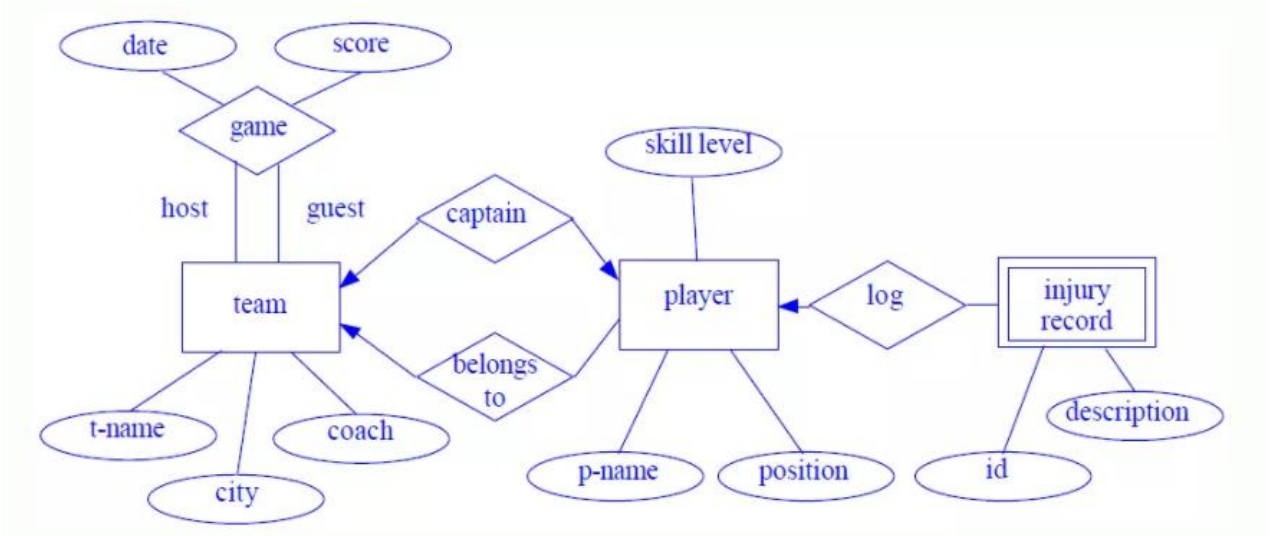
Suppose you are given the following requirements for a simple database for the National Hockey League (NHL):

The NHL has many teams, each **team** has a name, a city, a coach, a captain, and a set of players,

Each **player** belongs to only one team, each player has a name, a position (such as left wing or goalie), a skill level, and a set of **injury records**, a team captain is also a player,

A game is played between two teams (referred to as host team and guest team) and has a date (such as May 11th, 1999) and a score (such as 4 to 2).

Construct a clean and concise ER diagram for the NHL database.



A university registrar's office maintains data about the following entities:
1.courses, including number, title, credits, syllabus, and prerequisites;
2.course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom;
3.students, including student-id, name, and program;
4. Instructors, including identification number, name, department, and phone number. Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled. **Construct an E-R diagram for the registrar's office. Document all assumptions that you make about the mapping constraints.**

