Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, Vacaciones 2do.

Reproductor de Musica MP3 con archivo de entrada XML

201902888 – Ivan Alessandro Hilario

Resumen

El presente proyecto se centra en la implementación de tipos de datos abstractos, asi mismo con el fin de poder visualizar tipo de dato utilizando archivos de entrada con extensión XML. Un tipo de dato abstracto (TDA) es un conjunto de datos u objetos al cual se le asocian operaciones. El manejo de TDA se utiliza cuando se requiere de un manejo mas adecuado de una cantidad elevada de datos, los cuales precisan de un acceso eficaz y ordenado.

Un TDA puede ser implementado utilizando distintas estructuras de datos y que proveen la misma funcionalidad. Así mismo mostrado el presente proyecto creando e instanciando clase como Lista la cual fue heredada a la clase Biblioteca así mismo con la clase Nodo, demostrando y representando cada uno de los canciones implementando el manejo de reutilización de código dentro del programa así mismo demostrando el uso del paradigma de programación orientado a objetos.

Palabras clave

Tipos de datos abstracto, programación orientada a objetos, archivos XML, librería Grahpviz, estructuras de programación.

Abstract

This project is focused on the implementation of abstract data types, in order to be able to visualize data types using input files with XML extension. An abstract data type (ADT) is a set of data or objects to which operations are associated. The handling of TDA is used when a more adequate handling of a large amount of data is required, which need an efficient and orderly access.

An ADT can be implemented using different data structures that provide the same functionality. Also shown the present project creating and instantiating class as List which was inherited to the Library class as well as the Node class, demonstrating and representing each of the songs implementing the code reuse management within the program as well as demonstrating the use of the object-oriented programming paradigm.

Keywords

Abstract data types, Object-oriented programming, XML files, Grahpviz library, programming structures.

Introducción

En ciencias de la computación un tipo de dato abstracto (TDA) se define como un modelo matemático compuesto por una colección de operaciones definidas sobre un conjunto de datos para el modelo. Por lo cual, el uso de TDA para la realización de un analizador de datos es lo más ideal, ya que se cuenta con una manera más eficiente de acceder a los datos con el fin de poder operarlos a conveniencia del desarrollador.

El presente ensayo explica de mejor manera la forma en la que se implementaron las estructuras de datos utilizadas, asimismo expandir la teoría que no se puede plasmar en un proyecto de programación y explicar las razones por las que se optó por escoger ciertas tecnologías utilizadas en el desarrollo del proyecto.

Estructura de Datos

En ciencias de la computación, una estructura de datos es una forma particular de organizar datos en una computadora para que puedan ser utilizados de manera eficiente. Diferentes tipos de estructuras de datos son adecuados para diferentes tipos de aplicaciones, y algunos son altamente especializados para tareas específicas. Las estructuras de datos son un medio para manejar grandes cantidades de datos de manera eficiente para usos tales como grandes bases de datos y servicios de indización de Internet. Por lo general, las estructuras de datos eficientes son clave para diseñar algoritmos eficientes. Las estructuras de datos se basan generalmente en la capacidad de un ordenador para recuperar y almacenar datos en cualquier lugar de su memoria

Existen numerosos tipos de estructuras de datos, generalmente construidas sobre otras más simples:

a) Un vector es una serie de elementos en un orden específico, por lo general todos del mismo tipo (si bien los elementos pueden ser de casi cualquier tipo). Se accede a los elementos utilizando un entero como índice para especificar el elemento que se requiere. Las implementaciones típicas asignan palabras de memoria contiguas a los elementos de los arreglos (aunque no siempre es el caso). Los arreglos pueden cambiar de tamaño o tener una longitud fija.

b) Un vector asociativo (también llamado diccionario o mapa) es una variante más flexible que una matriz, en la que se puede añadir y eliminar libremente pares nombre-valor. Una tabla de hash es una implementación usual de un arreglo asociativo.

c) Un registro (también llamado tupla o estructura) es una estructura de datos agregados. Un registro es un valor que contiene otros valores, típicamente en un número fijo y la secuencia y por lo general un índice por nombres. Los elementos de los registros generalmente son llamados campos o celdas.

d) Una unión es una estructura de datos que especifica cuál de una serie de tipos de datos permitidos podrá ser almacenada en sus instancias, por ejemplo flotante o entero largo. En contraste con un registro, que se podría definir para contener un flotante y un entero largo, en una unión sólo hay un valor a la vez. Se asigna suficiente espacio para contener el tipo de datos de cualquiera de los miembros.

e) Un tipo variante (también llamado registro variante o unión discriminada) contiene un campo adicional que indica su tipo actual.

f) Un conjunto es un tipo de datos abstracto que puede almacenar valores específicos, sin orden particular y sin valores duplicados.

g) Un multiconjunto es un tipo de datos abstracto que puede almacenar valores específicos, sin orden particular. A diferencia de los conjuntos, los multiconjuntos admiten repeticiones.

