



“CONCEPTOS PARTE 1, DESARROLLO DE GUIA 1”

FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA.



ASIGNATURA: PROGRAMACION II

Catedrático: Ing. William Virgilio Zamora Girón

Instructor: Br. Sigfrido Ernesto Gómez Guinea.

Alumno: Cabeza Ruiz, Kevin Humberto

Cristhian

Grupo: #2

Ciclo I -2018



1- ¿Qué es la programación orientada a objetos (POO)?

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.

2- ¿Cuáles son los conceptos y nociones básicas de la POO?

- Clases. Una clase se compone por tres partes fundamentales: ☐ Nombre ☐ Atributos ☐ Métodos

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

☐ Estado

☐ Comportamiento

☐ Identidad

- Herencia.

- Encapsulamiento Se representa por 3 niveles :

☐ Público ☐ Protegido ☐ Privado

- Clases Abstractas.

- Interfaces.

- Polimorfismo.

3- Explique de manera clara que es:

- Polimorfismo. :Este tal vez sea uno de los conceptos de la programación orientada a objetos más 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 específica, se usa cuando se trabajan 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.

- Abstracción: Abstraer significa simplificar. A partir de ahora en adelante vamos a trabajar con clases y objetos, que son el elemento principal del lenguaje Java (y cualquier lenguaje orientado a objetos). Al principio se puede hacer un poco difícil de entender, pero con constancia y empeño se entenderá y veréis como la programación orientada a objetos ofrece muchísimas ventajas. - 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 más 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

4- ¿Por qué es preferible usar POO a programación estructurada?

La programación orientada a objetos trata de amoldarse al modo de pensar del hombre y no al de la máquina. El elemento básico de este paradigma no es la función (elemento básico de la programación estructurada), sino un ente denominado objeto. Un objeto es la representación de un concepto para un programa, y contiene toda la información necesaria para abstraer dicho concepto: los datos que describen su estado y las operaciones que pueden modificar dicho estado, y determinan las capacidades del objeto.

5- ¿Qué ventajas posee al usar JAVA para el desarrollo de aplicaciones?

Java ha sido probado, ajustado, ampliado y probado por toda una comunidad de desarrolladores, arquitectos de aplicaciones y entusiastas de Java. Java está diseñado para permitir el desarrollo de aplicaciones portátiles de elevado rendimiento para el más amplio rango de plataformas informáticas posible.

6-Desglose cada uno de los siguientes conceptos en el ámbito de la POO:

-Clase. 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 definir las 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.

-Método. Los métodos son como una Función. Lo que pasa que los métodos están dentro de una clase, es decir, que son funciones que solo utilizan los objetos de esta Clase, en este caso, los coches que creamos.

-Constructor. El constructor siempre tiene el mismo nombre que la Clase y se utiliza para construir objetos de esa Clase. Los constructores no tienen ningún valor de retorno, ni siquiera "void".

-Librería. Una librería en Java se puede entender como un conjunto de clases, que poseen una serie de métodos y atributos. Lo realmente interesante de estas librerías para Java es que facilitan muchas operaciones. De una forma más completa, las librerías en Java nos permiten reutilizar código, es decir que podemos hacer uso de los métodos, clases y atributos que componen la librería evitando así tener que implementar nosotros mismos esas funcionalidades.



-Registro. Ya se ha visto como variables simples pueden almacenar una sola pieza de información y como arreglos pueden almacenar un conjunto de ellas del mismo tipo y al mismo tiempo, estos dos mecanismos pueden manejar una gran variedad de situaciones, pero a menudo se necesita trabajar sobre datos de diversos tipos, en este caso ni variables escalares ni arreglos son adecuados.

-Dato. Los primeros lenguajes de programación no usaban objetos, solo variables. Una variable podríamos decir que es un espacio de la memoria del ordenador a la que asignamos un contenido que puede ser un valor numérico (sólo números, con su valor de cálculo) o de tipo carácter o cadena de caracteres (valor alfanumérico que constará sólo de texto o de texto mezclado con números).

-Valor. Las variables son el elemento esencial de la programación. En ellas será donde almacenemos todos los datos con los que vamos a trabajar. Vamos a aplicar los conceptos vistos anteriormente a la práctica, es decir, veremos cómo se crean y modifican datos en Java.

7-¿Cuáles son las características principales de los siguientes tipos de dato? (Limitantes, posibilidades, usos, etc).

-int. Tipo de dato entero. Su valor mínimo es -2,147,483,648 y el máximo 2,147,483,647 (incluido). Ejemplo: 12, 456, -315...

-double. Dato en coma flotante, para valores decimales. Ejemplo: 2.40 - 3.14 - 10.90.

