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 2021.

Procesamiento de imágenes con graphviz en Python.

201900462 – Xhunik Nikol Miguel Mutzutz

Resumen

Se presenta la solución para el manejo de archivos XML, el cual se procesa maneja objetos y es necesario crear una matriz para poder manipular los datos para lograr hacer varias operaciones con el mismo archivo, para eso usaremos las herramientas de Gestión de ficheros XML que se programó en el lenguaje Python utilizando la librería Element Tree, usando librerías que representan la información estructura en el fichero XML y cómo podemos obtener información de dicha estructura, como añadir o eliminar elementos o atributos y como modificar la información de guardada. Todo es una aplicación de consola, usando programación orientado a objetos y estructura de datos para almacenar la información y manejarla con EDD, la lista la cual se uso fue una lista ortogonal. para solucionar el problema a través de la programación, el cual se usaron varios paradigmas lo que nos permitió una solución rápida para la aplicación.

Palabras clave
Python, Lista, Matriz, Nodo, Acceso, Cabeceras, Objetos, Clases

Abstract

We present the solution for handling XML files, which is processed handles objects and it is necessary to create a matrix to manipulate the data to achieve several operations with the same file, for that we will use the XML file management tools that were programmed in the Python language using the Element Tree library, using libraries that represent the information structure in the XML file and how we can get information from that structure, how to add or delete elements or attributes and how to modify the stored information. Everything is a console application, using object-oriented programming and data structure to store the information and manage it with EDD, the list which was used was an orthogonal list.to solve the problem through programming, which used several paradigms which allowed us a quick solution for the app.

Keywords
Python, List, Matrix, Access, Headers, Objects, Classes.

Introducción

A continuación, se presentará la solución del problema del proyecto, el cual se usará el lenguaje de programación Python y el paradigma de programación orientado objetos, para el manejo de memoria se usará EDD (Estructura de datos), especialmente las listas ortogonales.

Junto a estas dos herramientas de programación se logró manejar la memoria del archivo y poder manipularla en diferentes clases, ya que se usó orientación a objetos lo cual nos facilitó y nos ayudó a tener ordenado nuestras sentencias de código.

Para este proyecto se realizo varias clases, las cuales manejan la interfaz, lógica y las operaciones solicitadas.

Desarrollo del tema

El problema que nos presenta es el siguiente:

“Las operaciones se aplican sobre una imagen contenida en la lista ordenada de imágenes, modificando la imagen sobre la cual se aplican.”

Para este problema se planteó la solución a través de diagramas para que el usuario pueda entender bien como se solucionó.

Para la creación de las matrices se planteó la siguiente solución el cual se desarrolló un diagrama para tener una comprensión mejor del usuario y pueda comprender como se efecto la aplicación:

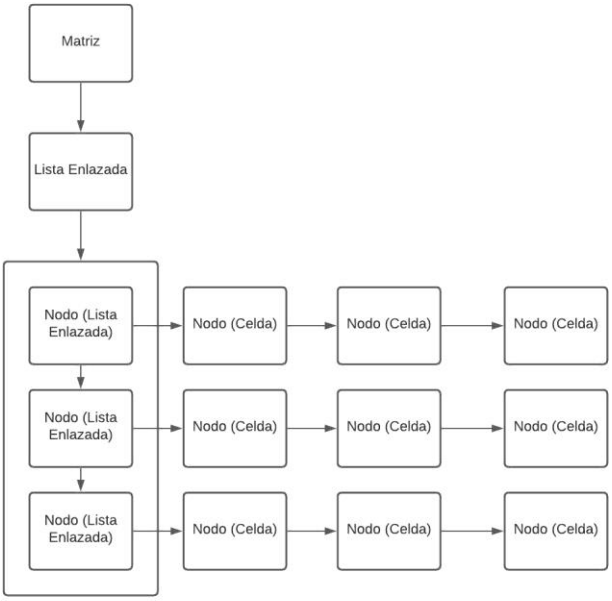

Figura 1. Diagrama.

Fuente: elaboración propia.

Se puede ver que esta aplicación, se utilizó varios nodos, los cuales sirvieron para manipular la información y hacer las operaciones las cuales fueron pedidas.

Las operaciones las cuales fueron pedidas para el usuario fueron las siguientes:

1. Rotación horizontal de una imagen
2. Rotación vertical de una imagen
3. Transpuesta de una imagen
4. Limpiar zona de una imagen
5. Agregar línea horizontal a una imagen
6. Agregar línea vertical a una imagen
7. Agregar rectángulo
8. Agregar triángulo rectángulo

Un ejemplo de la gráfica es en la figura 2. Pero para hacer las manipulaciones de la matriz

tuvimos que crear un nodo, el cual nos sirvió para recorrer las listas que nos pedía.

