

PHÂN TÍCH VÀ THIẾT KẾ HƯỚNG ĐỐI TƯỢNG

Th.S. Võ Đức Quang

Bộ môn KHMT&CNPM, Viện Kỹ thuật và công nghệ

✉ quangvd.cntt.dhv@gmail.com, quangvd@vinhuni.edu.vn

☎ 0989.891.418

Chương 5. Thiết kế hệ thống

- 5.1. Giới thiệu về thiết kế hệ thống
- 5.2. Thiết kế lớp và phương thức
- 5.3. Thiết kế quản trị dữ liệu
- 5.4. Thiết kế giao diện tương tác Người-Máy
- 5.5. Thiết kế kiến trúc vật lý
- 5.6. Bài tập

5.1. Giới thiệu về thiết kế hệ thống

- Giai đoạn phân tích → Xác định những yêu cầu cần thiết về nghiệp vụ của hệ thống
- Giai đoạn thiết kế → Tập trung vào cách xây dựng hệ thống
 - Tạo ra một kế hoạch chi tiết để thực hiện hệ thống
 - Cách lưu dữ liệu thế nào? Lưu trữ cái gì? Lưu ở đâu?
 - Cách người dùng sẽ giao tiếp với hệ thống?
 - Các thành phần vật lý sẽ có trong hệ thống
- Các giai đoạn của phân tích và thiết kế có mối quan hệ mật thiết với nhau, có thể yêu cầu nhiều về việc chuyển đổi qua lại
- Ví dụ: Sử dụng prototyping có thể phát hiện nhiều thông tin

5.1. Giới thiệu về thiết kế hệ thống

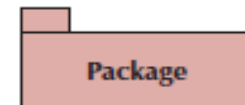
- Một số vấn đề lưu ý
 - Xác thực các mô hình đã phân tích ở giai đoạn trước
 - Phân vùng, Phân tầng để hệ thống phức tạp trở nên dễ hiểu hơn
 - Mô hình kiến trúc MVC (Model-view-controller)
 - Interface (Ví dụ: API-Application Programming Interface)
 - Các tầng đề xuất:
 - Foundation (e.g., container classes)
 - Problem domain (e.g., encapsulation, inheritance, polymorphism)
 - Data management (e.g., data storage and retrieval)
 - User interface (e.g., data input forms)
 - Physical architecture (e.g., specific computers and networks)

5.1. Sơ đồ gói – Package Diagrams

- Gói (Packages): tập các thành phần tương đồng cùng nhau
- Sơ đồ gói mô tả các gói và mối quan hệ giữa chúng
 - Giúp giảm sự phức tạp của việc mô hình hóa
 - Các mối quan hệ có thể có
 - Liên kết
 - Kết nhập
 - Phụ thuộc

A package:

- Is a logical grouping of UML elements
- Is used to simplify UML diagrams by grouping related elements into a single higher-level element.



