

Comenzado el	jueves, 8 de octubre de 2020, 08:10
Estado	Finalizado
Finalizado en	jueves, 8 de octubre de 2020, 09:58
Tiempo empleado	1 hora 47 minutos
Calificación	6,00 de 10,00 (60%)

Pregunta 1

Finalizado

Puntúa 2,00 sobre 2,50

- A) En un lenguaje de programación basado en un esquema de ejecución en pila, nombre los elementos que debe contener un registro de activación, a parte de los datos locales. Explique para qué sirve cada uno.
- B) Diga qué diferencias existen entre un lenguaje que está basado en un esquema en pila y uno que está basado en un esquema dinámico.

En un esquema de ejecución basado en pila el espacio es predecible, el espacio se deduce del código. Este esquema sirve para programas más potentes cuyos requerimientos de memoria no pueden ser calculados en traducción. La memoria a utilizarse es predecible y sigue la disciplina last-in-first-out. Las variables se alocan automáticamente y se desalocan cuando el alcance se termina. Se utiliza una estructura de pila para modelizar la estructura

Además de los datos locales son almacenados tantos registros de activación de esa rutina como llamados se hagan, otros elementos son los puntos de retorno, que se tiene que guardar en el registro de activación de la unidad llamada, y también el ambiente de referencia, que puede ser local o no local, cuando son no locales están ligadas a objetos almacenados en los registros de activación de otras unidades.

Con respecto a la diferencia entre un esquema de dinámico es que en este caso, el espacio es impredecible, los datos son alocados dinámicamente sólo cuando se los necesita durante la ejecución, no puede ser modelizado por una pila, y en este caso el programador puede crear objetos de datos en cualquier punto durante la ejecución del programa, y por último en el esquema dinámico los datos se alocan en la zona de memoria heap. Un ejemplo de esto son los punteros.

Comentario:

Existe la posibilidad de implementar recursión y rutinas anidadas.

Pregunta 2

Finalizado

Puntúa 2,50 sobre 2,50

```

1 #include <stdio.h>
2
3 int main()
4 {
5     int a=8;
6     if(a!=0){
7         if(a!=2){
8             if(a!=3){
9                 if(a==4){
10                    print("hola");
11                }
12            }
13        }
14        else{
15            if(a!=5){
16                print("chau");
17            }
18        }
19    }
20 }
```

Dado este código en C.

- a) Indique qué estructura de control es utilizada en este código y qué particularidades presenta conceptualmente. Mencione qué resultado devuelve el código presentado. Justifique la respuesta
- b) Presente alternativas de solución al código planteado utilizando otras estructuras de control mencionando las características más importantes tanto conceptualmente como en Lenguaje C en particular.

a) Se está utilizado una estructura de control de selección, esta estructura de control permite que el programador pueda expresar una elección entre un cierto número posible de sentencias alternativas, esta estructura de control es una de las que más ha evolucionado. En un principio los if, por ejemplo en Fortran se ejecutaba solo con una condición verdadera.

En este caso vemos una estructura de if anidados, y else en el caso de que no corte por una condición tome otro camino, para llegar a imprimir un valor, este tipo de estructuras hace que la legibilidad del código sea más complicada, y en cuanto a la mantenibilidad también lo es.

El programa imprimir "hola" si el valor de es distinto de 0, 2, 3 y es igual a 4, en el caso de que sea distinto de 0 pero igual a 2, para imprimir "chau" debe ser distinto de 5. Con el valor 8, entra al primer if de la línea 6, porque 8 es distinto de 0, luego entra al siguiente if de la línea 7 porque 8 es diferente de 2, luego sigue con el if de la línea 8 porque 8 es diferente de 3, pero cuando se encuentra con el if de la línea 10, como 8 no es igual a 4, el programa termina y no imprime ningún valor.

b) Como ya dijimos el programa imprimir "hola" si el valor de es distinto de 0, 2, 3 y es igual a 4, en el caso de que sea distinto de 0 pero igual a 2, para imprimir "chau" debe ser distinto de 5.

En este caso una alternativa que podría llegar a utilizarse a este conjunto de condiciones anidadas, es una cláusula de switch, por lo tanto en este caso, cada opción que se puede tener es etiquetada por valores constantes, entonces cuando coincida con una de las opciones del switch se ejecutarían las sentencias asociadas, que en este caso sería imprimir "hola" o "chau" en el caso que sea necesario. En el caso de C si al final del bloque de ejecución de una opción específica no se coloca la sentencia break, se sigue verificando con el resto de las opciones del switch, hay que tener en cuenta esto ya que el programa podría no realizar lo esperado. En C en el switch existe también una cláusula default, que se puede utilizar en el caso de que no matchee con ninguna de las opciones y se puede ejecutar el bloque de instrucciones que presente la opción default.