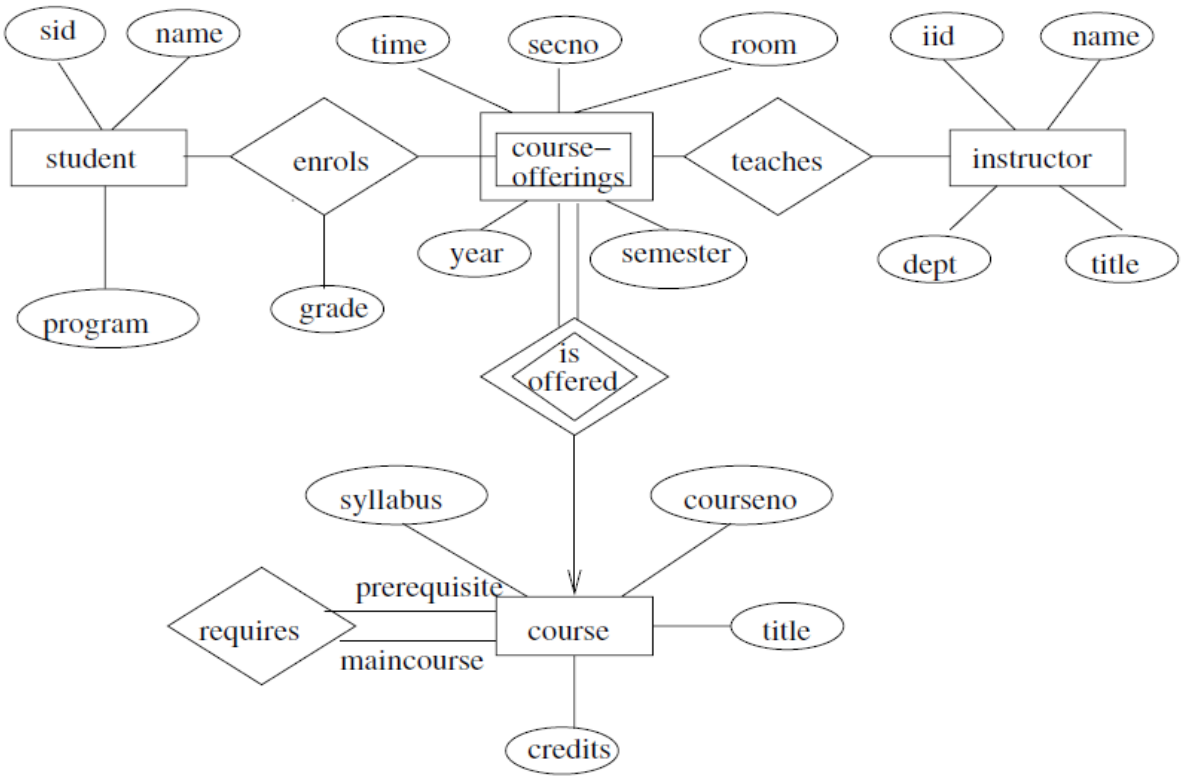
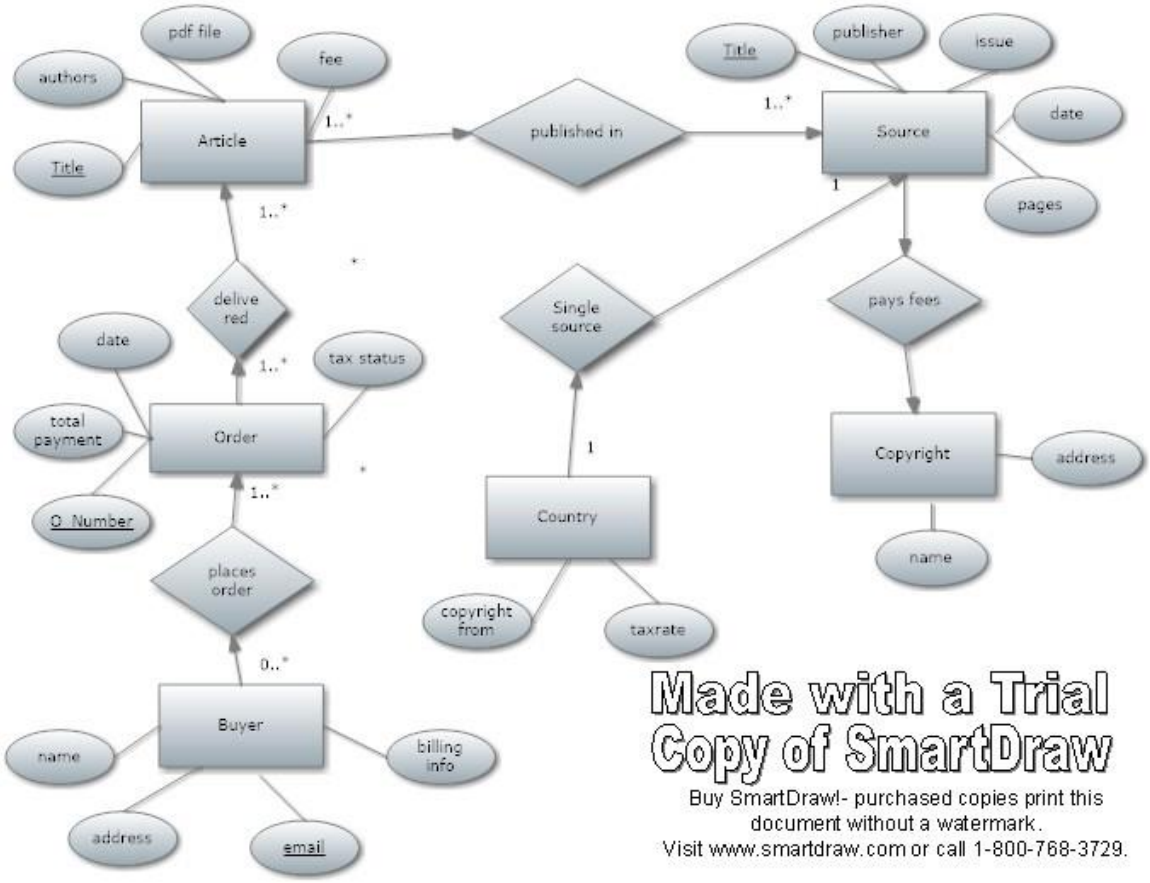


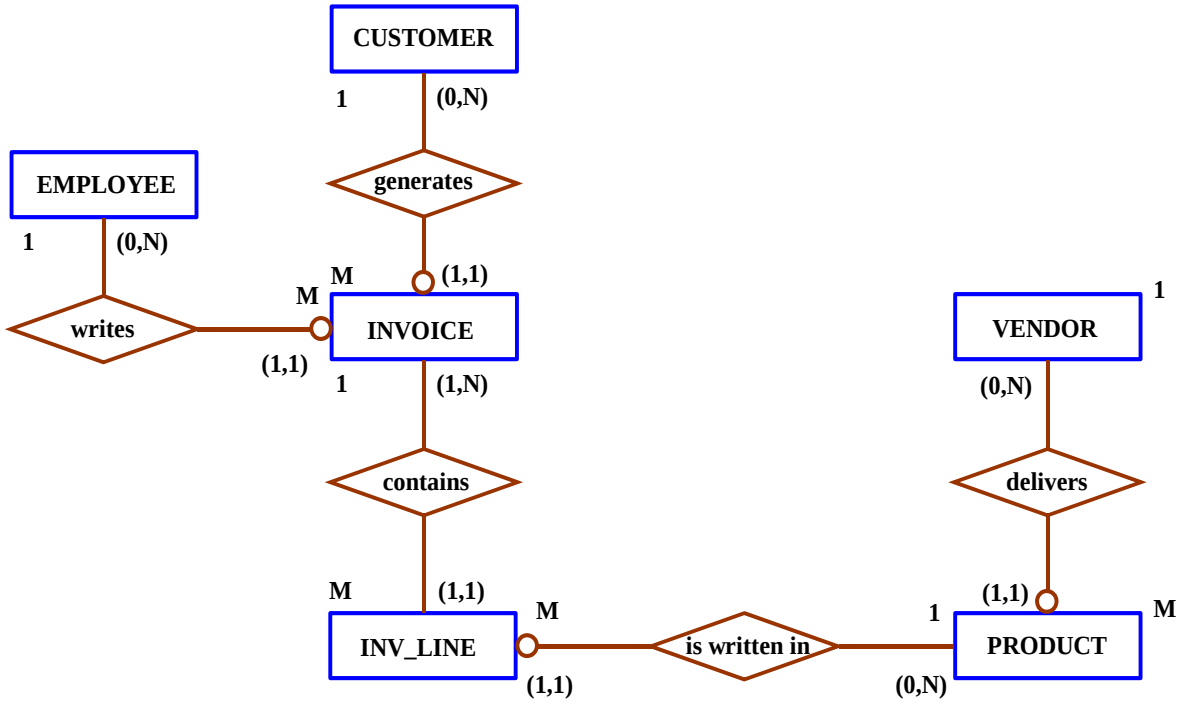
Figure 2.3 E-R diagram for a university.

Draw an **ER diagram** for the given scenario of buying an article.
Entities: Article, Source, Order, Copyright Agency, Country, Buyer
Attributes:
Article: Title(PK),authors, pdf file, fee
Source: Title(PK), publisher,issue,date, pages
Order: Order number(PK),total payment, date, tax status
Copyright Agency: name, address
Country: copyright from, taxrate
Buyer: name, address, email(PK), billing info
Following relationships are to be set:
a. Article is published in source. Many articles can be published in many sources.
b. Buyer places order. He can place zero or more orders.
c. Orders deliver articles. One article can be delivered in many orders and one order can deliver many articles as well.
d. Source pays fees to Copyright agency for every article published.
e. Every country has a single source of publication.

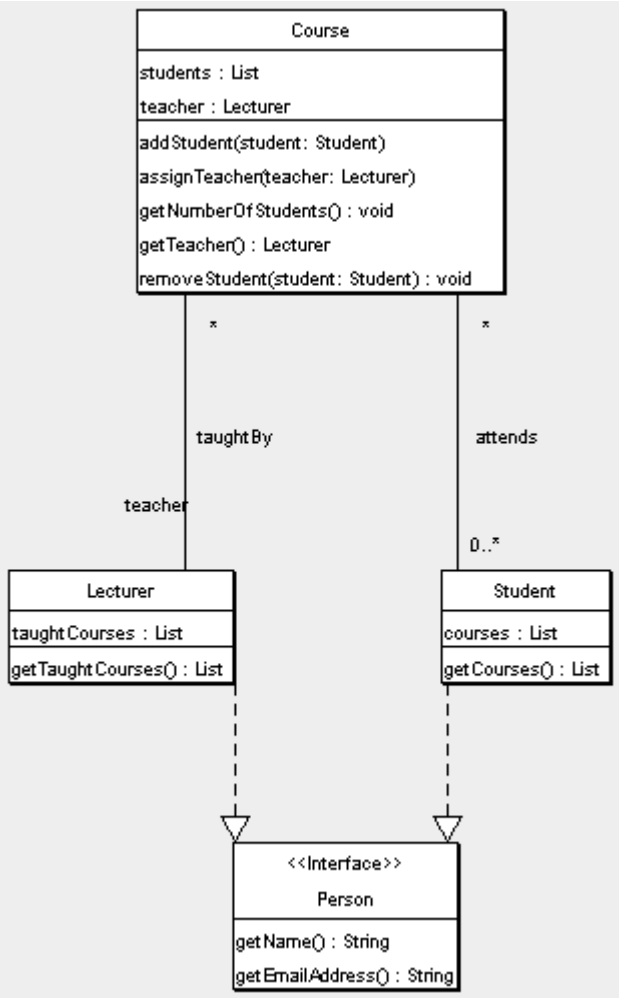


Create an ERD using the following requirements:

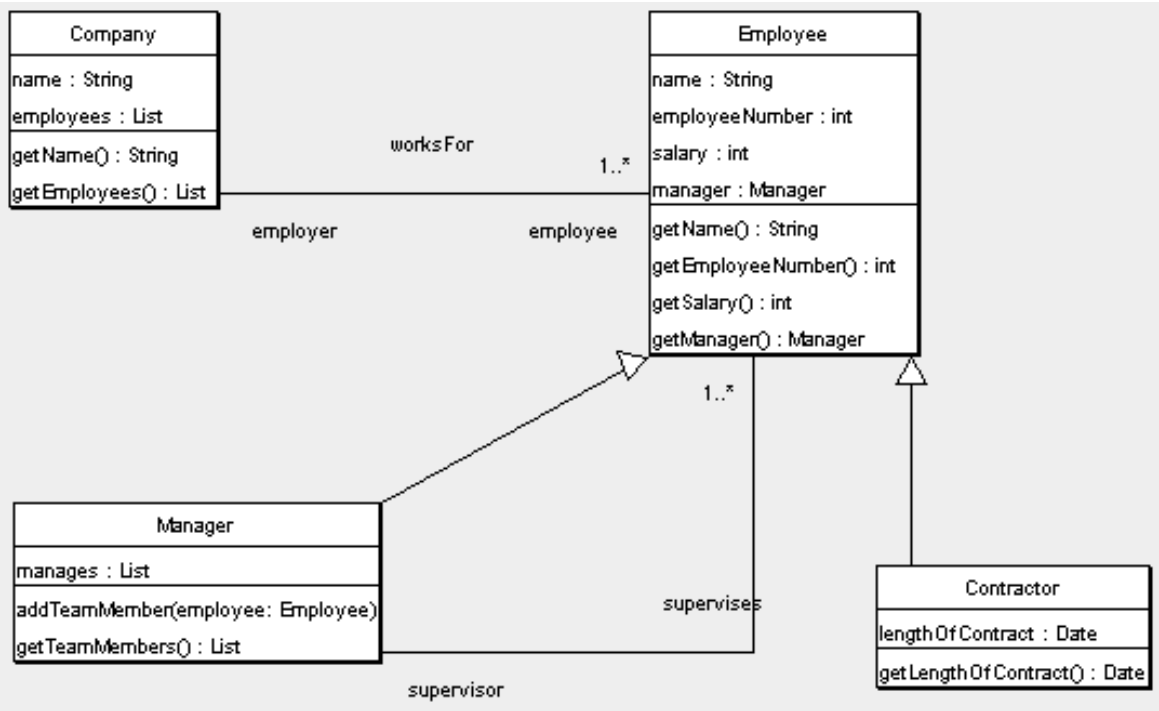
- An INVOICE is written by a SALESREP. Each sales representative can write many invoices, but each invoice is written by a single sales representative.
- The INVOICE is written for a single CUSTOMER. However, each customer can have many invoices.
- An INVOICE can include many detail lines (LINE), each of which describes one product bought by the customer.
- The product information is stored in a PRODUCT entity.
- The product's vendor information is found in a VENDOR entity.



