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.

GRAFICADOR Y ANALIZADOR DE OBJETOS DE TIPO DE DATO ABSTRACTO

201902781 – Rodrigo Antonio Porón De León

Resumen

El presente proyecto se centra en la implementación de tipos de datos abstractos (TDA), con el fin de poder visualizar este tipo de datos utilizando archivos XML como insumo para su solución. Un tipo de dato abstracto 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 más adecuado de una cantidad elevada de datos, los cuales precisan de un acceso eficaz y ordenado.

Un mismo TDA puede ser implementado utilizando distintas estructuras de datos y que proveen la misma funcionalidad. Por ejemplo, de una clase Nodo se puede heredar a una clase Lista, la cual también se puede heredar en una clase Matriz. Ya que una matriz es una lista de listas, implementando de ese modo la reutilización de código. Por lo tanto, se hace uso del paradigma de programación orientada a objetos para contar con las funcionalidades anteriormente explicadas para la solución de este proyecto.

Palabras clave

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

Abstract

This Project is focused on implementing abstract data with the purpose of visualizing this type of data using XML files as a resource for its solution. A type of abstract data is a set of data or objects to which operations are associated. The ADT management is used when it is required a more appropriate management of a large amount of data which would need an efficient and organized access.

The same ADT could be implemented by using different data structures which provide the same function. As an example, from one Node type it could be inherit a type of matrix as well. Due to a matrix being a list of lists, implementing the reusage of the programming code that way. Therefore, the programming paradigm is used oriented to objects in order to count on the functionalities previously explained to the solution of this project.

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.

Estructura de Datos en Programación

En programación, una estructura de datos puede ser declarada inicialmente escribiendo una palabra reservada, luego un identificador para la estructura y un nombre para cada uno de sus miembros, sin olvidar los tipos de datos que estos representan. Generalmente, cada miembro se separa con algún tipo de operador, carácter o palabra reservada.

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.

En el ejemplo anterior, “perro” es un Tipo de Dato Abstracto y todo el proceso de definirlo, implementarlo y mencionarlo es a lo que llamamos Abstracción de Datos.

Vamos a poner un ejemplo real de la programación. Supongamos que en algún Lenguaje de Programación Orientado a Objetos un pequeño programa saca el

área de un rectángulo de las dimensiones que un usuario decida. Pensemos también que el usuario probablemente quiera saber el área de varios rectángulos. Sería muy tedioso para el programador definir la multiplicación de 'base' por 'altura' varias veces en el programa, además que limitaría al usuario a sacar un número determinado de áreas. Por ello, el programador puede crear una función denominada 'Área', la cual va a ser llamada, con los parámetros correspondientes, el número de veces que sean necesitadas por el usuario y así el programador se evita mucho trabajo, el programa resulta más rápido, más eficiente y de menor longitud.

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.

Archivos XML

XML es el acrónimo de Extensible Markup Language, es decir, es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos.

El lenguaje de marcado es un conjunto de códigos que se pueden aplicar en el análisis de datos o la lectura de textos creados por computadoras o personas. El lenguaje XML proporciona una plataforma para definir elementos para crear un formato y generar un lenguaje personalizado.
Un archivo XML se divide en dos partes: prolog y body. La parte prolog consiste en metadatos administrativos, como declaración XML, instrucción de procesamiento opcional, declaración de tipo de documento y comentarios. La parte del body se compone de dos partes: estructural y de contenido (presente en los textos simples).

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 1. Ejemplo básico documento XML

Fuente: ResearchGate

¿Qué es HTML?

HTML (Hypertext Markup Language) es el lenguaje de marcado de documentos para construir páginas web. Por lo tanto, los comandos de formato utilizados en los contenidos para web se refieren a la estructura del mismo y al diseño que se mostrará en el navegador.

Es decir, los navegadores leen el documento con el formato HTML y lo procesan en la pantalla mediante el examen de los elementos HTML insertados en el documento, que se considera un archivo de texto con la información que se debe publicar. Por eso,

podemos generar un archivo HTML utilizando el Bloc de notas de nuestra computadora, por ejemplo.

Las instrucciones incorporadas se conocen como elementos que muestran la estructura y la presentación del documento en el navegador. Estos elementos se componen de las tags que definen el formato de un texto. Las tags suelen estar dos veces: tag inicial y tag final. Por ejemplo, para poner el texto en negrita, las etiquetas se usan al principio y </ strong> al final.

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.

- XML simplifica el intercambio de datos:
  Tanto los sistemas informáticos como las bases de datos contienen información en formatos incompatibles.

  Los datos XML se almacenan en formato de texto simple, lo que nos posibilita una forma independiente de almacenar datos. Esto facilita mucho la creación de datos que pueden ser compartidos por diferentes aplicaciones.

Graphviz

Graphviz es una librería o programa que permite visualizar gráficamente archivos, denotando que es de fuente abierta.

Está hecho especialmente para poder representar información estructural de diagramas gráficos y redes abstractas. Su uso se implementa mayormente en ingeniería de software, bioinformática, diseño de bases de datos, aprendizaje automático (machine learning), etc.

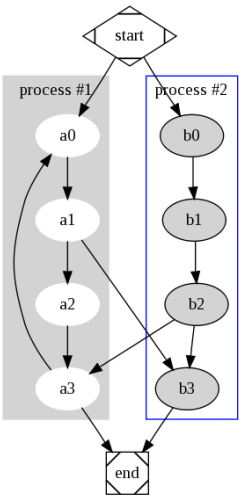

Figura 2. Gráfica generada con graphviz

Fuente: Graphviz

Programación Orientada a Objetos.

La Programación Orientada a Objetos (POO, en español; OOP, según sus siglas en inglés) es un

paradigma de programación que viene a innovar la forma de obtener resultados. Está basada en varias técnicas del sexenio: herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento.

Los métodos (comportamiento) y atributos (estado) están estrechamente relacionados por la propiedad de conjunto. Esta propiedad destaca que una clase requiere de métodos para poder tratar los atributos con los que cuenta. El programador debe pensar indistintamente en ambos conceptos, sin separar ni darle mayor importancia a alguno de ellos. Hacerlo podría producir el hábito erróneo de crear clases contenedoras de información por un lado y clases con métodos que manejen a las primeras por el otro. De esta manera se estaría realizando una "programación estructurada camuflada" en un lenguaje de POO.

Los conceptos de la POO tienen origen en Simula 67, un lenguaje diseñado para hacer simulaciones, creado por Ole-Johan Dahl y Kristen Nygaard, del Centro de Cómputo Noruego en Oslo. En este centro se trabajaba en simulaciones de naves, que fueron confundidas por la explosión combinatoria de cómo las diversas cualidades de diferentes naves podían afectar unas a las otras. La idea surgió al agrupar los diversos tipos de naves en diversas clases de objetos, siendo responsable cada clase de objetos de definir sus "propios" datos y comportamientos. Fueron refinados más tarde en Smalltalk, desarrollado en Simula en Xerox PARC (cuya primera versión fue escrita sobre Basic) pero diseñado para ser un sistema completamente dinámico en el cual los objetos se podrían crear y modificar "sobre la marcha" (en tiempo de ejecución) en lugar de tener un sistema basado en programas estáticos.)

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

Rockcontent, (2019). *XML: ¿qué es y para qué sirve este lenguaje de marcado?* https://rockcontent.com/es/blog/que-es-xml/

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

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.

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