



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

IT3100

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Bài 11: Tổng quan về PTTK HĐT – Biểu đồ lớp

Mục tiêu bài học

- Giới thiệu về Phân tích thiết kế hướng đối tượng
- Giới thiệu về ngôn ngữ mô hình hóa UML
- Giới thiệu các biểu đồ cơ bản
- Làm quen biểu đồ Lớp, Quan hệ giữa các lớp

Nội dung

1. Phân tích thiết kế hướng đối tượng
2. UML là gì ? Các biểu đồ UML cơ bản
3. Biểu đồ Lớp

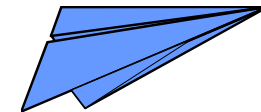
Nội dung

1. **Phân tích thiết kế hướng đối tượng**
2. UML là gì ? Các biểu đồ UML cơ bản
3. Biểu đồ Lớp

1.1. Tầm quan trọng của phân tích và thiết kế

❖ Hướng tiếp cận không có phân tích – thiết kế:

- Bắt đầu lập trình ngay khi có được yêu cầu
- Mất rất nhiều thời gian và tạo đi tạo lại nhiều mã nguồn
- Không có bất kỳ một kiến trúc nào
- Phải chịu khổ với những lỗi phát sinh



❖ Hướng tiếp cận có phân tích – thiết kế:

- Chuyển các yêu cầu của bài toán thành một bản thiết kế rõ ràng
- Tập trung vào phân tích các YÊU CẦU và thiết kế các MÔ HÌNH cho hệ thống TRƯỚC khi lập trình



1.2. Ưu điểm của việc PTTK

- Đơn giản hóa thế giới thực bằng các mô hình
- Mô tả đúng, đồng nhất cấu trúc, cách ứng xử của HT trong suốt quá trình xây dựng
- Đảm bảo mục đích và yêu cầu của HT được thỏa mãn trước khi xây dựng
- Cung cấp cho người dùng, khách hàng, kỹ sư phân tích, thiết kế, kỹ sư lập trình nhiều cái nhìn khác nhau về cùng một HT
- Ghi lại các quyết định của nhà phát triển để sử dụng sau này

1.3 Phương pháp PTTK HĐT (OOAD)

- ❖ OOAD được chia thành 2 giai đoạn
 - Phân tích hướng đối tượng (OOA)
 - Thiết kế hướng đối tượng (OOD)
- ❖ OOA là giai đoạn nhằm tạo ra các mô hình cơ bản (mô hình khái niệm) của hệ thống dựa theo những gì khách hàng yêu cầu về hệ thống của họ
- ❖ OOD sẽ bổ sung thêm các thông tin thiết kế chi tiết cho các mô hình nói trên

OOA

- ❖ Xác định yêu cầu phần mềm
- ❖ Đặc tả yêu cầu phần mềm thông qua mô hình các đối tượng và tương tác giữa chúng
- ❖ Tạo được mô hình có các thành phần là đối tượng và khái niệm đời thực, dễ hiểu với người dùng
- ❖ Mô hình hóa các thực thể, giữ nguyên cấu trúc, quan hệ, hành vi giữa chúng

OOD

- ❖ Thực thi các mô hình khái niệm là đầu ra của bước OOA
- ❖ Các khái niệm trong OOA được ánh xạ theo thành các lớp thực thi. Các ràng buộc, các giao diện được thiết kế. Kết quả là đặc tả chi tiết về hệ thống cần xây dựng, theo một công nghệ cụ thể được lựa chọn

OOD

❖ Tổ chức chương trình thành các tập hợp đối tượng cộng tác

- Mỗi đối tượng là thực thể của một lớp

❖ Thiết kế trên kết quả của OOA

- Cải thiện, tối ưu hóa thêm
- Thiết kế các
 - Phương thức (operations)
 - Thuộc tính (attributes)
 - Mối quan hệ giữa các lớp (classes)
- Đưa ra các biểu đồ tĩnh và động
 - Tĩnh: biểu thị các lớp và đối tượng
 - Động: biểu thị tương tác giữa các lớp & phương thức hoạt động

