**Universidad de Las Américas**

Facultad de Ingenierías y Ciencias Agropecuarias

*Ingeniería de Softwate*

**Informe de laboratorio**

1. **DATOS DEL ALUMNO:**

Dylan Clerque

1. **TEMA DE LA PRÁCTICA:**

El tema de la práctica es el estudio y comprensión del polimorfismo en la programación orientada a objetos.

1. **OBJETIVO DE LA PRÁCITCA**

El objetivo principal de la práctica es comprender el concepto de polimorfismo y su aplicación en el desarrollo de software para la creación de programas mejor estructurados.

1. **OBJETIVOS ESPECÍFICOS**

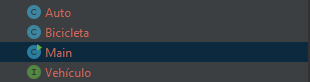
* Definir el concepto de polimorfismo y sus tipos.
* Comprender cómo se implementa el polimorfismo en diferentes lenguajes de programación.
* Explorar ejemplos prácticos de polimorfismo y su utilidad en el diseño de sistemas.

1. **MATERIALES Y MÉTODOS**

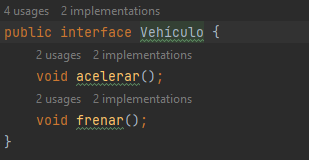
* IntelIj
* Ejemplos de Polimorfismo
* Método: Polimorfismo

1. **DESARROLLO DE LA PRÁCTICA Y RESULTADOS**

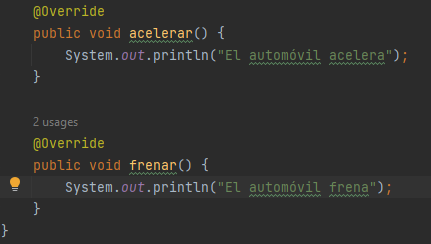
Se creo una interfaz y 2 clases para la implementación del polimorfismo



Interfaz Vehículo:



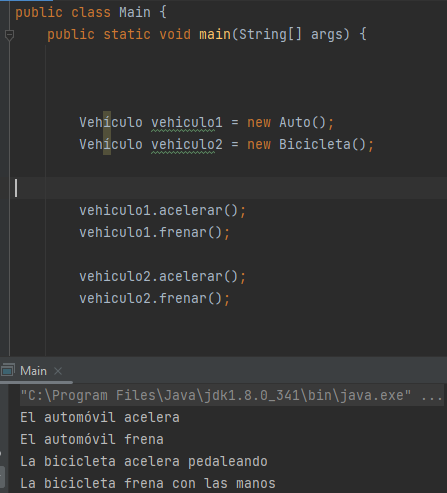
Clase Auto:



Clase Bicicleta:



Main y resultado:



1. **OPINIÓN PERSONAL**

* El polimorfismo permite escribir código más flexible y modular al tratar objetos de diferentes clases de manera uniforme. Esto facilita la creación de sistemas escalables y fáciles de mantener, ya que los cambios en una parte del código no requieren modificaciones extensivas en otras partes.
* Al utilizar el polimorfismo, es posible reutilizar el mismo código para manipular diferentes tipos de objetos, lo que reduce la duplicación de código y promueve prácticas de programación más eficientes. Esto conduce a un desarrollo de software más rápido y menos propenso a errores.
* El polimorfismo permite abstraer la complejidad de las relaciones entre objetos al ofrecer una interfaz común para interactuar con ellos. Esto simplifica el diseño y la comprensión del sistema, ya que los detalles de implementación se pueden ocultar detrás de interfaces bien definidas.
* Al utilizar el polimorfismo, es más fácil agregar nuevas funcionalidades al sistema sin afectar el código existente. Esto se debe a que las nuevas clases pueden implementar las mismas interfaces o heredar de las clases base existentes, lo que facilita la incorporación de cambios y la adaptación del sistema a nuevos requisitos.

1. **ANEXOS (Ai se requiere)**

[**https://github.com/DDCT2003/Polimorfismo**](https://github.com/DDCT2003/Polimorfismo)

**9. BIBLIOGRAFÍA (Si se requiere)**

¿Qué es el polimorfismo en Java? (2023, 16 mayo). *HubSpot*. <https://blog.hubspot.es/website/polimorfismo-java#:~:text=El%20polimorfismo%20hace%20referencia%20a,nombre%20para%20ejecutar%20funciones%20diferentes>.

*Polimorfismo en Java: Programación orientada a objetos*. (2023, 25 agosto). IfgeekthenNTTdata. https://ifgeekthen.nttdata.com/es/polimorfismo-en-java-programaci%C3%B3n-orientada-objetos

Fernández, E. C. (2024, 26 febrero). *Guía para comprender el polimorfismo en Java*. Tokio School. https://www.tokioschool.com/noticias/polimorfismo-java/