ALGORITMOS

Dato: conjunto de celdas o posiciones de memoria que tiene asociada un nombre ( identicador ) y un valor ( contenido).

Si especificamos que un dato es de un tipo le damos la siguiente información:

* Rango de los valores permitidos
* Conjunto de operaciones primitivas que se pueden aplicar.

Los tipos de datos permitidos en Java son:

* LOGICO: llamados boolean. Los valores puedes ser verdadero o falso, no tienen carácter numérico.
* CARÁCTER: llamados Char. Son de tipo carácter letras, números y símbolos.
* NUMÉRICOS: tenemos varios subconjuntos.
  + Valores enteros:
    - Byte: valor entero de pequeña capacidad
    - Short: valor entero de pequeña-media capacidad
    - Int: valor entero de capacidad media
    - Long: Valor entero de alta capaciadad
  + Valores reales de punto flotante:
    - Float: valor con decimales de capacidad media
    - Double: valor con decimales de alta capacidad

Las características de los datos serán las siguientes:

1. Boolean no tiene valor numérico solo nos dice si es verdadero o falso
2. Char contiene caracteres en código UNICODE
3. Bytem short, int y long son números enteros y en caso de que el resultado sea decimal produce un redondeo siendo positivos y negaticos
4. Float y doublé son números decimales
5. Void es usado para indicar la ausencia de un tipo de variable determinada.

2.2 VARIABLE Y CONSTANTES

Una variable es un datoque posee un valor y que es conocido en un programa o en un algoritmo por un nombre que es el identificador de la variable. El valor puede ser modificado en el seguimiento del algoritmo.

Para nombrar variables deberemos seguir las siguientes normas:

* La variable comience con una letra preferiblemente minúscula seguida de varias letras, números o el carácter especial \_
* No se permitirán espacion en blanco
* El sistema Java es sensible a mayúsculas y minúsculas

Las constantes son valores que no cambian en el desarrollo del algoritmo. Las constantes toman el primer y único valor se llama inicialización de la constante. Las constantes las conocemos como variables finales.

Un concepto relacionado son los valores constantes, es lo mismo pero aparece directamente en el algoritmo el valor con lo cual podemos referirnos a ellas únicamente con su valor, nombre o lo que sea.

Las variables, valores constantes o constantes pueden ser de varios tipos cadena, lógico entero, real…) Es lógico pensar en el caso de las variables que si la variable es entera solo podremos darle valores enteros y as ion todo.

Existen una serie de palabras que no podrán ser usadas en JAVA para designar las variables porque son de uso propio por JAVA.

CLASIFICACION DE LAS VARIABLES DE JAVA

Las variables son incognitas que pueden cambiar a lo largo de la realización del algoritmo. Tipos:

* VARIABLE DE TIPO PRIMITIVO: Definidas mediante un valor único que puede ser entero, punto flotante ( decimal ) carácter o booleano. Básicamente los 8 parametros que tenemos.
* VARIABLE REFERENCIA. Dan una información mas compleja referente a arrays u objetos de una determinada clase.

Una clasificación desde el punto de vista de su papel en el algoritmo:

* VARIABLES MIEMBRO DE UNA CLASE: Se definen en una clase fuera de cualquier método puedes ser primitivo o referencia.
* VARIABLES LOCALES: Se definen dentro de un método o mas en general dentro de cualquier bloque entre llaves. Se crean en el interior del bloque y se destruyen al finalizar el mismo pueden ser primitivo o referencia.

DECLARACION E INICIALIZACIO NDE VARIABLES CONSTANTES.

Para nombrar una variable se definirán especificacndo el tipo y el nombre de la variable ( para ambas clases ). Una vez realizada la reserva de memoria hemos de tener en cuenta que:

* Primitiva: Se inicializan a cero salvo boolean y char que si inicializan en false y \0 si no se especifica un valor en su declaración.
* Referencia: son inicializadas por defecto a un valor especial: null.

Exite una variable que no se inicializa a ningún valor por defecto, por tanto tendrán un valor desconocido ( variables locales ) que nos obligara a inicialarzas antes de usarlas.

OPERACIÓN EN JAVA

Las operaciones primitivas son aquellas operaciones que se realizan directamente en un lenguaje de programación. Si tenemos operaciones no primitivas tendremos que indicar como realizarlas. Su clasificación esta basada en el numero de datos que opera:

* Unarios: se aplican sobre un solo dato o argumento
* Binarios: Se aplican sobre dos datos o argumentos
* Enerarios: Se aplica sobre N argumentos.

Operadores a las funciones y operando a sus argumentos o datos.