Nội dung

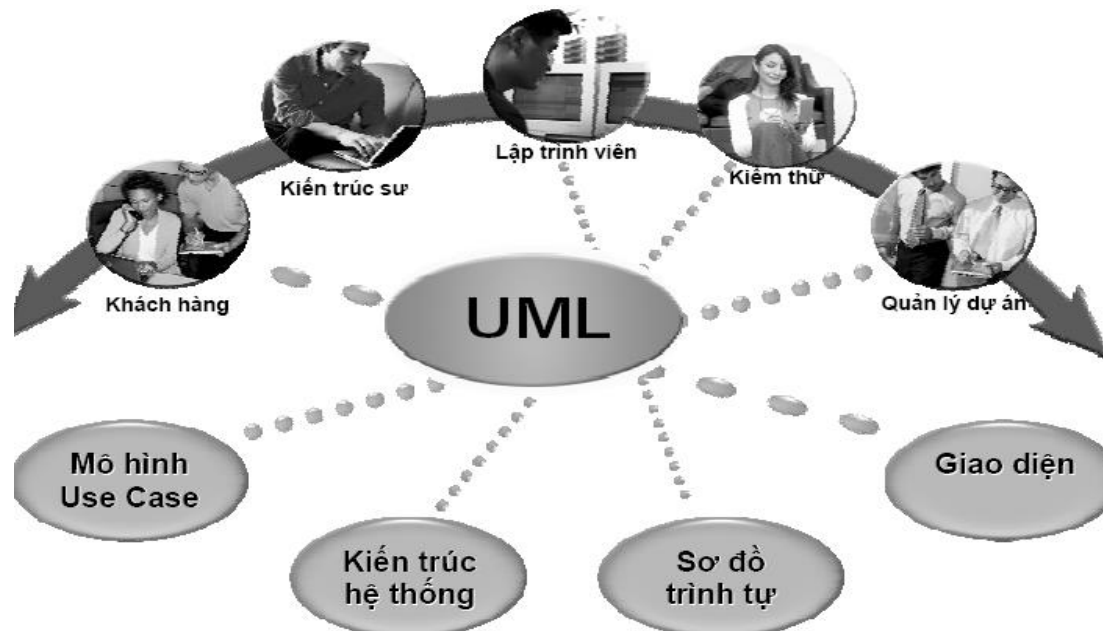
1. Phân tích thiết kế hướng đối tượng
2. UML là gì ? Các biểu đồ UML cơ bản
3. Biểu đồ Lớp

2. Khái niệm UML

- ❖ UML: viết tắt của “Unified Modeling Language” là một Ngôn ngữ mô hình hóa được thống nhất
 - ❖ UML là ngôn ngữ trực quan để:
 - trực quan hóa (visualizing)
 - đặc tả (specifying)
 - xây dựng (constructing)
 - tài liệu hóa (documenting)
- các cấu phần của một hệ thống phần mềm
- ❖ Giúp công việc phát triển được xử lý nhất quán, giảm thiểu lỗi xảy ra
 - Giúp dễ hình dung hơn cấu trúc của hệ thống
 - Hiệu quả hơn trong việc liên lạc, trao đổi

2. Khái niệm UML (2)

- ❖ Thiết lập một phương thức thống nhất để xây dựng và “vẽ” ra các yêu cầu và thiết kế hướng đối tượng trong quá trình PTTK phần mềm



2.1 Làm việc với UML

- ❖ Các mô hình UML có thể kết nối trực tiếp với rất nhiều ngôn ngữ lập trình.
 - Ánh xạ sang Java, C++, Visual Basic...
 - Các bảng trong RDBMS hoặc kho lưu trữ trong OODBMS
 - Cho phép các kỹ nghệ xuôi (chuyển UML thành mã nguồn)
 - Cho phép kỹ nghệ ngược (xây dựng mô hình hệ thống từ mã nguồn)
- ❖ Các công cụ UML
 - Công cụ mã nguồn mở: EclipseUML, UmlDesigner, StarUML, Argo UML...
 - Công cụ thương mại: Enterprise Architect, IBM Rational Software Architect, Microsoft Visio, Visual Paradigm for UML, SmartDraw...

