1. El proceso de solucion de problemas con una computadora conduce a la escritura de un programa y a su ejecución. El proceso de diseñar programas es un proceso creativo que posee una serie de pasos fundamentales que son:

* Análisis de problemas
* Diseño de algoritmos
* Codificación
* Compilacion
* Ejecución
* Verificación y depuración

Los dos primeros pasos conducen a un diseño detallado escrito en forma de algoritmo. Durante el tercer paso (codificación) se implementa el algoritmo en un código escrito en un lenguaje de programación. La fase de compilación y ejecución traduce y ejecuta el programa. En las fases de verificación y depuración el programador busca errores de las etapas anteriores y los elimina. Antes de conocer las tareas a realizar en cada paso, vamos a considerar el concepto y significado de algoritmo. Un algoritmo es un método para resolver un problema mediante una serie de pasos precisos, definidos y finitos.

1. Lenguaje: Lenguaje de programación En informática, cualquier lenguaje artificial que puede utilizarse para definir una secuencia de instrucciones para su procesamiento por un ordenador o computadora.

Es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

Programa: es un conjunto de instrucciones que al ejecutarlos cumple con varias tareas o funciones sin programas, estas maquinas no pueden funcionar o solucionar problemas estos programas se definen como software.

Compilador: Es un programa informático, que se encarga de traducir (compilar). En otras palabras es un software que se encarga de traducir el programa echo en un lenguaje de programación y lo pasa o transforma aun lenguaje de maquila el cual puede ser comprendido

Interpretador: ejecuta el programa y traduciendo cada sentencia de una secuencia de las subrutinas que ya han sido cargadas o compiladas en lenguaje maquina

1. La Programación Orientada a Objetos (POO) surge por primera vez en Noruega en 1967con un lenguaje llamado Simula 67desarrollado por Krinsten Nygaard y Ole-Johan Dahl, en el centro de cálculo noruego. Simula 67 introdujo por primera vez los conceptos de clases, corrutinas y subclases (conceptos muy similares a los lenguajes Orientados a Objetos de hoy en día); De ahí nace el famoso “HOLA MUNDO”.Después en los años70’s fue desarrollado SmallTalk en Simula en Xerox Parc, programa que fue desarrollado en Basic el cual refino las características de Simula 67, siendo éste un sistema completamente dinámico en el cual los objetos se podrían crear y modificar, en lugar de tener un sistema estático. Quien tuvo la idea fue D. Parnas cuando propuso la disciplina de ocultar la información. Su idea era encapsular cada una de las variables globales de la aplicación en un solo módulo junto con sus operaciones asociadas. En los años 80’s. “C” se amplia y cre “C++”, programa que contaba con 3 principales características: abstracción, soporte para la POO y el soporte de plantillas o programación genérica. Las características de Orientación a Objetos fueron agregadas en ese tiempo a muchos otros programas como: Ada, Basic, Lisp, Pascal, entre otros. Para crear nuevos lenguajes basados en métodos Orientados a Objetos, pero permitiendo algunas características imperativas de manera segura se creó el “Eiffelde Bertrand Meyer” quien fue un temprano y moderadamente acertado lenguaje con esos objetivos pero que de manera rápida fue remplazado por JAVA. En 1995 aparece “JAVA”, el más reciente lenguaje OO, desarrollado por la empresa“ SUN Microsystem”, que hereda conceptos de “C++”, pero los simplifica y evita la herencia multiple. Obtiene una rápida aceptación gracias a los applets, que son programas en JAVA insertados en páginas WEB dentro del código HTML. JAVA introduce también la programación concurrente y distribuida. En 1997-98 se desarrollan herramientas “CASE” orientadas a objetos (como el diseño asistido por computadora). Del 98 a la fecha se desarrolla la arquitectura de objetos distribuidos RMI, Corba, COM, DCOM.
2. Programación OO.

* La programación orientada a Objetos básicamente define una serie de conceptos y técnicas de programación para representar acciones o cosas de la vida real basada en objetos, a diferencia de otras formas de programación como por ejemplo la estructurada, con la POO trabajamos de manera distinta vinculando diferentes conceptos tales como clases, objetos, métodos, propiedades, estados, herencia, encapsulación entre otros, generando cada vez interrelaciones en nuestro desarrollo en pro del funcionamiento del sistema principal, definiendo el programa como un conjunto de estos objetos relacionados entre si.

Clases.

* Las clases son uno de los principales componentes de un lenguaje de programación, pues en ellas ocurren todos los procesos lógicos requeridos para un sistema, en si podemos definirlas como estructuras que representan objetos del mundo real, tomando como objetos a personas, lugares o cosas, en general las clases poseen propiedades, comportamientos y relaciones con otras clases del sistema

**Nombre :** Contiene el Nombre de la Clase.

**Atributos :** Representan las propiedades que caracterizan la clase.

**Métodos :** Representan el comportamiento u operaciones, la forma como interactúa la clase con su entorno.

Objeto

* Los objetos representan una entidad concreta o abstracta del mundo real, en programación básicamente se le conoce como la instancia de una clase en si es lo que da el sentido a estas.