1. Operadores Aritmeticos:

Son operadores binarios ( necesitamos dos operandos ) realiza: suma, resta, multiplicación, dividion y resto de la división entera. El resultado de toda operación tiene el mismo tipo que los operandos implicados.

1. Operadores de Asignacion:

Permiten asignar un valor a una variable. Las forma general de sentencia es: Variable=expresión.

1. Operadores unarios:

Los operadores mas y menos sirven para cambiar el signo de una variable, constante o expresión numérica.

1. Operadores instaceof:

Permite saber si un objeto pertenece a una determinada clase o no. Es un operador binario que contesta con boolean ( V o F ) .

1. Operador condicional:

Se trata de un operador ternario, Se trata de una condicional que responde con verdadero ( res1 ) o falso ( res2 )

1. Operadores incrementales:

ES un operador que incrementa o reduce en una unidad la variable a la cual están aplicados. Pueden usarse de dos maneras:

* Predece: primero se incrementa y luego se usa la variable:
* Siguiendo: primero usamos la variable y luego se incrementa.

1. Operadores relacionales:

Realizan comparaciones de igualdad, desigualdad y relación de menor o mayor. Se usa en bifurcaciones o bucles. El resultado es un valor boolean.

1. Operadores lógicos:

Son operaciones de tipo lógico o booleano. Viene determinado por la tabla de verdad. Existen 3 tipos de operadores lógicos:

* Conjuncion: si los dos son Verdadero el resultado es verdadero sino es falso.
* Disyuncion: es verdadero si algún operando es verdadero.
* Negación: Negara el valor que tenga el operando.

OPERADOR DE CONCATENACION DE CADENAS DE CARACTERES

El operados mas ( + ) se utiliza para concatenar cadenas de caracteres.

OPERADORES QUE ACTUAN 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.

PREDECENDCIA DE OPERADORES.

El orden en el cual se realizan las operaciones es fundamental por ello existe una lista del orden de prioridad de dichos operadores.

Los operadores binarios excepto los deasignacion se evalúan de izquierda a derecha. Los operadores de asignación de derecha a izquierda.

CONVERSIONES DE TIPO CASTING

La conversión entre tipos primiticos es mas sencilla. Java las realiza de forma automática de un tipo a otro de mas precisión.

Las conversiones de un tipo mayor a otro menor requieren una orden explicita del programador pq son conversiones inseguras que pueden dar lugar a errores. Estas conversiones se llaman cast.

EXPRESIONES

Es unacombinacion de variables, constantes, valores constantes, operadores, paréntesis y nombres de funciones especiales. El resultado de evaluarla de izquierda a derecha es el valor de la expresión.

Cuando deseamos que se evalue la expresión de forma distinta usamos los paréntesis.

* Las operaciones encerradas entre paréntesis se evalúan primero.
* Las operaciones aritméticas dentro de una expresión se evalúan según el orden de prioridad visto y en caso de igualdad de izquierda a derecha.

ELEMENTOS AUXILIARES DE UN ALGORITMO

Son variables que realizan funciones especificas dentro de un programa ( contadores, acumuladores e interruptores.

CONTADORES. Es una zona de memoria representada por un identificador cuyo valor se incremente aen una cantidad fija y esta asociado generalmente a un buclo. El valor inicial del contador es 0.

ACUMULADORES. Es una zona de memoria cuyo valor se incrementa sucesivas veces en cantidades variables. Siendo el valor inicial 0.

INTERRUPTORES. Es una zona de memoria que puede tomar dos valores exclusivos.

TEMA 3: CONCEPTOS FUNDAMENTALES DE ORIENTACION A OBJETOS

1.INTRODUCCION

1.1 Evolucion histórica.

La POO ( Programacion Orientada a Objetos ) tiene la idea fundamental que el trabajo pueda ser dividido de forma coherente en pequeñas unidades independientes de manera que la necesidad de coordinación sea minima.

Los programadores se dividen en:

* Los productores de componentes
* Los consumidores de componentes.

1.2. ¿ Que es irientado a objetos ¿

El software es una colección de objetos.