2.2 Các biểu đồ UML cơ bản

❖ Biểu đồ:

- là các hình vẽ bao gồm các ký hiệu phần tử mô hình hóa
- minh họa một thành phần cụ thể hay một khía cạnh cụ thể của hệ thống.

❖ Một mô hình hệ thống thường có nhiều loại biểu đồ, mỗi loại gồm nhiều biểu đồ khác nhau.

❖ Một biểu đồ là một thành phần của một hướng nhìn cụ thể

❖ Một số loại biểu đồ có thể là thành phần của nhiều hướng nhìn khác nhau

❖ UML thế hệ 2 có tới 13-14 loại biểu đồ. Trong một project, chỉ sử dụng những biểu đồ phù hợp nhất

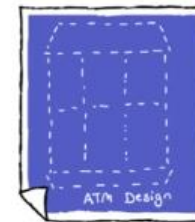
2.2. Các biểu đồ UML cơ bản

❖ Phân biệt:

- **Biểu đồ cấu trúc:** mô tả thành phần tĩnh, luôn có của hệ thống và mối quan hệ giữa chúng
- **Biểu đồ hành vi:** mô tả cách hoạt động của hệ thống.

Diagram Types

Two Main Categories:



Structural



Behavioral

what is contained in a system

what must happen in a system

2.3. Biểu đồ cấu trúc

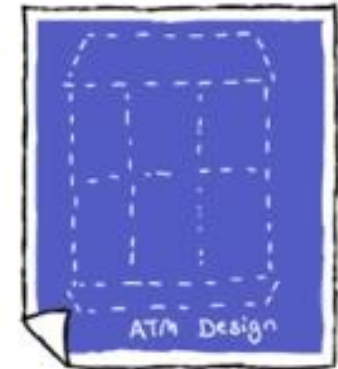
❖ Biểu đồ cấu trúc tĩnh

- Biểu đồ lớp (Class Diagram)
- Biểu đồ đối tượng (Object Diagram)
- Biểu đồ gói (Package diagram)

❖ Biểu đồ thực thi

- Biểu đồ thành phần (Component Diagram)
- Biểu đồ triển khai (Deployment Diagram)
- Biểu đồ cấu thành (Composite Diagram)

❖ Biểu đồ profile (Profile Diagram)



Structural

2.4. Biểu đồ hành vi

- ❖ Biểu đồ use case (Use Case Diagram)
- ❖ Biểu đồ hoạt động (Activity Diagram)
- ❖ Biểu đồ tương tác
 - Biểu đồ tổng quát (Interaction overview diagram)
 - Biểu đồ trình tự (Sequence Diagram)
 - Biểu đồ giao tiếp/cộng tác (Communication/Collaboration Diagram)
- ❖ Biểu đồ trạng thái (State Diagram)
- ❖ Biểu đồ thời gian (Timing Diagram)



