

Programación orientada a objetos

El paradigma orientado a objetos permite organizar el Software como una colección de objetos que constan tanto de datos como de comportamiento.

Clase: Proporciona una interfaz bien definida, representa un concepto claro, ser completa y estar bien documentada, código robusto, no debe fallar.

Estructura de una clase:

Estados (datos): Valores que tiene el objeto

Métodos (comportamientos): formas en que el objeto puede interactuar

Objeto: Instancia de una clase, puede haber muchos objetos de una misma clase, no puede haber 2 clases iguales

Encapsulamiento: Ocultar la mecánica del objeto
permite ocultar la implementación real del objeto

Interfaz: No se conoce la implementación real de las funciones, con el nombre se entiende lo que hace

Encapsulamiento termino utilizado para describir la forma en que la interfaz se separa de la implementación

Herencia: Permite derivar una clase de otra clase

Polimorfismo: Formas múltiples en programación orientada a objetos, se refieren a múltiples formas del mismo método donde se puede usar

over-Riding: ocurre en subclases, declara métodos con el mismo nombre,

over-Load: utiliza el mismo nombre de método

Clase abstracta: Clase incompleta, descripción de las operaciones, pero falta la implementación, no se puede crear una instancia de clase abstracta, solo se puede usar a través de herencia

Curso de Java

Variable: permite almacenar información, tiene un identificador, el valor que almacena puede modificarse con el tiempo

no se recomienda utilizar acentos para nombre de variables, así como no usar palabras reservadas

+ permite sumar y concatenar cadenas

Var: declarar cualquier tipo de dato

Si primero encuentra una cadena concatena lo demás

Parenthesis: toma máxima prioridad en una operación en la evaluación

\n: Salto de línea

\t: tabulador, es un espacio

\b: Retroceso, retrocede una posición según la cantidad de caracteres de retroceso

Scanner: Capturar información que escribe el usuario

Console: variable para solicitar información del usuario por medio del método `NextLine`