

Presentación

Asignatura:

Programación Para
Mecatrónicos

Nombre:

Franjelis Luciano

Matrícula:

2023-1509

Tema:

Cap. 1 del libro
Fundamento de
Programación.

Profesor:

Carlos Antonio
Pichardo Viuque

Cap. 1: del Libro Fundamento de Programación, (Algoritmos, diagrama de flujo y Programas en C):

NAME: Franzulis Juliano PAGES: 1 SPEAKER/CLASS: Programación de DATE-TIME: 23/03/2024

Title: Cap 1: Algoritmos, diagrama de flujo y Problemas en C

Keyword
algoritmo
Proceso
Pasos
Resultado

Topic: Cap 1.1 Problemas y algoritmos

Notes: Un algoritmo es una serie de pasos usados para conseguir un resultado específico. Nuestra vida está llena de algoritmos solo que están tan automatizados en hacer las diferentes cosas y no contemplamos los pasos para realizarlos y entender una PC, celular, etc.

Consta de 3 secciones
Problema

Datos de entrada

Procesamiento de los datos

Impresión de los resultados

Questions

¿Qué es un algoritmo?
¿Importancia del mismo?
Es una serie de pasos a seguir, su importancia es que nos ayuda a resolver algo.

Análisis del prob. Construcción del algorit. Verificación del alg.

Las características del algoritmo son:
Precisión (los datos deben estar claros).
Determinismo (debe asegurarse los mismos resultados de los datos de entrada).
Finitud (el algoritmo independiente de la complejidad del mismo, siempre debe tener longitud finita).

Summary:

En este apartado hemos visto la importancia de los algoritmos, el cual se compone de una serie de pasos que nos ayudan a llegar a un resultado específico, además que resulta muy útil ya que nuestra vida en la computación está rodeada de ellos.

NAME
Francis Juliano

PAGES
2

SPEAKER/CLASS
Programación MC

DATE - TIME
23/05/2024

Title: Diagrama de flujo

Keyword

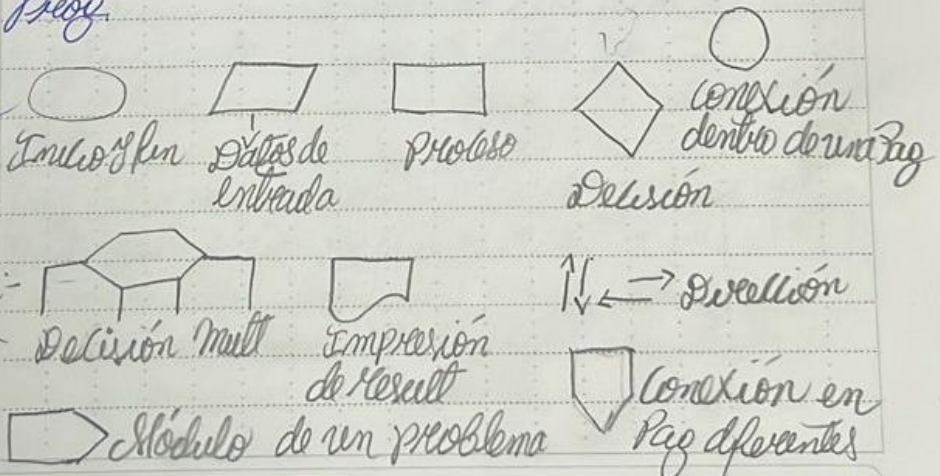
Gráfico
Representación
Pasos
Procesos

Topic: Diagrama de flujo

Notes: Representa la esquematización gráfica de un algoritmo. En realidad muestra gráficamente los pasos o procesos a seguir para alcanzar la solución de un problema. La construcción correcta del mismo es muy importante, ya que a partir de este se escribe el programa en un lenguaje de programación determinado. El diagrama de flujo se debe construir de manera independiente al lenguaje de programación. El diagrama es la solución del problema, el Programa la implementación en un lenguaje de prog.

Questions

¿Cuál es la importancia de este diagrama?
Muestra de manera gráfica la solución a un problema



Summary: El diagrama de flujo es una herramienta vital puesto que muestra soluciones de manera gráfica, además de que cada elemento está puesto ahí por un propósito.

