

ED

Departamento de Computación
ITAM

1 Conceptos de Programación Orientada a Objetos

Tipos Primitivos . int, float, double, char.

Clase En Java todos son clases menos los tipos primitivos. Una clase es una manera de definir un tipo propio.

Una clase define las características abstractas de un objeto, incluyendo sus características (atributos) y comportamiento (métodos). Una clase es el plano de construcción de los objetos.

Las clases proveen modularidad y estructura a un programa

Objeto Es una instanciación de una clase. Armando manzanero es una instanciación de la clase persona. Las clases solitas no se pueden usar. Hay que instanciarlas. Los objetos tienen estado, valores a sus atributos, las clases son patrones, machotes

Encapsulamiento: El proceso de compartimentalizar los elementos de una abstracción que constituyen su estructura y comportamiento. El encapsulamiento sirve para separar la interface de una abstracción de su implementación.

El encapsulamiento esconde los detalles de una clase

Abstracción Una abstracción es una simplificación de la realidad mediante la omisión de detalles sin importancia

Interface La interface es el conjunto de operaciones que pueden ser utilizados por un cliente de la clase. Una clase puede implementar tantas interfaces como quiera (a diferencia de la herencia en donde las clases solo puede tener un progenitor)

Herencia La herencia es el proceso mediante el cual se genera una subclase. Una subclase es una versión especializada de una clase e.g. músico puede ser una subclase de persona. Las subclases heredan atributos y métodos de la clase "madre" y pueden introducir los propios. No se heredan atributos y métodos privados.

Una subclase siempre puede usarse en donde quiera que se use la superclase

En Java sólo se permite heredar de una clase. Todas las clases heredan de la clase Object de manera implícita. Todas tienen a su disposición métodos de la clase Object como lo son toString() e equals(). Sin embargo estas se suelen sobre escribir (equals ve si dos objetos son aliases: si es que son el mismo con diferente nombre).

Overload o Sobrecarga: Es tener métodos con el mismo nombre pero diferente número o tipo en los parámetros. Constructores vacíos o con parámetros. Un método de nombre public void Ordena(int Arreglo[]) y otro public void Ordena(int Arreglo[][]) dentro de una misma clase

Override o sobre-escribir: Una subclase puede decidir redefinir la funcionalidad de algunos métodos heredados. Por ejemplo, si la clase Perro tiene el método ladra que produce un sonido estandar, la subclase Poodle que hereda de Perro puede sobre escribir el método ladra para cambiar el tono del

ladrido a uno chillón.

Polimorfismo: Muchas formas. Un método que define como parámetro instancias de una clase específica, puede recibir instancias de cualquier subclase. Por ejemplo un método diseñado para calcular los impuestos que debe pagar una persona y define como parámetro a una instancia de la clase persona, puede recibir a un músico siempre y cuando la clase músico herede de persona. Lo interesante es que se escoge automáticamente el método apropiado al objeto para pagar impuestos, pero el texto del programa es el mismo. Con el mismo texto se logran diferentes comportamientos (polimórfico). (Esto se logra con dynamic binding: determinar cuál es el método apropiado a la hora de la ejecución). Una interface se puede usar para lograr polimorfismo; por ejemplo, puedo tener un arreglo de elementos Comparables

Data type y Abstract Data Type Un tipo de datos es un conjunto de valores y las operaciones sobre éstos. Ej. int, double (estos son datos primitivos y los únicos que no son objetos en JAVA). Una clase es un tipo de datos definido por el programador.

Los tipo de datos abstractos establecen las propiedades que debe tener un tipo de datos sin implementarlo.

Composition vs Aggregation Si A esta compuesto por elementos tipo B destruir A implica la destrucción de B. Destruir una biblioteca implica la destrucción de sus libros. Si A es un agregado de elementos tipo B, destruir A no implica la destrucción de B. Una orden de compra es un agregado de productos. Una película es un agregado de actores.

Static Los métodos y atributos estáticos existen por clase no por objeto. Hay una copia para todas las instancias, por lo mismo no se pueden sobre-escribir. No se requiere instanciar una clase para utilizar su miembros estáticos.

Visibilidad Public, Private y Protected:

- Public visible a todas las clases
- Private visible solo a la clase a la que pertenecen
- Protected visible a la clase y sus subclases
- Sin Modificador visibles a todas las clases dentro del paquete

Variables Genéricas Son una clase especial de variable que te ahorra especificar el tipo exacto de una variable en el código y te permite usar el mismo código para variables de diferente tipo siempre y cuando cumplan con las condiciones que se le imponen a la variable genérica; por ejemplo, que sea comparable. Así puedes escribir código para ordenar un arreglo sin especificar de qué es el arreglo más allá de decir que es de objetos que se pueden comparar entre si.

Clases Abstractas : Son aquellas que tienen uno o más métodos abstractos. Un método abstracto no tiene implementado el código asociado al comportamiento del método. Sólo tienen el nombre y parámetros. No se pueden instanciar (pues los métodos abstractos no tienen código) pero pueden ser super clase de otra que para no ser a su vez abstracta debe implantar los métodos abstractos heredados. A diferencia de las interfaces, una clase abstracta puede tener implementaciones en algunos de los métodos declarados.