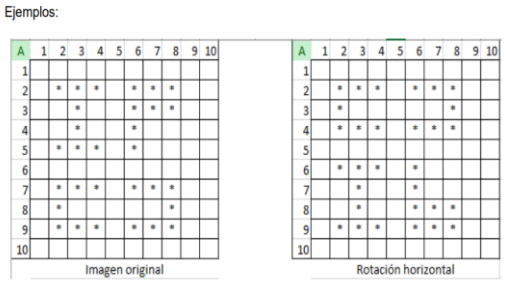

Figura 1. Matriz

Fuente: Enunciado del proyecto.

Para esto se tuvo que crear para clases para un mejor manejo de objetos y aprovechar sus atributos de las matrices para efectuar las operaciones y así poder mostrarlas en el panel,

Para que el cliente pueda comprender como se efectuó la estructura de las clases, se muestra a continuación un diagrama de clases para que el usuario tenga una idea de como se desarrolló en Python:

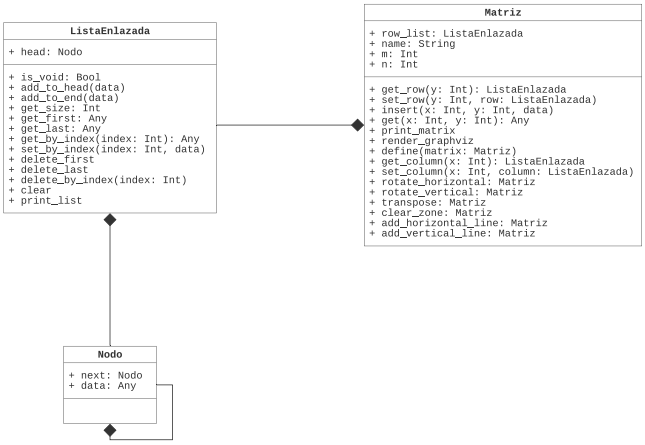

Figura 2. Diagrama de Clases UML

La estructura de datos utilizada consta de una lista de cabeceras, las cuales representan las filas de la matriz, cada una de ellas permite acceder a las celdas internas de cada fila, simulando de esta forma el funcionamiento de las listas nativas del lenguaje Python.

Para acceder a un nodo en específico se obtiene el nodo raíz de la fila respectiva y luego se recorre la estructura hasta encontrar la celda correspondiente a su coordenada ‘y’, es decir su columna.

Cada nodo esta enlazado con el siguiente, de esta manera solo hace falta almacenar el nodo “head” de la lista de filas, para poder acceder a todo el contenido de la matriz.

Para ejecutar un acceso, se obtiene el nodo “row_list” que contiene todas las filas, posteriormente se navega por todos los nodos hasta encontrar la fila, una vez ahí se obtiene el nodo “head” desde el cual se navega toda la estructura hasta encontrar la ubicación que corresponde al índice dado, de esta manera se simula el funcionamiento de una lista al tener acceso a los datos solo con los índices de posición dentro de la estructura.

Para ejecutar las operaciones de transformación se definieron funciones dentro de la clase matriz, las cuales devolvían las matrices con la operación ejecutada, luego esa matriz resultado sobrescribió la matriz inicial, de esta manera haciendo persistentes los cambios.

Para las operaciones donde 2 matrices estaban involucradas se valido el tamaño de las mismas, de tal forma que solo puedan ejecutarse dichas operaciones si ambas matrices son del mismo tamaño, luego desde una función externa se hicieron comparaciones entre las matrices

utilizando los índices para acceder a las mismas, creando de esta forma las funciones Unión e Intersección.

Conclusiones

Este proyecto se concluyó que la programación orientada a objetos es muy fundamental y es muy importante para desarrollar cualquier aplicación, ya que por sus varias ventajas ayudan mucho a poder tener un mejor orden del programa, pero también lo importante es que la mejor forma de manipular los datos de una matriz es a través de las estructuras de datos, específicamente las listas ortogonales y junto con la herramienta Graphviz se logra graficar cualquier matriz desde las librerías.

Referencias bibliográficas

Máximo 5 referencias en orden alfabético.

https://www.geeksforgeeks.org/Python-gui-tkinter/

https://www.geeksforgeeks.org/Python-tkinter-tutorial/

https://www.geeksforgeeks.org/introduction-to-tkinter/

https://www.geeksforgeeks.org/sparse-matrix-representation/

Extensión: de cuatro a siete páginas como máximo

Adicionalmente, se pueden agregar apéndices con modelos, tablas, etc. Que complementan el contenido del trabajo.