Al igual que las clases se componen de tres partes fundamentales:

**Estado:** Representa los atributos o características con valores concretos del objeto.

**Comportamiento:** Se define por los métodos u operaciones que se pueden realizar con el.

**Identidad:** Es la propiedad única que representa al objeto y lo diferencia del resto.

Herencia.

* La herencia en java representa lo que conocemos de herencia en el mundo real, básicamente mediante esta obtenemos las características o rasgos comunes de nuestros padres o abuelos, en java es el mismo enfoque permitiendo la creación de nuevas clases basadas en clases ya existentes, con las cuales podemos obtener las características de las clases padres, heredando campos, atributos, métodos o funcionalidades.

Encapsulamiento.

* Este concepto es uno de los mas importantes en términos de seguridad dentro de nuestra aplicación, la encapsulación es la forma de proteger nuestros datos dentro del sistema, estableciendo básicamente los permisos o niveles de visibilidad o acceso de nuestros datos Se representa por 3 niveles :

**Público:** Se puede acceder a todos los atributos o métodos de la clase.

**Protegido:** Se puede acceder a los atributos o métodos solo en la misma jerarquía de herencia.

**Privado:** Solo se puede acceder a los atributos o métodos de la clase en la que se encuentran.

Clases Abstractas.

* La abstracción permite resaltar la parte mas representativa de algo, ignorando detalles para centrarse en lo principal.



La imagen es muy fácil de identificar, con base a ella podemos crear una clase persona, o la clase hombre, humano entre otras, pero obviamente vemos que la imagen no tiene elementos como ojos, nariz, boca, rostro en general, ni dedos, pies, manos o cuello....... pero entonces porque decimos que es una persona?.........Precisamente aquí estamos aplicando el concepto de abstracción, ya que nos fijamos en lo mas representativo de algo, en este caso vemos que se tiene una cabeza, tronco, brazos y pies, con esto es suficiente para saber que es una persona sin fijarnos en los detalles mencionados anteriormente.

Las clases abstractas permiten crear métodos generales con un comportamiento común para otras clases concretas sin importar sus características ni el comportamiento que usen para dichos métodos.

Polimorfismo.

* Este tal vez sea uno de los conceptos de la programación orientada a objetos mas usados pero muchas veces sin saber que se aplica ya que el concepto inicialmente puede ser un poco confuso, básicamente mediante el polimorfismo programamos de forma general en lugar de hacerlo de forma especifica, se usa cuando se trabajen con la herencia y objetos de características comunes los cuales comparten la misma superClase y árbol jerárquico, al trabajar con este concepto optimizamos y simplificamos en gran medida nuestro trabajo.

básicamente podemos definirlo como la capacidad que tienen los objetos de comportarse de múltiples formas sin olvidar que para esto se requiere de la herencia, en si consiste en hacer referencia a objetos de una clase que puedan tomar comportamientos de objetos descendientes de esta.

1. El contexto del Enfoque Orientado a Objetos (EOO) un objeto es una entidad que encapsula datos (atributos) y acciones ofunciones que los manejan (métodos). También para el EOO un objeto se define como una instancia o particularización de una clase.

Los objetos de interés durante el desarrollo de software no sólo son tomados de la vida real (objetos visibles o tangibles),también pueden ser abstractos. En general son entidades que juegan un rol bien definido en el dominio del problema. Un libro, una persona, un carro, un polígono, son apenas algunos ejemplos de objeto.

Cada objeto puede ser considerado como un proveedor de servicios utilizados por otros objetos que son sus clientes. Cada objeto puede ser a al vez proveedor y cliente. De allí que un programa pueda ser visto como un conjunto de relaciones entre proveedores clientes. Los servicios ofrecidos por los objetos son de dos tipos: Los datos, que llamamos atributos. Las acciones o funciones, que llamamos métodos.

1. **Tipos de datos**.

Los datos se clasifican en diversas categorías, según el tipo de máquina o del lenguaje en uso. Generalmente podemos encontrar las siguientes categorías :

* Numéricos
* Lógicos
* Cadenas

**Datos Numéricos**

Son aquéllos que representan una cantidad o valor determinado. Su representación se lleva a cabo en los formatos ya conocidos (enteros, punto y fracciones decimales si estas existen).

Estos pueden representarse en dos formas distintas :

* Tipo Numérico Entero (integer).
* Tipo Numérico Real (real).

**Enteros**

Es un conjunto finito de los números enteros. Los enteros son números completos, no tienen componentes fraccionarios o decimales y pueden ser negativos y positivos.

