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.

DOCUMENTACIÓN PROYECTO 1

202002882 – Jonatan David Reyna Monterroso

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 miniDOM, esta nos ayuda a representar la información estructura en el fichero xml y a obtener la información guardada en el archivo xml. Asi como una manipulación entera de los datos ya sea al agregar, eliminar o modificarlos. Esto se logró creando una aplicación de consola, usando programación orientada a objetos y estructura de datos para almacenar la información y manejarla con EDD, el cual nos servirá para construir un programa y solucionar el problema, el cual se usaron varios paradigmas lo que nos permitió una solución rápida para la app.

Palabras clave

- XML (Lenguaje de Marcado Extensible)
- Lista enlazada
- Nodos
- Matriz
- Estructuras de Datos

Abstract

The solution for handling XML files is presented, which processes and handles objects and it is necessary to create a matrix in which we are able to manipulate the data to achieve several operations with the same file, for that we will use the XML file management tools that are implemented within Python using the miniDOM library, using this helps us the structure information in the xml file and we are also able to obtain the information stored in the xml file. As well as a whole manipulation of the data either when adding, eliminating or modifying them. This was achieved by creating a console application, using object-oriented programming and data structure to store the information and manage it with EDD, which will help us to build a program and solve the problem, which used several paradigms which allowed us a quick fix for the app.

Keywords

- *XML (eXtensible Markup Language)*
- *Linked List*
- *Nodes*
- *Matrix*
- *Data Structures*

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).

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 usaron diferentes métodos y clases que posteriormente serán explicadas a más detalle.

Desarrollo del tema

El objetivo del protecto es implementar una solución en Python haciendo uso de estructuras de programación secuenciales, cíclicas y condicionales y principalmente que se logre dominar los archivos XML. La Agencia Guatemalteca de Investigación Espacial (AGIE) ha diseñado un nuevo robot de exploración, llamado r2e2, que tiene la habilidad de explorar nuevos terrenos, este nuevo robot puede moverse en todo tipo de terrenos, sólo necesita más combustible para moverse en terrenos accidentados, y menos combustible en terrenos planos. El único problema con r2e2 es que solo puede moverse ortogonalmente, es decir, únicamente puede moverse en dirección Norte, Este, Sur y Oeste de su posición. Se espera que podamos desarrollar el algoritmo que r2e2 necesita para identificar el camino con menor consumo de combustible para moverse por el área de exploración.
Para todo este proceso usamos programación orientado a objetos, el cual creamos varias clases. Esto nos ayudó a crear nuestra estructura de datos por diferentes áreas, las clases fueron las siguientes:

- Clase Lista Enlazada
- Clase Nodo
- Clase Matriz

Primero el usuario tiene que ingresar la ruta del archivo xml el cual se guarda y se procesa usando la librería miniDOM, con esto se hace el “parse” con lo cual usando un proceso cíclico, vamos recorriendo el contenido del archivo buscando los tags que queremos. Luego en cada tag, obtenemos sus nodos “hijos” los cuales tienen los datos que necesitamos para posteriormente añadirlos a la matriz ortogonal con la que trabajamos.

```
def lecturaarchivo(data: ListaEnlazada):
    # global domTree, rootNode
    doc = input("Ingrese la ruta del archivo: ")
    domTree = parse(doc)
    root = domTree.documentElement
    terrenos = root.getElementsByTagName("terreno")

    for terreno in terrenos:
        nombre = terreno.getAttribute("nombre")
        dimension = terreno.getElementsByTagName("dimension")[0]
        m = dimension.getElementsByTagName("m")[0].childNodes[0].data
        n = dimension.getElementsByTagName("n")[0].childNodes[0].data

        posicioninicio = terreno.getElementsByTagName("posicioninicio")[0]
        x_i = posicioninicio.getElementsByTagName("x")[0].childNodes[0].data
        y_i = posicioninicio.getElementsByTagName("y")[0].childNodes[0].data

        posicionfinal = terreno.getElementsByTagName("posicionfin")[0]
        x_f = posicionfinal.getElementsByTagName("x")[0].childNodes[0].data
        y_f = posicionfinal.getElementsByTagName("y")[0].childNodes[0].data
        matrizej = Matriz(nombre, int(m), int(n))
        matrizej.print_matrix()
        data.add_to_end(Matriz(nombre, int(m), int(n)))
```

Figura 1. Lectura inicial de los datos

Esta matriz se conforma de una lista enlazada, la cual a su vez está formada de varias listas. Para manipularla usamos métodos que nos permiten interactuar con la matriz buscando índices, lo cual permite ingresar elementos a una lista indicando únicamente el valor de un índice.

Para realizar la manipulación de los datos, es decir para permitir que los datos sean accesibles desde todos los ámbitos, se declararon 2 listas enlazadas en el inicio del programa, las cuales se pasan como parámetros a todas las funciones que puedan manipular los datos, esto se hace aprovechando la funcionalidad de Python, que permite pasar por referencia todos los valores, de tal manera que, si un

valor es modificado por una función, este realmente es modificado para todas las funciones y/o clases que requieran de su acceso

```python
def menu(data: ListaEnlazada):
    ans = True
    while ans:
        print("----------------Bienvenidos-------------")
        print("|    1.Cargar Archivo                  |")
        print("|    2.Procesar Terreno                | ")
        print("|    3.Escribir Archivo de Salida      |")
        print("|    4.Mostrar Datos del Estudiante    |")
        print("|    5.Generar Gráfica                 |")
        print("|    6.Salir                           |")
        print("---------------------------------------")
        ans = input("Eliga una opción: ")
        if ans == "1":
            lecturaarchivo(data)
        elif ans == "2":
            procesarterreno()
            print("Archivo procesado con exito")
            print()
        elif ans == "3":
            print("Se escribió la ruta especifica")
        elif ans == "4":
            mostrar_info()
        elif ans == "5":
            print("Grafica generada")
        elif ans == "6":
            print("Cerrando aplicación...")
            ans = False
        else:
            print("Opción inválida\n")
```

Figura 2. Función Menú

Conclusiones

Este proyecto se concluyó que la programación orientada a objetos es muy fundamental para el desarrollo de cualquier aplicación con la cual manejemos grandes cantidades de datos.

También podemos concluir que es importante aprender a manejar diferentes tipos de archivos para almacenamiento de datos, asi como XML ya que son un tipo de archivo que se sigue usando activamente en la industria y esto nos da una introducción a un problema real que se nos puede presentar en un futuro y con este proyecto somos capaces de identificar dónde podemos usar diferentes tipos de TDA para hacer una aplicación más eficiente.

Referencias Bibliográficas

Corporation, P. S. (20 de Agosto de 2021). *Python*. Obtenido de https://docs.python.org/3/library/xml.dom.minidom.html

Jones, C. A. (2002). *Python and XML - O'Reilly.*

Singh. (9 de Julio de 2021). *TechGeekBuzz*. Obtenido de https://www.techgeekbuzz.com/python-xml-parser-tutorial/