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 2022.

PROYECTO 1 IPC2

201906572 – Allan Ricardo Barillas Sosa

Resumen

El presente proyecto fue realizado con listas realizadas mediante tda utilizando todos los conocimientos adquiridos anteriormente en POO.

El utilizar la memoria dinámica en la programación resulta un tanto incomodo al principio, ya que se desconoce como guiar correctamente los apuntadores hacia otros objetos llamados nodos en una lista.

Las listas utilizadas fueron una circular simple y una lista ortogonal, tomando en cuenta que cada nodo de la lista tiene los apuntadores correspondientes hacia los demás nodos, y los cuales debemos guardarlos y cuidarlos para no cometer el error de perder el apuntador.

Al referirnos a perder el apuntador, es perder la forma de acceder a un nodo, principalmente en la lista ortogonal sabiendo que cada nodo tiene 4 apuntadores.

Abstract

The present project was executed with lists made with adt(abstract data types), using all the knowledge acquired previously in POO.

While speaking of dynamic memory in, The use of such tool is somewhat uncomfortable at the beginning phase,since it is not known how tocorrectly guide the pointers to other objects called nodes in a list. and by that, proceeding even further into projects can be slow and complex.

The implemented lists used for this project were a simple circular list and an orthogonal list, while considering that each node of the list has the corresponding pointers to the other existing nodes, and which we must keep and take care of them in order to avoid any mistakes related to losing the pointer.

When refering to losing a pointer, it means that we lost the way to access the node, mainly in the orthogonal list knowing that each node has 4 pointers.

Palabras clave

Lista circular sencilla: es aquella en la que sus nodos se encuentran enlazados únicamente por una liga, yel último nodo de la lista apunta hacia el primer nodo de la lista.

Lista Ortogonal: es aquella en la que sus nodos se encuentran encadenados por cuatro ligas.

Nodo: Apuntador al nodo cuya dirección de memoria está en nodo.

Keywords

Simple circular list*: It is the one that has every node bonded by a link, the last node in the list points the direction of the first node in the list.*

Orthogonal list*: It is the one in which all nodes are chained by four leagues.*

Node*: Pointer to the node whose memory address isat node*

Introducción

Una estructura de datos en general es una colección de datos, unidos y contiguos por la memoria dinámica del computador, llevando consigodistintos nodos apuntando al consiguiente cada nodo de una lista es un mismo tipo de dato u objetollamado en la programación.

Estructuras de datos dinámicas: Son aquellas en las que su ocupación en memoria puede aumentar o disminuir durante el tiempo de ejecución de un programa. A su vez las estructuras de datos dinámicas se pueden clasificar en lineales y no lineales.

El tipo de datos abstractos (TDA) es un modelo que describe datos u objetos y actividades que se pueden realizar en él. Y se le llama abstracto porque la intención es que quien lo use no necesite conocer los detalles de la representación interna o la ejecución de las actividades, lo que permite la colocación de datos y actividades y gracias a esto se puede ocultar, cómo TDA es. implementado.

Desarrollo del tema

Este programa fue desarrollado a través de una solución integral que implementó tipos de datos hipotéticos (TDA) e imágenes (Graphviz) bajo el concepto de Planificación Orientada a Objetos (POO).

El tipo de datos de resumen es una estructura que describe datos u objetos y actividades que se pueden realizar en él. Y se le llama abstracto porque la intención es que quien lo use no necesite conocerlos detalles de la representación interna o la ejecución de las actividades, lo que permite la colocación de datos y actividades y gracias a esto sepuede ocultar, cómo TDA es. implementado.

Debido a esto, podemos usar TDA para separar el código del código que está implementando.

Al usar TDA, puede crear listas que son estructuras de datos flexibles, lo que significa que se les da espacio y mantienen el espacio según sea necesario. Las estructuras más complejas y ampliamente utilizadas, como la configuración de lista, se pueden definir a partir de la lista. A veces, los gráficos se definen como listas cerradas. También se utiliza como columnas de lista para tablas.

Estructuras lineales: Son aquellas en las que el orden se define como los conjuntos de elementossobre los que se construye la relación predecesora y sucesora. Las diferentesestructuras de datos de acuerdo con este concepto difieren en el funcionamiento de los componentes de adquisición y el funcionamientode la estructura. Hay tres estructuras de filasprincipales: columnas, colas y listas.

