1st design pattern: *Abstract Factory*

In our scenario, the abstract factory pattern can be used to provide an interface for creating families of related objects, in this case, different types of employees

● Abstract Factory:

 - The `EmployeeFactory` interface or abstract class serves as the abstract factory. It declares methods for creating various types of employees without specifying their concrete classes.

● Concrete Factories:

 - Implementations of the `EmployeeFactory` interface would include `FinanceWorkerFactory`, `HRWorkerFactory`, `DentistFactory`, and `ReceptionistFactory`. Each concrete factory is responsible for creating a specific type of employee.

● Products (Employees):

 - Each type of employee (`FinanceWorker`, `HRWorker`, `Dentist`, `Receptionist`) implements the `Employee` interface. This interface defines methods such as getWorkingHours(), getAge(), getPhoneNumber(), getAddress(), getName(), getEmail(), and getID(`.

 - Each concrete factory (`FinanceWorkerFactory`, `HRWorkerFactory`,`DentistFactory`, and `ReceptionistFactory`) is responsible for instantiating the corresponding type of employee.

This approach promotes loose coupling and flexibility, allowing for easier extension and maintenance of the system, especially as new types of employees may be added in the future.
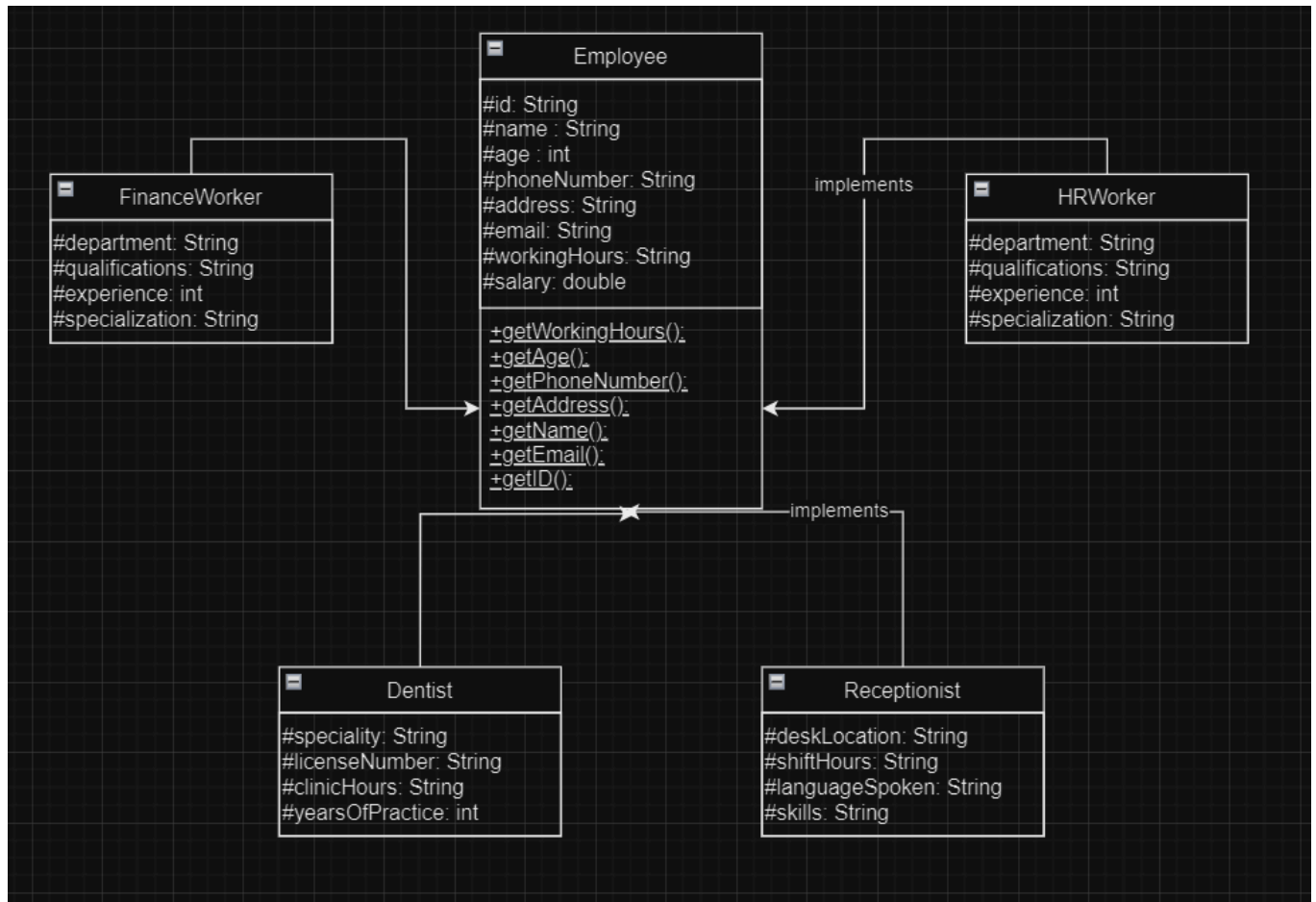
Design Pattern - Observer

In a dental clinic management system, a patient card contains patient information and a treatment plan for each patient. This patient card can be edited by a dentist. When the patient card is updated, various components (NotificationService,PatientCardView), such as notification services,UI need to react to these changes.

Components

1. PatientCard:
   - Holds patient data and treatment plans.
   - Notifies observers when data changes.
2. Observer Interface:
   - Defines the update method for observers.
3. Concrete Observers:
   - PatientCardView: Updates the UI when the patient card changes.

- NotificationService: Sends notifications when the patient card is updated.
4. Dentist:
   - Updates the patient card, triggering notifications to observers.