Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
IPC2, 2do. Semestre 2023.

MANUAL TECNICO - PROYECTO NO. 1 -

201212891 – Edgar Rolando Ramírez López

Resumen

El código consiste en un programa Python que gestiona señales de audio en formato XML mediante una estructura de listas enlazadas y la generación de gráficos utilizando Graphviz. El archivo nodo.py define la clase Nodo, que representa los nodos individuales de la lista enlazada, almacenando información como nombre, tiempo, amplitud y valor de las señales. La clase ListaEnlazada, definida en listaSimple.py, gestiona la lista enlazada y proporciona métodos para agregar, buscar, eliminar y mostrar señales, así como generar gráficos visuales de la lista. El archivo principal main.py coordina la interacción con el usuario, permitiendo la carga de archivos XML, el procesamiento de datos para generar matrices binarias y la escritura de nuevos archivos XML con datos procesados. En resumen, el programa permite cargar, procesar y visualizar señales de audio de manera eficiente y flexible.

Palabras clave
Python, funciones, clases, listas, librerías.

Abstract

The code consists of a Python program that manages audio signals in XML format using a linked list structure and generates graphs using Graphviz. The nodo.py file defines the Node class, which represents individual nodes of the linked list, storing information such as the name, time, amplitude, and value of the signals. The ListaEnlazada class, defined in listaSimple.py, manages the linked list and provides methods for adding, searching, deleting, and displaying signals, as well as generating visual graphs of the list. The main file, main.py, coordinates user interaction, allowing the loading of XML files, processing data to generate binary matrices, and writing new XML files with processed data. In summary, the program efficiently and flexibly allows for the loading, processing, and visualization of audio signals.

Keywords
Python, functions, class, lists, libraries.

Introducción

El programa es un sistema de gestión de señales de audio en formato XML implementado en Python. Utiliza una estructura de listas enlazadas para almacenar información detallada de cada señal, incluyendo nombre, tiempo, amplitud y valor. Además, ofrece la capacidad de cargar archivos XML, procesar los datos para generar matrices binarias y generar visualizaciones gráficas de las señales. Este sistema proporciona a los usuarios una herramienta versátil para trabajar con señales de audio, desde la carga inicial de datos hasta su procesamiento y representación gráfica, lo que lo hace útil en diversas aplicaciones relacionadas con el procesamiento de señales y la visualización de datos.

Desarrollo del tema

En el desarrollo de la práctica se utilizó el lenguaje Python 3.11.1, el IDE utilizado fue Visual Studio Code y también se subió la practica a GitHub. El paradigma utilizado en la práctica fue la programación con orientación a objetos. Las librerías utilizadas fueron xml.etree.ElementTree, graphviz y subprocess.

a. Python

Python es un lenguaje de programación interpretado multiparadigma, multiplataforma y de código abierto. Diseñado para facilitar la programación orientada a objetos, permite trabajar con un gran número de estructuras de datos como diccionarios, listas, tuplas y conjuntos. Es compatible con muchos sistemas operativos, como Windows, Mac OS, Linux, y también se puede usar para el desarrollo web, scripting y desarrollo de aplicaciones. Ofrece una extensa biblioteca estándar con muchas utilidades prácticas. Está aumentando su presencia en la industria de la tecnología y está ganando popularidad por su facilidad de uso y simplicidad.

b. Visual Code Studio (VSCODE)

Un IDE es un entorno de desarrollo integrado, es una aplicación de software que ayuda a los programadores a desarrollar código de software de manera eficiente. Visual Studio Code es un editor de código fuente de código abierto y gratuito desarrollado por Microsoft para MacOS, Linux y Windows. Incluye soporte para desarrollo web, lenguajes de edición de línea de comandos y lenguajes de scripting modernos, así como lenguajes de programación como HTML, CSS, CoffeeScript, PHP, Ruby y Python. Ofrece características como resaltado de sintaxis, completado inteligente, intenciones y muchos más. La eficiencia y la facilidad de uso de Visual Studio Code han hecho de él una herramienta de desarrollo muy utilizada en todo el mundo.

c. GitHub

GitHub es una plataforma de alojamiento web que ayuda a los desarrolladores a almacenar y mantener el código fuente para proyectos de software. Utiliza el sistema de control de versiones Git para permitir a los usuarios colaborar en el desarrollo de proyectos, realizar revisión de código, realizar tareas de seguimiento de errores y realizar pruebas. También ofrece una variedad de herramientas de integración de terceros, navegación web y soporte para proyectos públicos y privados.

d. Librería xml.etree.ElementTree

Biblioteca estándar de Python que se utiliza para analizar y manipular documentos XML de una manera simple y eficiente. Proporciona una interfaz de alto nivel para acceder a elementos y atributos XML, así como para crear y modificar documentos XML. ElementTree facilita la lectura y escritura de archivos XML, lo que lo hace útil para procesar datos estructurados en formato XML en aplicaciones Python.

e. Librería subprocess

Módulo en Python que proporciona una interfaz para trabajar con procesos (programas o comandos) externos al script de Python en sí. Permite a los programas de Python crear, ejecutar y comunicarse con otros procesos o ejecutar comandos del sistema operativo en la línea de comandos.

f. Librería Graphviz

Herramienta de software utilizada para crear y renderizar gráficos y diagramas, especialmente diagramas de grafos. Esta librería proporciona un conjunto de herramientas y funciones que permiten la creación y visualización de estructuras de datos gráficas de manera programática.

g. XML (Extensible Markup Language)

Lenguaje de marcado diseñado para almacenar y transportar datos de manera estructurada. XML se utiliza comúnmente para representar información en forma de texto legible tanto por humanos como por máquinas.