Behavioral

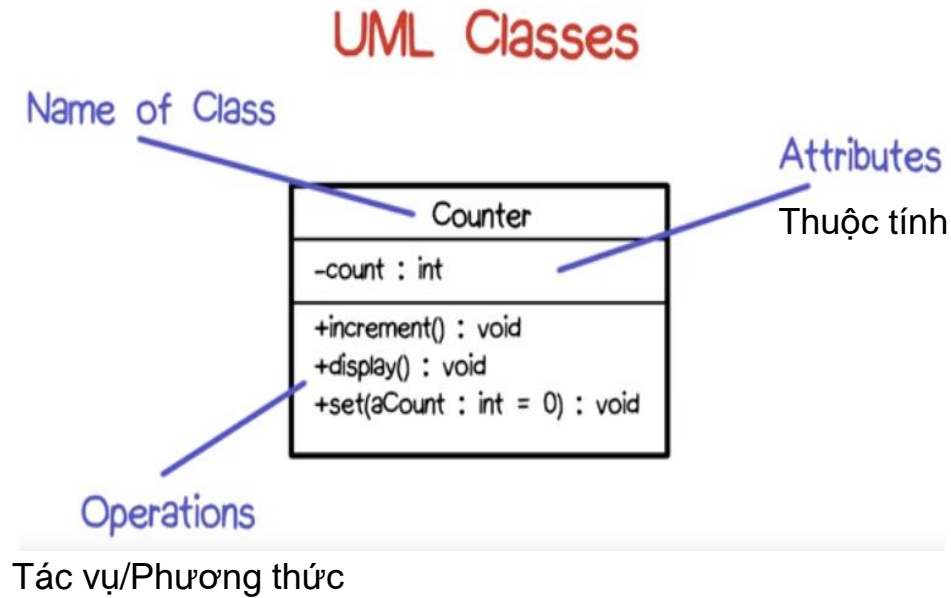
Nội dung

1. Phân tích thiết kế hướng đối tượng
2. UML là gì ? Các biểu đồ UML cơ bản
3. **Biểu đồ Lớp**

3.1 Biểu đồ lớp là gì?

- ❖ Biểu đồ lớp chỉ ra sự tồn tại của các lớp và mối quan hệ giữa chúng trong bản thiết kế logic của một hệ thống
 - Chỉ ra cấu trúc tĩnh của mô hình như lớp, cấu trúc bên trong của chúng và mối quan hệ với các lớp khác.
 - Chỉ ra tất cả hoặc một phần cấu trúc lớp của một hệ thống.
 - Không đưa ra các thông tin tạm thời.
- ❖ Khung nhìn tĩnh của một hệ thống chủ yếu hỗ trợ các yêu cầu chức năng của hệ thống.

3.2 Biểu đồ lớp



3.1 Các ký hiệu biểu đồ lớp

- ❖ Lớp: tên lớp, thuộc tính, tác vụ/phương thức
- ❖ Phạm vi truy cập: ai được truy cập các thông tin trong lớp
 - - : private: không được truy cập từ ngoài lớp
 - + : public: truy cập được từ bất kỳ lớp/phương thức ngoài lớp
 - # : protected: truy cập được từ các lớp thừa kế

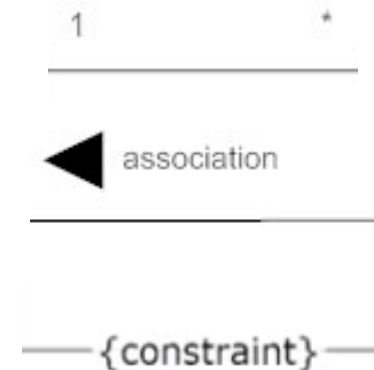
Class Name
attributes
operations()

Class Name
- private attributes + public attributes # protected attributes
- private operation + public operation # protected operation

3.1 Các ký hiệu biểu đồ lớp (2)

- Quan hệ giữa các lớp:
 - Liên kết (Association): có kết nối, có sử dụng
 - Bội số quan hệ (Multiplicity)
 - Có hướng (navigability): chỉ rõ hướng liên kết
 - Ràng buộc (constraint): có điều kiện về liên kết

