Instituto Tecnológico Superior de San Andrés Tuxtla.

Alumnos:

Grupo:

303 − A

Fecha:

27 de Noviembre de 2000

Vo.Bo.

_____

UNIDAD 6: OTRAS ESTRUCTURAS DE DATOS

GRAFOS

CONCEPTO

Un grafo es un diagrama que consiste de puntos (llamados nodos) unidos por líneas (llamadas arcos). Cada arco en un grafo se especifica por medio de un par de nodos.

Según la figura anterior, los nodos serian (A,B,C,D,F) y los arcos[(A,B),(A,C),(A,D),(C,D),(C,F)]. Si los pares de nodos que conforman los arcos son pares ordenados, el grafo se denomina grafo directo o digrafo.

La cabeza de cada flecha representa el segundo nodo en el par de nodos ordenados que hacen un arco, mientras que la cola de la flecha representa el primer nodo en el par.

Un nodo *n* se dice que es incidente a un arco *x*, si *n* es uno de los dos nodos en el par ordenado de nodos que componen *x*. (también se dice que *x* es incidente a *n*). El grado de un nodo es el numero de arcos incidentes a el.

Un nodo *n* es adyacente al nodo *n* si existe un arco de *m* a *n.*

Una relación *R* en un conjunto *A* es un grupo de pares ordenados de elementos de A. Si (*x*,*y*) es un miembro de la relación *R*, entonces *x* se dice que está relacionado con *y* en *R*.

Una relación se puede representar por un grafo en el cuál los nodos representan el conjunto fundamental y los arcos representan los pares ordenados de la relación.

- **10**

5 8 17

6

LAZO

Es cuando un arco sale de un nodo y regresa al mismo nodo sin tocar otro nodo diferente.

LADOS PARALELOS

Es cuando dos arcos salen de un mismo nodo y ambos llegan a otro nodo

GRAFO SIMPLE

Es aquel grafo que no tiene ni lazos, ni lados paralelos.

GRAFOS CONEXOS

Caminos y circuitos:

Un camino de longitud n de V a W es una sucesión de lados que va de V a W y la cual tiene n lados distintos entre sí.

Un camino simple de longitud n de V a W es una de la forma(V0,V1,V2...Vn) en donde V0=V y Vn =W,y VV0,V1,Vn son distintos entre sí.

Un circuito o ciclo es un camino de V aV.

Un circuito simple es un circuito de la forma : (V0,V1,V2...Vn) en donde V0=Vn y V0,V1...Vn−1 son distintos entre sí.

c

b g

d

a

e

f

De acuerdo a la figura anterior se tiene:

SUCESION DE LADOS ¿CAMINO? ¿CAMINO SIMPLE? ¿CIRCUITO? ¿CIRCUITO SIMPLE?

(a,b,d,c,b,a) NO NO NO NO

(f,e,b,d,c,b,a) SI NO NO NO

(f,e,b,d) SI SI NO NO

(b,f,e,b,d,c,b) SI NO SI NO

(e,f,b,e) SI NO SI NO

CIRCUITO DE HAMILTON

En un grafo se tiene un circuito de Hamilton cuando se inicie en un nodo y termina en el mismo pasando exactamente una vez por cada nodo.

CIRCUITO DE VENN EULER

Este circuito se da en grafos donde se realiza un camino donde se puede pasar solo una vez por cada arco.

El factor de peso o Ponderación se da en un grafo en el cual hay datos asociados con un arco.

1

- **10**

7

3 1

5 8 17

1 5

6

A continuación identificaremos algunas operaciones básicas que son utiles para manejar grafos. La operación union (a,b) agrega un arco del nodo al nodo b si este no existe. Union wt (a,b,x) agrega un arco de a a b con el factor de peso x en un grafo con factor de peso. Remv(a,b) y remvwt (a,b,x) remueve un arco desde a hasta b si este existe (revwt también coloca x a su factor de peso). La función adjacent (a,b) retorna el valor de verdadero si b es adyacente a a y falso en caso contrario.

VALENCIA

Es la numero de arcos coincidentes en un nodo.