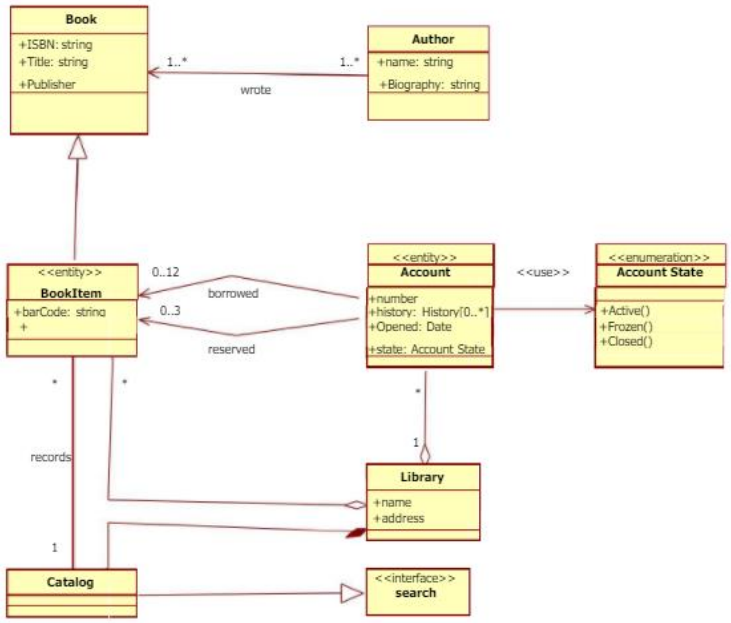
Draw a **class diagram** using StarUML for the scenario given below.
This is an example that **models University Courses**. Assume three classes' such as course, lecturer, student and an interface person.Each course objects maintains a list of student on that course and lecturer who has been assigned to teach that course. The courseobject has behavior that allows adding and removing student to and from course, assigning the teacher and getting a list of currentlyassigned student and currently assigned teacher. A teacher may teach several courses but a course only has a single teacher .A lecturer object maintains a list of courses that it teaches, course isattended by 0 or more student and student may attend multiple courses. A person interface will have getName() and getEmailAddress () methods both lecturer and student are shown tobe the type of person.



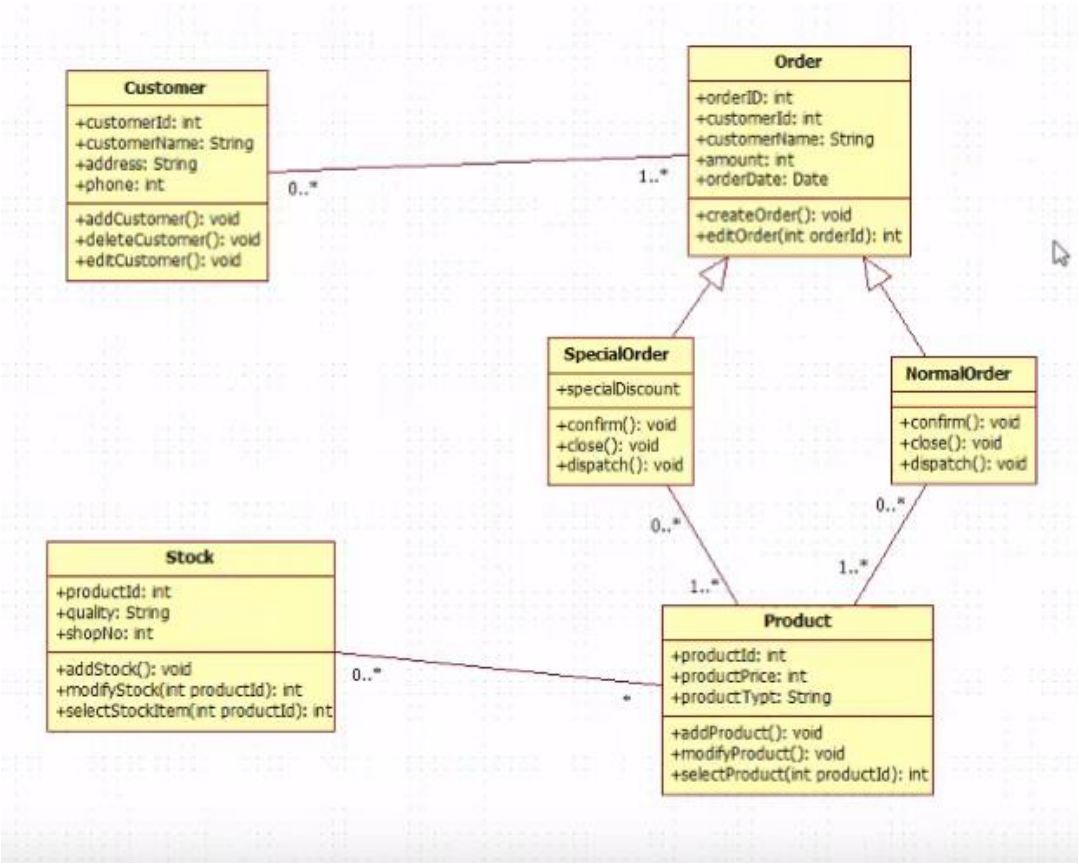
Draw a **class diagram** using StarUML for the scenario given below.
This scenario is from system that models companies for a payroll or reporting system. Company object has properties such as name and employees_list and getName and getEmployees as its behavior. Employee object includes employee no, name, salary and manager as its properties getName (), getEmplyoeeNo () ,getSalary() getManager() as its methods. getManager() accepts object of manager. Company may have one or more employees. A manager object keeps manages as list property and add TeamMember(employee_list) and getTeamMember() as itsbehaviors. One or more employee can be managed by manager objects. Some employees are contractual employees who are within a lieu of a contractor object. A contractor object may have length_of _contract as its property and getLength() as its behavior.