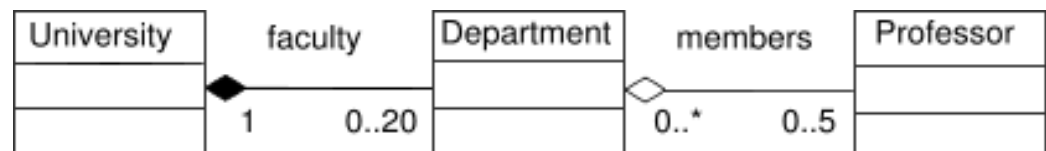
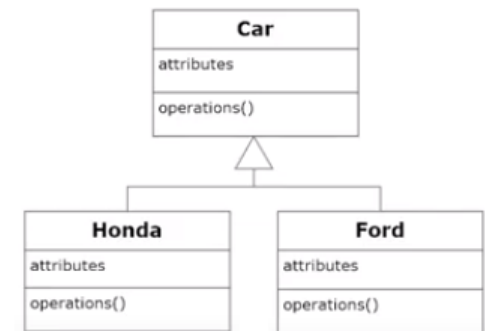
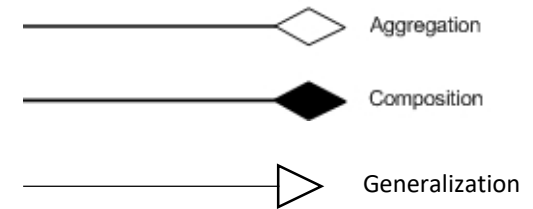
Indicator		Meaning
0..1		Zero or one
1		One only
0..*		0 or more
1..*	*	1 or more
n		Only n (where n > 1)
0..n		Zero to n (where n > 1)
1..n		One to n (where n > 1)



3.1 Các ký hiệu biểu đồ lớp (3)

■ Quan hệ giữa các lớp (tiếp):

- Kết tập (Aggregation): mô tả quan hệ toàn thể - bộ phận hay còn gọi là quan hệ “is a part of”.
 - Trong quan hệ Aggregation, lớp toàn thể chỉ có thể truy cập vào các thành phần Public của lớp bộ phận.
- Cấu thành (composition): là một dạng kết tập, bộ phận không thể tồn tại nếu toàn thể bị hủy bỏ.
- Tổng quát hóa (generalization): khi một lớp là một dạng cụ thể của một lớp khác. Hay còn gọi là quan hệ “is a kind of”
 - Lớp con thừa hưởng các thuộc tính/phương thức public và protected của lớp mẹ



VD: Hệ thống đăng ký khóa học

❖ Các lớp cơ bản

CloseRegistrationForm
+ open() + close registration()

Student
+ get tuition() + add schedule() + get schedule() + delete schedule() + has pre-requisites()

