Ministerio de Educación Superior, Ciencia y Tecnología (Mescyt)
Centro de Tecnología y Educación Permanente TEP
Pontificia Universidad Católica Madre y Maestra (PUCMM)
Diplomado en Programación en Lenguaje JAVA

Facilitador: Ing. Eudris Cabrera

Glosario de Términos Relacionados a Java

Java :
Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90.

JDK (Java Development Kit):
Herramienta de desarrollo para Java.

JRE (Java Runtime Environment):
Intérprete de código Java.

Java Virtual Machine:
Es una máquina virtual capaz de ejecutar bytecode Java.

Programación Orientada a Objetos (POO u OOP según sus siglas en inglés):
Es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora.

Clase:
Definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.

Objeto:
Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos).

Método:
Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se

desencadena tras la recepción de un "mensaje".

Propiedad o Atributo:
Contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.

Evento:
Un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto).

Mensaje:
Una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.

Paquete:
Es un contenedor de clases que permite agrupar las distintas partes de un programa cuya funcionalidad tienen elementos comunes.

Excepción:
Una excepción es un cierto tipo de error o una condición anormal que se ha producido durante la ejecución de un programa.

Constructor:
Es un método que lleva el mismo nombre de la clase pero no tiene especificado ningún tipo de retorno.

La creación de un objeto debe realizarse a través de un constructor, también sirven para asignar valores iniciales a las propiedades de un objeto.

Encapsulamiento:
Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema.

Principio de Ocultación:
Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una interfaz a otros objetos que especifica cómo pueden interactuar con los objetos de la clase.

Herencia:
Un objeto puede extender las características de otro objeto. Una clase puede ser "hija" de otra clase y a la vez esa clase puede ser "padre" de otra, al derivar una clase de otra esta toma sus atributos y comportamientos (dependiendo del nivel de acceso).
La herencia se realiza en Java utilizando la palabra reservada **extends.**

Control de Acceso:
Cuando se crea una nueva clase en Java, se puede especificar el nivel de acceso que se quiere para las variables de instancia y los métodos definidos en la clase:

- **public:**
  Cualquier clase desde cualquier lugar puede acceder a las variables y métodos de instancia públicos.
- **protected:**
  Sólo las subclases de la clase y nadie más puede acceder a las variables y métodos de instancia protegidos.
- **private:**
  Las variables y métodos de instancia privados sólo pueden ser accedidos desde dentro de la clase. No son accesibles desde las subclases.

Sobreescritura de Métodos:
Una subclase puede redefinir o sobreescribir un método de su superclase cuando define un método con las mismas características (nombre, número y tipo de argumentos) que el método de la superclase.

Sobrecarga de Métodos:
La sobrecarga de métodos es la creación de varios métodos con el mismo nombre pero con diferentes firmas y definiciones. Java utiliza el número y tipo de argumentos para seleccionar cuál definición de método ejecutar.

Clase Abstracta:
Una clase abstracta es una clase de la cual no se pueden instanciar objetos, de estas clases sólo se pueden heredar otras, en desarrollo se utilizan simplemente para derivar a otras clases más especializadas.

Interfaz:
Una interfaz es una clase únicamente para implementación en otras, en una interfaz se indican los prototipos de los métodos que debería tener un objeto pero el cuerpo del mismo debe ser escrito en las clases que la implementan.

En los lenguajes que lo implementan se utiliza la palabra reservada interface para indicar una interfaz e **implements** para su implementación.

Clase final:
Las clases final son clases que no se pueden heredar, es la última parte de la estructura de clases, en los lenguajes que las implementan se utiliza la palabra reservada final.

Polimorfismo:
El polimorfismo se refiere a la posibilidad de definir múltiples clases con funcionalidad diferente, pero con métodos o propiedades denominados de forma idéntica, que pueden utilizarse de manera intercambiable mediante código cliente en tiempo de ejecución.

Método Nativo:
Java proporciona un mecanismo para la llamada a funciones C y C++ desde nuestro código fuente Java. Para definir métodos como funciones C o C++ se utiliza la palabra clave native.

Arreglo:
Un arreglo es un objeto contenedor que almacena un número fijo de valores de un solo tipo. La longitud de un arreglo se establece cuando se crea el arreglo.

Ciclo "for" mejorado:
El ciclo for mejorado es mucha veces llamado ciclo foreach, porque es usado para procesar cada elemento en un arreglo o colección.

Colección:
Una colección es todo aquello que se puede recorrer (o **“iterar”**) y de lo que se puede saber el tamaño. En java se representan mediante la interfaz **Collection.**

List:
Un List, o simplemente lista, es una **Collection**.
La diferencia que tiene una lista con una Collection es que la lista mantiene un orden arbitrario de los elementos y permite acceder a los elementos por orden.

ArrayList:
Es una implementación de **List**, la ventaja de **ArrayList** sobre un **array** común es que

es expandible, es decir que crece a medida que se le añaden elementos (mientras que el tamaño de un array es fijo desde su creación).

Set
Es una collection, un **Set** agrega una sola restricción: No puede haber duplicados.

HashSet
Existen varias implementaciones de **Set**. La más comúnmente usada es **HashSet.**

Map
Un Map es un conjunto de valores, con el detalle de que cada uno de estos valores tiene un objeto extra asociado. A los primeros se los llama **“claves”** o **“keys”**, ya que nos permiten acceder a los segundos.

Eudris Cabrera Rodriguez

Ingeniero Telemático
Consultor / Desarrollador Informático
LinkedIn: http://www.linkedin.com/in/eudriscabrera
Revisado Marzo 2016, Santiago de los Caballeros, R. D.