Las valencias de los nodos de este grafo son:

A=3

B=1

C=3

D=2

F=1

APLICACIONES

Una de las aplicaciones mas importantes es de hallar el camino mas corto hacia un destino, ya sea de una ciudad a otra, de unos departamentos a otros, para el recorrido de arboles, sirve para la representación de algoritmos, etc.

Un ejemplo de esto es la tarea de freir un huevo:

FRAMES

PRINCIPIOS DE CONMUTACION DE PAQUETES

CONMUTACION DE PAQUETES: Método de transmisión de mensajes a través de una red de comunicación, en la que los mensajes largos se subdividen en pequeños paquetes. Los paquetes se transmiten después como en conmutación de mensajes.

CONMUTACION DE CIRCUITOS: Método de comunicación en el que se establece un camino de comunicación entre dos dispositivos a través de uno o más nodos de conmutación intermedios.

TECNICA DE CONMUTACION

Es la forma como trata la red los paquetes para llevar a cabo el envío hacia el destino. Existen dos tipos de soluciones usadas en las redes actuales: datagramas y circuitos virtuales.

En la técnica del datagrama cada paquete se trata de forma independiente, sin ninguna referencia a los paquetes precedentes. Por ejemplo, si un nodo de conmutación de paquetes se encuentra momentáneamente inactivos, se perderán todos sus paquetes en cola.

VENTAJAS DE LOS DATAGRAMAS

- No existe la fase de establecimiento de llamada.
- Mayor flexibilidad.
- El envío de datagramas es inherente más seguro.

En la técnica de circuitos virtuales se fija una ruta previa al envío de algún paquete.

La principal característica de la técnica de circuitos virtuales radica en el establecimiento de la ruta previa a la transferencia de datos. Esto no significa que se trate de un camino dedicado como el caso de conmutación de circuitos.Un paquete continua siendo almacenado en cada nodo y puesto en cola sobre una línea de salida. La diferencia con la técnica de datagramas es que, con circuitos virtuales, el nodo no necesita tomar una decisión de encaminamiento para cada paquete, si no que este se realiza una sola vez para todos los paquetes que usan dicho circuito virtual.

COMPARACION DE LAS TECNICAS DE CONMUTACION DE CIRCUITOS Y CONMUTACION DE PAQUETES

PRESTACIONES

En la siguiente figura se realiza una sencilla comparación entre la conmutación de circuitos y las dos técnicas de conmutaciòn de paquetes. Esta figura muestra la transmisión de un mensaje a través de 4 nodos, desde una estación emisora conectada al nodo 1 hasta una estación destino conectada al nodo 4. En esta figura existen 3 tipos de retardo.

- Retardo de propagación: es el tiempo de propagación de una señal desde un nodo hasta el siguiente.

El tiempo es generalmente despreciable, ya que la velocidad típica de las señales elextromagneticas a través de un cable es de 2 x 108 m/s.

- Tiempo de transmisión: Es el tiempo necesario para la transmisión de un paquete de datos. Por ejemplo, para una línea de 10 Kbps x seg. La transmisión de 10000 bits requiere 1 segundo.
- Retardo de nodo: Es el tiempo que necesita un nodo para realizar el procesamiento involucrado en la conmutación de datos.

La tecnica de conmutación de paquetes mediante circuitos virtuales parece muy similar a la de conmutación de circuitos.

