de listas enlazadas, aprender su estructura y como estas se comportan, como están estructuradas y poder implementar estos conocimientos en otros futuros proyectos ya que es un método eficiente y así tener control de los datos que se manejan dentro de nuestro programa. Utilizar los datos XML y sabes cómo están estructurados, utilizar las herramientas que Python nos ofrecen, como las librerías ElementTree y Minidom, dos librerías implementadas en Python para el manejo de archivos XML, un tipo de archivo similar a HTML, ya que este se maneja con etiquetas. Utilizar las clases y saber cómo estas están estructuradas, conectadas y saber como manejar estas relaciones dentro de nuestros proyectos, utilizar los conocimientos aprendidos en clase para mejorar nuestro proyecto y darle una solución optima y eficaz a cada uno de los problemas que se nos presentan.

Palabras clave

Enlazada: Coger o juntar [una cosa] con lazos.

Xml: Es un lenguaje de marcado similar a HTML. Significa Extensible Markup Language (Lenguaje de Marcado Extensible) y es una especificación de W3C como lenguaje de marcado de propósito general.

Html: El Lenguaje de Marcado de Hipertexto (HTML) es el código que se utiliza para estructurar y desplegar una página web y sus contenidos.

Python: Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional.

Abstract

The resolution of a project can be very simple if we know the concepts with which we can solve it. The optimal way to solve the problem posed by project 1 is to use lists, but not to instantiate lists that the same programming language, in this case Python, offers us, but rather to create our own linked list methods, learn its structure and how they behave, how they are structured and to be able to implement this knowledge in other future projects since it is an efficient method and thus have control of the data that is handled within our program. Use XML data and you know how it is structured, use the tools that Python offers us, such as the ElementTree and Minidom libraries, two libraries implemented in Python for handling XML

files, a type of file similar to HTML, since it handles with labels. Use the classes and know how they are structured, connected and know how to manage these relationships within our projects, use the knowledge learned in class to improve our project and give it an optimal and effective solution to each of the problems that are presented to us.

Keywords

Linked: To catch or join [something] with ties.

Xml: It is a markup language like HTML. It stands for Extensible Markup Language and is a W3C specification as a general-purpose markup language.

Html: Hypertext Markup Language (HTML) is the code used to structure and display a web page and its content.

Python: It is a multiparadigm programming language, since it partially supports object orientation, imperative programming and, to a lesser extent, functional programming.

Introducción

Las listas enlazadas son una estructura de datos las cuales pueden ser utilizadas e implementadas en distintas estructuras como una herramienta eficiente con la cual podemos manejar de una manera más optima los datos de nuestro programa. Las listas enlazadas son esencial mente constituidas por una secuencia de nodos en los cuales se almacena nuestra información, campos de datos arbitrarios referenciados, estos están conectados entre sí de una manera unidireccional, en caso de las listas simples y bidireccional, hacia adelante y hacia atrás, en caso de ser listas doblemente enlazadas. Las listas enlazadas pueden ser implementadas en distintos lenguajes de programación, los cuales pueden variar en su construcción, pero siendo esencialmente la misma estructura. Las listas enlazadas permiten un correcto orden de almacenamiento de datos, con los cuales las estructuras de los datos permiten un correcto almacenamiento y manejo de ellos en la implementación dentro de una estructura de datos.

Desarrollo del tema

El desarrollo de un proyecto comienza planteando un problema, para poder llegar a una solución a ese problema se debe hacer un esquema de nuestro problema y las posibles soluciones a este.