Title: Diagrama de flujo

Keyword	Topic: Reglas Para la construcción de Diag. de flujo.
<p>Secuencia Flujo Diagrama Simple Estructurados</p>	<p>Notes: El diagrama usa símbolos que se conectan entre sí para mostrar una solución de algoritmos a seguir. Algunas reglas importantes para construir un diagrama de flujo son: empezar y terminar claramente, utilizar líneas rectas para indicar el flujo, conectar todas las líneas entre sí, construir de arriba hacia abajo y de izquierda a derecha, usar una notación independiente del lenguaje de programación, agregar comentarios para clarificar tareas complejas y usar conectores si es necesario. Los datos que procesa una computadora se clasifican en simple y estructurados. Tipos de datos en C++</p> <p>int → Enteros (descripción), Rango (-32,768 a 32,767)</p> <p>float Reales " " (3.4 x 10⁻³⁸ a 3.4 x 10³⁸)</p> <p>long Enteros de largo alcance " " (-2,147,483,648 a 2,147,483,647)</p> <p>double Reales de doble precisión " " (1.7 x 10⁻³⁰⁸ a 1.7 x 10³⁰⁸)</p> <p>char Caracter (descripción) Rango (Número del teclado, números o símbolos especiales que van entrecomillados entre comillas)</p>
Questions	
<p>¿Qué impacto tiene que la información o los símbolos no estén correctos? La información no se entenderá</p>	

Summary: El diagrama de flujo se rige en base a una regla de símbolos que son necesarios para que el mismo funcione correctamente y su información sea exacta. Su estructura puede ser simple o compuesta y cada uno tiene una serie de datos necesarios para que funcione bien.

Title: Cap 1: Algoritmos, Diagrama de flujo y Programa en C

Keyword: Identificadores
Programación
Lenguaje C
letra

Topic: Identificadores

Notes: Un identificador está formado por letras, dígitos y el carácter de subrayado (-). Los datos que procesa una computadora ya sean simples o estructurados se deben almacenar en celdas o celdas de memoria para usarlos posteriormente. A dichas celdas o celdas de memoria se les asigna un nombre para reconocerlas, para esto es necesario el identificador. Siempre hay que comenzar con una letra. El lenguaje de programación C se distingue entre minúscula y mayúscula por lo tanto `aux` y `Aux` son dos identificadores diferentes. Cabe señalar que hay nombres que no se pueden utilizar por ser palabras reservadas del lenguaje C, dichos nombres

Questions

¿Para qué sirve el identificador?

Para identificar se presentan a continuación

los datos que procesa una computadora en las celdas o memoria

<code>auto</code>	<code>default</code>	<code>goto</code>	<code>register</code>	<code>struct</code>
<code>break</code>	<code>do</code>	<code>float</code>	<code>return</code>	<code>switch</code>
<code>case</code>	<code>double</code>	<code>for</code>	<code>short</code>	<code>typedef</code>
<code>char</code>	<code>else</code>	<code>if</code>	<code>signed</code>	<code>union</code>
<code>const</code>	<code>enum</code>	<code>int</code>	<code>sizeof</code>	<code>while</code>
<code>continue</code>	<code>extern</code>	<code>long</code>	<code>static</code>	<code>volatile</code>

Summary: Hemos visto una parte esencial dentro del lenguaje C y de la programación, esto es necesario para identificar los datos procesados por una computadora en las celdas o celdas de memoria.

Title: Cap 1: Algoritmos, diagramas de flujo, y Programas en C

Keyword

Datos
Ejecución
Cambiar
Constante

Topic: Constantes y Variables

Notes: Las constantes son datos que no cambian durante la ejecución del programa. Para nombrar las constantes usamos identificadores, existen tipos de constantes de todos los tipos de datos, de tipo entero, real, carácter, cadena de caracteres, etc. Las mismas se deben definir antes de comenzar el programa principal y estas no cambian durante su ejecución.

Const int num1 = 20; // num1 es una constante de tipo entero

Const int num2 = 15; // num2 " " " " " " " "

Const float real = 2.18; // real " " " " " " Real

Const char cat = 'P'; // cat " " " " " " Carácter

Questions

Diferencia entre variable y constante durante la ejecución de un programa. La constante por su nombre no cambia pero la variable puede cambiar.

En ese mismo sentido, las variables son objetos que pueden cambiar su valor durante la ejecución de un programa, usan identificadores. Al igual que las constantes, pueden existir tipos de variables de todo tipo de datos, estas se declaran en el programa principal y funciones, una vez declaradas las variables, estas reciben un valor a través del bloque de asignación. La asignación es una operación destructiva puesto que si tenía un valor, se destruye al asignar un nuevo

Summary:

El lenguaje de programación C es contenedor de muchos elementos, entre estos, variables y constantes, que son datos que cambian, o permanecen fijos durante la ejecución de un programa (respectivamente).

