
Diseño Proyecto 1

202202161 – Juan Pascual Itzep Coguo

Resumen

La aplicación de estructura de datos específicamente para un sistema requerido que maneja matrices de frecuencias de una señal de datos usando programación orientada a objetos, leyendo archivos XML, también escribiéndolos y creando graficas con la herramienta Graphviz, estas herramientas se utilizaron para generar una señal de audio reducida de la señal de audio ingresada, en este documento encontrara la explicación de POO aplicada a los requerimientos del sistema, la explicación de la lógica utilizada para la implementación de Estructura de Datos y las funciones específicas y necesarias para la creación de las listas enlazadas simples y la forma en que se utilizó para verlas como matrices de tiempo y amplitud de una señal de audio, también encontrara una breve explicación de un archivo XML y la librería utilizada para el Proyecto, finalmente encontrara una breve descripción de la herramienta requerida por el enunciado del Proyecto para la creación de gráficos, específicamente grafos.

Palabras clave

POO

Estructura de Datos

Listas Enlazadas

Abstract

The data structure application specifically for a required system that handles frequency matrices of a data signal using object-oriented programming, reading XML files, also writing them and creating graphs with the Graphviz tool, these tools were used to generate a data signal. reduced audio of the entered audio signal, in this document you will find the explanation of OOP applied to the system requirements, the explanation of the logic used for the implementation of the Data Structure and the specific and necessary functions for the creation of the linked lists simple and the way in which it was used to see them as time and amplitude matrices of an audio signal, you will also find a brief explanation of an XML file and the library used for the Project, finally you will find a brief description of the tool required by the Project statement for the creation of graphs, specifically graphs.

Keywords

POO

Data Structure

Introducción

El principal fin de las computadoras ha sido el eficiente manejo de información que muchas veces será de grandes extensiones y para crear un funcionamiento específico referente a la información, se deben crear sistemas específicos. En este caso se desea implementar un sistema que permita leer archivos XML con los tiempos, amplitud y frecuencia de una señal de audio, guardar y manejar la información empleando TDA, Estructura de Datos y POO para conseguir señales de audio reducidas mediante coincidencias entre las matrices y finalmente graficar dichas matrices empleando Graphviz, todo este sistema realizado en el lenguaje de programación Python.

Desarrollo del tema

Existen innumerables maneras de diseñar e implementar un sistema para dar solución a un único problema, en lo que refiere a programación de software POO es uno de los paradigmas de programación mas eficientes y con el que se implementó el sistema requerido.

a. Implementación de POO

Por los requerimientos del proyecto y también para facilitar la programación del sistema se utilizo POO(Programación Orientada a Objetos), el cual brevemente trata sobre la segmentación de un sistema en objetos que tienen atributos y funciones, los cuales se relacionan entre si para completar las tareas del sistema.

Por la información que se maneja en el sistema se crearon 4 tipos de objetos.

El objeto `senal`(de señal) es el mas general y engloba la información de los demás objetos, `senal` tiene como atributos el nombre de la señal, los tiempos y amplitud de la frecuencia, tambien contiene listas enlazadas para la lista de datos, lista de patrones y lista de datos de la matriz reducida.

El segundo objeto `celda_datos` tiene como atributo el valor del tiempo, de la amplitud y el dato o valor de una frecuencia, este tipo de objeto se almacena en lista de datos o lista de patrones del objeto `senal`.

El tercer objeto `identificador_patron` tiene como atributos: el valor del tiempo como fila de una matriz de tiempo y amplitud, y el patrón concatenado de la lista de patrones, se almacena en una lista de identificadores que solamente se crea durante la función que crea la lista de datos de la matriz reducida de una señal y en esta misma se elimina.

El cuarto objeto `celda_matriz_reducida` tiene como atributos: el grupo de las filas sumadas, el patron de las filas sumadas el cual debe ser el mismo, el dato resultante de sumar cada celda de las filas que tienen el mismo patron, el valor de la amplitud en la que se encuentra, la cantidad de filas sumadas y el valor de la fila tiempo menor entre las sumadas, se guarda en lista de datos de la matriz reducida como atributo del objeto `senal`.

Estos son los únicos objetos utilizados para completar todas las tareas necesarias del sistema requerido ya que todas las funciones se encuentran en las Estructuras de datos.

b. Ciclos While

Antes de establecer la implementación de Estructura de Datos específicamente para el sistema, es bueno recordar que es un ciclo while, pues podría asegurar

que este ciclo es el mas importante para la implementación de las Estructuras de Datos.

Un ciclo While como su nombre lo indica es una función que realiza una serie de instrucciones mientras una condición se cumpla, con este concepto realmente claro se puede entender como aplicar estructuras de datos en un sistema.

c. Implementación de Estructura de Datos

El principal propósito del enunciado del proyecto es que nosotros como estudiantes comprendamos y seamos capaces de aplicar Estructura de Datos.