1.3. La consecuencia

* Posibilidad de representar directamente las entidades del mundo real en los entornos informáticos sin necesidad de deformarlos ni descomponerlos.
* Posibilidad de reutilizar y extender las aplicaciones existentes.
* Trabajo con entornos de desarrollo potentes.
* Disponibilidad de herramientas de comunicación hombre.maquina visuales de gran calidad
* Facilidades de prototipado rápido de aplicaciones
* Facilidad de utilización de paralelismo en el momento de la implementación

2.FACTORES CRUCIALES QUE MIDEN LA CALIDAD DEL SOFTWARE

* Factores externos: Perceptibles por el usuario ( de cualquier nivel ):
  + Correcion: habilidad de ajustarse a los requerimientos
  + Robustez: habilidad de funcionar aun bajo condiciones anormales.
  + Extensibilidad: facilidad de adaptarse a cambios en los requerimientos
  + Reutilidad: habilidad para ser reutilizado todo, o en parte, en nuevos desarrollos
  + Eficiencia: habilidad en el uso optimo de los recursos hardware.
* Factores internos: perceptibles por los desarrolladores del producto
  + Formalidad ( corrección + robustez ): debe ser fácil crear software que funcione correctamente y fácil garantizar que esto es asi
  + Modularidad ( reutilización + extensibilidad ):: debe crearse el menor software posible. Debe ser fácil de modificar.

Existe una cualidad no mencionada: facilidad de mantenimiento, ya sea por modificaciones en las especificaciones de los sistemas informáticos o por la depuración de errores a posteriori.

3. ¿Qué ES LA PROGRAMACION ORIENTADA A OBJETOS?

Un método de implementación en el que los programas se organizan como colecciones cooperativas de objetos. Cada uno de los cuales representan una instancia de alguna clase, y cuyas clases son todas miembros de una jerarquía de clases unidas mediante relaciones de herencia.

1. Utiliza objetos, no algoritmos, como bloques de construcción lógicos
2. Cada objeto es una instancia de una clase
3. Las clases se relacionan unas con otros por medio de relaciones de herencia

El concepto de objeto es una colección de elementos de datos, junto con las funciones asociadas para operar sobre esos datos. Sin embargo la potencia real de los objetos reside en el modo en que los objetos puden definir otro objetos. Este proceso, se denomina herencia y es el mecanismo que ayuda aa construir programas que se modifican fácilmente y se adaptan a aplicaciones diferentes.

4. OBJETOS

Un objeto representa un elemento individual e identificable, real o abstractom con un comportamiento bien definidifo en el dominio del problema.

El estado se refiere a las propiedades o atributos que caracterizan al objeto. Cada atributo tiene un valor en algún dominio.

El comportamiento de un objet viene determinado por la forma en la que el objeto interactua con el sistema. La forma de actuar es enviándole un mensaje , que activara un comportamiento ( método ) que determina su forma de actuar.

La identidad del objeto es la propiedad característica que los distingue del resto, no se la damos la tiene intrínseca.

La implementación de un objeto:

* Mantiene una memoria privada que describe las propiedades del objeto y sus atributos.
* Disponen de un conjunto de operaciones que actúan sobre dicha memoria privada, definen su comportamiento.
* Un objeto tiene una identidad.

La forma en la cual afecta el método es mediante el envio de mensajes. Receptor que es el objeto y selector que es el comportamiento. Como resultado del envio de un mensaje dependerá el mismo de:

* Del estado en que se encuentre dicho objeto
* Del método involucrado en el mensaje
* De la transformación que este mensaje pueda soportar.

Un método tiene total visibilidad sobre los atributos del objeto El único camino para acceder al estado de un objeto es por activación de algún método. El conjunto de métodos es capaz de responder es su protocolo o interfaz que definen su conducto vs al estimulo recibido.

5.CLASES

Una clse describe el comportamiento de una familia de objetos. Es una plantilla para crear objetos que tienen:

* Los mismos atributos
* Responden a los mismos mensajes.

A los objetos de una clase se les llama instancias de clase. A los atributos que describen la memomria privada de una objeto se les suele llamar, datos miembro, variable de instancia o slot.

A los métodos de comportamiento se les suele llamar funciones miembro o métodos de instancia.

Ventajas de la utilización de clases:

* Cada clase puede ser creada de modo independiente
* Cada clase puede probarse de modo independiente
* Asegura la consistencia de los datos pues ofrece un interfaz para su manejo
* La implementación queda escondida al usuario de la clase ( lo mismo que la implementación de los enteros queda oculta a los que los usan )
* Puede variarse la implementación sin tener que cambiar los programas que las utilizan
* Es altamente reutilizable

En resumen:

* Todas las instancias de una clase responden al mismo conjunto de mensajes con los mismos métodos
* Todas las instancias de una clase tienen las mismas variables de instancia pero cada una con sus valores

1. ECUACION FUNDAMENTAL DE LA POO

PROGRAMACION ORIENTADA A OBJETOS = TIPOS ABSTRACTOS + HERENCIA + POLIMORFISMO

* 1. Tipos abstractos de datos

Los tipos abstractos de datos permiten encapsulación ( guardan datos y operaciones de forma conjunta ) y ocultación ( protegerlos)

Las ventajas que apostan son:

* Implementacion escondida al cliente
* Los TADs ( tipos abstractos de datos ) se generan independiente,ente
* Se pueden probar independientemente
* Aseguran la consistencia de los datos
* Aumentan la reutilización de código
  1. Herencia

Es una técnica que permite incrementar la reusabilidad:

* Maneja eficientemente relaciones
* Crea nuevas clases a partir de generalizar o especializar otras clases ya existentes.

