Definiciones de algunos conceptos basicos

Objetos: Un objeto en Java es una instancia concreta de una clase, que contiene datos y funciones que trabajan juntos para realizar una tarea o representar una entidad en un programa.

Atributos: Los atributos son variables definidas en una clase que contienen datos asociados a un objeto de esa clase. Son una parte fundamental de la descripción de un objeto y permiten que el objeto tenga un estado y comportamiento específicos.

Método: Un método en Java es una colección de instrucciones o bloques de código que se ejecutan cuando se llama a ese método. Los métodos son parte de una clase y se utilizan para definir el comportamiento de un objeto.

Clase: Una clase en Java es una plantilla o modelo que describe las propiedades y comportamientos de un conjunto de objetos relacionados. Una clase se utiliza para definir objetos que se pueden crear y utilizar en un programa.

Instancia: Una instancia en Java es un objeto creado a partir de una clase específica. Cada instancia tiene acceso a los atributos y métodos definidos en la clase y tiene sus propios valores de atributos independientes de otras instancias.

Herencia: La herencia en Java es un mecanismo que permite que una clase adquiera las propiedades de otra clase. La clase que hereda se llama subclase, y la clase de la cual se heredan las propiedades se llama superclase. La herencia es una técnica de programación orientada a objetos que permite la reutilización del código y la organización jerárquica de las clases.

Polimorfismo: El polimorfismo en Java se refiere a la capacidad de objetos de diferentes clases de comportarse de manera similar o intercambiable en un programa. Se utiliza para mejorar la reutilización del código y la flexibilidad del diseño y se puede implementar a través de la sobrecarga de métodos y la herencia.

Existen dos tipos principales de polimorfismo en Java:

1. Polimorfismo de sobrecarga de métodos: Se refiere a la capacidad de una clase para tener múltiples métodos con el mismo nombre, pero diferentes parámetros. Esto permite que un método tenga diferentes comportamientos dependiendo de los parámetros que se le pasen.
2. Polimorfismo de herencia: Se refiere a la capacidad de una clase para heredar los métodos y atributos de su clase padre. Esto permite que las subclases puedan reutilizar el código y comportarse como su clase padre, pero también pueden agregar nuevos métodos y atributos o modificar los ya existentes.

Interfaz: una interfaz en Java es un contrato que especifica un conjunto de métodos y constantes que deben ser implementados por cualquier clase que implemente dicha interfaz. Las interfaces se utilizan para lograr la abstracción y la separación de preocupaciones en el diseño de un programa, y se definen utilizando la palabra clave "interface".

Propiedad: el término "propiedad" no se utiliza comúnmente. En su lugar, se utilizan los términos "atributo" o "campo" para referirse a las variables de una clase que almacenan los datos asociados a un objeto. Los atributos de una clase se definen dentro de la clase y se acceden y modifican mediante métodos especiales llamados "getters" y "setters".

Ejemplo para poder trabajar con fechas y horas:

package test;

import java.time.\*;

import java.time.format.DateTimeFormatter;

public class TestFechas {

public static void main(String[] args) {

LocalDate facha; // solo la fecha

LocalTime hora; // solo la hora

LocalDateTime fechaYHora; // fecha y hora

// fechaYHora = LocalDateTime.now();

// imprimo fecha y hora del momento en el que se ejecuta

System.out.println("fechaYHora = " + fechaYHora);

// Dar formato deseado

System.out.println("");

DateTimeFormatter formato;

formato = DateTimeFormatter.ofPattern("dd/MM/yyyy HH:mm");

System.out.println("Fecha y hora: " + fechaYHora.format(formato));

//Setenado de ante mano una hora

fechaYHora = LocalDateTime.of(2023, 10, 18, 00, 00);

System.out.println("Cumple de la loca <3 = " + fechaYHora.format(formato));

}

}

Foto de apache netBeans:

