Abstracción: expresar las características esenciales de un objeto, así como su comportamiento

¿Cuáles son sus características?
¿Qué acciones puede realizar?

Usuario

Características

- ID
- Nombre
- Fecha de Nacimiento
- Correo
- Fecha de Registro ...

Acciones

- Revisar cuenta
- Depositar a cuenta
- Retirar de cuenta
- Transferir a tercero ...

Clase

Una plantilla que define las características y acciones de los objetos de un cierto tipo.

- Las características se representan por medio de propiedades.
- Las acciones por medio de métodos.

Objeto

Un objeto es una entidad concreta basada en una clase.

Es una instancia de una clase.

Es una variable del tipo de la clase.

Constructores

Los constructores son métodos de una clase que nos permiten crear instancias (crear objetos) de la clase.

- Mismo nombre de la clase.
- No definen un valor de retorno.
- No hay un límite para la cantidad de constructores.

Encapsulamiento

Se refiere al ocultamiento del estado (el conjunto de propiedades) de una clase, para que no sea modificado directamente.

Sólo se puede acceder (obtener o modificar) a las propiedades a través de métodos.

De esta manera se protege el estado de la clase.

Modificadores de acceso

Nivel de acceso	Descripción
public	Sin restricción. Cualquier elemento tiene acceso.
private	Sólo tienen acceso los elementos de la clase.
protected	Los elementos de la clase y de clases hijas tienen acceso.
internal	Cualquier elemento dentro del ensamblado (proyecto) tiene acceso; '**public**' a nivel del ensamblado.

Sobrecarga de métodos

La sobrecarga significa que, en una clase, existen dos o más métodos con el mismo nombre.

La diferencia entre ellos es la cantidad y/o el tipo de parámetros.

Clases estáticas

Una clase estática no se puede instanciar; no se pueden crear objetos de ella.

Para acceder a los miembros de una clase estática se debe referenciar a la clase misma.

Herencia

La herencia es un mecanismo mediante el cuál se puede construir una jerarquía de clases.

En esta jerarquía, se definen clases hijas (o subclases) que heredan las propiedades y métodos de su clase padre.

- Usuario
  - Cliente
    - Personal
    - Institucional
  - Empleado
    - Gerente
  - Administrador
    - TI

Herencia

Al heredar las propiedades y métodos de los padres, se posibilita la reutilización de código.

A su vez, cada clase hija define propiedades y métodos propios, que la diferencian del padre.

Conforme se baja en la jerarquía, se desarrollan clases cada vez más específicas.

Sobre escritura de métodos

La sobre escritura ocurre cuando una o más clases hijas implementan un método de la clase padre.

El método es el mismo (mismo nombre y parámetros), pero tiene una implementación distinta en las clases hijas.

Sobre escritura de métodos

La sobre escritura es una aplicación del polimorfismo (muchas formas).

Es la capacidad de los objetos de implementar de manera diferente a un mismo método.

Se produce el mismo efecto u objetivo básico, pero la implementación es distinta.

Clases abstractas

Una clase abstracta no se puede instanciar; sólo sirve como clase base (padre).

Su propósito es proveer una definición común que múltiples clases derivadas pueden compartir.

- Puede definir elementos abstractos.
- Estos elementos deben ser implementados por las clases hijas.

Interfaces

Una interfaz tampoco se puede instanciar.

Su propósito también es proveer una definición común que múltiples clases pueden compartir.

- Todos sus elementos son abstractos.
- Una clase que implementa a una interfaz debe implementar a todos sus elementos.

Una clase puede heredar de sólo una clase, sea abstracta o no.

Sin embargo, una clase puede implementar múltiples interfaces.