Create a **class diagram** (Use StarUML) for "library management" using the classes with their attributes and operation given below. Also set the appropriate relationship between the classes using the relationship tools from the toolbox following the overview of the system given below. Overview of the system:- It has a class "Book". Book has authors so it has an "Author" class. In order to collect book information it has "BookItem" class which uses some of the properties from book class. It needs an account for reserving book by the user so it has an "Account class." In account class there is an attribute named state which uses an enumeration named "AccountState". It also has a class "Library" to manage the account, user and the books. It has a user class to manage the user detail that has an account in the library and he can borrow and return books to library. The system also has an interface "Search" where the user searches the book he needed from the "Catalog" class.



Draw a **class diagram** using StarUML for the scenario given below: This is an example that models **"ORDER MANAGEMENT"**. The Customer object has properties such as CustomerId,CustomerName, Address and Phone and methods such as AddCustomer(),DeleteCustomer() and EditCustomer(). Orderobject includes OrderId, CustomerId, CustomerName, ProductId, Amount and OrderDate as its property and CreateOrder() and EditOrder(OrderId) as its behavior. A customer can place one or many orders. Further there are SpecialOrder object andNormalOrder object which have same methods CreateOrder(), confirm(), close(), dispatch() whereas the SpecialOrder object alsohas one property named SpecialDiscount. **Special Order and Normal Order** objects are both kinds of order and are thereforeshown to inherit from order entity. Moreover the system also has Product entity having attributes such as ProductId, ProductPrice, ProductType and methods such as AddProduct(), ModifyProduct() and SelectProduct(ProductId). Stock object has properties likeProductId, Quality and ShopNo and behavior such as addStock(), ModifyStock(ProductId) and slectStockItem(ProductId). Note that specialOrder and NormalOrder has 1 or more product whereas stock has many products.



Draw a **class diagram** using StarUML for the scenario given below:
This is an example that models **"Hospital Management**". The ward object of this system has attributes such as name, patient-gender andcapacity. Note that patient-gender is a gender type which is an enumeration containing enums male and female. The system also has **Patient** entity with attributes such as patient_id, admitted,sickness _history, prescriptions,special_reqs and allergies and gender which is again a gender enumeration type. And operations such as getPatient() and deletePatient(Patient_id).Ward is a division of a hospital object having attributes such as name address and phone number. In **hospital** there are number of wards each of which may be empty or have one or more patient. Each ward has unique name. This ward is shared by patients who need a similar kind of care. Each patient is on a single ward. The system also has **Doctor** entity which is further classified into **Consultant Doctor** and **Junior Doctor**. The doctors in the hospital are organized into Teams entity with attribute team_name. Each team can have two or more doctors. Each patient is under the care of a single team of doctors. A patient may be treated by any number of doctors but all the doctors must belong to same team that cares for the patient. Note that team is own by the hospital.