NAME
Elojelis Juliano

PAGES
6

SPEAKER/CLASS
Programación del

DATE - TIME
23/05/2024

Title: Cap 1: Algoritmos, diagrama de flujo y Programa en C

Keyword
operadores
numero
lenguaje

Topic: Operadores

Notes: Los operadores son necesario para realizar operaciones. Existen diferentes tipos de operaciones entre operandos. Numeros, constantes o variables. El resultado de una operación aritmética siempre es un número. Dado que \mathbb{C} distingue entre los tipos de operandos (int y float), en la siguiente tabla se presentan los operadores aritméticos y ejemplo.

Operador	Operación	Ejemplo	R
arit			
+	Suma	$x = 4.5 + 3;$ $v = 4.5 + 3;$	7.5
-	Resta	$x = 4.5 - 3;$ $v = 4.5 - 3;$	1.5
	Multiplicación	$x = 4.5 * 3;$ $v = 4.0 * 3;$ $v = 4.5 * 3;$	12 12 13.5
	División	$x = 4 / 3;$ $x = 4.0 / 3.0;$ $v = 4 / 3;$ $v = 4.0 / 3.0;$ $v = \text{float } 4 / 3;$ $v = \text{float } 5 / 6;$	$x = 1$ 1.0 1.33 1.33 1.33 1.33

Questions
¿Qué son los operadores?
A aquellos que hacen posible que se realice las operaciones.

Summary: Para que las operaciones sean posibles se necesitan los operadores, encargados de determinar la operación a realizar.

By Carlos Pichardo Quique

Title: Cap 1: Algoritmos, diagrama de flujo y Programación C

Keyword

Lenguaje
clarifica
estructuras

Topic: Operadores de incremento y decremento

Notes: Estos operadores (+), (-) son propios del lenguaje C y su aplicación es muy importante porque simplifica y clarifica la estructura de los programas. Se pueden usar antes o después de la variable. Los resultados son diferentes, por ej la siguiente tabla considera que x y y son variables de tipo entero (int x, y).

Questions

Para qué
serven los
operadores
++ , --?

Clarificar las
estructuras de
los programas.

Operador	Operación	Ejemplos	Resultados
++	Incremento	$x = 7;$	$x = 7$
		$y = x++;$	$y = 7$
			$x = 8$
		$x = 7;$	$x = 7$
--	Decremento	$y = ++x;$	$y = 8$
			$x = 8$
		$x = 6;$	$x = 6$
		$y = x--;$	$y = 6$
			$x = 5$
		$x = 6;$	$x = 6$
		$y = --x;$	$y = 5$
			$x = 5$

Summary: Estos operadores son necesarios al momento de realizar operaciones por ej al momento donde es necesario elevar o disminuir el valor de una variable en uno, como en bucles o para realizar operaciones de seguimiento y control en un programa.

Title: Cap 1: Algoritmos, Diagrama de flujo y Programa en C

Keyword: Expresiones lógicas y operaciones relacionales

Expresiones
lógicas
verdaderas
falsas
0, 1
Caracteres

Notes: Las expresiones lógicas están constituidas por números constantes o variables y operadores lógicos o relacionales. El valor que pueden tomar estas expresiones es 1 en caso de ser verdaderas o 0 en caso de ser falsas. Se usan frecuentemente tanto en las estructuras selectivas, como en las repetitivas. En las selectivas se emplean para seleccionar un camino determinado, y en las repetitivas para determinar si el ciclo continúa o se interrumpe.

Questions

¿Qué son expresiones lógicas?
Son aquellas que indican la veracidad de una operación

Las operaciones relacionales se usan para comparar a operandos, que pueden ser números, caracteres, cadena de caracteres, constantes o variables. Ej:

=	Igual a	Res = 'a' == 'a'	Res 0
!=	Diferente de	Res = 'a' != 'b'	Res 1
<	Menor que	Res = 4 < 15	Res 1
>	Mayor que	Res = 22 > 1	Res 1
<=	menor o igual	Res = 15 <= 2	Res 0
>=	mayor o igual	Res = 35 >= 20	Res 1

Summary: Tanto las expresiones lógicas como las operaciones relacionales, sirven para comparar operandos e indicar si la operación es falsa o verdadera.

By Carlos Pichardo Vinque

NAME
Georgeliz Juliano

PAGES
9

SPEAKER/CLASS
Programación

DATE - TIME
23/05/2024

Title: Cap 1: Algoritmos diagramas de flujo y Programas en C

Keyword

Condición
Conjunción
Disyunción
Negación

Topic: Operadores lógicos

Notes: Permiten formular condiciones complejas a partir de condiciones simples, son de conjunción (&&), disyunción (||) y negación (!)