Comparación de técnicas de conmutación en comunicaciones

Conmutación de circuitos	*Conmutacion de paquetes mediante datagramas*	*Conmutación de paquetes mediante circuitos virtuales.*
Ruta de transmisión dedicada	Ruta de transmisión no dedicada	Ruta de transmisión no dedicada
Transmisión de datos continua	Transmisión de paquetes	Transmisión de paquetes
Suficientemente rápida para aplicaciones interactivas	Suficientemente rápida para aplicaciones interactivas	Suficientemente rápida para aplicaciones interactivas
Los mensajes no se almacenan	Los paquetes pueden almacenarse hasta su entrega	Los paquetes se almacenan hasta su entrega.
Las rutas se establecen para la conversación entera	Se establece una ruta para cada paquete	Las rutas se establecen para la conversación entera
Retardo de establecimiento de llamada; retardo de transmisión despreciable.	Retardo de transmisión de paquetes	Retardo de establecimiento de llamada; retardo de transmisión de paquetes.
Uso de señal de ocupación si el extremo llamado esta ocupado	Si un paquete no se ha entregado debe ser notificado al emisor	El emisor debe ser avisado en caso de que se niegue la conexión
La sobrecarga puede bloquear el establecimiento de llamadas; no existe retardo en las llamadas establecidas.	La sobrecarga aumenta en el retardo de paquete	La sobrecarga puede bloquear el establecimiento de llamadas; aumenta el retardo de paquetes.
Nodos de conmutación electromecánicos o computarizados	Nodos de conmutación pequeños	Nodos de conmutación pequeños
El usuario es el responsable de la protección ante la pérdida de un mensaje	La red puede ser la responsable de paquetes individuales.	La red puede ser responsable de secuencias de paquetes.
Generalmente no existe conversión de velocidad o código	Conversión de velocidad y código	Conversión de velocidad y código
Ancho de banda de transmisión fijo	Uso dinámico del ancho de banda	Uso dinámico del ancho de banda
Ausencia de bits suplementarios tras el establecimiento de llamada	Uso de bits suplementarios en cada paquete.	Uso de bits suplementarios en cada paquete.

Funcionamiento interno y externo.

Una de las características mas importantes de una red de conmutación de paquetes es el uso de datagramas o de circuitos virtuales. Hay dos niveles o dimensiones de sus características. En la interfaz entre estación y nodo de red, puede ofrecer tanto un servicio de circuito virtual como uno de datagrama. Todos los paquetes enviados por la red son identificados como pertenecientes a una conexión lógica dada y son numerados secuencialmente.

Estas decisiones de diseño interno y externo no necesitan ser coincidentes:

- Circuito virtual externo, circuito virtual interno: cuando el usuario solicita un circuito virtual se crea un camino dedicado a través de la red, siguiendo todos los paquetes de la misma ruta
- Circuito virtual externo, datagrama interno: la red maneja separadamente cada paquete, de manera que los paquetes correspondientes a un mismo circuito virtual siguen caminos diferentes, aunque, eso si, son enviados en orden secuencial.
- Datagrama externo, datagrama interno: cada paquete se trata de forma independiente, tanto desde el punto de vista del usuario, como desde el de la red.
- Datagrama Externo, circuito virtual interno: el usuario externo no ve ninguna conexión, limitándose a enviar paquetes a lo largo del tiempo. La red, sin embargo, establece una conexión lógica entre estaciones para el envío de paquetes, pudiendo mantener la conexión durante un largo periodo de tiempo para satisfacer futuras necesidades.

Secuencia de eventos: protocolo X.25

- A solicita un circuito virtual a B mediante el envio de un paquete petición de llamada (Call Request) al DCE de A. El paquete incluye las direcciones fuente y destino asi como el número a usar en este nuevo circuito virtual. Las futuras transmisiones de entrada y salida seran identificadas con este número de circuito virtual.
- La red encamina esta petición de llamada al DCE (equipo de circuitos de datos) de B.
- El DCE de B recibe el paquete petición de llamada y envia un paquete llamada entrante (incoming Call) a B. Este paquete tiene el mismo formato que el de petición de llamada pero con diferente número de circuito virtual. Este número es seleccionado por DCE de B entre el conjunto de números locales libres.
- B indica la aceptación de la llamada mediante el envio del paquete llamada aceptada (Call Acepted), especificando el mismo número de circuitos virtuales que el del paquete llamada entrante.
- El DCE de A recibe el paquete llamada aceptada y envia el paquete llamada establecida (call Connected) a A. Este paquete tiene el mismo formato que el de llamada aceptada pero el mismo número de circuito virtual que el paquete petición de llamada original.
- A y B se envían paquetes de datos y de control entre si, haciendo uso de sus respectivos números de circuitos virtual.
- A (o B) envia un paquete petición liberación (Clear Request) para liberar el circuito virtual y recibe un paquete confirmación de liberación (Clear confirmation).
- B (o A) recibe un paquete Indicación de Liberación (Clear Indication) y transmite uno confirmación de liberación.

