Contents

[I. Yêu cầu chung 2](#_Toc488669499)

[II. Nội dung đề bài 2](#_Toc488669500)

[Ex01 – Environment Setup (1d) 2](#_Toc488669501)

[Ex02 – Java Core & OOP (1d) 2](#_Toc488669502)

[Ex03 – Oracle & SQL (1d) 3](#_Toc488669503)

[Ex04 – JDBC, Log & Exception (2d) 6](#_Toc488669504)

[Ex05 – HTML & CSS (1d) 7](#_Toc488669505)

[Ex06 – JavaScript & JQuery (1d) 7](#_Toc488669506)

[Ex07 - JSP & MVC Project (2d) 8](#_Toc488669507)

[Ex08 – Collections (1d) 8](#_Toc488669508)

[Ex09 – Servlet & Rest APIs (1d) 9](#_Toc488669509)

[Ex10 – AngularJS (3d) 9](#_Toc488669510)

[Ex11 – Spring (2d) 9](#_Toc488669511)

[Ex12 – Hibernate (2d) 9](#_Toc488669512)

[Ex13 – Tích hợp (2d) 9](#_Toc488669513)

# I. Yêu cầu chung

Các nội dung cần nộp cho mỗi bài bao gồm:

* Thư mục dự án (bao gồm source codes, các file cấu hình & thư viện dùng trong project)
* File Readme.txt, chứa các nội dung hướng dẫn mở, build codes, chạy test và các hướng dẫn khác

Các nội dung nộp cần đưa vào 1 file nén với tên AccName-ExNN.zip (trong đó AccName là tên Account GIT đã đăng ký, NN là số thứ tự của bài tập; ví dụ KienNT-Ex02.zip).

# II. Nội dung đề bài

## Ex01 – Environment Setup (1d)

Viết ứng dụng HelloWorld gồm

* 1 Java class,
* 1 page JSP nhập tên user,
* 1 JSP hiển thị lời chào (khởi tạo bởi Java class)

## Ex02 – Java Core & OOP (1d)

**Mô tả chung**

Viết chương trình để có thể lưu trữ và quản lý được thông tin về 3 loại xe (**Car**) khác nhau: Xe đời mới (**Modern Car**), Xe đời trung (**Medium Car**) ,Xe đời cũ (**Old Car**)

* Mỗi loại Xe đều có các thuộc tính là: Biển kiểm soát (**Number Plate**), Năm sản xuất (**Year of manufacture**), Hãng Sản xuất (**Brand**), Đã mua bảo hiểm(**Have Insurance**)
* Riêng xe đời mới có thêm thuộc tính là không có/có thiết bị định vị (**Have positioning device**), xe đời trung có thêm thuộc tính không/có tay lái trợ lực (**Have power steering**) và xe đời cũ thì có thêm thuộc tính số năm đã lưu hành (**Action Duration**).
* Theo quy định, các loại xe có các thuộc tính tương ứng như sau:

|  |  |
| --- | --- |
| **Car Type** | **Year of manufacture** |
| **Modern Car** | >= 2005 |
| **Medium Car** | 1996 - 2004 |
| **Old Car** | <= 1995 |

Table 1: Mô tả phân loại xe

**Yêu cầu chức năng**

Chương trình bao gồm các chức năng

1. Khởi tạo: tạo được danh sách gồm 10 chiếc xe có tên xe từ Car #(xy+1) đến Car #(xy+10), trong đó:

* xy là số lượng xe hiện tại đang được lưu (trong file hoặc CSDL)
* Các thuộc tính được tạo ngẫu nhiên, bao gồm: **Number Plate** (5 chữ số ngẫu nhiên), **Year of manufacture** (tạo ngẫu nhiên từ 1980-2012), **Brand** (tạo ngẫu nhiên trong danh sách TOYOTA, BMW, HUYNDAI), **Have Insurance** (tạo ngẫu nhiên true/false).
* Thông tin của 10 chiếc xemới tạo cần được lưu lại (file hoặc CSDL).

