You

4 principles in oop

Copilot

Certainly! In object-oriented programming (OOP), there are **four fundamental principles** that guide software design. Let's explore each of them:

1. **Abstraction**:
    - Abstraction involves **hiding implementation details** and exposing only essential features to the user.
    - It allows you to create **reusable, simple, and easily changeable code** by focusing on high-level concepts.
    - For example, think of a function that hits an API. Instead of dealing with specific details for each API type, you can abstract it by passing a URL and an HTTP method, making your code more maintainable and reusable.
2. **Encapsulation**:
    - Encapsulation is about **bundling data (attributes) and methods (functions) together** into a single unit (an object).
    - It helps protect data from unauthorized access by providing **access control** (public, private, protected).
    - By encapsulating related functionality, you create **modular and organized code**.
3. **Inheritance**:
    - Inheritance allows you to create a new class (a child or derived class) based on an existing class (a parent or base class).
    - The child class **inherits properties and methods** from the parent class.
    - It promotes **code reuse** and establishes a **hierarchical relationship** between classes.
4. **Polymorphism**:
    - Polymorphism means that objects can take on **different forms**.
    - It allows you to use a single interface (method or function) to handle different data types or classes.
    - For example, you can call the same method on different objects, and each object behaves differently based on its specific implementation.

Remember, these principles help create **clean, maintainable, and efficient** object-oriented code. 🚀