OBJETOS.

El termino *Programación Orientada a Objetos* (POO) es difícil de definir, ya que es un desarrollo de técnicas de programación desde principios de la década de los setenta, aunque es en la década de los 90's cuando se ha dado el auge en el desarrollo, difusión uso y popularidad. No obstante se puede definir POO como una técnica o estilo de programación que utiliza objetos como bloque general de instrucción.

Los objetos son en realidad un *tipo abstracto de datos* (TAD), que es un tipo de datos definido por el programador junto con un conjunto de operaciones que se pueden realizar sobre ellos. Se denomina *abstracto* para diferenciarlos de los tipos de datos fundamentales (char, Integer, Real, etc.) o básicos definidos.

Al igual que los tipos de datos definidos por el usuario, un objeto es una colección de datos junto con las funciones asociadas, utilizadas para operar sobre estos datos. La potencia real reside en las propiedades que soportan: *herencia, encapsulación (o encapsulamiento) y polimorfismo.* Junto con los conceptos de *objetos, clases, métodos* y *mensajes*

¿Qué es un objeto?

Un objeto es *la unidad que contiene datos y las funciones que operan sobre estos datos*. A los elementos de un objeto se les conoce como *miembros*. Las funciones que operan sobre ellos se les denominan *métodos*; y los datos se denominan *miembros datos*. Los objetos de un programa se comunican mediante *el paso o envío de mensajes*

¿Qué son objetos en POO?

La respuesta es cualquier entidad del mundo real que se pueda imaginar.

• Objetos físicos Automóviles. Aviones en una terminal aérea Componentes de un circuito Animales mamíferos • Elementos de interfaces gráficas de usuario Ventanas. Iconos Menús Objetos gráficos Ratón Teclado.	• Estructuras de datos Arrays Pilas Colas Arboles binarios • Tipos de datos definidos por el usuario. Números complejos. Hora del día Puntos de un plano

Ejemplo:

Supongamos que se dispone de un objeto tal como una ventana en la pantalla, y se desea definir las 4 operaciones requeridas para mover el objeto ventana en la pantalla. (Mover derecha, izquierda, abajo, arriba). Utilizando métodos de programación convencional se puede escribir un procedimiento independiente para cada operación como un mecanismo idóneo para dividir operaciones. Pero desgraciadamente los datos son independientes para de cada uno de los procedimientos.

Examinemos el método de definir el objeto visual utilizando del método de orientación a objetos; así combinamos datos y procedimientos en un único paquete. Primero creamos un objeto llamado ventana, esta ventana contienen los procedimientos que definen como se mueve una ventana, dependiendo de la orden de entrada, será el procedimiento a ejecutar.

Los objetos soportan una serie de características específicas de los mismos:

- Se agrupan en tipos denominados *clases*

- Contienen *datos internos* que definen su estado actual.
- Soportan *Ocultación* de datos.
- Pueden *heredar* propiedades de otros objetos.
- Pueden *comunicarse* con otros objetos enviando o pasando *mensajes*.
- Tienen *métodos* que definen su *comportamiento.*

Los datos y las funciones se encapsulan en una única entidad. Los datos están ocultos y solo mediante las funciones miembro es posible acceder a ellos.

Un método gráfico para representar los objetos se representa en la siguiente figura.

Diagrama de un objeto

El cuadro (perímetro del mismo) representa el límite o frontera entre el interior y el exterior de un objeto, en el interior del objeto se encuentran las variables locales (campos miembro) y las funciones, un campo se representa por un cuadro rectangular, una función por un hexágono. Los campos y funciones se rotulan con sus nombres. Todos los campos miembro y las funciones que están completamente en el interior del cuadro objeto son ocultos desde el interior, lo que significa que están encapsulados. Los campos o funciones se extienden fuera del cuadro son accesibles desde el exterior y actúan de interfaz. El acceso a las características miembro (campos y funciones) es posible a través de la interfaz del objeto.

