1. **Person (Abstract Class)**

* **Fields:** 
  + name:String
  + Age : int
  + Type:String
  + Gender:String
  + Address:String
  + contactNnmber : int
* **Methods**
  + setName(name: String): void
  + setAge(age: int): void
  + setType(type: String): void
  + setGender(gender: String): void
  + setAddress(address: String): void
  + setContactNumber(contactNumber: int): void
  + getName(): String
  + getAge(): int
  + getType(): String
  + getGender(): String
  + getAddress(): String
  + getContactNumber(): int
  + getDetails(): String

**2. Patient (Inherits from Person):**

* **Fields:** 
  + patientID: int
  + admissionDate: String
  + (dischargeDate: String
  + medicalHistory: String
  + hosiptal: Hosiptal
* **Methods:**
  + setPatientID(patientID: int): void
  + setAdmissionDate(admissionDate: String): void
  + setDischargeDate(dischargeDate: String): void
  + setMedicalHistory(medicalHistory: String): void
  + getPatientID(): int
  + getAdmissionDate(): String
  + getDischargeDate(): String
  + getMedicalHistory(): String
  + addMedicalHistory(medicalHistory: String): void

**3. Doctor (Inherits from Person)**

* **Fields:** 
  + doctorID: int
  + specialization: String
  + yearsOfExperience: int
  + schedule: String
  + hosiptal: Hosiptal
* **Methods:**
  + setDoctorID(doctorID: int): void
  + setSpecialization(specialization: String): void
  + setYearsOfExperience(yearsOfExperience: int): void
  + setSchedule(schedule: String): void
  + getDoctorID(): int
  + getSpecialization(): String
  + getYearsOfExperience(): int
  + getSchedule(): String
  + diagnosePatient(p: Patient): void

**4. Hospital**

* **Fields:** 
  + hospitalID: int
  + hospitalName: String
  + location:: String
  + departments: List<String>
  + capacity: int
* capacity: int**Methods:**
  + addDoctor(d: Doctor): void
  + addPatient(p: Patient): void
  + getDoctorList(): List<Doctor>
  + getPatientList(): List<Patient>

**5. MedicalRecord**

* **Fields:** 
  + recordID: int
  + diagnosis: String
  + treatment: String
  + prescribedMedications: Stirng
  + doctorNotes: String
  + hosiptal: Hosiptal
* **Methods:**
  + updateRecord(diagnosis: String, treatment: String): void
  + getRecordDetails(): String

**6. Billing Interface**

* **Methods:**
  + generateBill(p: Patient): double
  + processPayment(p: Patient, amount: double): void

**7. BillingDepartment (Implements Billing Interface)**

* **Fields:** 
  + billingId: int
  + totalAmount:double
* **Methods:**
  + generateBill(p: Patient): double
  + processPayment(p: Patient, amount: double): void

**Relationships:**

* **Person** is an abstract base class for **Patient** and **Doctor**.
* **Hospital** contains (aggregates) **Doctor** and **Patient**.
* **MedicalRecord** relates to **Patient**.
* **BillingDepartment** implements **Billing** interface for payment processing.