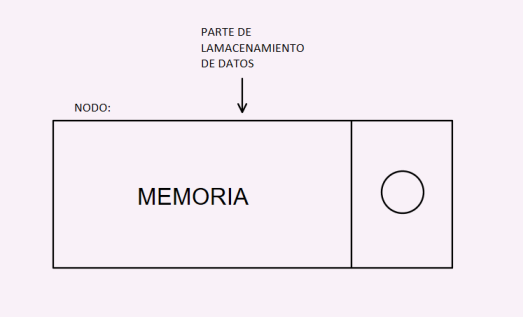
El proyecto al cual se le debe dar una solución, en este caso es en base a la estructura de una lista enlazada, a la cual se plantea utilizar las listas para poder almacenar los datos de un archivo de entrada el cual se debe poder manejar para que el usuario pueda hacer uso de los datos que cargo con un archivo. La implementación de las listas es una acción recurrente en el ámbito de la programación, pero a todo esto las personas, usuario y programadores, que hacen uso de estas no saben el proceso que lleva detrás toda la implementación de una lista.

Se plantea el problema para poder resolver durante el proyecto y se crean las restricciones “no utilizar las listas ya implementadas dentro del lenguaje de programación, para esto el estudiante debe desarrollar por si mismo una estructura de listas, ya sea listas enlazadas, doblemente enlazadas, circulares, etc. Para poder llegar a la solución del problema.

La implementación de una lista lleva diferentes partes para su creación y utilización dentro del código, no sobra decir que la creación de este código puede ser reutilizado, ya que es una estructura muy versátil que puede reutilizarse en distintos casos sea su necesidad. La creación de una lista debe llevar una estructura con la cual manejaremos los datos y poder acceder a ellos en un orden especifico o como el usuario/programador necesite de esta, siendo la principal parte de una lista un “Nodo” el cual es la parte de la lista en la cual se almacena la información requerida por el implementador, en la clase nodo se almacenan los distintos datos que se utilizaran

durante su ejecución, en pocas palabras un nodo es la parte de memoria dentro de la lista.

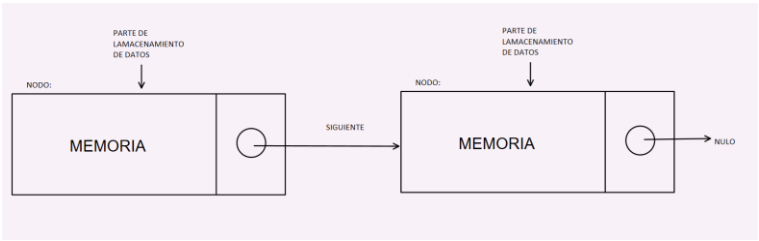
Estructura de un nodo:



Fuente: elaboración propia 2020

La segunda parte con mayor relevancia en la creación de una lista enlazada es la creación de la lista, que se tienen los nodos estos deben esta enlazados entre si o hacia el siguiente nodo, con lo cual se utiliza un apuntador, su función es estar siempre consiente del nodo que le sigue y el ultimo de ellos, debe estar apuntando a “nulo”, en caso de ser una lista simple, o estar apuntando hacia el primer nodo creado, en caso de una lista doblemente enlazada.

Estructura de una lista:



Fuente: elaboración propia 2022

Ya sabiendo la estructura de una lista podemos implementarla dentro de nuestro código para poder manejar los datos requeridos dentro de nuestro proyecto.

Para este proyecto como archivo de entrada se utilizó una estructura XML, que consiste en una estructura similar a la de un HTML. Esta estructura nos permite manejar los datos por medio de etiquetas haciendo más fácil su implementación y obtención de datos para nuestro programa. Como lenguaje para la realización de este proyecto se utilizo Python, ya que este lenguaje es muy versátil y portable, haciendo que todo lo que se realice en su estructura pueda ser utilizado en otros dispositivos sin necesidad de agregar, quitar o editar la estructura ya implementada, siendo esto gracias a su portabilidad ya que trabaja con una máquina virtual.

Gracias al lenguaje Python se pueden utilizar las librerías que este posee para la utilización de el archivo XML, que son Elementree y Minidom, don librerías que manejan los datos de un XML por medio de sus etiquetas.