Un programa consta de un número de objetos que se comunican entre si , mediante métodos que son a su vez, llamadas funciones miembro del objeto invocado.

CLASES

En POO se suele decir que los objetos son miembros de una clase, una clase es un tipo definido por el usuario que determina las estructuras de datos y las operaciones asociadas con ese tipo. Las clases son como plantillas o modelos que describen como se construyen cierto tipo de objetos. Cada vez que se construye un objeto de una clase se crea una instancia de esa clase.

Una clase es una colección de objetos similares y un objeto es una instancia de una definición de una clase. una clase es simplemente un modelo que se utiliza para describir uno o más objetos del mismo tipo

La comunicación del objeto se realiza mediante el paso de mensajes. El envío de un mensaje a una instancia de una clase produce la ejecución de un método. El paso de mensajes es la invocación o llamada de una función miembro de un objeto.

MENSAJES: ACTIVACIÓN DE OBJETOS.

Los objetos pueden ser activados mediante la recepción de mensajes. Un *mensaje* es simplemente la petición de un objeto a otro objeto para que este se comporte de una manera determinada, ejecutando uno de sus métodos. La técnica se conoce como *paso de mensajes* y se representa de la siguiente manera:

Estructuralmente, un mensaje consta de 3 partes: la identidad del objeto receptor, el método (función miembro) del receptor cuya ejecución se ha solicitado y cualquier otra información adicional que el receptor pueda necesitar para ejecutar el método requerido.

Los mensajes juegan un papel vital en POO; sin ellos, los objetos que se definan no se podrán comunicar con otros objetos. Desde un punto de vista convencional, el paso de mensajes no es mas que sinónimo de llamada a una función.

HERENCIA

Una característica muy importante de los objetos y las clases es la herencia. La herencia es la propiedad que permite a los objetos construirse a partir de otros objetos. Las clases se dividen en subclases.

El principio de este tipo de división es que cada subclase comparte características comunes con la clase de la que se deriva. Además de las características compartidas con otros miembros de la clase, cada subclase tiene sus propias características particulares.

La idea de herencia se muestra en la siguiente figura:

Obsérvese que en la figura que las características A y B que pertenecen a la clase base, también son comunes a todas las clases derivadas, y a su vez estas clases derivadas tienen sus propias características.

De modo similar, en POO, una clase se puede dividir en subclases. Las clases que se definen a partir de la clase base, compartiendo sus características y añadiendo otras nuevas, se denominan clases derivadas.

Las clases derivadas pueden heredar código y datos de su clase base añadiendo su propio código especial y datos a la misma.

La herencia impone una relación jerárquica entre clases en la cual una clase hija hereda de su clase padre. Si una clase solo puede recibir características de otra clase base, la herencia se denomina herencia simple.

Si una clase recibe propiedades de mas de una clase base, la herencia se denomina herencia múltiple.

POLIMORFISMO.

Significa la cualidad de tener mas de una forma. El polimorfismo se refiere al hecho de que una misma operación puede tener diferentes comportamientos en diferentes objetos. Diferentes objetos reaccionan la mismo mensaje de modo diferente. Supongamos un número de figuras geométricas que responden todas al mensaje dibujar. Cada objeto reacciona a este mensaje visualizando su figura en una pantalla de visualización. Obviamente, el mecanismo real para visualizar los objetos difiere de una figura a otra, pero todas las figuras realizan esta misma tarea en respuesta al mismo mensaje.

REUTILIZACIÓN

Una vez que una clase se ha escrito, creado, depurado y utilizado, se puede difundir entre otros programadores para que puedan utilizarla en sus propios programas. Esta propiedad se denomina reutilización o reutilizabilidad. Es similar a la forma en que una biblioteca de funciones de un lenguaje procedimental se puede incorporar en programas diferentes.

En POO, el concepto de herencia proporciona una importante extensión a la idea de la reutilizabilidad. Un programador puede tomar una clase existente, y sin modificarla, añadir características y posibilidades adicionales a la misma. Estas operaciones se realizan por derivación de una clase nueva a partir de una clase existente. La nueva clase heredará las propiedades de la antigua, pero es posible añadir nuevas características propias.

