

Proyecto Pokémon: Uso de Polimorfismo, Clases Abstractas e Interfaces

Introducción

El proyecto Pokémon simula un sistema en el que entrenadores pueden capturar y entrenar Pokémon. Se aprovechan conceptos de programación orientada a objetos, incluyendo polimorfismo, clases abstractas e interfaces para crear una estructura flexible y extensible.

Objetivos del Proyecto

1. Demostrar el uso de clases abstractas para definir comportamientos comunes.
2. Utilizar polimorfismo para manejar diferentes tipos de Pokémon de manera uniforme.
3. Implementar interfaces para definir comportamientos adicionales.

Clases y Estructuras del Proyecto

1. Clase Abstracta Pokémon

La clase abstracta Pokémon define los comportamientos y atributos comunes a todos los Pokémon.

2. Clase Pikachu

La clase Pikachu extiende la clase abstracta Pokémon y proporciona implementaciones específicas para los métodos abstractos.

3. Interfaz Capturable

La interfaz Capturable define un comportamiento adicional para los Pokémon que pueden ser capturados.

4. Clase Entrenador

La clase Entrenador representa a un entrenador de Pokémon y contiene una lista de Pokémon.

Uso del Polimorfismo, Clases Abstractas e Interfaces

El polimorfismo se demuestra a través de la capacidad de tratar a diferentes tipos de Pokémon de manera uniforme usando la clase abstracta Pokémon. Por ejemplo, un

Entrenador puede tener una lista de objetos Pokémon y llamar a los métodos atacar () y defender() sin preocuparse del tipo específico de Pokémon.

Las clases abstractas se utilizan para definir los métodos y atributos comunes que deben ser implementados por todas las subclases de Pokémon. Esto garantiza que todas las subclases tengan los métodos necesarios, como atacar () y defender(), mientras permite que cada subclase proporcione su propia implementación específica.

Las interfaces, como Capturable, se utilizan para definir comportamientos adicionales que algunas, pero no todas, las subclases pueden implementar. En este caso, la interfaz Capturable define un método capturar () que puede ser implementado por cualquier clase de Pokémon que pueda ser capturado.

Ejemplo de Uso

En una clase principal, se podría instanciar un Entrenador y agregarle varios Pokémon. Luego, se puede iterar sobre el equipo de Pokémon del entrenador, llamando a métodos como atacar(), defender() y capturar(), demostrando así el uso del polimorfismo.