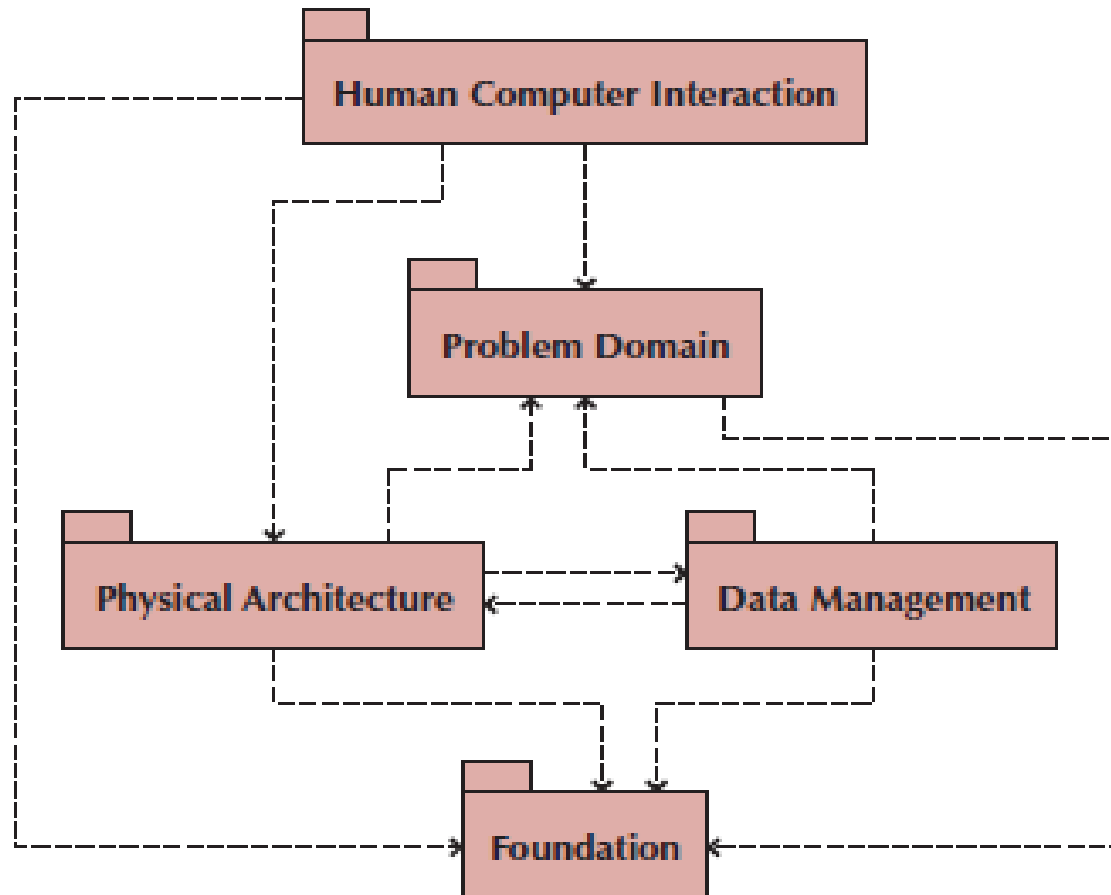
A dependency relationship:

- Represents a dependency between packages: If a package is changed, the dependent package also could have to be modified.
- Has an arrow drawn from the dependent package toward the package on which it is dependent.



5.1. Sơ đồ gói – Package Diagrams

- Ví dụ



5.1. Sơ đồ gói – Package Diagrams

- Cách xây dựng
 - Đặt ra khung cảnh cần mô tả biểu diễn
 - Xác định các cụm lớp cùng nhau dựa trên các mối quan hệ được chia sẻ
 - Tạo các gói từ các cụm đã xác định
 - Xác định mối quan hệ phụ thuộc giữa các gói
 - Bố trí và vẽ sơ đồ
 - Xác minh và xác nhận sơ đồ gói

