

PROGRAMACIÓN ESTRUCTURADA

UNIDAD 2

LA ESTRUCTURA SECUENCIAL

CLASE 3:

ESTRUCTURA SECUENCIAL

PRESENTACIÓN

En esta unidad presentaremos las estructuras básicas del paradigma lógico de la programación estructurada. Este tema implica conocer todo el marco teórico estudiado en las unidades anteriores por lo que le solicitamos que vuelva a las mismas tantas veces como sea necesario para favorecer su comprensión.

Recuerde que se aproxima la evaluación parcial con la Propuesta para la Integración del Módulo I, por lo que le sugerimos que practique en forma intensiva y consiente.

Las estructuras básicas son tres:

La secuencial,

La condicional

La cíclica o repetitiva

En esta clase presentaremos la primera estructura que es la secuencial en la cual las distintas operaciones se realizan en una secuencia y con un orden lógico necesario para lograr un fin.

En la siguiente clase veremos la segunda estructura que es la condicional, que permite determinar qué camino elegir ante un suceso o hecho que exija una toma de decisión. Dentro de esta estructura veremos dos opciones:

- Una es el IF que permite bifurcar dentro de dos opciones - si/no, verdadero/falso - dependiendo de una condición. Esta operación puede realizarse en forma secuencial entre ellas o anidar un IF dentro de otro IF. Veremos que muchas veces es lo mismo realizar una u otra relación de IF, mientras que en otros casos sólo es posible una de ellas.
- La otra opción dentro de esta estructura es la conocida como selección, SELECT, SWITCH dependiendo del lenguaje a utilizar. Es una alternativa para escribir en forma prolija y rápida los distintos caminos a seguir dependiendo del valor que tome una variable o dato a evaluar. A diferencia del IF en la selección tendremos N opciones a seguir dependiendo del valor que tome el dato evaluado.

Posteriormente en las ultimas clases antes de la primera evaluación parcial veremos la tercera estructura que es la estructura de repetición o cíclica. Esta permite repetir n veces una misma operación, que se realizará bajo dos condiciones posibles: por ciclos exactos o por ciclos no exactos.

- Por ciclos exactos –en inglés FOR- es posible si se conoce de antemano cuántas veces se debe repetir la operación.

- La segunda condición se da cuando no es posible anticipar cuántas veces se repetirá una operación, ya que depende de un dato externo que ingresa o del valor que tome una variable interna del programa. Estos ciclos son conocidos como WHILE y DO- WHILE, según sea el caso o el gusto del programador.

Con el fin de mejor comprensión de los temas, le proponemos la realización gradual de los Trabajos Prácticos que abonan a la resolución del Trabajo Práctico Requerido que plantearemos en esta unidad.

La evaluación de este último nos brindará información referida al nivel de conocimientos alcanzado por usted, tanto en el aspecto lógico como de conocimiento del lenguaje C.

Tenga en cuenta que tanto en la Evaluación Parcial que se aproxima como en la evaluación final de la asignatura deberá resolver situaciones en las que tendrá que aplicar su lógica y utilizar los recursos que le brinda el paradigma estructurado y el lenguaje C.

Anticípese y realice una ejercitación regular y permanente. Controle sus producciones; evalúe sus aprendizajes, dificultades, limitaciones y obstáculos y consulte a su tutor/a.

Por todo lo expresado hasta aquí es que esperamos que usted, a través del estudio de esta unidad, adquiera capacidad para:

- Construir soluciones eficientes a distintos problemas.
- Seleccionar las herramientas más convenientes para cada uno de ellos.
- Realizar la depuración de errores.
- Realizar diseños que consideren el criterio de dar al usuario una solución amigable y coherente con sus necesidades.

BIENVENIDOS A LA CLASE NÚMERO 3

En esta clase comenzaremos de lleno a ver la primera estructura del paradigma estructurado, la estructura secuencial

Junto a esta estructura veremos

- Definición de una variable
- El uso de asignación de valores a una variable
- Operaciones básicas con una variable

Como indica la definición de un programa ***“secuencia finita de instrucciones en un orden lógico que nos permite la resolución de un problema en particular”***, esta estructura es simplemente eso, donde inicialmente veremos operaciones simples, primitivas, y luego veremos que podremos ir haciendo secuencias secuenciales entre otras estructuras

A continuación usted deberá:

1. Leer la teoría de una estructura secuencial y analizar la resolución de un ejercicio con esta estructura
2. Leer el libro de Johannes Aguilar donde trata este tema
3. Ejercitación del tema
4. Ante dudas ver el video






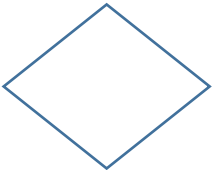
DIAGRAMACION RESPETANDO EL PARADIGMA ESTRUCTURADO

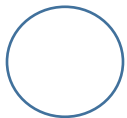

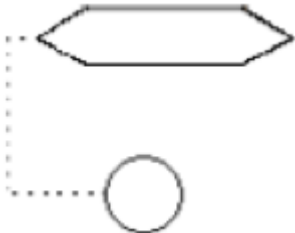
Esta es una forma más de representar la lógica de programación.