2. Quản lý các gói bảo hiểm: chương trình đang lưu danh sách các loại kiểu gói bảo hiểm (Package Type) như bảng bên dưới

|  |  |
| --- | --- |
| **Package type** | **Cho loại xe (Car Type)** |
| A | Đời mới (**Modern Car**) |
| B | Đời trung (**Medium Car)** |
| C | Đời cũ (**Old Car)** |

Table 2: Mô tả các gói bảo hiểm

Yêu cầu: cho phép tạo ra các gói bảo hiểm (**Insurance\_Package**) cụ thể để bán, mỗi lần tạo ra 10 gói có tên từ “Insurance Package#AB+1” đến “Insurance Package#AB+10”. Package\_Type cho mỗi gói bảo hiểm được chọn ngẫu nhiên từ các loại gói bảo hiểm đang có.

* AB là số lượng gói bảo hiểm đang có (đang được lưu trong hệ thống)
* Thông tin của các gói bảo hiểm đã tạo cần được lưu lại trong hệ thống (file hoặc CSDL)

3. Quản lý việc phân bổ gói bảo hiểm: cho phép người dùng chọn một xe cụ thể và mua một Insurance\_Package bất kì (trong danh sách các gói bảo hiểm chưa dùng).

* Nếu xe được chọn đã mua bảo hiểm rồi thì thông báo “Unavailable Buying!”
* Nếu**Car Type** của xekhông thể mua được Package tương ứng thì báo “Invalid Package!”
* Nếu Xe chưa mua bảo hiểm và Gói bảo hiểm khớp với loại xe thì hiện lên thông báo “**Successful Buying**”, đồng thời set thuộc tính **Have Insurance** là “**true**”.
* Các trường hợp còn lại thì hiện thông báo là “**Unsuccessful Buying** ”

4. Hiển thị: hiển thị danh sách các xe đã lưu

* Hiển thị tất cả hoặc chỉ những xe theo tiêu chí phân loại (đời cũ, đời trung hoặc đời mới).
* Các cột thông tin bao gồm Tên, Number Plate, Năm sản xuất, Loại Xe, Trạng thái bảo hiểm.

**Yêu cầu thiết kế**

* Dùng cơ chế lập trình hướng đối tượng để quản lý các thông tin, cụ thể theo chức năng dưới
* Chức năng 1, 2: cần thiết kế các classes để quản lý thông tin
  + Các xe ở mỗi loại
  + Các kiểu gói bảo hiểm
  + Các gói bảo hiểm
* Chức năng 3: cần xây dựng method assignAssurance() trong các class quản lý xe để thực hiện việc phân bổ gói bảo hiểm. Tham số truyền vào là thông tin gói bảo hiểm sẽ phân bổ
* Chức năng 4: xây dựng method showCarInfo() trong các class quản lý xe để hiển thị thông tin mỗi xe trong danh sách.

## Ex03 –Oracle & SQL (1d)

Cho cơ sở dữ liệu như sau



Trong đó:

* Bảng NHACUNGCAP lưu trữ dữ liệu về các đối tác cung cấp hàng cho công ty
* Bảng MATHANG lưu trữ dữ liệu về các mặt hàng hiện có trong công ty.
* Bảng LOAIHANG phân loại các mặt hàng hiện có.
* Bảng NHANVIEN có dữ liệu là thông tin về các nhân viên làm việc trong công ty.
* Bảng KHACHHANG được sử dụng để lưu giữ thông tin về các khách hàng của công ty.

Khách hàng đặt hàng cho cty thông qua các đơn đặt hàng. Thông tin chung về các đơn đạt hàng được lưu trữ trong DONDATHANG (mỗi một đơn đặt hàng phải do một nhân viên của cty lập và do đó bảng này có quan hệ với bảng NHANVIEN)

Thông tin chi tiết của các đơn đặt hàng (đặt mua hàng gì, số lượng, giá cả, …) được lưu trữ chi tiết trong bảng CHITIETDATHANG. Bảng này có quan hệ với DONDATHANG và MATHANG.

**Yêu cầu 1**

Viết các lệnh SQL để tạo cấu trúc bảng và ràng buộc bảng

**Yêu cầu 2**

Viết các câu lệnh SQL để tạo dữ liệu test + các câu lệnh SQL để giải quyết các bài toán sau => output gồm 2 file .sql: 1 file tạo dữ liệu test, 1 file chứa các câu lệnh query cho các yêu cầu bài toán.