5.1. Chiến lược thiết kế hệ thống

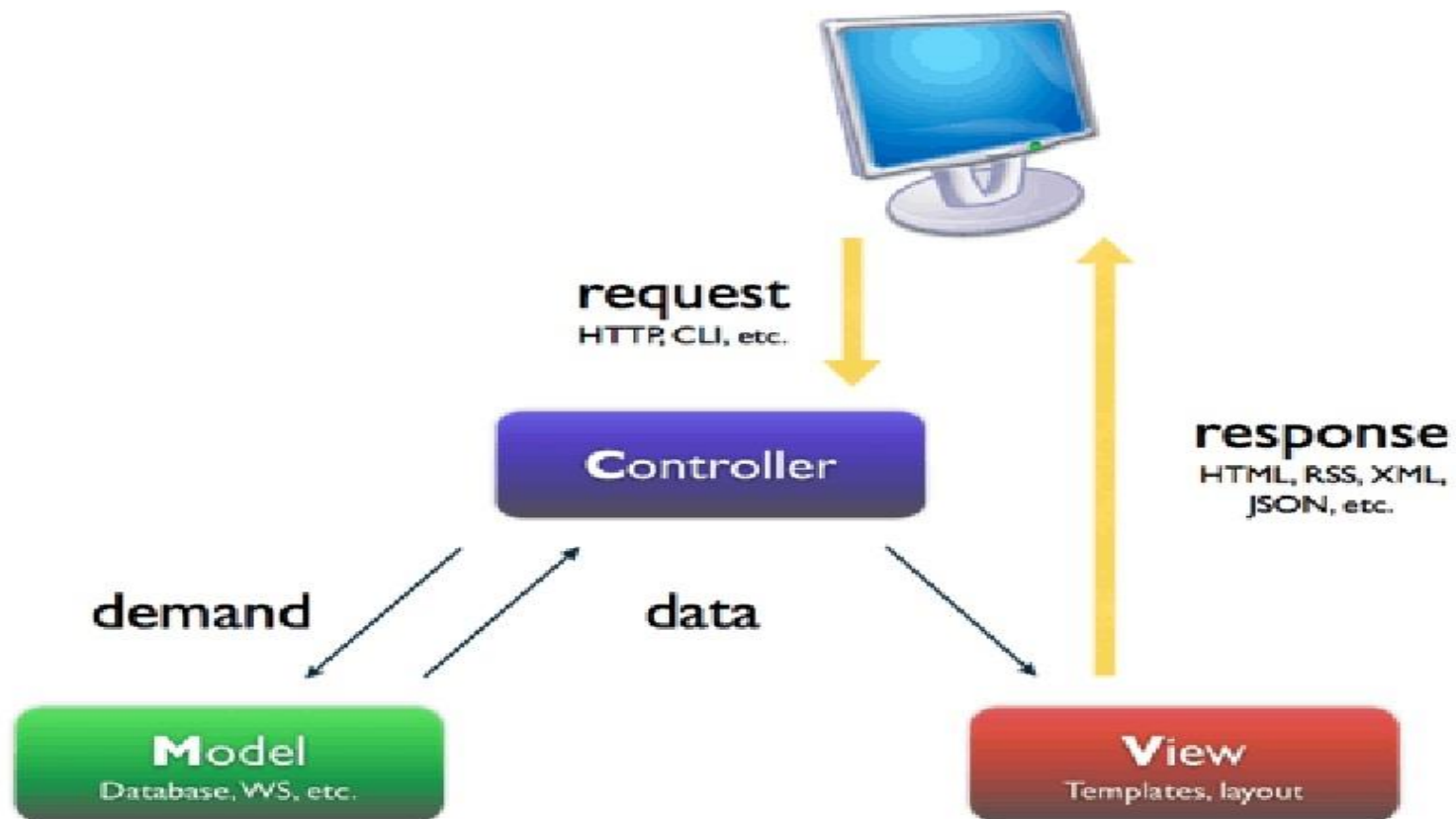
- Thiết kế và xây dựng tùy chỉnh từ đầu
 - Cho phép đáp ứng các yêu cầu chuyên môn cao
 - Cho phép linh hoạt và sáng tạo trong việc giải quyết vấn đề
 - Thay đổi thành phần dễ dàng hơn
 - Phát triển kỹ năng nhân sự cơ hữu
 - Có thể gây gánh nặng quá mức cho nhân viên
 - Có thể thêm rủi ro
- Sử dụng phần mềm đóng gói (nguồn đóng, nguồn mở)
 - Có thể phải can thiệp thay đổi cho phù hợp
- Tích hợp hệ thống
 - Xây dựng một hệ thống mới bằng cách kết hợp các gói, hệ thống cũ và phần mềm mới
 - Vấn đề: tích hợp dữ liệu, xây dựng API
- Thuê bên ngoài
 - Mất kiểm soát
 - Thỏa thuận, phối hợp