-String. Dentro de un objeto de la clases String o StringBuffer, Java crea un array de caracteres de una forma similar a como lo hace el lenguaje C++. A este array se accede a través de las funciones miembro de la clase.

-Object. La clase Object, como ya se ha indicado anteriormente, es la clase raíz de todo el árbol de la jerarquía de clases Java, y proporciona un cierto número de métodos de utilidad general que pueden utilizar todos los objetos.

-char. El tipo de dato char es un solo carácter Unicode. Por ejemplo: 'c', 'a', '7', etc.

8-¿Que es la clase math en JAVA?

La clase Math tiene miembros dato y funciones miembro estáticas, El modificador final indica que los valores que guardan no se pueden cambiar, son valores constantes.

9-¿Qué es una base de datos?

Java Database Connectivity, más conocida por sus siglas JDBC, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

“CONCEPTOS PARTE 1, DESARROLLO DE GUIA 2”

1. *¿Qué es un vector dentro de JAVA?*

Según Wikipedia un vector es una zona de almacenamiento continuo, que contiene una serie de elementos del mismo tipo.

Los vectores se utilizan para almacenar información de una manera más ordenada y de esta forma será más fácil y rápido de acceder a la información cuando la necesitemos.

2. *¿Qué es un array dentro de JAVA?*

Un array es un medio de guardar un conjunto de objetos del mismo tipo. Se accede a cada elemento individual del array mediante un número entero denominado índice. 0 es el índice del primer elemento y n-1 es el índice del último elemento, siendo n, la dimensión del array..

3. *¿Cómo se diferencia un vector de un array?*

Un vector es similar a un array, la diferencia estriba en que un vector crece automáticamente cuando alcanza la dimensión inicial máxima. Además, proporciona métodos adicionales para añadir, eliminar elementos, e insertar elementos entre otros dos existentes.

Los arrays unidimensionales se conocen con el nombre de vectores. Los arrays bidimensionales se conocen con el nombre de matrices.

4. *Liste los diferentes tipos de vectores y arrays que pueden utilizarse en JAVA (`int vector [];`)*

Se pueden crear arrays:

☐ De enteros de tipo `int`

☐ De reales de tipo `float`

Vector:

☐ Sentencia `import`.

☐ `String`

☐ `Object`

5. *¿Qué utilidad concreta tiene cada una de estas estructuras de datos dentro de la programación?*

Un array es una estructura de datos que contiene una colección de datos del mismo tipo.



Ejemplos: Temperaturas mínimas de los últimos treinta días. Valor de las acciones de una empresa durante la última semana.

Propiedades de los arrays.

Los arrays se utilizan como contenedores para almacenar datos relacionados (En vez de declarar variables por separado para cada uno de los elementos del array).

Todos los datos incluidos en el array son del mismo tipo. Se pueden crear arrays de enteros de tipo int o de reales de tipo float, pero en un mismo array no se pueden mezclar datos tipo int y datos de tipo float.

6. ¿Qué es el tipo de datos object dentro de JAVA?

La clase Object, como ya se ha indicado anteriormente, es la clase raíz de todo el árbol de la jerarquía de clases Java, y proporciona un cierto número de métodos de utilidad general que pueden utilizar todos los objetos. La lista completa se puede ver en la documentación del API de Java, aquí solamente se tratarán algunos de ellos; por ejemplo, Object proporciona:

- ☐ Un método por el que un objeto se puede comparar con otro objeto
- ☐ Un método para convertir un objeto a una cadena
- ☐ Un método para esperar a que ocurra una determinada condición
- ☐ Un método para notificar a otros objetos que una condición ha cambiado
- ☐ Un método para devolver la clase de un objeto

7. ¿Qué es una matriz y que tipo de datos puede manejar?

Una matriz es una estructura de datos, o más técnicamente, un espacio de memoria que permite almacenar una colección de elementos, todos del mismo tipo. La diferencia con los arreglos está en que, en las matrices, los elementos no están organizados linealmente sino que su organización es bidimensional, es decir, en filas y columnas. Conviene imaginar una matriz como una organización de celdas de memoria, o casillas, en cada una de las cuales se puede guardar un elemento de la colección.

En este formato aparecen en mayúsculas y entre los caracteres < y > los componentes que el programador puede determinar. Así por ejemplo, si se quiere declarar una matriz con nombre mat, de dimensión 15x4 y que pueda almacenar datos de tipo carácter, se debe escribir la siguiente línea.

Mat: matriz[15][4] de carácter

Según el formato anterior, el programador debe bautizar la matriz (ponerle un nombre significativo), debe decir cuál es su dimensión, y también debe decir de qué tipo son los elementos que almacenará