**Reales**

Consiste en un subconjunto de los números reales. Estos números siempre tienen un punto decimal y pueden ser positivos o negativos. Un número real consiste de un número entero y una parte decimal

**Cadenas**

Son los datos que representan información textual (palabras, frases, símbolos, etc). No representan valor alguno para efectos numéricos. Pueden distinguirse porque son delimitados por apóstrofes o comillas.

Se clasifica en dos categorías :

* Datos tipo carácter (char)
* Datos tipo Cadena (string)

**Datos Tipo Carácter**

Es un conjunto finito y ordenado de caracteres que la computadora reconoce. Un dato de este tipo contiene solo un carácter

**Datos Tipo Cadena (string)**

Es un sucesión de caracteres que se encuentran delimitados por una comilla (apóstrofe) o dobles comillas, según el tipo de lenguaje de programación. La longitud de una cadena de caracteres es el número de ellos comprendidos entre los separadores o delimitadores

**Lógicos**

También se le denomina Booleano, es aquél dato que solo puede tomar uno de dos valores: Falso y verdadero. Se utiliza para representar las alternativas (si/no) a determinadas condiciones. Por ejemplo, cuando se pide si un valor entero sea primo, la respuesta será verdadera o falsa, según sea.

Las categorías y tipos que se mencionaron anteriormente se conocen como Tipos Simples, puesto que no poseen una estructura compleja. En forma adicional, cada lenguaje puede proporcionar la utilización de Tipos Compuestos, siendo estos, datos que tienen una estructura predeterminada.

1. **Variable**

Una variable, en Java, es un identificador que representa un espacio de memoria que contiene información. La información es del tipo con que se declaro la variable.

Hay 2 categorías de tipos para variables, los tipos primitivos (como int, short, byte, long, char, boolen y double) y las referencias a objetos (como String, Array y otros objetos).

Java tiene ademas, tres tipos de variables, que son:

* De instancia
* De clase
* Locales

Las variables de instancia, se usan para guardar los atributos de un objeto particular.

Las variables de clase, son variables que guardan el mismo valor, para todos los objetos de una clase determinada.

Las variables locales, se utilizan dentro de los métodos, en Java, las variables locales pueden declararse en el momento en que serán utilizadas, y una buena costumbre es inicializar las variables al momento de declararlas.

**Expresiones**

Una expresión es una combinación de valores y operaciones que, al ser evaluados, entregan un valor.

Algunos elementos que pueden formar parte de una expresión son: valores literales (como 2, "hola" o 5.7), variables, operadores y llamadas a funciones

**Constantes**

Una constante es un dato cuyo valor no puede cambiar durante la ejecución del programa. Recibe un valor en el momento de la compilación y este permanece inalterado durante todo el programa.

Como ya se ha comentado en el tema sobre las partes de un programa, las constantes se declaran en una sección que comienza con la palabra reservada const. Después de declarar una constante ya puedes usarla en el cuerpo principal del programa. Tienen varios usos: ser miembro en una expresion, en una comparación, asignar su valor a una variable, etc.

1. Aun no
2. El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación.

**PARA QUE FUNCIONA**

Hay muchos paradigmas o modelos para la resolución de problemas en la informática, que es el estudio de algoritmos y datos. Hay cuatro categorías de modelos para la resolución de problemas: lenguajes imperativos, funcionales, declarativos y orientados a objetos (OOP). En los lenguajes orientados a objetos, los algoritmos se expresan definiendo 'objetos' y haciendo que los objetos interactúen entre sí. Esos objetos son cosas que deben ser manipuladas y existen en el mundo real. Pueden ser edificios, artefactos sobre un escritorio o seres humanos.

Los lenguajes orientados a objetos dominan el mundo de la programación porque modelan los objetos del mundo real. UML es una combinación de varias notaciones orientadas a objetos: diseño orientado a objetos, técnica de modelado de objetos e ingeniería de software orientada a objetos.

UML usa las fortalezas de estos tres enfoques para presentar una metodología más uniforme que sea más sencilla de usar. UML representa buenas prácticas para la construcción y documentación de diferentes aspectos del modelado de sistemas de software y de negocios.

Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene. UML no es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML. UML guarda una relación directa con el análisis y el diseño orientados a objetos.

1. Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria.

bibliografía

<http://ragui-johana.blogspot.com/2012/04/resolucion-de-problemas-mediante-el.html>

<https://www.ecured.cu/Lenguaje_de_Programaci%C3%B3n>

<https://www.ecured.cu/Programa_inform%C3%A1tico>

<https://es.scribd.com/document/237708929/Antecedentes-de-La-Programacion-Orientada-a-Objetos>

<http://codejavu.blogspot.com/2013/05/conceptos-de-programacion-orientada.html>

<https://sites.google.com/site/paradigmasdelais/4-2-el-enfoque-orientado-a-objetos>

<https://andresmtzg.wordpress.com/2012/09/27/tipos-de-datos/>