Estructuras no lineales. Son aquellas en las que no existe una relación estrecha entre sus componentes, es decir. Un elemento puede estarvinculado a cero, uno o más elementos. Hay dosdiseños principales que no son de línea: árboles y gráficos.

Listas ortogonales:
Una lista ortogonal es aquella en el que los nodos están conectados por cuatro enlaces, es decir, cada nodo está conectado horizontalmente dos veces y cada nodo está conectado verticalmente dos veces.La lista ortogonal se puede diseñar como una líneao un círculo. Estos tipos de listas se pueden utilizarpara representar matrices.

Listas sencillas circular:
Una lista sencilla es aquella cuyos nodos solo están conectados al nodo, es decir, cada nodo apunta al siguiente nodo de la lista y cada nodo se redirige al nodo anterior de la lista, con la excepción del primer nodo y el último nodo de la lista. En un círculo, el último nodo de la lista apunta al primer nodo de la lista.

Las listas pueden ser operadas por algoritmos que deben desarrollarse según el tipo de lista. Algunas de

las actividades básicas que se pueden realizar enla lista son, como se muestra en el diagrama de clases de las dos listas que tenemos, también le llamamos métodos de las clases.

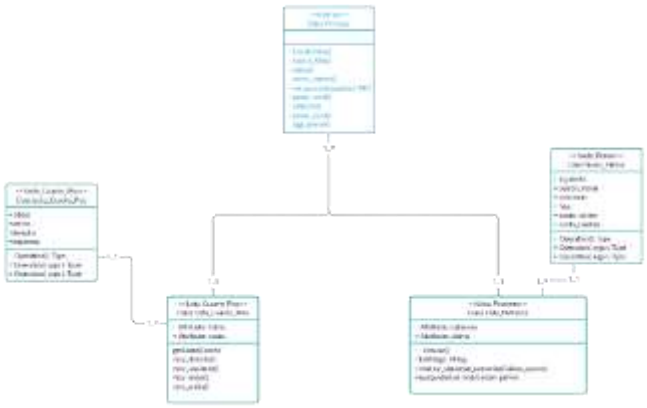

Figura 1. Diagrama de Clases

Fuente: elaboración propia.

Conclusiones

- Para el desarrollo de este proyecto es necesario tener claro los conceptos de tipos dedatos abstractos y visualización a través de graphviz.
- El uso de matrices doblemente enlazadas facilita la navegación de una estructura de varias listas altrabajar con tipos de datos abstractos.
- Es necesaria la implementación orientada a objetos en el desarrollo de este tipo de proyectos en donde se hace uso de TDA ya que hace más fácil la manipulación de estos.
- Se debe tener en cuenta que el proyecto se desarrolló con los siguientes datos técnicos: Lenguaje utilizado: Python
- IDE utilizado: Visual Studio Code Sistema operativo: Windows 10(64 bits)

Referencias bibliográficas

Luis Joyanes Aguilar, Ignacio Zahonero Martínez, Estructura de Datos: Algoritmos,abstracción y objetos, McGraw-Hill interamericana 1998.

El sistema de los objetos, 10ed, México

Luis Joyanes,Fundamentos de Programación: Algoritmos, estructuras de datos y Objetos, Aguilar, 3ed Madrid, McGraw-Hill Interamericana de españa

Schah, Stephen R ,Ingenieria De Software Clasica Y Orientada A Objetos, 6 ed. México, McGraw Hill Interamericana.

Eric J Braude, Tr: María C murcias, RevTec. Roberto valdiviaBeutelspacher, Ingeniería de Software: Una Perspectiva Orientada a Objetos, México, Alfaomega.

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 2022.

Muestra de algunos cálculos

Forma de calcular los movimientos necesarios para mover dos nodos

						pos fin - pos in -1				
			temp.arriba	**auxi.arriba**				1		
	1	**temp.anterior**	**auxi**	temp	**auxi.sig**					
3			**temp.abajo**	**auxi.abajo**					2xesp+1	2 * nds -3
		diferente de None	temp.arriba	auxi.arriba		2 nds	0 esp	1	1	1
	4	temp.anterior	temp	auxi	auxi.sig	3nds	1 esp	3	3	3
			temp.abajo	auxi.abajo		4nds	2 esp	5	5	5
						5nds	3 esp	7	7	7
						6nds	4 esp	9	9	9
						7nds	5 esp	11	11	11