1. Hãy cho biết có những khách hàng nào lại chính là đối tác cung cấp hàng của công ty (tức là có cùng tên giao dịch).
2. Những đơn đặt hàng nào yêu cầu giao hàng ngay tại cty đặt hàng và những đơn đó là của công ty nào?
3. Những mặt hàng nào chưa từng được khách hàng đặt mua?
4. Những nhân viên nào của công ty chưa từng lập bất kỳ một hoá đơn đặt hàng nào?
5. Trong năm 2003, những mặt hàng nào chỉ được đặt mua đúng một lần
6. Hãy cho biết mỗi một khách hàng đã phải bỏ ra bao nhiêu tiền để đặt mua hàng của công ty?
7. Mỗi một nhân viên của công ty đã lập bao nhiêu đơn đặt hàng (nếu nhân viên chưa hề lập một hoá đơn nào thì cho kết quả là 0)
8. Cho biết tổng số tiền hàng mà cửa hàng thu được trong mỗi tháng của năm 2003 (thời được gian tính theo ngày đặt hàng).
9. Hãy cho biết tổng số lượng hàng của mỗi mặt hàng mà cty đã có (tổng số lượng hàng hiện có và đã bán).
10. Nhân viên nào của cty bán được số lượng hàng nhiều nhất và số lượng hàng bán được của nhân viên này là bao nhiêu?
11. Mỗi một đơn đặt hàng đặt mua những mặt hàng nào và tổng số tiền mà mỗi đơn đặt hàng phải trả là bao nhiêu?
12. Hãy cho biết mỗi một loại hàng bao gồm những mặt hàng nào, tổng số lượng hàng của mỗi loại và tổng số lượng của tất cả các mặt hàng hiện có trong công ty là bao nhiêu?
13. Thống kê xem trong năm 2003, mỗi một mặt hàng trong mỗi tháng và trong cả năm bán được với số lượng bao nhiêu.
14. Cập nhật lại giá trị NGAYCHUYENHANG của những bản ghi có giá trị NGAYCHUYENHANG chưa xác định (NULL) trong bảng DONDATHANG bằng với giá trị của trường NGAYDATHANG.
15. Cập nhật giá trị của trường NOIGIAOHANG trong bảng DONDATHANG bằng địa chỉ của khách hàng đối với những đơn đặt hàng chưa xác định được nơi giao hàng (có giá trị trường NOIGIAOHANG bằng NULL)
16. Cập nhật lại dữ liệu trong bảng KHACHHANG sao cho nếu tên công ty và tên giao dịch của khách hàng trùng với tên công ty và tên giao dịch của một nhà cung cấp nào đó thì địa chỉ, điện thoại, fax và email phải giống nhau.
17. Tăng lương lên gấp rưỡi cho những nhân viên bán được số lượng hàng nhiều hơn 100 trong năm 2003
18. Tăng phụ cấp lên bằng 50% lương cho những nhân viên bán được hàng nhiều nhất.
19. Giảm 25% lương của những nhân viên trong năm 2003 ko lập được bất kỳ đơn đặt hàng nào
20. Giả sử trong bảng DONDATHANG có them trường SOTIEN cho biết số tiền mà khách hàng phải trả trong mỗi dơnđặt hàng. Hãy tính giá trị cho trường này.

Xoá khỏi bảng MATHANG những mặt hàng có số lượng bằng 0 và không được đặt mua trong bất kỳ đơn đặt hàng nào.

**Yêu cầu 3**

1. Tạo thủ tục lưu trữ để thông qua thủ tục này có thể bổ sung thêm một bản ghi mới cho bảng MATHANG (thủ tục phải thực hiện kiểm tra tính hợp lệ của dữ liệu cần bổ sung: không trùng khoá chính và đảm bảo toàn vẹn tham chiếu)
2. Tạo thủ tục lưu trữ có chức năng thống kê tổng số lượng hàng bán được của một mặt hàng có mã bất kỳ (mã mặt hàng cần thống kê là tham số của thủ tục).
3. Viết trigger cho bảng CHITIETDATHANG theo yêu cầu sau:

* Khi một bản ghi mới được bổ sung vào bảng này thì giảm số lượng hàng hiện có nếu số lượng hàng hiện có lớn hơn hoặc bằng số lượng hàng được bán ra. Ngược lại thì huỷ bỏ thao tác bổ sung.
* Khi cập nhật lại số lượng hàng được bán, kiểm tra số lượng hàng được cập nhật lại có phù hợp hay không (số lượng hàng bán ra không được vượt quá số lượng hàng hiện có và không được nhỏ hơn 1). Nếu dữ liệu hợp lệ thì giảm (hoặc tăng) số lượng hàng hiện có trong công ty, ngược lại thì huỷ bỏ thao tác cập nhật.