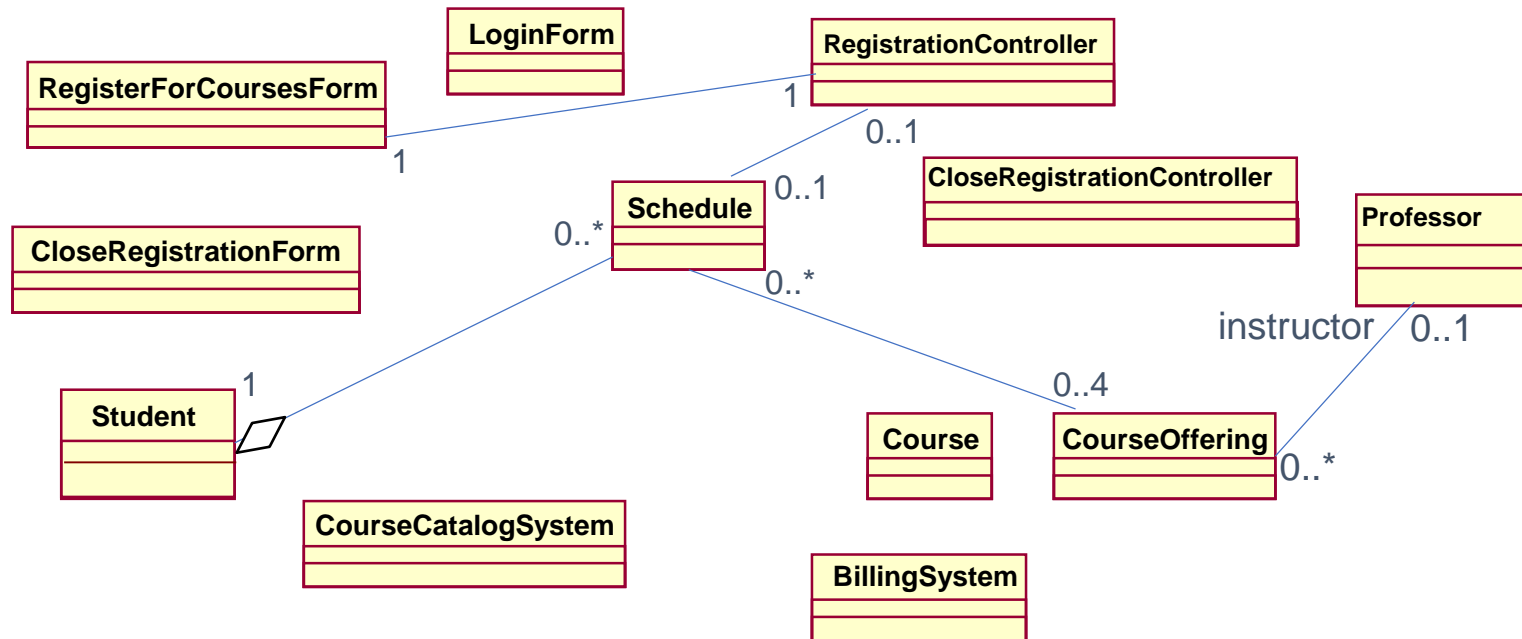
Schedule
- semester
+ commit() + select alternate() + remove offering() + level() + cancel() + get cost() + delete() + submit() + save() + any conflicts?() + create with offerings() + update with new selections()

CloseRegistrationController
+ is registration open?() + close registration()

Professor
- name - employeeID : UniqueId - hireDate - status - discipline - maxLoad
+ submitFinalGrade() + acceptCourseOffering() + setMaxLoad() + takeSabbatical() + teachClass()

VD: Hệ thống đăng ký khóa học (2)

❖ Biểu đồ lớp sơ lược

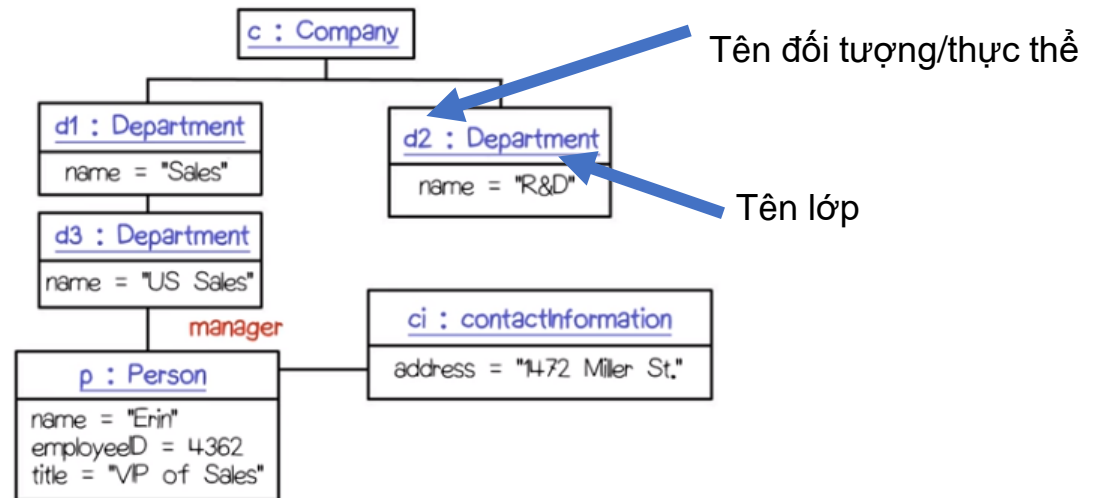


3.2. Biểu đồ đối tượng

- ❖ Giống biểu đồ lớp, thay vì biểu diễn lớp, biểu đồ đối tượng biểu diễn các thực thể thật của lớp và quan hệ giữa chúng