La estructura de datos utilizada en el proyecto son las listas enlazadas, las cuales son objetos llamados Nodo, que pueden guardar un dato u otro objeto y tienen apuntadores a otros Nodos que también tienen datos y apuntadores, de esa manera es que se enlaza, es como ver un tren y sus vagones, el apuntador seria la conexión entre los vagones del tren.

En este proyecto se empleó una lista enlazada simple, este tipo de listas enlazadas contiene nodos que únicamente tienen un apuntador hacia el siguiente nodo.

Al momento de recorrer una lista enlazada simple únicamente se puede ir hacia el siguiente nodo hasta que la lista termine.

Para recorrer una lista enlazada se utiliza un nodo temporal que comience en el primer nodo y con un ciclo while se actualiza el nodo temporal con el nodo siguiente mientras que el nodo temporal no llegue al final.

Para este proyecto se crearon dos tipos de listas enlazadas simples, una para los objetos de tipo senal y otra para las listas de objetos que guarda una señal.

Para las listas de objetos que guarda una señal se desarrollaron varias funciones para el correcto funcionamiento del sistema requerido.

Las principales funciones son: insertar los datos de manera ordenada si es que en el archivo XML los tiempos y amplitudes vienen desordenados, crear las celdas para los datos que no se encuentren en el archivo XML dándoles un valor de 0, crear la lista de patrones y crear la matriz reducida la cual es la más importante.

La lógica de insertar las celdas o datos de manera ordenada recorre la lista mientras valúa si los tiempo y amplitudes son mayores o menores a los tiempos y amplitudes de la celda a insertar.

Para la función de crear las celdas de datos faltantes se recorre la lista y los valores de los tiempos y amplitudes de 1 hasta el valor mas grande tiempo y amplitud validando que para cada celda de la matriz de tiempo y amplitud exista una celda ingresada por el archivo XML, si no es así se crea e ingresa una celda en esa posición con un valor de frecuencia cero.

La función crear lista de patrones recorre la lista de datos y si el valor del dato es mayor a 0 se crea un objeto con el mismo tiempo y amplitud, pero con un valor de 1 para la posterior generación de patrones y se guarda en la lista de patrones.

Para generar la matriz reducida, la función comienza recorriendo la matriz de patrones y concatenando los datos de cada tiempo y amplitud mientras que el valor del tiempo coincida con el valor del tiempo anterior.

Ejemplo: hay 3 datos(1 o 0) dos con tiempo 1 y amplitudes: 1 y 2, el tercer dato con tiempo: 2 y amplitud: 1, los datos con el mismo tiempo se concatenarán y se guardarán en una lista de identificadores, el siguiente buscará otros datos con

el mismo tiempo y también se guardará en la lista de identificadores.

Posteriormente de crear la lista de identificadores la función recorrerá esta misma lista 2 veces en 2 ciclos while anidados, y buscará que 2 cadenas de patrones coincidan.

Si coinciden y es la primera coincidencia(pueden haber mas de dos filas que tengan el mismo patrón), con un ciclo For para recorrer todos los valores de las columnas de amplitud se buscan los valores en la lista de datos con el tiempo y amplitud específico y se suman todos los datos y se agregan a la lista de matriz reducida, como lo pide el proyecto, si no es la primera coincidencia se suman obtiene los valores de la nueva coincidencia y se suman a las sumas anteriores y si se recorre toda la lista con un patrón y no coincide con ninguna fila, se buscan los datos y se agregan a la lista de matriz reducida.

Finalmente se ordena la lista de matriz reducida dependiendo del numero de filas sumadas y el valor del tiempo menor de las filas sumadas, las filas de la matriz reducida quedan con los valores de mas filas sumadas primero y los valores de tiempo_smaller mas pequeño como primeros también.

Esta estructura de datos contiene mas funciones, pero las mas importantes son las explicadas aquí.

d. Archivos XML

Estos archivos se dividen como un árbol, tiene una raíz, de esta raíz crecen ramas y de las ramas más ramas cada una con su propia información.

Este sistema recibe la información por medio de archivos XML con Señales como raíz y señal como una rama que tiene ramas de datos de la matriz de datos.

Esa información se utiliza para llenar la lista de datos.

Para la lectura y escritura de archivos XML existen varias librerías, la utilizada en este proyecto es ElementTree.

e. Graphviz

Graphviz es una herramienta que permite utilizar cadenas de texto con códigos específicos para crear grafos.

En este proyecto se utilizó Graphviz para representar gráficamente las listas de datos y matriz reducida.

Conclusiones

Es importante comprender el funcionamiento de las Estructuras de Datos pues en eso se basa toda la informática, en guardar datos, manipularlos y representarlos de manera fácil y rápida.

¿Crear muchas listas enlazadas puede ser contraproducente para un sistema en lo que refiere a velocidad del manejo de datos masivos?

Anexos