1. Viết trigger cho bảng CHITIETDATHANG để sao cho chỉ chấp nhận giá hàng bán ra phải nhỏ hơn hoặc bằng giá gốc (giá của mặt hàng trong bảng MATHANG)

## Ex04 – JDBC, Log & Exception (2d)

Viết một chương trình Java (Java console) cho phép các giáo vụ quản lý việc đào tạo ở một trung tâm đào tạo theo các mô tả như sau:

* Khi vào ứng dụng, giáo vụ được yêu cầu nhập các thông tin gồm: account, first-name, last-name, email. Những giá trị này bắt buộc, account là duy nhất trong ứng dụng.
* Trường hợp account đã tồn tại thì ứng dụng sẽ lưu lại thông tin giáo vụ, các thông tin đã bị thay đổi cũng như thời gian truy cập của giáo vụ.
* Sau khi đăng nhập thành công, giáo vụ được chuyển đến màn hình gồm danh sách các học viên (ID, Họ tên, email, SĐT, lớp, ghi chú) của các lớp minh phụ trách. Từ đây cho phép giáo vụ có thể
  + Lọc danh sách học viên theo lớp,
  + Tìm hoặc sắp xếp học viên theo 1 trong các tiêu chí họ, tên, email, SĐT
  + Thêm, sửa, xóa một học viên cụ thể

Yêu cầu bổ sung:

* Giao diện cần dễ đọc, thao tác
* Khi nhập thông tin cần có các kiểm tra tương ứng
* Cần xử lý đầy đủ các ngoại lệ có thể xảy ra trong ứng dụng
* Có cơ chế log các thông tin ra file ở các mức: debug , cảnh báo (warn), lỗi (error)

## Ex05 – HTML& CSS (1d)

Dựa vào nội dung website tin tức cụ thể với nội dung trong ngày giao bài, dựng lại nội dung các trang (ở trạng thái ngày tham khảo) để xây dựng 1 website tin tức gồm 3 trang như liệt kê bên dưới.

(1) Home Page

(2) Trang chuyên mục

(3) Chi tiết 1 tin/bài, hiển thị sau khi click link trên (1) hoặc (2)

Yêu cầu

* **Không được để lại code thừa, code không có ý nghĩa trong các file CSS, HTML**
* Mỗi người được giao làm 1 site riêng ko trùng nhau, sử dụng HTML, CSS và JavaScript
* Sử dụng chung các styles để trong 1 file CSS => output gồm 3 file HTML, 1 file CSS

Các website tin tức lựa chọn

|  |  |  |  |
| --- | --- | --- | --- |
| <http://vnexpress.net>  <http://vietnamnet.vn>  <http://tuoitre.vn>  <http://thanhnien.vn>  <http://tintuc.vn> | <http://dantri.com.vn>  <http://ictnews.vn>  <http://cafef.vn>  <http://cafebiz.vn>  <http://vneconomy.vn> | <http://bbc.com>  <http://cnn.com>  <https://techtalk.vn>  <http://genk.vn/>  <http://www.techz.vn/> | <http://vnmedia.vn/>  <https://quantrimang.com>  <http://24h.com.vn>  <http://tintuconline.com.vn>  … |