5.1. Chiến lược thiết kế hệ thống

- Lựa chọn dựa trên các yếu tố
 - Xác định công cụ và kỹ năng cần thiết để phát triển nội bộ
 - Xác định các gói hiện có đáp ứng nhu cầu của người dùng
 - Xác định vị trí các công ty có thể xây dựng nó theo hợp đồng
 - Kết hợp tính khả thi về kỹ thuật, kinh tế và tổ chức

5.1. Mô hình MVC

- MVC (Model-View-Controller)



5.1. Mô hình MVC

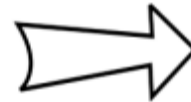
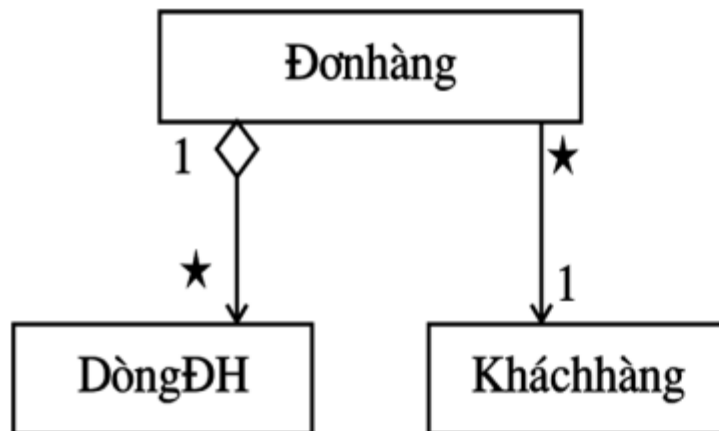
- MVC (Model-View-Controller)
 - Model (Mô hình), đó là lớp thực thể (ví dụ: Khách hàng, Đơn hàng) thuộc tầng nghiệp vụ
 - Đây là nguồn gốc của thông tin được hiển thị trên giao diện người dùng
 - Tuy nhiên, giao diện không truy cập trực tiếp vào đối tượng thực thể, mà tầng ứng dụng sẽ gom (tạo) thông tin từ các đối tượng thực thể vào các "tài liệu" (document), và giao diện người dùng truy cập vào tài liệu để lấy thông tin, thường thì có sự tương ứng 1-1 giữa khung nhìn (view) và tài liệu
 - View (Khung nhìn), đó là phần hiển thị của giao diện mà người dùng trông thấy
 - Controller (Bộ điều khiển), đó là phần đảm trách việc quản lý động thái (hành vi) của giao diện, thực hiện sự chuyển đổi trạng thái của giao diện theo các sự kiện kích hoạt từ phía người dùng. Có sự tương ứng 1-1 giữa View và Controller

Chương 5. Thiết kế hệ thống

- 5.1. Giới thiệu về thiết kế hệ thống
- 5.2. Thiết kế lớp và phương thức
- 5.3. Thiết kế quản trị dữ liệu
- 5.4. Thiết kế giao diện tương tác Người-Máy
- 5.5. Thiết kế kiến trúc vật lý
- 5.6. Bài tập