h) Un grafo es una estructura de datos conectada compuesta por nodos. Cada nodo contiene un valor y una o más referencias a otros nodos. Los grafos pueden

utilizarse para representar redes, dado que los nodos pueden referenciarse entre ellos. Las conexiones entre nodos pueden tener dirección, es decir un nodo de partida y uno de llegada.

i)Un árbol es un caso particular de grafo dirigido en el que no se admiten ciclos y existe un camino desde un nodo llamado raíz hasta cada uno de los otros nodos. Una colección de árboles es llamada un bosque.

j) Una clase es una plantilla para la creación de objetos de datos según un modelo predefinido. Las clases se utilizan como representación abstracta de conceptos, incluyen campos como los registros y operaciones que pueden consultar el valor de los campos o cambiar sus valores.

Figura 1. Diagrama de clases.

Fuente: elaboración propia.

Características de XML

• El XML separa datos de HTML: Si necesita mostrar datos dinámicos en su documento HTML, tendrá que dedicarle mucho trabajo a editarlos cada vez que los datos cambien. Con el XML, los datos se pueden almacenar en archivos XML separados. De esa manera, puedes usar HTML para la visualización y el diseño. Con algunas líneas de código JavaScript, puedes leer un archivo XML externo y actualizar el contenido de los datos de tu página web.

El diseño XML se centra en la simplicidad, la generalidad y la facilidad de uso y, por lo tanto, se utiliza para varios servicios web. Tanto es así que hay sistemas destinados a ayudar en la definición de lenguajes basados en XML, así como APIs que ayudan en el procesamiento de datos XML – que no deben confundirse con HTML.

```
<persona>
   <nombres>Elsa</nombres>
   <apellidos>Zambrano</apellidos>
   <fecha-de-nacimiento>
      <día>18</día>
      <mes>6</mes>
      <año>1996</año>
   </fecha-de-nacimiento>
   <ciudad>Pamplona</ciudad>
</persona>
```

Figura 2. Estructura XML.

Fuente: ResearchGate

. Tipo de dato abstracto

El concepto de tipo de dato abstracto (TDA, Abstract Data Type), fue propuesto por primera vez hacia 1974 por John Guttag y otros, pero no fue hasta 1975 que por primera vez Barbara Liskov lo propuso para el lenguaje CLU.

Los Lenguajes de Programación Orientados a Objetos son lenguajes formados por diferentes métodos o funciones y que son llamados en el orden en que el programa lo requiere, o el usuario lo desea. La abstracción de datos consiste en ocultar las características de un objeto y obviarlas, de manera que solamente utilizamos el nombre del objeto en nuestro programa. Esto es similar a una situación de la vida cotidiana. Cuando yo digo la palabra “perro”, usted no necesita que yo le diga lo que hace el perro. Usted ya sabe la forma que tiene un perro y también sabe que los perros ladran. De manera que yo abstraigo todas las características de todos los perros en un solo término, al cual llamó “perro”. A esto se le llama ‘abstracción’ y es un concepto muy útil en la programación, ya que un usuario no necesita

mencionar todas las características y funciones de un objeto cada vez que este se utiliza, sino que son declaradas por separado en el programa y simplemente se utiliza el término abstracto (“perro”) para mencionarlo.
Para lograr esto, se crea el método Área de una manera separada de la interfaz gráfica presentada al usuario y se estipula ahí la operación a realizar, devolviendo el valor de la multiplicación. En el método principal solamente se llama a la función Área y el programa hace el resto.

Al hecho de guardar todas las características y habilidades de un objeto por separado se le llama Encapsulamiento y es también un concepto importante para entender la estructuración de datos. Es frecuente que el Encapsulamiento sea usado como un sinónimo del Ocultación de información

Conclusiones

Las estructuras de datos son esenciales para organizar datos en una computadora y que posteriormente se puedan utilizar eficientemente.

Implementar un tipo de dato abstracto permite la comprensión de la correcta utilización del paradigma de programación orientado a objetos.

Manejar datos matricialmente y convertirlos visualmente en un grafo favorece la precepción y facilita el aprendizaje para futuros eventos en los cuales se vean involucrados grandes cantidades de datos.

Referencias bibliográficas

UChile, (2021). Tipos de datos abstractos.
https://users.dcc.uchile.cl/

Wikipedia, (2021). Tipo de dato abstracto. https://es.wikipedia.org/wiki/Tipo_de_dato_abstract o.

Wikipedia, (2021). Estructura de datos. https://es.wikipedia.org/wiki/Estructura_de_datos#:~:text=En%20ciencias%20de%20la%20computaci%C3%B3n,ser%20utilizados%20de%20manera%20eficiente.&text=Por%20lo%20general%2C%20las%20estructuras,clave%20para%20dise%C3%B1ar%20algoritmos%20eficientes.

Extensión: de cuatro a siete páginas como máximo

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