
PROYECTO 1

201900157 – Kevin Alexis Lopez Contreras

Resumen

La solución del problema presentado se soluciono de manera que el costo total de la transmisión de datos para el procesamiento de todas las aplicaciones sea minimizado, para llegar a este resultado de minimizar el costo de transferencia se utilizó estructuras de datos abstractos (TDA's) esto para poder tener en la memoria espacio no desea sin ser utilizado ya que permite tener un mejor rendimiento en el programa, la escalabilidad del problema puede ser grande si no se optimiza de una forma adecuada los datos abstractos para poder optimizar de una forma adecuada los datos se requirió utilizar listas circulares simples enlazadas ya que minimiza la memoria el uso de la memoria del computador y el programa actúa de una manera eficiente disminuyendo los costos el paradigma orientado a objetos es muy útil para poder manejar listas circulares simples enlazadas

ya que su manejo con este paradigma lo hace más eficiente.

Palabras clave

Lista Circular Simple: la lista no tiene fin ya que al llegar al último nodo empieza de nuevo la lista. Se accede a la lista mediante el primer nodo o también llamado inicio de la lista

TDA's: Un tipo de datos abstracto se define por su comportamiento (semántica) desde el punto de vista de un usuario , de los datos, específicamente en términos de posibles valores, posibles operaciones sobre datos de este tipo y el comportamiento de estas operaciones

Matrices: Una matriz es una matriz rectangular de números (u otros objetos matemáticos) para los que se definen operaciones como la suma y la multiplicación .

Orientado a objetos: es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos se utilizan como metáfora para emular las entidades reales del negocio a modelar.

Muchos de los objetos prediseñados de los lenguajes de programación actuales permiten la agrupación en bibliotecas o librerías, sin embargo, muchos de estos lenguajes permiten al usuario la creación de sus propias bibliotecas.

Está basada en varias técnicas del sexenio: herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento.

Su uso se popularizó a principios de la década de 1990. En la actualidad, existe una gran variedad de lenguajes de programación que soportan la orientación a objetos.

Abstract

The solution of the presented problem was solved in such a way that the total cost of data transmission for the processing of all applications is minimized, to arrive at this result of minimizing the transfer cost, abstract data structures (TDA's) were used, this for Being able to have space in memory you do not

want without being used since it allows to have a better performance in the program, the scalability of the problem can be great if the abstract data is not optimized in an adequate way to be able to optimize the data in a suitable way It required the use of simple linked circular lists since it minimizes memory, the use of computer memory and the program acts in an efficient way reducing costs. The object-oriented paradigm is very useful to be able to handle simple linked circular lists.

Keywords

Simple Circular List: the list has no end since when it reaches the last node the list starts again. The list is accessed through the first node or also called the start of the list

ADTs: An abstract data type is defined by its behavior (semantics) from the point of view of a user, of the data, specifically in terms of possible values, possible operations on data of this type and the behavior of these operations

Matrices: A matrix is a rectangular matrix of numbers (or other mathematical objects) for which operations such as addition and multiplication are defined.

Object-oriented: it is a programming paradigm that innovates the way to obtain results. The objects are used as a metaphor to emulate the real entities of the business to be modeled.

Many of the predesigned objects in current programming languages allow grouping into libraries or libraries, however, many of these languages allow the user to create their own libraries.

It is based on various techniques of the six-year term: inheritance, cohesion, abstraction, polymorphism, coupling and encapsulation.

Its use became popular in the early 1990s. Today, there is a wide variety of programming languages that support object orientation.

Introducción

El objetivo o la utilidad de la idea de usar TDA's es la de conseguir una mayor flexibilidad. Y esto lo logramos mediante el concepto más general de abstracción. Como vimos en el ejemplo, el código que utiliza el TDA no conoce ni depende de la implementación de las operaciones.

La abstracción es probablemente el concepto más general y más importante de la programación (o cualquier otra resolución de problemas).

En parte tiene que ver con la famosa idea del divide

y conquista. Porque *separamos* los problemas, hacemos que diferentes partes de la aplicación se

La idea de modularización se puede pensar como la de ocultamiento y encapsulamiento, pero a aplicadas a un nivel más amplio o general de la aplicación. No solo a una unidad, como vimos a un Tipo u objeto, si no a un conjunto de éstos.

Al agruparlos y dividirlos definimos módulos.

Desarrollo del tema

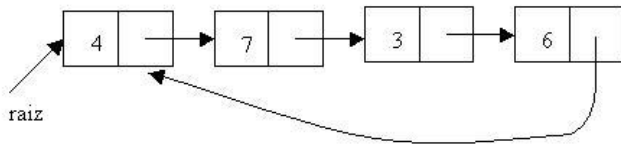
Las estructuras de datos se utilizaron en esta solución porque era la más factible de implementar ya que los datos que se utilizaran son adecuados para ingresarlos a una lista circular simple ¿y por qué este tipo de lista?

Pues bien la lista circular simple enlazada se adecua a esta solución porque el programa maneja matrices ya entonces estos datos pueden retornar de una forma mas sencilla y también se maneja de una forma mucho más correcta los espacios de memoria que no utiliza el programa, en algunos programas tienen ese peculiar problema que el programa es muy lento y esto puede ocurrir por muchos factores pero el principal factor de que ocurra esto es los espacios de memoria que se quedan sin utilizar y esto el equipo de cómputo lo compila o lo interpreta y esto genera más demora para que se ejecute el

programa por esta razón se utilizó una lista circular simple enlazada.

Para agregar cada nodo a una lista se requieren de funciones cíclicas esto ayuda a agregar los datos de una forma adecuada sin fugas de memoria dinámica.

Figura 1. Lista circular simple enlazada



fuelle: tutorialesprogramacionya.com

Conclusiones

La implementación del paradigma orientado a objetos fue factible ya que con programación orientado a objetos el programa se optimizó el tiempo de carga.

La utilización de funciones cíclicas se implementó correctamente ya que no hubo ninguna fuga de memoria esto quiere decir que los nodos se insertaron de una forma adecuada.

Adicionalmente, pueden incluirse preguntas abiertas a la reflexión y debate, temas concatenados con el tema expuesto o recomendaciones para profundizar en la temática expuesta.

Se visualizan los TDA's creados por la herramienta Graphviz de una forma que el usuario los pueda entender y analizar.

Utilizar archivos XML como insumos para la lógica y comportamiento de la solución.

Referencias bibliográficas .

Hernández, R., Lázaro, J. C., Dormido, R., & Ros, S. (2001). Estructuras de datos y Algoritmos. Prentice Hall.

Joyanes Aguilar, L. (1998). Fundamentos de Programación, Algoritmos y Estructuras de Datos. McGraw-Hill.