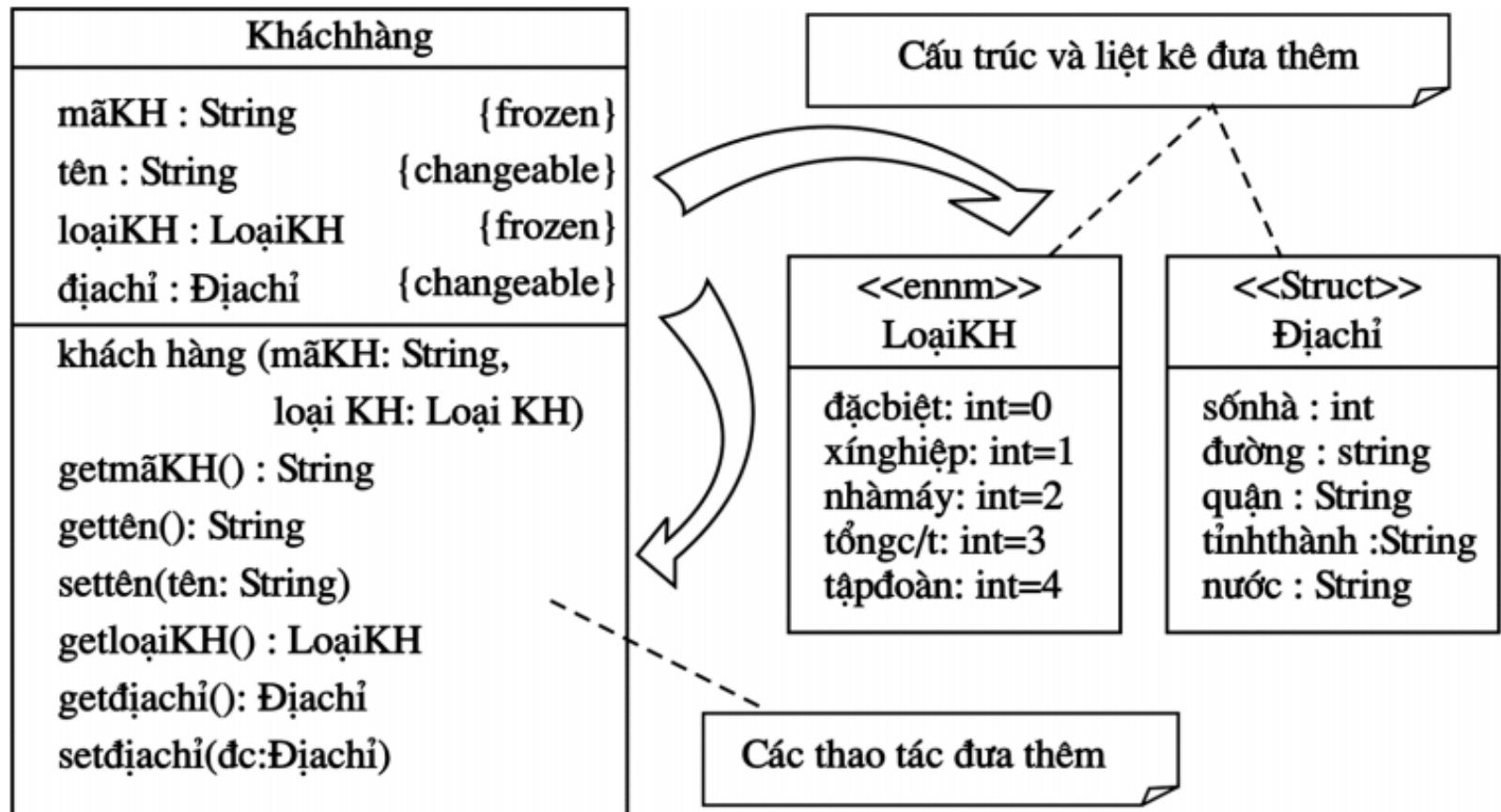
5.2. Thiết kế lớp

- Ánh xạ liên kết, xây dựng thuộc tính



Đơn hàng
ngườimua: Khách hàng dsdòngĐH: Vector<Dòng ĐH>
ngườimua(): Khách hàng thêmđòngĐH(d: DòngĐH) bótdòngĐH (d:DòngĐH) dsdòngĐH():Vector<DòngĐH>

5.2. Thiết kế lớp



5.3. Thiết kế quản trị dữ liệu

- Lưu trữ lâu dài các thông tin về một đối tượng
- Các vấn đề liên quan:
 - Lựa chọn cách lưu trữ: Tập? Cơ sở dữ liệu quan hệ (RDBMS)? OODBMS? noSQL DBMS
 - Ánh xạ lớp sang bảng CSDL