Ej Operador lógico

Operador	Operación	Ej	Resp
! (Negación)		$x = (! (7 > 15))$	$x = 1$
		$(10) > 1 \times 1$	1
&& (Conjunción)		$y = (! 0)$	$y = 1$
		$x = (35 > 20) \&\& (20 <= 25)$	$x = 1$
		1×1	
		$y = 0 \&\& 1$	$y = 0$
(Disyunción)		$x = (35 > 20) (20 <= 18)$	$x = 1$
		$1 \times 1 0$	1
		$y = 0 1$	$y = 1$

Questions

¿Cuántos operadores lógicos existen?

3

Los operadores usan una tabla de verdad que indican comparaciones entre las premisas

Summary: El operador lógico es necesario para realizar operaciones lógicas entre 2 o mas valores, realizar comparaciones y tomar decisiones basadas en condiciones lógicas dentro de los programas.

Title: Cap 1: Algoritmo Diagrama de flujo y Programación

Keyword: Operador
coma
encadenar
expresión

Topic: Operador coma y Precedencia entre operadores

Notes:

La coma es usada como operador para encadenar diferentes expresiones. Considerando que las variables X , V , Z , y Y son del tipo entero, observemos

Expresión	Equivalencia	Resultado
$X(V=3, V \times 5);$	$V=3$	$V=3$
	$X=V \times 5$	$X=15$
$X=(V+=5, V \times 3)$	$V=V+5;$	$V=8$
	$X=5 \times 3$	$X=2$
	$Y=(15 > 10);$	$Y=1$
	$Z=(2 > Y);$	$Z=1$
	$X=Y \& \& Z;$	$X=1$
	$Y=(!(4 > 5));$	$Y=1$
	$Z=(35 > 40) \& \& X;$	$Z=0$
	$X=(! (Y \& \& Z));$	$X=1$

Questions

¿Por qué es necesario la precedencia de los operadores?
 Para que los programas se ejecuten correctamente

Las operaciones ^{no} se ejecutan de derecha a izquierda, es al revés, pero los operadores se aplican según su precedencia

Summary:

La coma ayuda a que la información esté bien organizada y los términos bien encadenados. Las precedencias de operadores ayudan a que toda cosa esté en su lugar y funcione bien.

Title: Cap 1: Algoritmos Diagrama de flujo, Programación

Keyword: correcta Topic: Construcción de diagrama de flujo

Construcción
escribir

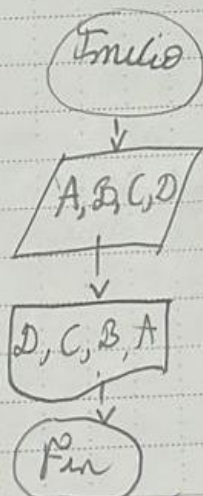
Notes: La correcta construcción del diagrama de flujo, ya que a partir del mismo se debe escribir el programa en su lenguaje de programación determinado.

Un ej: Construir un diagrama de flujo que, al recibir todos los datos A, B, C, D que representen números enteros, escriba lo mismo en orden inverso.

Ej

Questions

¿Por qué es necesario el diagrama de flujo?
Por su claridad y comprensión



Summary:

La correcta construcción del diagrama de flujo es la parte esencial; puesto que de ello depende la comunicación, análisis de los documentos y depuración de procesos y algoritmos.

By Carlos Pichardo Vinque

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
François Juvino	12	Programación	23/05/2024

Title: Cap 1: Algoritmos, Diagramas de flujo y Programación C

Keyword

conjunto
instrucciones
lenguaje
programación
reglas

Topic: Programas

Notes: Es un conjunto de instrucciones que sigue la computadora para obtener un resultado efectivo y específico. El programa se escribe en el lenguaje de programación C en ese caso a partir del diseño de un diagrama de flujo escrito con anterioridad.

El lenguaje de programación está constituido por un conjunto de reglas sintácticas específicas y semánticas. Las reglas sintácticas especifican la formación de instrucciones válidas, mientras que las semánticas especifican el significado de estas instrucciones.

#include <stdio.h>

Program 1.1

El siguiente es el primer programa escrito en el lenguaje C.

void main (void)

{

printf ("mi primer programa en C");

Las instrucciones deben estar dentro de un bloque y finalizar con puntos y coma excepto en algunos casos.

