

POO

Pilares

- Abstracción (Mecanismo de descripción)
- Encapsulamiento
- Herencia
- Polimorfismo

Abstracción

Mecanismo para describir el entorno de un problema a solucionar

El mundo se compone de **objetos** que se interrelacionan para solucionar un problema

Un objeto es una “idea” (Teoría de las formas o teoría de las ideas – Sócrates, Platón, Aristóteles)

Un objeto se compone de :

- Características estructurales (El mecanismo que utiliza un objeto para almacenar su “estado”. El estado guarda información del objeto (atributos – en programación son variables que almacena un dato)

Definiciones importantes

- **Dato:** Unidad mínima para armar información. Un caracter, un número, un símbolo desde el punto de vista minimalista. Pragmático puede ser una palabra.
- **Información:** Una colección de datos, organizada con sentido lógico. Ejemplo: Un libro.
- **Conocimiento:** “Lo que un usuario aprendió al tener acceso a la información”

Los atributos

- Característica estructural de un objeto
- Almacenan los estados del objeto
- Un **estado** es un dato, asociado a un *tipo de dato*. En programación un atributo es una variable del objeto.
- **Variable**: Posición de memoria asociada a un nombre que guarda un dato.

Tipos de dato

- Hay dos tipos de dato: Primitivos (TPD), abstractos (TAD).
- Cuando se almacena un dato en la memoria de una máquina: conservando bits en ese espacio de memoria.
- Los tipos primitivos son los básicos que se almacenan en la RAM, dependiendo de la arquitectura de hardware de la máquina. Actualmente 64 bits.
- Dependiendo de la agrupación de bits y el **sentido** que se le quiera dar a esa agrupación se conforman los tipos TPD.

TPD

Número de bits	Como se interpretará	Nombre* (java)
8 (BYTE)	Número	byte
8	Símbolo (ASCII)	char
16 (WORD)	Número	short
16	Carácter (Unicode)	char
32	Número entero	int
32	Número Real (Coma flotante)	float
64	Número entero	long
64	Número Real (Coma flotante)	double

TAD

- Es un tipo que se forma agrupando los inicialmente TPD.
- Si el agrupamiento es homogéneo, se conocen como arreglos o arrays. Homogéneo, porque todos las “celdas o posiciones” del arreglo son del mismo tipo de dato.
- Si el agrupamiento es heterogéneo, son los registros o estructuras (en C struct), clases (en POO). Son heterogéneos porque sus componentes pueden ser de diferente tipo de dato.

Los métodos

- Características de comportamiento de un objeto
- Los comportamientos cambian los estados de los atributos.
- En la práctica de programación, estos son las funciones, procedimientos, contrato de la clase, subrutinas.
- Pueden tener entradas (variables llamadas parámetros o argumentos), de diferente tipo de dato.
- Pueden generar un valor de salida (funciones), o no(procedimientos): tipo void.

Encapsulamiento

- Significa encerrar, rodear... encapsular.
- Cuando se hace abstracción, dependiendo del “mundo” (mi objetivo de solución), pues no todo es necesario modelar.
- Se elimina lo que no se utilice en MI modelo, ya que no es necesario en la interrelación de los objetos en mi solución.
- Unidad de encapsulamiento: clase (donde se agrupa las características de los objetos).
- Diferentes metodologías para abstraer y luego encapsular (ingeniería de requerimientos).

encapsulamiento

- Como se va a estructurar las características de los objetos e inclusive el mismo objeto:
- Cada lenguaje OO tiene sus mecanismos para aplicar esta “estructuración”.
- 2 formas de aplicar encapsulamiento: visibilidad y alcance.

Encapsulamiento - visibilidad

- Se refiere a que se puede acceder o no de las características.
- La gran mayoría de los lenguajes utilizan palabras reservadas para aplicarla.

En el caso de java son:

private: No se permite el acceso a la característica del objeto

public: Se puede acceder a las características del objeto

protected: No se permite el acceso a la característica del objeto
(modificación dependiendo de la herencia)

friendly (no es palabra):

Encapsulamiento - alcance

- Modifica la forma en que se van a comportar las características de los objetos.
- Cada lenguaje aplica sus propios mecanismos para lograr estas modificaciones (java lo hace con palabras reservadas, C# en algunos casos usan sentencias encima a lo que se quiere modificar)
- En java: static, final, abstract, volatile, synchronized, transient
- También existen encapsulamientos de alcance propios del lenguaje: en java native.

Encapsulamiento - alcance

- `static` (aplica a: atributos, métodos): El estado o comportamiento es universal.
- `final` (aplica a: clase, atributos, métodos): *const* en otros lenguajes. “No se puede aplicar” en una situación donde lo normal aplica.
- `abstract` (aplica a: clase, métodos): es lo contrario a `final`. “Se tiene que aplicar”
- `volatile` (aplica a: atributos): en hilos, el valor siempre es reescrito o se mantiene actualizado.
- `synchronized` (aplica a: métodos): en hilos, solo uno ejecuta ese comportamiento y bloque el acceso a otros hilos.
- `transient` (aplica a: atributos): en serialización (convertir a a bytes), no se envía por el flujo de bytes.

Herencia

- Crea clases a partir de clases existentes
- Tipos: simple (se hereda de un solo padre), múltiple (se hereda de varios padres)
- Formas de heredar: en gral lo privado NO se hereda, public (protected public), protected (todo es protected), private(public->protected, protected->private)
- En la herencia múltiple se puede presentar herencia repetida.

Polimorfismo

- Poli = mucho, morfos = formas
- Formas de hacer polimorfismo: Interno, externo
- Polimorfismo interno: un método puede tener varios comportamientos (Sobrecarga).
- Polimorfismo externo: un método puede cambiar su comportamiento al hacer herencia (Sobreescritura)
- Variables polimórficas (ejemplo de aplicación en el Principio de Sustitución de Liskov)