B

R

L

U

A

D

C

F

Caliente el aceite

Engrase el sarten

Consiga el aceite

Consiga un huevo

F

G

P

A

F

B

N

G

I

M----------

A

D

C

F

Ponga el huevo en el sarten

Espere hasta que este hecho

Retire el huevo

A

D

C

F

B

Mover derecha

Mover izquierda

Mover abajo

Mover arriba

Objeto ventana implementado mediante procedimientos El objeto ventana

Objeto

Miembros Dato

Miembros función

interior

I

N

T

E

R

F

A

Z

exterior

Variables

(funciones miembro)

Métodos

(Miembros dato)

Mensaje TEST

Objeto B

(Objeto receptor)

Objeto A

(Objeto Emisor)

Resultados (de la ejecución del método Test)

Clase base

Característica B

Característica A

Característica A

Característica A

Carac. D

Carac. B

Carac. A

Característica B

Característica B

Carac. F

Carac. C

Clases derivadas

Carac. E

Paquete de aceptación de llamada

2

3

4

1

Pqt 1

Pqt 1

Pqt 1

Pqt 2

Pqt 2

Pqt 2

Pqt 3

Pqt 3

Pqt 3

paquete de petición de llamada

Paquete de confirmacion

Datos de usuario

Retardo de programacion

Retardo de procesamiento

Señal de petición de llamada

Señal de aceptación de llamada

Señal de confirmación

enlace

enlace

enlace

1

2

3

4

Nodos:

2

3

4

1

Pqt 1

Pqt 1

Pqt 1

Pqt 2

Pqt 2

Pqt 2

Pqt 3

Pqt 3

Pqt 3

Conmutación de paquetes datagramas

Conmutación de paquetes mediante circuitos virtuales

Conmutación de circuitos

Red de comunicación de paquetes

A

1.3

1.2

1.1

2.3

2.2

2.1

B

C

1.3

1.2

1.1

2.3

2.1

2.2

C.3

C.1

C.2

B.2

B.3

B.1

C

B

C.1

C.2

C.3

B.1

B.2

B.3

A

Red de comunicación de paquetes

Circuito virtual externo. Se establece una conexión lógica entre dos estaciones.

Los paquetes se marcan con un número de circuito virtual y un número de secuencia. Los paquetes se reciben en orden.

Datagrama externo. Cada paquete se transmite independientemente. Los paquetes se marcan con una dirección de destino y pueden recibirse en forma desordenada

1

A

2

3

5

4

6

C

B

VC#1

VC#2

Circuito virtual interno. Se define y se marca una ruta para los paquetes entre dos estaciones todos los paquetes de dicho circuito virtual siguen la misma ruta y se reciben en orden secuencial

Datagrama interno: la red trata de forma independiente cada paquete. Los paquetes se marcan con una dirección destino y pueden recibirse desordenadamente en el nodo destino

6

B

C

5

3

2

1

4

A

2

1

1

2

3

3

Petición de llamada

Llamada entrante

Llamada aceptada

Comunicación establecida

Datos R=0, S=0

Datos R=0, S=1

Datos R=0, S=2

Datos R=2, S=0

Datos R=1, S=3

Datos R=1, S=4

Preparado para recibir R=3

Datos R=5, S= 1

12

Petición de liberación

12

Confirmación de Liberación

12

Datos R=0, S=0

Datos R=0, S=1

Datos R=2, S=0

Datos R=0, S=2

Preparado para recibir R=3

Datos R=1, S=3

Datos R=1, S=4

Datos R=5, S=1

Indicación de liberación

confirmación de liberación

Sistema de usuario A

Interfaz

Usuario de red

A inicia una llamada virtual a B

Sistema de usuario B

Interfaz

Usuario de red

B acepta la llamada

Cuando se informa a A que se ha establecido la comunicación, este puede comenzar a transmitir paquetes de datos

Los paquetes se entregan en orden

B no dispone de paquetes de datos con los que conforman el paquete s=2, de mono que envía un paquete de control

A comienza a liberar el circuito virtual