Ya conociendo las estructuras a utilizar y las herramientas que el lenguaje seleccionado para la creación de nuestro programa con las cuales le daremos solución a nuestro problema y crear una solución optima se comienza a crear una estructura basada en Clases, las cuales nos permiten distribuir las tareas que se realizaran para nuestro programa.

Principalmente se creo un menú en el cual el usuario podría acceder a la carga de datos o terminar el programa, al seleccionar la carga de datos se abre una ventana con la librería TKinter, librería grafica que nos permite mostrar ventanas, botones, textos, etc. En la cual se selecciona un archivo XML. Al tener el archivo cargado en el sistema este es descompuesto en sus diferentes datos por sus etiquetas con la librería minidom, esto haciendo mas accesibles los datos. Estos son enviados a una lista enlazada creada para esta estructura de datos, teniendo tres diferentes listas que manejan los datos del archivo cargado.

La primera lista maneja los datos generales de nuestro archivo, siendo el que guarda los datos de las etiquetas principales del archivo, con los datos

agregados a nuestra estructura Nodo por medio de la implementación de ciclos se crea una lista con los datos de principales de nuestro archivo.

La segunda lista contiene los datos tipo nombre de nuestro archivo, los cuales son accedidos por medio de ciclos ya que estos están contenidos dentro de cada uno de nuestros datos principales del archivo, siendo uno de estos el principal problema, acceder a los archivos secundarios de cada dato principal. Utilizando ciclos de manera recurrente se logro acceder a cada uno de los datos de cada etiqueta principal, eligiendo un dato principal a la vez y por medio de otro ciclo recorrer y agregar a la segunda lista cada uno de los datos que se consiguieron de la etiqueta principal.

La tercera lista, al igual que la segunda, contenía los datos de la primera lista seleccionando por medio de ciclos, a diferencia de la segunda esta implementa una simulación de matriz, ya que para este proyecto se solicito que los datos se manejaran en manera de matriz, haciendo que cada nodo contuviese un numero de referencia el cual se tomaría como el índice que marca el lugar al cual pertenece dentro de la matriz, recorrer cada uno de ellos y comprobar que estos fuesen correctos.

Al lograr que todos los datos estén almacenados dentro de las listas se implementan un método dentro de las clases para poder visualizar los datos y que estos estén correctamente almacenados y poder manejarlos de manera mas eficiente. Todo esto conlleva a que el usuario tenga la certeza de que los datos que almacena estén correctamente almacenados.
Cunado todos los datos se encuentran almacenados se despliega un segundo menú dentro de la primera opción, esto nos muestra las acciones que podemos realizar con los datos almacenados. Las opciones que presentan cada uno de estos es con los datos que se encuentran ya almacenados en el sistema, y el usuario debe seleccionar el dato a modificar o con el cual interactuara. Para esto se crea un método con el cual se verifica que todos los datos que se encuentran dentro de las listas estén en los correctos y que el usuario pueda seleccionar uno de ellos, verificándolos por medio de una implementación de errores, que muestra si los datos que el usuario quiere manejar se encuentran en el sistema o este ingrese datos que no sean correspondientes a nuestros datos almacenados.

La implementación de manejo de errores es algo muy importante dentro de la programación ya que nos ayuda a crear confiabilidad en nuestros proyectos y programas realizados, creando y fomentando esta practica hacemos un trabajo mas eficiente y de mejor calidad.

La finalidad del programa es modificar los datos del archivo creando una opción óptima para la cantidad de movimiento y cambios que se realicen en nuestra matriz, siendo la implementación de listas una manera optima de trabajar con estos datos y poder aprender más sobre cómo manejarlos.

Conclusiones

Las listas enlazadas son un recurso esencial en la programación la cual nos enseña como esta conformada una lista, como las que vienen implementadas en los lenguajes de programación, ya que muestra las estructuras base de ellas.