Object Diagram

Conveys **objects** and **links** instead of **classes** and **relationships**



Bài tập 1

- **Xây dựng phần mềm quản lý đăng ký dạy và học ở trường đại học như sau:**
- Các sinh viên và giảng viên được lưu trữ thông tin vào trong phần mềm này với các nội dung về tên, tuổi, định danh cá nhân, mã số sinh viên hoặc mã cán bộ. Giảng viên còn có thông tin về cấp bậc (**level** với dạng số nguyên từ 1 trở đi); và mã số thuế (**tax**).
- Mỗi giảng viên có thể dạy nhiều lớp (**Course**) và có thể chủ nhiệm nhiều sinh viên. Mỗi sinh viên cũng đăng ký nhiều lớp học (**Course**), ứng với mỗi sinh viên có một bảng điểm (**Table**) và mỗi sinh viên được chủ nhiệm bởi một giảng viên.
- Bảng điểm của một sinh viên lưu trữ thông tin điểm của từng lớp học của sinh viên đăng ký.
- Hãy xây dựng biểu đồ lớp

Bài tập 2

- **Chương trình quản lý Thông tin trong một nhà ga được mô tả như sau:**
- Hoạt động chuyên chở trong nhà ga gồm nhiều đoàn tàu. Mỗi đoàn tàu có một số hiệu riêng, thông tin về ga đích đến của đoàn tàu và lịch trình chạy của đoàn tàu (giờ khởi hành và giờ dự kiến đến ga đích).
- Một đoàn tàu gồm nhiều toa tàu. Mỗi toa thuộc một trong hai loại toa chở khách hoặc toa chở hàng. Mỗi toa tàu có một số hiệu duy nhất và trọng lượng không tải tính bằng tấn (khi không chở khách hay hàng hoá). Mỗi toa chở khách còn có thông tin riêng về số lượng khách tối đa có thể chở.
- Khi tàu vào ga toa chở khách có thêm các hoạt động: thêm khách lên toa, bớt khách xuống toa.
- Thông tin về hành khách đi tàu gồm có họ tên, số chứng minh nhân dân, đoàn tàu và toa tàu mà họ mua vé. ga lên tàu và điểm xuống.
- Hãy xây dựng biểu đồ lớp

Bài tập 3

- Một phần mềm Quản lý xe buýt tại bến xe được mô tả như sau:
- Một xe buýt (**Bus**) chạy được tối đa 30 chuyến/ngày (**Trip**). Mỗi chuyến chứa tối đa 80 hành khách (**Person**). Hành khách được chia làm hai loại: hành khách mua vé theo từng lượt đi (**Customer**) và hành khách mua vé tháng (**Passenger**).
- *Tất cả các hành khách* đều được định danh bằng tên (**name**) và số chứng minh thư (**citizenCard**). Khách mua vé tháng có thêm thông tin mã vé ID.
- *Các xe buýt* có thông số về số lượng ghế ngồi (**numberOfSeats**) khác nhau. Trong lớp **Bus**, người ta xây dựng phương thức public **isEnabledToLeaveStation(Trip t)**, trả về **true** nếu số hành khách trên chuyến xe buýt **t** bé hơn hoặc bằng 80% số ghế ngồi. Lớp **Bus** là lớp toàn thể, lớp **Trip** kết tập trong nó với tên vai trò là **trips**.
- Các xe buýt có thông số để định danh, đây là một con số. Trong lớp **Trip**, người ta xây dựng phương thức public **availableSeats()** trả về số lượng các ghế trống có trên chuyến xe.
- Người ta cài đặt trong lớp **Trip** phương thức mang tên **numberOfPassenger()**, trả về số lượng các khách sử dụng vé tháng có trên chuyến xe.
- Hãy xây dựng biểu đồ lớp