Estructura del Código

nodo.py

Este archivo define una clase llamada Nodo. Los nodos se utilizan para construir una lista enlazada que almacena información sobre señales de audio. Cada nodo contiene los siguientes atributos:

- **nombre:** El nombre de la señal de audio.
- **tiempo:** El tiempo de la señal de audio.
- **amplitud:** La amplitud de la señal de audio.
- **valor:** El valor de la señal de audio.
- **siguiente:** Una referencia al siguiente nodo en la lista enlazada.

Este archivo define la estructura básica de los nodos que se utilizan para almacenar y manipular la información de las señales de audio.

listaSimple.py

Este archivo define una clase llamada ListaEnlazada que representa una lista enlazada que almacena nodos que representan señales de audio. La clase tiene los siguientes métodos:

- **__init__(self):** El constructor de la clase inicializa una lista enlazada vacía.
- **agregar(self, nombre, tiempo, amplitud, valor):** Este método agrega un nuevo nodo al final de la lista enlazada con los valores proporcionados.
- **buscar(self, nombre):** Este método busca un nodo con un nombre de señal específico en la

lista enlazada y devuelve True si se encuentra o False si no se encuentra.

- **delete_node(self, nombre, tiempo, amplitud):** Este método elimina un nodo específico de la lista enlazada según su nombre, tiempo y amplitud.
- **mostrar_nombres(self):** Este método muestra los nombres únicos de las señales almacenadas en la lista enlazada.
- **mostrar_senal(self, nombre_senal):** Este método muestra la información de una señal específica, incluyendo su nombre, tiempo, amplitud y valor.
- **writeNodes(self, f):** Este método es utilizado internamente para escribir nodos en un archivo.
- **generateGraphvizCode(self, nombre_senal):** Este método genera un archivo DOT de Graphviz para visualizar la lista enlazada como un gráfico.

La clase ListaEnlazada proporciona funcionalidad para agregar, buscar, eliminar y mostrar nodos en la lista enlazada. También puede generar un gráfico visual de la lista enlazada.

main.py

Este es el archivo principal del programa que interactúa con el usuario y coordina todas las operaciones. Aquí se definen las siguientes funciones y variables:

- **estudiante():** Esta función imprime información sobre el estudiante.
- **elegirArchivo():** Esta función permite al usuario elegir un archivo XML para cargar datos.
- **cargar_datos_desde_xml(xml_file):** Esta función carga datos desde un archivo XML proporcionado y almacena la información en una lista enlazada.
- **procesarxml():** Esta función procesa los datos de las señales almacenadas en la lista enlazada y realiza cálculos específicos para generar matrices binarias y otras estructuras de datos.
- **escribir_xml_final(matrices_finales_por_senal):** Esta función escribe los datos procesados en un nuevo archivo XML.
- **if __name__ == "__main__":** El código principal del programa comienza aquí. Proporciona un menú de opciones al usuario para cargar archivos, procesar datos, generar gráficos y más.

El flujo principal del programa es interactuar con el usuario a través del menú y realizar las operaciones seleccionadas, como cargar datos, procesarlos, generar gráficos y escribir archivos XML.

Conclusiones

El programa ofrece una solución integral para la gestión de señales de audio en formato XML mediante el uso de listas enlazadas y funcionalidades de procesamiento de datos. Esto permite a los usuarios cargar, procesar y visualizar señales de manera eficiente. Su versatilidad y capacidades hacen que sea una herramienta valiosa para una variedad de aplicaciones relacionadas con el procesamiento de señales y la representación gráfica de datos de audio.

Referencias bibliográficas

Welcome to. (2023, 15 febrero). Python.org. https://www.python.org/

Documentation for Visual Studio Code. (2021, 3 noviembre). https://code.visualstudio.com/docs

Hello World. (s. f.). GitHub Docs. https://docs.github.com/en/get-started/quickstart/hello-world

J. (2022, 16 enero). *Funciones en Python: Definición de función y para qué se utilizan*. J2LOGO. https://j2logo.com/python/tutorial/funciones-en-python/

Bustamante, S. J. (2021, 21 febrero). *Guía de funciones de Python con ejemplos*. freeCodeCamp.org. https://www.freecodecamp.org/espanol/news/guia-de-funciones-de-python-con-ejemplos/

Programación orientada a objetos. (s. f.). https://www.ibm.com/docs/es/spss-modeler/saas?topic=language-object-oriented-programming

J. (2022b, enero 16). *Programación orientada a objetos (POO) en Python*. J2LOGO. https://j2logo.com/python/tutorial/programacion-orientada-a-objetos/