Task

Alert System of accounting office

Team 5

Members:

- Paulo Ramos
- Josue Rojas

Core OOP Concepts for the Project

1. Class
   A blueprint that defines the properties and behaviors (attributes and methods) of objects.
   Example: Task, User, Manager, Assistant, Invoice, Alert.

2. Object
   An instance of a class that represents a specific entity in the system.
   Example: A specific task like “Prepare Tax Report” or an assistant named “John Doe”.

3. Encapsulation
   The concept of hiding internal data and only exposing necessary functionality through methods (getters and setters).
   Example: Task status or deadline are private attributes accessed only through getStatus() or setStatus().

4. Inheritance
   Allows one class to acquire the properties and behaviors of another.
   Example: Manager and Assistant can inherit from a general User class.

5. Polymorphism
   Allows methods to behave differently based on the object that calls them.
   Example: generateReport() could be implemented differently for Manager and Assistant.

6. Abstraction
   The process of simplifying complex systems by modeling classes that represent real-world entities.
   Example: You don’t need to show how the alert is generated internally—just provide a sendAlert() method.

7. Association
   Defines a relationship between classes.

Example: A Manager assigns Tasks to Assistants.

8. Aggregation
   A “has-a” relationship where one class contains another but can exist independently.
   Example: A Manager has many Tasks, but tasks can exist even if the manager is removed.

9. Composition
   A stronger “has-a” relationship where the contained object cannot exist without the container.
   Example: An Invoice cannot exist without its related Task.

10. Interface (or Abstract Class)
    Defines a contract for classes to implement specific methods.
    Example: An interface Notifiable with a method sendNotification() that both Manager and Assistant implement differently.

Additional Useful Concepts

- Constructor: Method used to initialize objects when they are created.
- Method Overloading: Two methods with the same name but different parameters.
- Method Overriding: Redefining a parent class method in a subclass.
- Static Members: Shared attributes or methods among all objects of a class.
- Exception Handling: Used to manage runtime errors, like when generating invoices or missing deadlines.