Comentario:

Pregunta 3

Finalizado

Puntúa 0,50 sobre 2,50

Se busca desarrollar un programa para la búsqueda de relaciones interpersonales de aquellos padecieron covid-19. En este sentido, se cuenta con diferentes hipótesis sobre el comportamiento interpersonal de acuerdo a los diferentes medios de transporte con el que se mueve la persona, factores estadísticos que permiten suponer determinadas relaciones (tuvo contacto cercano/físico con el padre, con la madre, etc) y supuestos estandarizados de comportamiento.

1- Indique qué paradigma utilizaría para una desarrollar una solución justificando la respuesta.

2- Mencione qué cuestiones debería tener en cuenta si quisiera plantear la misma solución en otro paradigma elegido por ud. Justifique la respuesta.

1) Utilizaría el paradigma orientado a objetos, dado que el dominio del problema hace referencia a las relaciones que existen entre las personas, también se habla de un comportamiento, esto me hace referencia a que se podrían modelar objetos, que van a contar con un estado interno y un comportamiento, y que estos objetos se van a comunicar mediante mensajes, podemos usar clases para modelar las estructuras de datos y las operaciones asociadas con este tipo.

2. Supongamos que también se podría utilizar un paradigma funcional, se debería tener en cuenta el conjunto de datos que se va a utilizar, las funciones que se van a utilizar para poder resolver el problema.

Comentario:

El paradigma más acertado sería el. Lógico ya que por su naturaleza esta preparado para realizar predicciones y análisis de comportamiento.

Pregunta 4

Finalizado

Puntúa 1,00 sobre 2,50

```
main.c
1  #include <stdio.h>
2  int y=4;
3
4  void A()
5  { static int x=1;
6    x = x + y;
7    printf ("El valor de x es: %d", x);
8  }
9
10 int main()
11 {
12     int z;
13     printf("Evaluación teórica");
14     for (z=1 ; z < 3; z++ )
15     { A(); }
16     return 0;
17 }
```

Sea el siguiente código escrito en lenguaje C:

a) Diga qué tipo de variables observa en el código: 1- de acuerdo al tiempo de vida y 2- de acuerdo al alcance. Cada tipo de variable que nombre debe estar acompañada de una descripción de sus características.

b) Si la sentencia fuera `const int x=1;` ¿sería lo mismo tanto conceptualmente como en el resultado que arroje el programa?. Justifique la respuesta en forma completa, basándose en conceptos vistos.

a) Temos las siguiente variables:

`int y=4`, en este caso el tiempo de vida de la variable (l-valor) va a ser desde el momento que se aloque el programa hasta que termine. Con respecto al alcance de variable en este caso podríamos decir que es estático, otra opción es decir que es una variable global, por lo tanto va a poder ser referenciada en cualquier punto del programa.

`void a()`, en este caso podemos definirlo como automático, el mismo será cargado en memoria y una vez que se termine de usar será desalocado.

Dentro de `A()`, tenemos la variable `static int x=1`, en este caso la variable va a ser visible en el scope de `A` durante su ejecución, en cuanto al alcance es una variable local de `A`, de misma manera que esta variable vivirá en el momento que se ejecute `A`, pero luego se deslocalará cuando termine el flujo de ejecución.

En `main` tenemos la variable `int`, en este caso es una variable que tiene un alcance global, dado que es una referencia creada en el programa principal, el alcance será estático ya que se carga con el programa, y se desaloca cuando este finaliza.

b) Al declararla como constante estamos diciendo que tendrá el mismo valor a lo largo de todo el programa, en este caso se está creando una variable `static`, el uso de `static` dentro de la función permite que el valor de la variable persista entre las llamadas, por lo tanto conceptualmente sería lo mismo.

Comentario:

La variable estática `x` en `A` permanece en el. Ambiente porque se liga en compilación. Su tiempo de vida es más que el programa.

En el caso de. `Main` la variable `int Z` es local a `main` y en todo caso(dependiendo del lenguaje) su alcance podría ser dinámico.

El. Concepto de estático y global son diferentes. El. Primero hace referencia al tiempo de vida mientras que el segundo al alcance. Si pasa a ser constante dejaría de existir cuando termina `A`.