multiple branches
Each dental branch has an address, phone number, and a manager.

Scenario:

Consider the following problem statement:

A dental company, "Bright Smiles", has multiple branches (i.e., dental clinics) and would like you to create a software application to help manage the business processes and dental services. Each dental branch has an address, phone number, and a manager. A dental branch offers dental services to patients. Examples of services include cleaning, implants, crowns, fillings, and more. Each of the services has a cost. The clinic keeps track of its patients and staff. The staff includes managers, receptionists, hygienists, and dentists. The patient needs to book an appointment before coming to the clinic. Upon checkout, the clinic charges the patient depending on the services she/he has received. Also, a 5% value-added tax (VAT) is added to the final bill.

1. DentalCompany:
      i.   Name: string
      ii.   Associations: Branch (1 to many)
2. Branch:
   - Address
   - phone number
   - and manager.
   - dental services:List[Services]
   - staff:List[Staff]
   - patients:List[Patient]
   - Associations: DentalService (1 to many), Staff (1 to many), Patient (1 to many)
3. Services
   - Name: string
   - Cost: float
   -
4. Person:
- firstName: string
- lastName: string
- phoneNumber:string
- id: string
      a. StaffMember (inherits from Person):

i.   Attributes:
ii.   Manager (inherits from StaffMember):
iii. Receptionist (inherits from StaffMember):
iv.   Hygienist (inherits from StaffMember):
v.   Dentist (inherits from StaffMember):
vi.   Patient (inherits from Person):
  - appointments: List[Appointment]

5. Appointment:
- patient: Patient
- dateTime:datetime
- services:List[Services]
- totalCost:float

1. Composition: Branch and Service .

This type of relationship means that one class has control over the lifetime of another class.

2. Association: Appointment and Service
3. Binary Association: Staff and Person
4. Unary Association: Patient and Appointment
5. Aggregation: DentalCompany and Branch

The Branch class has a composition relationship with the Service class. A branch offers one or more dental services, such as cleaning, filling, etc. The Branch class has a services attribute that stores a list of Service objects.

The Branch class has a composition relationship with the Staff class. A branch has one or more staff members, such as receptionists, hygienists, and dentists. The Branch class has a staff attribute that stores a list of Staff objects.

The Branch class has a composition relationship with the Patient class. A branch has one or more patients who receive dental services. The Branch class has a patients attribute that stores a list of Patient objects.

The Appointment class has an association relationship with the Service class. An appointment consists of one or more dental services. The Appointment class has a services attribute that stores a list of Service objects.

The Appointment class has an association relationship with the Patient class. An appointment is booked by a patient. The Appointment class has a patient attribute that stores a Patient object.

The Person class has a generalization relationship with the Staff and Patient classes. A person can be either a staff member or a patient. The Staff and Patient classes inherit from the Person class.

The Receipt class has a composition relationship with the Appointment class and an aggregation relationship with the Patient class. A receipt consists of one or more appointments and is issued to a patient. The Receipt class has an appointments attribute that stores a list of Appointment objects and a patient attribute that stores a Patient object.

Staff class and patient inherit from person class, and Manager, Receptionist, Hygienist, and Dentist inherit from the Staff

In summary :

- Branch class: composition relationship with Service class, Staff class, and Patient class
- Staff class: inherits from the Person class
- Patient class: inherits from the Person class
- Appointment class: association relationship with Service class and Patient class
- Person class: generalization relationship with Staff class and Patient class
- Manager class: inheritance from Staff class
- Receptionist class: inheritance from Staff class
- Hygienist class: inheritance from Staff class
- Dentist class: inheritance from Staff class
- Receipt class: composition relationship with Appointment class and aggregation relationship with Patient class

DentalCompany

-name:string
branch:List[Branch]

+getName():string
+addBranch(branch:List[Branch])
+get_branches():List[Branch]

1

*

Services

-serviceName:string
-cost:float

+setServiceName(serviceName:string:string)
+getServiceName():string
+setCost(cost:float)
+getCost():float

Branch

-name:string
-address:string
-phoneNumber:string
-manager:Manager
-services:List[Services]
-staff:List[Staff]
-patients:List[Patient]

+setName(name:string)
+getName():string
+ setAddress(address: str)
+ getAddress(): str
+ setPhoneNumber(phoneNumber: str)
+ getPhoneNumber(): str
+ setManager(manager: Manager)
+ getManager(): Manager
+ addService(service: Service)
+ getServices(): List[Service]
+ addStaff(staff_member: StaffMember)
+ getStaff(): List[Staff]
+ addPatient(patient: Patient)
+ getPatients(): List[Patient]

Appointment

-patient:Patient
-appointment_time:appointment_time
-services:List[Services]
-totalCost:float

+setPatient(patient:Patient)
+getPatient():Patient
+setAppointment_time(appointment_time:appointment_time)
+getppointment_time():appointment_time
+setServices(services:List[Services])
+getServices():List[Services]
+getTotalCost():float
…

Staff (inherits from Person)

-employeeID:string

+setEmployeeID(employeeID:string)
+ getEmployeeID(): str

inherits from Staff

Manager	Receptionist	Hygienist	Dentist

Person

-firstName:string
-lastName:string
-phoneNumber:string

+setFirstName(firstName:string)
+getFirstName():string
+setLasname(lastName:string)
+getLastName()string
+setPhoneNumber(phoneNumber:string)
+getPhoneNumber():string

has a

Receipt

-patient: patient
-appointment:appointments
-DentalCompany

…..

Patients

-ID
-appointment:List[Appointments]

+setID(ID:string)
+getID():string
+addAppointment(appointment: Appointments]
+getAppointment():List[Appointments]