5.3. Thiết kế quản trị dữ liệu

- Sự tương đương giữa lớp và bảng

Mô hình đối tượng

Lớp

Thuộc tính có kiểu đơn

Thuộc tính có kiểu phức tạp

Đối tượng

OID

Liên kết và Kết nối

Thừa kế

Mô hình quan hệ

Bảng

Cột

Các cột, hay Bảng và khoá ngoài

Bộ-n

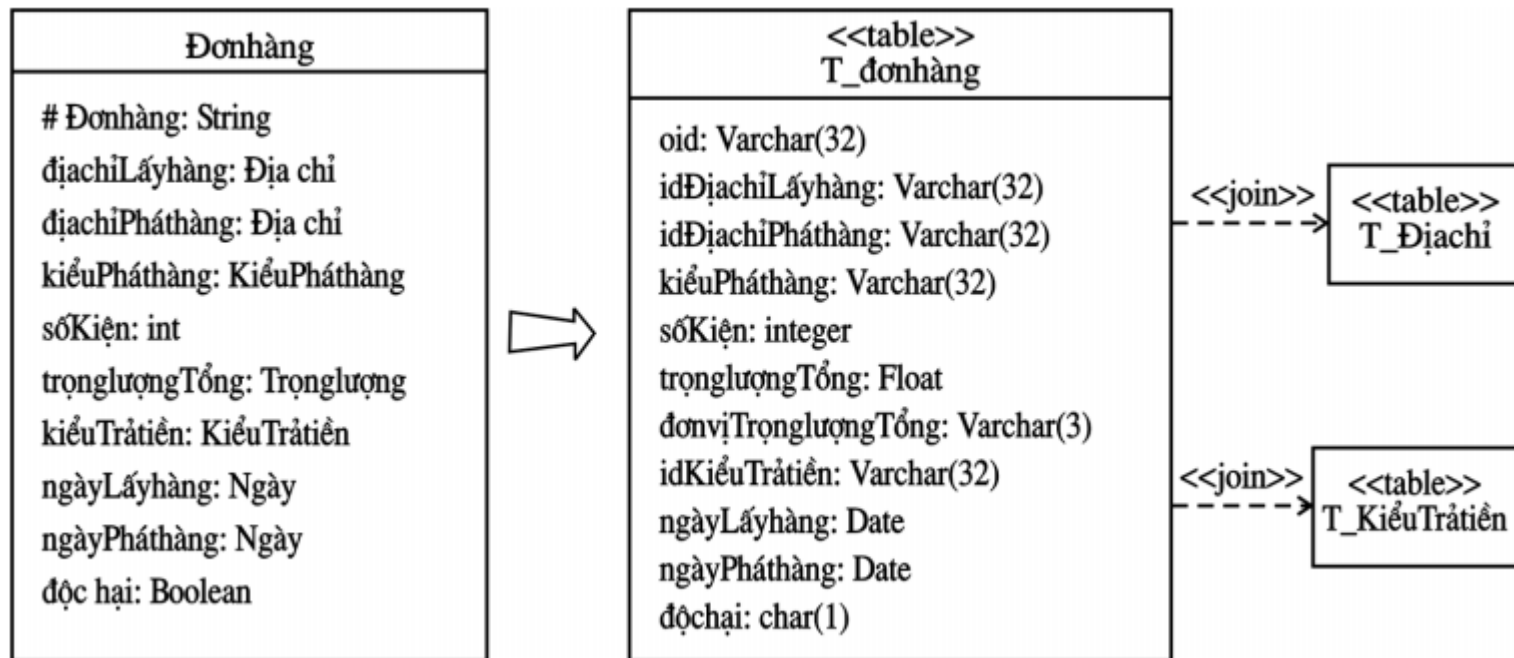
Khoá chính

Khoá ngoài hay bảng

Khoá chính đồng nhất trên nhiều bảng

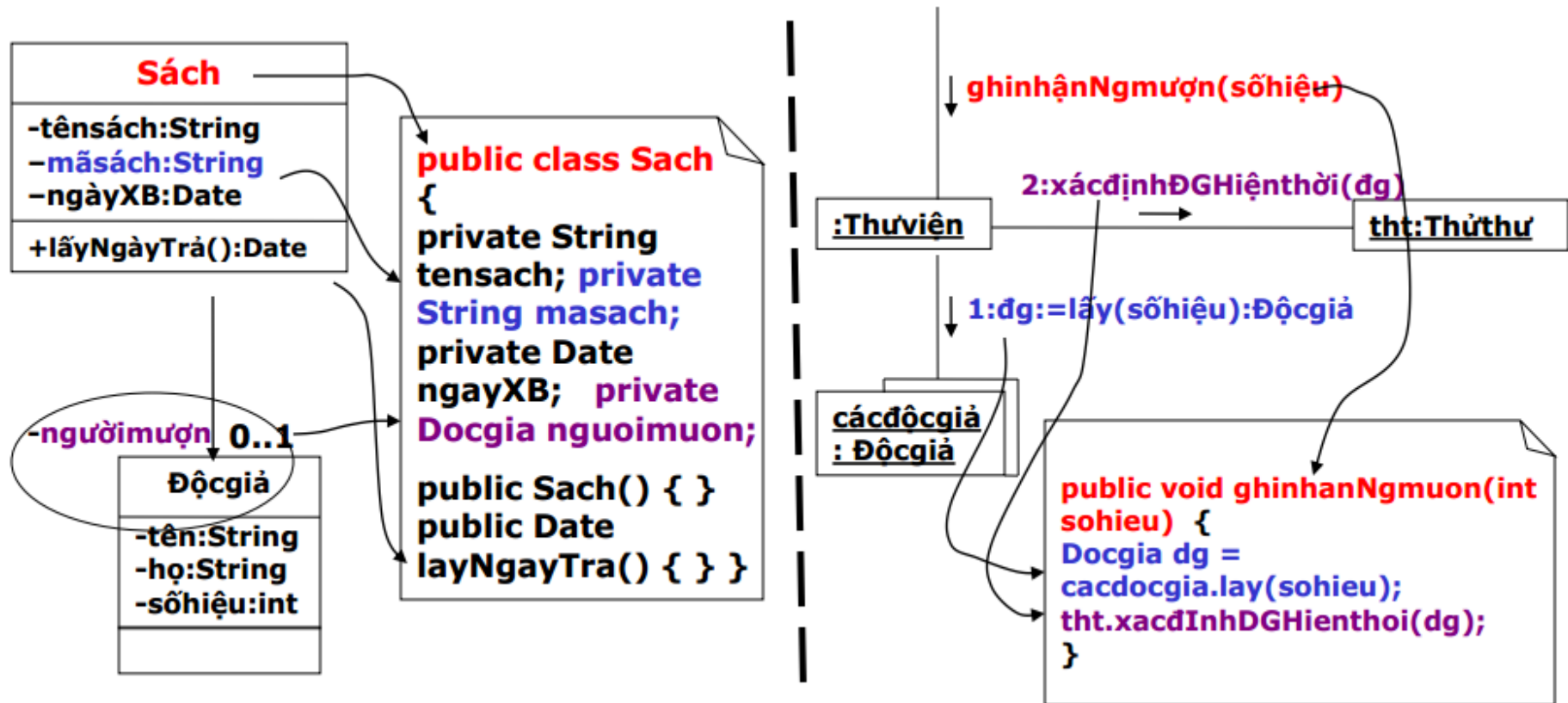
5.3. Thiết kế quản trị dữ liệu

- Ví dụ:



5.3. Thiết kế quản trị dữ liệu

- Sinh mã nguồn



5.4. Thiết kế giao diện tương tác người - máy

- Thiết kế giao diện xác định cách hệ thống sẽ tương tác với các thực thể bên ngoài (ví dụ: khách hàng, người dùng, các hệ thống khác)
 - Machine-Machine → tích hợp hệ thống
 - Human-machine (*)
- Phục vụ tương tác Người dùng-Hệ thống (UI)
 - Nguyên tắc thiết kế UI?
 - Quy trình?
 - Cách làm?
 - Điều hướng?
 - Đầu vào (input)?
 - Đầu ra (output)?