Un proceso es una secuencia de instrucciones que ocupan una cantidad de recursos del computador y permiten la solución de un problema, ya sea en función de un solo proceso o de varios subprocesos que al combinarse, correcta y lógicamente, generan la solución deseada.

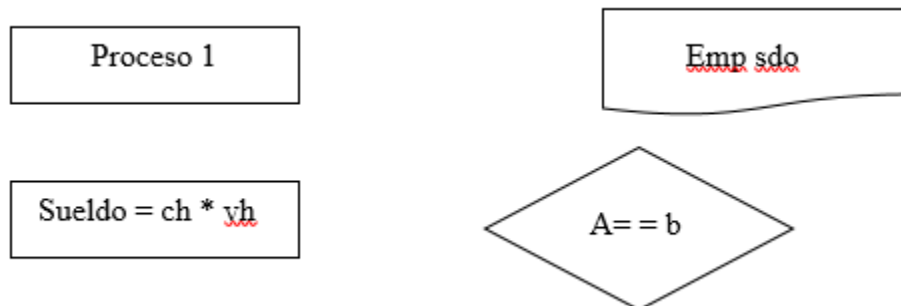
La diagramación que utilizaremos consiste en dibujar a todos sus elementos, que se encuentran ordenados en forma secuencial de arriba hacia abajo.

Simbología definida para el desarrollo del diagrama de flujo bajo el estándar ANSI.

SÍMBOLO	SIGNIFICADO
	Inicio y Fin del Diagrama
	Entrada de Datos. (Se homogeniza el uso de este símbolo para entradas desde cualquier origen incluyendo el teclado. En caso de ser necesario identificar el origen se aclara dentro del símbolo)
	Proceso. Indica una acción o proceso. (p.e . cambio de valor de variables, asignaciones y operaciones matemáticas)
	Llamada a Subrutina
	Salida. (Se homogeniza el uso de este símbolo para salidas hacia cualquier destino Pantalla o Impresora. En caso de ser necesario identificar el destino se aclara dentro del símbolo.)
	Decisión. Compara dos datos y dependiendo de su resultado lógico (verdadero o falso) se toma la decisión de seguir por un camino del diagrama u otro.

	Conector (in page). Enlaza dos partes del programa de la misma página. Si el diagrama tuviera varios, se distinguen por el número que llevan en su interior.
	Conector (out page). Indica el enlace de dos partes de un programa en distintas páginas.
EL SIGUIENTE SÍMBOLO SOLE SE APLICA A LOS DIAGRAMAS DENOMINADOS DE ALTA ABSTRACCIÓN PARA SIMPLIFICAR LA COMPRENSIÓN DEL USO DE ESTRUCTURAS ITERATIVAS	
	Iteración. Indica que una instrucción o grupo de instrucciones deben ejecutarse varias veces.

En todos los casos dentro del símbolo se escribe la instrucción, el procedimiento que se llama, etc.



Recordemos lo que dice el teorema de Bohm y Jacopini

El teorema de Böhm y Jacopini establece, que un **programa propio** puede ser escrito utilizando únicamente tres tipos de estructuras de control, las cuales son: estructuras secuenciales, estructuras de selección y estructuras de repetición.

Para que la programación sea estructurada, los programas han de ser propios. Un programa se define como **propio** si cumple las siguientes condiciones:

- Si tiene un solo punto de entrada y un solo punto de salida
- Si todas las sentencias del algoritmo son alcanzables, esto es, existe al menos un camino que va desde el inicio hasta el fin del algoritmo
- Si no posee ciclos infinitos

De este **teorema** se deduce que, si los algoritmos se diseñan empleando exclusivamente dichas estructuras de control, los algoritmos, y por consecuencia los programas derivados de ellos, serán propios.

- Tipos de Estructuras

Dentro de la programación estructurada reconocemos tres estructuras básicas

Estructuras secuenciales
Estructuras condicionales
Estructuras iterativas o de repetición

Forma de dibujar las distintas operaciones

Secuenciales

Dado el valor de la hora y la cantidad de horas trabajadas por un empleado, calcular su sueldo