## Ex06 – JavaScript & JQuery (1d)

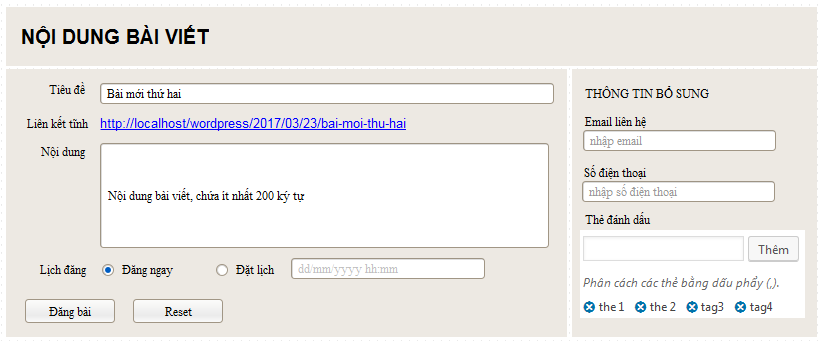
Thiết kế chức năng CMS (dùng HTML, CSS, JS) để xem danh sách và nhập thông tin bài viết gồm 2 trang (màn hình) như mô tả bên dưới

**A. Danh sách bài viết**

* Hiển thị >=10 bài viết samples với các cột thông tin: Chọn, Tiêu đề bài viết, Danh mục, Lịch đăng, Trạng thái, Thao tác
* Cột trạng thái hiển thị trạng thái bài viết, gồm các trạng thái: Duyệt, Chặn hoặc Đặt lịch
* Cột thao tác hiển thị các icons với các actions tương ứng như sau (xử lý dùng JS)
  + Icon Duyệt: hiển thị khi bài viết đang ở trạng thái Chặn hoặc Đặt lịch; khi click vào thì sẽ chuyển trạng thái bài viết sang Duyệt (có alert xác nhận trước khi thực hiện)
  + Icon Chặn: hiển thị khi bài viết đang ở trạng thái Duyệt hoặc Đặt lịch; khi click vào thì sẽ chuyển trạng thái bài viết sang Chặn (có alert xác nhận trước khi thực hiện)
  + Icon Xem: click vào thì sẽ hiển thị trang Chi tiết, hiển thị ở trạng thái Read-Only (ko cho phép user sửa đổi các nội dung + ko hiển thị các nút Đăng bài hay Reset)
  + Icon Edit: click vào thì sẽ hiển thị trang Chi tiết, hiển thị ở trạng thái Editable (cho phép user sửa đổi các nội dung + hiển thị các nút Đăng bài và Reset)
  + Icon Xóa: click vào thì cho phép user xóa bài viết tương ứng (xác nhận trước khi xóa)
* Cho phép chọn 1 hoặc nhiều bài viết trong danh sách để Duyệt, Chặn hoặc Xóa theo batch
* Sau khi duyệt, chặn, xóa thì cần cập nhật lại danh sách bài viết tương ứng (dùng JS)

**B. Màn hình Chi tiết bài viết:**

Hiển thị dưới dạng PopUp với giao diện mô phỏng như sau (cần bổ sung CSS styles để màn hình đẹp/thân thiện hơn)



Các yêu cầu:

(1) kiểm tra bằng JavaScript như sau (sau khi click nút Đăng bài):

1. Tiêu đề bài viết ko được trống
2. Liên kết tĩnh: tự động được generate ra, sau khi nhập tiêu đề bài viết (là valid URL)
3. Nội dung: bắt buộc nhập, có >=200 ký tự, không bao gồm các ký tự đặc biệt (>, <, &, “)
4. Khi chọn Lịch đăng = Đăng ngay thì ô textbox bên cạnh sẽ được enabled (và ngược lại); Dữ liệu nhập vào phải theo đúng định dạng ngày-giờ là dd/MM/yyyy hh:mm
5. Cần kiểm tra định dạng email và số điện thoại nhập vào, bắt buộc nhập định dạng đúng
6. Nhập nội dung thẻ đánh dấu xong enter thì từ nào chưa có trong DS bên dưới sẽ được tự động bổ sung; click dấu x trong DS bên dưới sẽ tự động xóa từ đó khỏi DS;

*Trong quá trình nhập, nếu bấm nút Đăng bài mà có bất kỳ vấn đề nào vi phạm các yêu cầu trên thì sẽ thông báo lỗi tương ứng; Nếu bấm nút Reset thì sẽ xóa tất cả các nội dung đã nhập*

(2) Sau khi click reset thì tất cả các dữ liệu đã nhập trở về trạng thái ban đầu (như mới load trang)