5.4. Thiết kế giao diện tương tác người - máy

- Dựa vào các công cụ tạo lập giao diện người dùng (GUI – Graphical user interface – design tools) ta thành lập sớm và nhanh một nguyên mẫu giao diện người dùng (GUI prototype), có tính thăm dò sự phù hợp, nhằm các mục đích:
 - Tạo ra một môi trường làm việc cụ thể, dễ tiếp xúc, dễ làm thử, làm cho người dùng trở nên yên tâm hơn, và năng động hơn trong việc đóng góp cho việc phát triển hệ thống
 - Qua quá trình dùng thử, ta thu thập được nhiều ý kiến phản hồi có ích từ phía người dùng
 - Sớm phát hiện được các yêu cầu hay chức năng bị bỏ sót, sớm nhìn thấy các điểm yếu, chỗ khó khăn nhất của hệ thống
- **Thiết kế giao diện không phải là các ảnh chụp màn hình hệ thống sau cài đặt!**

5.4. Thiết kế giao diện tương tác người - máy

- Như đã biết, cứ mỗi cặp (tác nhân, ca sử dụng) liên quan, có ít nhất một lớp biên để chuyển đổi các thông tin vào/ra
 - Thể hiện của lớp biên chính là giao diện mà bây giờ ta cần phải mô tả
- Muốn mô tả các lớp biên, ta xem lại từng bước trong kịch bản của mỗi ca sử dụng – để xác định các phần tử của giao diện:
 - Xét nội dung của tương tác giữa tác nhân và hệ thống
 - Các thông tin vào và ra
 - Các hành động được yêu cầu

5.4. Thiết kế giao diện tương tác người - máy

- Bản mô tả một giao diện thường chứa các điểm sau:
 - Tên của giao diện;
 - Mô tả tóm tắt nội dung của giao diện;
 - Mức độ phức tạp của giao diện (phức tạp/chuẩn/đơn giản);
 - Ghi chú thêm (nếu có);
 - Thiết kế chi tiết của giao diện, tùy thuộc vào 4 loại giao diện sau:
 - Các giao diện hội thoại (với người dùng)
 - Các giao diện hiển thị thông tin (kết quả xử lý, báo cáo,...)
 - Các giao diện dữ liệu từ/đến các hệ thống ngoài
 - Các giao diện chức năng đến các hệ thống ngoài

5.4. Thiết kế giao diện tương tác người - máy

- Làm nguyên mẫu giao diện
 - Bắt đầu càng sớm càng tốt, ngay sau khi phân tích các ca sử dụng
 - Đối với phiên bản đầu của nguyên mẫu giao diện, thì các trường là rỗng hoặc mang giá trị minh họa, các nút và các phần tử tương tác có thể chưa cần có hiệu ứng tương tác rõ ràng
 - Qua các vòng lặp tiếp theo (của việc làm nguyên mẫu), thì giao diện dần trở nên sinh động hơn (dần đi tới phương án cuối)
 - Như vậy, người dùng có thể làm việc thử với các nguyên mẫu giao diện

Chương 5. Thiết kế hệ thống

- 5.1. Giới thiệu về thiết kế hệ thống
- 5.2. Thiết kế lớp và phương thức
- 5.3. Thiết kế quản trị dữ liệu
- 5.4. Thiết kế giao diện tương tác Người-Máy
- 5.5. Thiết kế kiến trúc vật lý
- 5.6. Bài tập