

PILAS Y COLAS Y SU IMPLEMENTACIÓN EN OBJECT PASCAL.

PILES AND QUEUES AND THEIR IMPLEMENTATION IN OBJECT PASCAL.

Autores: MsC. Marcos Antonio León Fonseca.
MsC. Noralys Muñiz Maldonado.
MsC. Juan Antonio Fonseca Hernández

Resumen:

En este artículo el autor describe como se implementan las pilas y las colas en el lenguaje de programación Object Pascal. El mismo es un resultado del proyecto de investigación "Una alternativa de superación, la educación a distancia por correo electrónico" que se desarrolló en el departamento de Informática por un equipo de investigadores perteneciente al mismo y del cual formó parte el autor.

Summary:

In this article the author describes how to implement Piles and Queues in the programming language "Object Pascal". It is the result of an investigation project "A supuration alternative, distance education by Email, developed by a group of researches of the Informatics department where the author belongs.

Introducción:

Cuando se utiliza un lenguaje de programación para la solución de un problema, se debe tener en cuenta cómo representar y operar sobre los datos de ese problema de acuerdo a las estructuras de datos del lenguaje que se va a utilizar. Para poder realizar esto no solo es necesario conocer los tipos de datos predefinidos en el lenguaje sino también aquellos que se pueden construir a partir de esos datos simples.

Dentro de los diferentes tipos de datos que se pueden instrumentar en el lenguaje Object pascal se encuentran las listas, término que hace referencia a una colección lineal de elementos.

Las listas permiten insertar y borrar elementos en cualquier lugar de la misma, al principio, en el medio o al final; pero hay algunas situaciones frecuentes en programación en las que es necesario restringir las inserciones y borrados de los elementos solo al principio o al final. Dos de las estructuras de datos que son útiles en tales situaciones son las pilas y las colas.

Una pila es una estructura lineal en la que los elementos pueden ser añadidos o eliminados solo por el final y una cola es una lista lineal en la que los elementos solo pueden ser añadidos por un extremo y eliminados por el otro.

Este trabajo tiene como propósito describir cómo implementar estas estructuras en el lenguaje de programación Object Pascal.

Desarrollo:

Las pilas se pueden representar en la memoria de la computadora de varias formas, una de ellas es a través de una lista enlazada en la cual cada componente apunta a su predecesor excepto el primer componente que apuntará a Nil para indicar el final de la lista.

Esta forma de representación puede ser caracterizada por:

Type

```
Pila = ^Lista;  
Lista = Record  
    Elemento : Tipo de dato;  
    Predecesor : Pila;  
end;
```

donde Pila es la variable puntero, Lista la variable referenciada y Tipo de dato es el tipo de dato que corresponde a la información que se desea almacenar.

La pila se formará creando cada nuevo componente y enlazándolo después a su predecesor.

Las operaciones de inserción y extracción de elementos en la pila pueden definirse entonces por:

```
Procedure Insertar (Var Ultimo : Pila; Informacion : Tipo de dato);
```

```
Var
```

```
    Nuevo : Pila;
```

```
begin
```

```
    New (Nuevo);
```

```
    Nuevo.Elemento := Informacion;
```

```
    Nuevo.Predecesor := Ultimo;
```

```
    Ultimo := Nuevo;
```

```
end;
```

```
Procedure Extraer (Var Ultimo : Pila; Var Informacion : Tipo de dato);
```

```
begin
```

```
    If Ultimo <> Nil Then
```

```
        begin
```

```
            Informacion := Ultimo.Elemento;
```

```
            Ultimo := Ultimo.Predecesor;
```

```
        end
```

```
end;
```

Antes de usarlos se debe declarar dos variables: una de tipo Pila y la otra de tipo Tipo de dato para almacenar en ellas los valores que devuelven los

procedimientos en las variables Ultimo e Información. Inicialice la variable de tipo Pila a Nil.

Esta forma de representación de las pilas mediante lista enlazadas es muy útil cuando es necesario indicar el orden de procesamiento de los datos y no se conoce la cantidad de elementos a procesar ya que a través de ella es posible crear y destruir variables dinámicamente.

Para las colas se utilizan las mismas representaciones que para las pilas, solamente se añade un apuntador al último elemento.

Esta forma de representación puede ser caracterizada por:

Type

```
Cola = ^Lista;  
Lista = Record  
    Elemento : Tipo de dato;  
    Proximo : Cola;  
end;
```

Punteros = Record

```
Inicio : Cola;  
Fin : Cola;  
end;
```

En la formación de la cola cada nuevo componente que se cree será el último por lo que deberá apuntar a Nil y el que era el último hasta ese momento apuntará al nuevo componente pasando a ser el penúltimo.

La operación de inserción se puede definir por:

Procedure Insertar (Var Ultimo : Punteros; Informacion : Tipo de dato);

Var

```
    Nuevo : Cola;  
begin  
    New (Nuevo);  
    Nuevo.Elemento := Informacion;  
    Nuevo.Proximo := Nil;  
    If Ultimo.Fin = Nil then  
    begin  
        Ultimo.Inicio := Nuevo;  
        Ultimo.Fin := Nuevo;  
    end  
    else  
    begin  
        Ultimo.Fin.Proximo := Nuevo;  
        Ultimo.Fin := Nuevo;  
    end;  
end;
```

y la de extracción por:

```
Procedure Extraer (Var Ultimo : Punteros; Var Información : Tipo de dato);  
begin  
  If Ultimo.Inicio <> Nil Then  
    bgin  
      Información := Ultimo.Inicio.Elemento;  
      Ultimo.Inicio := Ultimo.Inicio.Proximo;  
    ed;  
  end;
```

Antes de usarlos se debe declarar una variable de tipo Punteros y otra de tipo Tipo de dato e inicializar los punteros Inicio y Fin de la variable Punteros a Nil.

Al igual que en el caso de las pilas, esta forma de representación de las colas es muy eficiente en el caso de que sea necesario indicar el orden de procesamiento de los datos y no se conozca la cantidad de elementos a procesar.

Conclusiones:

La implementación de pilas y colas mediante listas enlazadas posibilita la representación eficiente de los datos en situaciones donde es necesario indicar el orden de procesamiento de los mismos y no es posible prever la cantidad de elementos a procesar por cuanto este tipo de representación permite crear y destruir variables dinámicamente.

Bibliografía:

Katib mora, miguel. Lenguajes de programación y Técnicas de compilación/ Miguel Katib Mora.-- Ciudad de la Habana: Editorial Pueblo y Educación, 1986.- - 421 p.

GOTTFRIED, BYRON S. Programación en Pascal/ Byron S Gottfried.—Ciudad de la Habana: Editorial Pueblo y Educación, 1989.—397 p.

LIPSCHUTZ, SEYMOUR. Estructura de datos/ Syemour Lipschutz.—Ciudad de la Habana: Edición Revolucionaria, 1989.—390 p.