## Ex07 - JSP& MVC Project (2d)

Xây dựng module BackEnd web để quản lý danh mục ô tô, tin tức của 1 doanh nghiệp

* Quản lý tin tức bao gồm các thông tin và logics như đề cập trong bài tập **Ex07**
* Quản lý ô tô bao gồm cả thông tin về ô tô và các gói bảo hiểm áp dụng cho các ô tô. Các logics chi tiết được đề cập trong bài tập **Ex02**

Yêu cầu:

* Giao diện xây dựng đơn giản, rõ ràng và thao tác
* Thực thi xây dựng ứng dụng dạng Web-based, theo mô hình MVC (JSP Model 2), tham khảo <https://www.javatpoint.com/model-1-and-model-2-mvc-architecture>

## Ex08 – Collections (1d)

Sử dụng các đối tượng Collections phù hợp để xây dựng module FrontEnd web cung cấp thông tin tới khách hàng của doanh nghiệp với các dữ liệu được đề cập trong bài **Ex07**

1/ Bổ sung chức năng thực hiện "mua bảo hiểm" cho các ô tô

2/ Bổ sung chức năng hiển thị tin tức: danh sách (có sắp xếp, phân trang) và chi tiết tin

3/ Xây dựng cơ chế caching cho việc QL tin tức (ko thường xuyên truy vấn DB nếu dữ liệu ko có thay đổi)

Như vậy module FrontEnd web cần xây dựng gồm các trang sau:

* Home: hiển thị danh mục tin tức, DS các tin hot, danh sách các ô tô đang chưa có bảo hiểm
* Link từ Home: hiển thị DS tất cả các tin, tất cả các ô tô (sắp xếp theo thứ tự tạo)
* Có login, từ đó cho phép user mua bảo hiểm cho 1 ô tô cụ thể (có theo dõi lại)
* Tin tức chi tiết

## Ex09–Servlet & Rest APIs (1d)

Tách việc xử lý dữ liệu và thao tác cơ sở dữ liệu (các Model) trong bài tập **Ex07** ra thành project riêng dưới dạng các RESTful APIs viết bằng Java Servlet.

Lưu ý: chưa cần tích hợp ngay, có thể test được các APIs thông qua Postman (thuộc Chrome)

## Ex10–AngularJS (3d)

1/ Tự tìm hiểu và làm các ví dụ (có nộp lại bài) theo hướng dẫn trên <https://www.w3schools.com/angular/default.asp>

2/ Chuyển đổi các trang FrontEnd như mô tả trong bài Ex08 sang dùng Angular JS ở phía Controller & View (phía Model có thể tạm gán cứng dữ liệu trả về)

3/ Chuyển đổi các trang BackEnd như mô tả trong bài **Ex07** sang dùng Angular JS ở phái Controller và View, phía Model thì gọi đến các Servlet API đã viết trong bài **Ex09**

## Ex11 – Spring (2d)

1/ Chuyển đổi các APIs đã viết trong bài **Ex09** sang sử dụng Spring thay vì Servlet

2/ Tách việc xử lý dữ liệu và thao tác cơ sở dữ liệu (các lớp Model) trong bài **Ex08** ra thành project riêng, cung cấp dịch vụ dưới dạng các RESTful APIs viết bằng Spring

## Ex12 – Hibernate (2d)

1/ Cập nhật bài tập Ex04, chuyển từ việc dùng JDBC sang Hibernate cho việc thao tác CSDL (dùng class để cấu hình Hibernate)

2/ Cập nhật bài tập Ex11, chuyển từ việc dùng JDBC sang Hibernate cho việc thao tác CSDL (dùng file xml để cấu hình Hibernate)

## Ex13 – Tích hợp (2d)

Hơpoàn thiện hệ thống như mô tả trong các bài **Ex07** và **Ex08** (lưu ý kế thừa các kết quả từ các bài Ex09-Ex12), trong đó chia làm 3 module chạy độc lập nhau, phía BackEnd có chức năng đăng nhập (phải đăng nhập và được cấp quyền mới được thao tác trên chức năng tương ứng)

* Server: các API viết sử dụng framework Spring, thao tác cơ sở dữ liệu thông qua Hibernate
* FrontEnd và BackEnd
  + Các giao diện thao tác triển khai dùng Angular JS, HTML, CSS, JS, JQuery;
  + Xử lý dữ liệu và thao tác với cơ sở dữ liệu gọi tới các Restful API phía Server