Las librerías utilizadas para el manejo de los archivos XML son una herramienta las cuales se basan enteramente en la estructura por etiquetas las cuales podemos utilizar dentro de nuestro programa

simplemente implementado estas librerías, siendo la más amigable, en lo que cabe a la instancia de ella, minidom ya que es más intuitiva y simple en lo que utilización se habla, pero siendo elemntree la mas confiable ya que se pueden acceder a los datos de una manera más precisa.

La utilización de clases es importante en la creación de un programa, ya que nos deja ver e implementar las ideas de una manera mas ordenada y sencilla, siendo lo mas acercado a la realidad.

Referencias bibliográficas

- Introducción a XML - XML: Extensible Markup Language | MDN. (s. f.). Mdn Web Docs. Recuperado 6 de marzo de 2022, de https://developer.mozilla.org/es/docs/Web/XML/XML_introduction
- Definición de nodo — Definicion.de. (s. f.). Definición.de. Recuperado 6 de marzo de 2022, de https://definicion.de/nodo/#:%7E:text=La%20programaci%C3%B3n%20inform%C3%A1tica%20considera%20que,de%20referencia%20para%20otro%20nodo.
- Estructuras de datos: listas enlazadas, pilas y colas. (s. f.). Estructuras de datos: listas enlazadas, pilas y colas. Recuperado 6 de marzo de 2022, de https://calcifer.org/documentos/librognome/glib-lists-queues.html

Anexos:

Nodo:

```
class Nodo:

    def __init__(self, nombre, fila, columna, volteo, cambio):
        self.nombre = nombre
        self.fila = fila
        self.columna = columna
        self.volteo = volteo
        self.cambio = cambio
        self.Siguiente = None

    def __str__(self):

        return str(self.nombre, self.fila, self.columna, self.volteo, self.cambio)
```

Fuente: elaboración propia 2022.

Lista:

```
#formacion de las listas enlazadas
class ListasEnlazadas:

    def __init__(self):
        self.Primero = None
        self.Tamaño = 0

    def Agregar(self, nombre, fila, columna, volteo, cambio):

        NuevoNodo= Nodo( nombre, fila, columna, volteo, cambio)

        if self.Tamaño == 0:
            self.Primero = NuevoNodo
        else:
            current = self.Primero
            while current.Siguiente != None:
                current = current.Siguiente
            current.Siguiente = NuevoNodo

        self.Tamaño+=1
        return NuevoNodo
```

Fuente: elaboración Propia 2022

Impresión de una lista:

```
def Imprimir_Nombres_Pisos(self):

    puntero = self.Primero
    indice = 1

    while puntero != None:
        print(str(indice)+'. '+puntero.nombre)
        puntero = puntero.Siguiente
        indice +=1
    print('')
```

Fuente: elaboración propia 2022

Implementación Minidom:

```
#utilizar minidom para poder utilizar datos del archivo
doc = minidom.parse(link)
```

Fuente: elaboración propia 2022.

Utilizando minidom para obtener los datos del documento:

```
#datos extraidos del documento xml
nombre = doc.getElementsByTagName("piso")
lineas = doc.getElementsByTagName("R")
columnas = doc.getElementsByTagName("C")
volteo = doc.getElementsByTagName("F")
cambio = doc.getElementsByTagName("S")
patron = doc.getElementsByTagName("patrones")


    #envio de iformacion a lista enlazada
for x in range(len(nombre)):

    name = nombre[x].attributes['nombre'].value #nombre del piso
    line = lineas[x].firstChild.data #cantidad de lineas
    colum = columnas[x].firstChild.data #cantidad de columnas
    swap= volteo[x].firstChild.data #costo giro
    switch = cambio[x].firstChild.data #costo cambio
    cod = patron[x].getElementsByTagName("patron")

    lista.Agregar(name, line, colum, swap, switch) #agregar datos a lista principal
```

Fuente: elaboración propia 2022.