ALGORITMIA. ELEMENTOS DE PROGRAMA

1. INTRODUCCION A LA ALGORITMIA

Un algoritmo es un método para resolver un problema. Es un conjunto finito de reglas que dan una secuencia de operaciones para resolver un tipo especifico de problema.

* Finito: acaba siempre tras un numero finito de pasos
* Preciso: estar comupuesto por un conjunto ordenado de acciones, especificas en cada caso rigurosamente y sin ambigüedad
* Estar definido: Si realizamos dos veces el proceso daría el mismo resultado.

Secuencia o instrucción es uno de los pasos de un algoritmo.

* 1. Diseño de algoritmos

La algoritmia es una rama de la ciencia que investiga en los algoritmos. Los aspectos a tener en cuenta:

* Analisis de los procesos para encontrar un método de resolución único.
* Optimizacion de los algoritmos.
* Determinacion de la validez de los algoritmos.
  1. Metodos de representación de algoritmos.
* Metodo informal
* Metodos formales:
  + Pseudocodigo:
  + Diagramas:
    - Diagrama de flujo u organigramas
    - Diagrama de Nassi-Schneiderman

1. ELEMENTOS DE UN ALGORITMO

La memoria de un ordenador se divide en casillas o celdas de igual tamaño. Cada casilla es accesible mediante una dirección a la que se denomina dirección de memoria o posición de memoria.

2.1. Datos y tipos de datos

Un dato es un conjunto de celdas o posiciones de memoria que tiene asociado un nombre ( identificador ) y un valor ( contenido ). Hay dos partes nombre y contenido. Un dato es de un determinado tipo se da la siguiente información:

* El rango de los valores
* El conjunto de operaciones primitivas que se puede aplicar

Datos primitivos:

* Logico o booleman: Verdadero o falso.
* Carácter: es un único carácter. Caracteres alfabetivos, numéricos y especiales.
* Numericos:
  + Tipos para enteros: byte, short, int y long.
  + Tipos para valor real de puntoflotante: Float y doublé.

2.2 Variables y constantes

Una variable es undato que posee un valor y que es conocido e un programa o en un algoritmo por un nombre ( identificador de la variable).

Una constante contiene valores que no deben cambar a lo largo del desarrollo del algoritmo. La inicialización es el primer y único valor. Tambien se denominan variables finales.

2.3 Clasificacion de las variables de Java

Una variable es un nombre que contiene un valor que puede ir cambiando a lo largo del programa.

* Variable tipos primitivos:
* Variable referencia:

Desde el punto de vista del papel en el programa:

* Variable miembro de un clase
* Variable Locales

2.4 Declaracion e inicialización de variables y constantes.

Definimos la variable con el nombre y diciendo el tipo de variable que es. Las inicializadas de tipo primitivo a 0 ( las numéricas) false ( boolean ) y /0 si es que no se especifica una inicialización diferente. Las variables tipo referencia se inicializan en null.

Las variables locales no se inicializan a ningún valor por defecto,tienen un valor desconocido.

La declaración de variables es <tipo><nombre-variable>

2.5 Operadores de Java

Una operación primitiva es aquella que se realiza directamente en un lenguaje de programación concreto. Estas operaciones primitivas u operadores se clasifican según el numero de datos:

* Unarios
* Binarios
* Enerarios

Operadores a las funciones y operandos a los argumentos.

2.5.1 Operadores aritméticos:

Son binarios: suma, resta, multiplicación, división y resto de la división

El resultado de una operación tiene el mismo tipo que los operandos implicados.

2.5.2. Operadores de asignación

Asignan un valor a una variable. Operador igual. Cuando se asigna a una variable una expresión decimos dos cosas:

* Evaluar la expresión
* Almacenar el resultado de la evaluación en la variable.

La operación de asignación es destructiva, por lo que cualquier valor que tuviese la variable de asignación se pierde y se reemplaza por el nuevo.

Existen abreviaturas del operador = que realizan operaciones acumulativas sobre una variable.

2.5.3. Operadores unarios

Operador mas y menos cambiar el signo de la variable, constante o expresión numérica.

2.5.4 Operación instaceof

Permite saber si un objeto pertenece a una deterninada clase o no. Su forma general es: objectName instanceof ClassName Devolviendo un resultado boolean ( true o false )

2.5.5. Operador condicional

Permite realizar bifurcaciones. Devolviendo res1 ( true ) o res2 (false). Es el único operador ternario

2.5.6 Operadores incrementales

Existe el operador incremento ( ++) y decremento ( -- ). Consiste en reducir o incrementar una unidad. Pueden ir delante o detrás de nuestra variable:

* Predecer: Primero se incrementa a variable y lugo se utiliza
* Posterior: Primero se utiliza la variable y luego se incrementa

2.5.7 Operadores relacionales

Sirven para realizar comparaciones de igualdad.ass, desigualdad y relación de mayor y menor. El resultado siempre será boolean.

Son muy usados en bifurcaciones y en los bucles.

2.5.8. Operadores lógicos

Son para datos lógicos o booleano determinados por:

* Conjuncion: Verdadero si ambos operandos son verdaderos si no falso.
* Disyuncion: Devolvera verdadero siempre que un operando sea verdadero
* Negacion: devuelve el contrario al valor aplicado

2.5.9. Operador de concatenación de cadenas de caracteres

El operador + se utiliza para la concateniacion de cadenas de caracteres.

2.5.10 Operadores que actúan a nivel de bits.

Se utilizan para definir señales o flags, variables de tipo entero en las que cada uno de sus bits indican si una opción esta activada o no.

2.5.11 Precedencia de los operadores

2.6 Conversiones de tipo ( casting )

Se realiza cambios de una variable de un tipo a otra. Conversiones implícitas de un tipo a otro de mas precisión. Las conversiones de una variable mayor a otra menor son conversiones inseguras denominándolas Cast.

Una conversión especifica es el cambio de un numero entero a un carácter y viceversa.

1. EXPRESIONES

Es una combinación de variables, constantes, valores constantes, operadores, paréntesis y nombre de funciones especiales. Las expresiones se evalúan de izquierda a derecha.

1. ELEMENTOS AUXILIARES DE UN ALGORTIMO

4.1 Contadores

Es una zona de memoria representada por un identificador cuyo valor incrementa en una cantidad fija, asociado generalmente a un bucle. Antes de empezar debemos inicializarlo.

4.2 Acumuladores

Es una zona de memoria cuyo valor se incrementa sucesivas veces en cantidades variables. Se inicializa en 0 o 1.

4.3 Interruptores

Es una zona de memoria que tiene 2 valores exclusivos ( 0 y 1, -1 y 1, verdadero o falso, on y off,etc)