Questions

¿Qué pasa si se pone algún signo mal por alguna equivocación?

El programa no funciona.

Summary:

La programación como hemos visto, es un proceso vital que conlleva una serie de pasos y reglas sintácticas que hacen posible la funcionalidad del mismo por lo que las instrucciones deben estar dentro de un bloque { } y finalizar (;) excepto en estructuras repetitivas o nombres de funciones.

NAME
Blaizelus Luciani

PAGES
13

SPEAKER/CLASS
Programación del

DATE - TIME
23/05/2024

Title: Cap 1: Algoritmos, diagrama de flujo y Programación

Keyword

efectos
importantes
impresión

Topic: Caracteres de control

Notes: Estos producen efectos importantes en la impresión de resultados, continuación se muestra mejor

Carácter de control Explicación

\n

Permite pasar una nueva línea

\t

// tabular horizontalmente

\v

// // verticalmente

\f

Indica al inicio de página

\a

// sonido de alerta

\'

Escibe un apóstrofo

\"

// comillas

\"

// Diagonal invertida

Questions

¿Cuál es el uso de estos caracteres?

Se usan para controlar las impresiones, mejorar la presentación de datos o hacer otras funciones

Es la instrucción

Print("xx\txy\nzz\tRR\nww"); produce

este otro

xx	xy
zz	RR
ww	

Summary:

Los caracteres de control juegan un papel muy importante puesto a que son fundamentales para controlar el flujo y la presentación de datos en sistemas informáticos, además de facilitar la interacción entre el usuario y el sistema.

By Carlos Pichardo Vinque

Title: Cap 1 Algoritmos, diagramas de flujo y Programos en C

Keyword	Topic: Formato de Variables																				
Formato Estructura Lectura Instrucciones	<p>Notes: En el lenguaje C, el formato de estructura y lectura de las variables cambia de acuerdo al tipo de datos que estas puedan tener. La especificación del formato es obligatorio al escribir instrucciones de lectura (long) y escritura (printf).</p> <p>Ej</p> <table border="1"> <thead> <tr> <th>Formato</th><th>Explicación</th></tr> </thead> <tbody> <tr> <td>%u</td><td>Escribe enteros sin signo de bytes (unsigned int)</td></tr> <tr> <td>%d %i</td><td>do 2 bytes (int)</td></tr> <tr> <td>%ld</td><td>Imprime enteros de largo alongo (long)</td></tr> <tr> <td>%f</td><td>Escribe decimales de 4 bytes (float)</td></tr> <tr> <td>%lf</td><td>Escribe decimales de doble precisión 8 bytes (double)</td></tr> <tr> <td>%e</td><td>Imprime en forma exponencial</td></tr> <tr> <td>%g</td><td>Imprime en f o %e en función del tamaño del número</td></tr> <tr> <td>%c</td><td>Escribe un caracter de un byte (char)</td></tr> <tr> <td>%s</td><td>Escribe una cadena de caracteres que termina con '\0'</td></tr> </tbody> </table> <p>Es definir variables. float x=6.2555, z=7.2576, int y=4, t=-5, instrucción printf("%f %d %f %d", x, y, z, t);</p> <p>Produce este resultado = 6.255500 4 7.257600 -5</p>	Formato	Explicación	%u	Escribe enteros sin signo de bytes (unsigned int)	%d %i	do 2 bytes (int)	%ld	Imprime enteros de largo alongo (long)	%f	Escribe decimales de 4 bytes (float)	%lf	Escribe decimales de doble precisión 8 bytes (double)	%e	Imprime en forma exponencial	%g	Imprime en f o %e en función del tamaño del número	%c	Escribe un caracter de un byte (char)	%s	Escribe una cadena de caracteres que termina con '\0'
Formato	Explicación																				
%u	Escribe enteros sin signo de bytes (unsigned int)																				
%d %i	do 2 bytes (int)																				
%ld	Imprime enteros de largo alongo (long)																				
%f	Escribe decimales de 4 bytes (float)																				
%lf	Escribe decimales de doble precisión 8 bytes (double)																				
%e	Imprime en forma exponencial																				
%g	Imprime en f o %e en función del tamaño del número																				
%c	Escribe un caracter de un byte (char)																				
%s	Escribe una cadena de caracteres que termina con '\0'																				
Questions																					
¿Qué es el formato de variable?																					
Es la representación a almanenamiento de un valor de una variable en la memoria de una PC																					

Summary: La especificación del formato es obligatorio debido a que pueden cambiar de acuerdo al tipo de datos que pueda tener.

By Carlos Pichardo Vinque