Para crear clases reutiliza la conducta de otra. Lo hacemos añadiendo, suprimiendo o modificanco métodos o añadiendo o suprimiendo variables de instancia.

La herencia puede ser:

* Simple: Es aquel tipo de herencia en la cual un objeto ( clase ) puede tener solo un ascendiente, una subclase puede heredar datos y métodos de una única clase
* Multiple: Es aquel tipo de herencia en la cual una clase puede tener mas de un ascendiente inmediato

Una clase abstracta es una clase que proporciona un interfaz común y básico a sus herederas. De ella no obtenemos ninguna instancia. Definira métodos con el cuerpo vacio o métodos con comportamientos generales a todas las subclases.

* 1. Polimorfismo

Es el uso de un nombre o un símbolo para representar o significar mas de una acción. Es la capacidad de una entidad de referencias distintos elementos en distintos instantes. Tipos:

* Por sobrecarga de funciones: Dos funciones con el mismo nombre y distintos argumentos son junciones distintas. Dos funciones con el mosmo nomrbe y los mismos argumentos pero definidas en clases distintas son funciones distintas.
* Por vinculación dinámica: paramtrico:
  + En tiempo de compilación no puede determinarse el tipo de algunas variables o expresiones
  + El compolliador puede que no sepa a que clase pertenece un objeto antes de la ejecuacion
  + Por tanto hasta tanto no se determine la clase a la que pertenece el objeto, no podrá sustituir el cuerpo. Por tanto, esta vinculación tiene que hacerse en tiempo de ejecución.

1. METODOS Y VARIABLES DE CLASE

Variable de clase: una variable común compartida por todos los objetos de una clase

Metodos de clase: métodos definidos para la clase que controlan el acceso a las variables de clase.

1. LENGUAJES

Los lenguajes orientados a objetos los podemos clasificar en tres grandes grupos:

* PUROS
* EXTENSIONES
* CODIGOS PORTABLES

TEMA 4: EL LENGUAJE DE JAVA

1. Evolucion Historica
2. ¿Qué es Java 2?

Java 2 es la tercera versión del lenguaje Java de programación, son extensiones y ampliaciones.

1. Caracteristicas de Java

Las principales son:

* Simple: Esta basado en C++, no exiten los punteros ni aritmética de punteros, las cadenas de caracteres son objeto y la administración de memoria es automática.
* Orientada a Objetos:
  + Encapsulacion, herencia y polimorfismo
  + Interfaces para suplir la carencia de herencia multiple
  + REsolucion dinámica de métodos
* Distribuido:
  + Extensas capacidades de comunicaciones.
  + Permite actuar con http y ftp
  + El lenguaje no es distribuido, pero incopora facilidades para construir programas distribuidos
  + Se están incorporando características para hacerlo distribuido.
* Robusto: Realiza continuos chequeos en tiempo de compilación y en tiempo de ejecución. Chequeos de punteros nulos, chequeo de limite en vectores, excepciones, verificación de código binario, recolección automática de basuras…
* Seguro:
  + No hay punteros
  + El cast ( promoción ) hacia lo general es implícito
  + Los bytecodes ( ficheros compilados con extensión .class ) pasan varios test antes de ser ejecutados
* Arquitectura neutral ( mismo código en distintas arquitecturas )
  + Define la longitud desus independientemente de la plataforma
  + Construye sus interfaces en base a un sitema abstracto de ventanas.
* Interpretado
  + Para conseguir la independencia del S.O. genera bytecodes
  + El interprete toma cada bytecode y lo interpreta
  + El mismo interprete corre en distintas arquitecturas
* Multihebras:
  + Permite crear tareas o hebras.
  + Sincronizacion de métodos
  + Comunicación de tareas
* Dinamica:
  + Conecta los modulos que intervienen en una aplicación en el momento de su ejecución
  + No hay necesidad de enlazar previamente.

1. El Entorno de Desarrollo de Java

Para compilar y ejecutar los programas existen dos opciones:

* Utilizar un entorno integrado de desarrollo IDE. Reune varios programas necesarios para el desarrollador, en uno solo, en el caso de java.
* Emplear el software básico de desarrollo SDK de Oracle.