1. Introducción Teórica:

* Polimorfismo:

El polimorfismo es un concepto de la programación orientada a objetos que permite que un objeto pueda tomar diferentes formas o comportamientos según el contexto en el que se utiliza. Esto se logra mediante la capacidad de los objetos de una clase base para ser tratados como objetos de sus clases derivadas. El polimorfismo facilita el diseño de software modular y extensible.

* Herencia:

La herencia es un principio de la programación orientada a objetos que permite que una clase (llamada subclase o clase derivada) herede los atributos y métodos de otra clase (llamada superclase o clase base). La subclase puede extender el comportamiento de la superclase al agregar nuevos métodos o sobrescribir los métodos existentes.

* Sobrecarga de Métodos:

La sobrecarga de métodos es un concepto que permite definir múltiples métodos con el mismo nombre en una clase, pero con diferentes parámetros. Esto permite que los métodos realicen acciones similares pero con diferentes tipos de datos o cantidades de parámetros.

* Polimorfismo Paramétrico (Generics en Java):

El polimorfismo paramétrico, también conocido como generics en Java, es un mecanismo que permite escribir clases y métodos que funcionan con un tipo de datos genérico. Esto proporciona flexibilidad y reutilización del código al permitir que los tipos de datos sean especificados como parámetros.

* Polimorfismo de Inclusión:

La habilidad para redefinir por completo el método de una superclase en una subclase es lo que se conoce como polimorfismo de inclusión (o redefinición).En él, una subclase define un método que existe en una superclase con una lista de argumentos (si se define otra lista de argumentos, estaríamos haciendo sobrecarga y no redefinición).Un ejemplo muy básico en donde la clase Bishop sobreescribe el método move. Esto es el polimorfismo de inclusión.

2. Ejemplos de Código:

* Herencia:
* Polimorfismo de Inclusión:
* Sobrecarga (Overloading):
* Polimorfismo Paramétrico (Generics en Java):
* Polimorfismo (en general):

<https://github.com/Franciscorh19/PrograFrancisco>

3. Análisis Comparativo:

* Diferencias entre Polimorfismo y Sobrecarga de Métodos:

El polimorfismo permite que un objeto pueda tomar diferentes formas o comportamientos según el contexto en el que se utiliza, mientras que la sobrecarga de métodos permite definir varios métodos con el mismo nombre pero diferentes parámetros en una misma clase.

* Diferenciar entre Sobrecarga (Overloading) y Redefinición (Overriding) de Métodos:

Overloading significa que un mismo método tiene diferentes firmas mientras que Overriding es el mismo método, por tanto misma firma, al que diferentes clases conectan a través de la herencia. En algunos textos encontramos otra explicación en donde se resume la sobrecarga como un ejemplo de polimorfismo en tiempo de compilación y la sobreescritura como un ejemplo de polimorfismo en tiempo de ejecución.

Respuestas a las preguntas adicionales:

* ¿Qué es el término firma?

La firma de un método consiste en su nombre y la lista de tipos de sus parámetros. Dos métodos con la misma firma pero en diferentes clases no están relacionados y no constituyen sobrecarga.

* ¿Diferencias entre los términos Overloading y Overriding?

Overloading (sobrecarga) se refiere a definir múltiples métodos con el mismo nombre pero diferentes parámetros en una misma clase, mientras que Overriding (redefinición) se refiere a proporcionar una implementación específica para un método que ya está definido en la superclase en una subclase.

* ¿Se pueden sobrecargar métodos estáticos?

Sí, se pueden sobrecargar métodos estáticos en Java. Esto significa que se pueden definir varios métodos estáticos con el mismo nombre pero diferentes listas de parámetros.

* ¿Es posible sobrecargar la clase main() en Java?

No, la clase main() en Java no puede ser sobrecargada. La firma del método main() debe ser exactamente public static void main(String[] args).