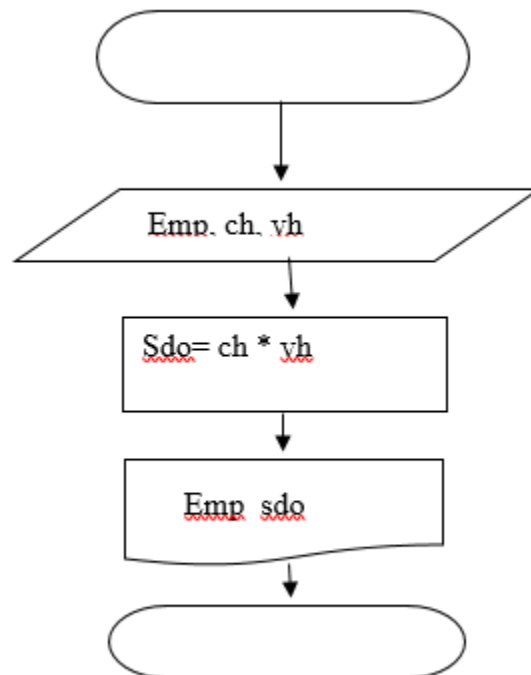
Datos : valor de la hora..... vh

cantidad de horas trabajadas..... ch

nro de empleado..... emp

Resultado: sueldo..... sdo

Proceso: $sdo = ch * vh$



En seudo código sería:

Comienzo

Ingresar "el nro de empleado"

Leer, emp

Ingresar "la cantidad de horas"

Leer, ch

Ingresar "el valor de la hora"

Leer vh

Sdo = ch * vh

Imprimir "el empleado, emp, cobra, sdo, pesos"

fin

En lenguaje C , DEV C++ , sería

```

1  #include <stdio.h>
2  #include <conio.h>
3  #include<iostream>
4  #include<stdlib.h>
5
6  int main()
7  {
8      int emp,ch,vh,sdo;
9      system("cls");
10     printf("ingrese el nro de empleado ");
11     scanf("%d",&emp);
12     printf("ingrese la cantidad de horas ");
13     scanf("%d",&ch);
14     printf("ingrese el valor de la hora ");
15     scanf("%d",&vh);
16     sdo=ch*vh;
17     printf("el empleado %d, cobra %d pesos \n",emp,sdo);
18     system("pause");
19 }
20

```

//*****

CON FUNCIONES

En C de borland

```

1  #include <stdio.h>
2  #include <conio.h>
3  #include<stdlib.h>
4
5  /////////////////////////////////////////////////// DECLARACIONES O PROTOTIPOS DE FUNCIONES
6  void carga_datos(int*, int*, int*);
7  int calculo (int, int);
8  void informe (int, int);
9  void FIN (void);
10
11 int main()
12 {
13     int emp,ch,vh,sdo;
14     system ("cls");
15     //////////////////////////////////// LLAMADAS A FUNCIONES
16     carga_datos(&emp,&ch,&vh);
17     sdo=calculo(ch,vh);
18     informe(emp,sdo);      // TAMBIEN SE PUEDE REEMPLAZAR LAS DOS ULTIMAS
19     FIN();                 // LLAMADAS POR -----> informe (emp, calculo(ch,vh));
20 }
21
22
23 //////////////////////////////////// DEFINICIONES DE FUNCIONES
24
25 void carga_datos(int *x, int *y, int *z)
26 {
27     printf("ingrese el nro de empleado ");
28     scanf("%d",x);
29     printf("ingrese la cantidad de horas ");
30     scanf("%d",y);
31     printf("ingrese el valor de la hora ");
32     scanf("%d",z);
33 }
34
35 ////////////////////////////////////
36 int calculo (int x, int y)
37 {
38     return (x *y);
39 }
40
41 ////////////////////////////////////
42 void informe (int x, int y)
43 {
44     printf("el empleado %d, cobra %d pesos",x,y);
45 }
46
47 ////////////////////////////////////
48 void FIN(void)
49 {
50     printf("\n\n TIPEE UNA TECLA PARA FINALIZAR  ");
51     getch();
52 }

```

EJERCICIOS DE VARIABLES Y ASIGNACIONES

1. Ingresar dos valores enteros, sumarlos e imprimir esta suma
2. Ingresar tres valores, imprimir la suma total, sólo sabe sumar de a dos.operandos por vez
3. Ingresar tres valores, sumarlos e imprimir esa suma, se puede sumar de a varios operandos.en la misma operacion
4. Ingresar los lados de un triángulo calcular su perímetro e imprimirlo
5. Ingresar dos lados de un triángulo rectángulo y calcular, la hipotenusa, el perímetro, la superficie.imprima los resultados solicitados
6. Ingresar los lados de un rectángulo y calcular su diagonal principal, superficie y perímetro.imprima los resultados solicitados
7. Ingresar dos valores, calcular su suma, su producto y la resta del 1ro menos el 2do valor ingresado, imprimir los resultados
8. Ingresar el valor de la hora y el tiempo trabajado por un operario, calcular su sueldo e imprimirlo
9. Ingresar el tiempo trabajado por un operario y considerando que el valor de la hora es de 10 pesos, calcular su sueld e imprimirlo
10. Una concesionaria de autos desea liquidar el sueldo a cada vendedor pagando \$ 500 por mes más un plus del 10 % del precio sobre cada vehículo vendido y un valor constante de 50 pesos por cada uno de ellos, se ingresa el valor del vehiculo y cuantos vehiculos dee ese tipo vendio, calcular su sueldo e imprimirlo