**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo

Description automatically generated

**BÁO CÁO ĐỒ ÁN**

**NGÔN NGỮ LẬP TRÌNH TIÊN TIẾN**

**Giảng viên hướng dẫn: ThS. Trần Văn Định**

**Sinh viên thực hiện: Nguyễn Phúc An 19110312**

**Bành Đăng Khoa 19110378**

**Lâm Hoàng Phúc 19110433**

TP. Hồ Chí Minh, tháng 12 năm 2021

# MỤC LỤC

# Cơ sở lý thuyết

## Lập trình hướng đối tượng.

- Lập trình hướng đối tượng (**Object Oriented Programing** – OOP) là một phương pháp để thiết kế một chương trình bởi sử dụng các lớp và các đối tượng.

- Java là một ngôn ngữ lập trình hướng đối tượng vì vậy nó cũng hỗ trợ các đặc tính của lập trình hướng đối tượng:

* Đa hình (Polymorphism): Khi một nhiệm vụ được thực hiện bởi nhiều cách khác nhau, tính chất này được gọi là đa hình. Ví dụ có nhiều các để thuyết phục các khách hàng khác nhau, để vẽ một cái gì đó như hình tròn, hình chữ nhật, ...

Trong java, để áp dụng tính đa hình chúng ta sử dụng phương thức orverloading hoặc overriding.

* Thừa kế (Inheritance): Khi một đối tượng được truyền lại tất cả các thuộc tính và phương thức của đối tượng cha được gọi là kế thừa. Kế thừa giúp tái sử dụng lại mã nguồn. Nó được sử dụng cho đa hình lúc runtime.
* Đóng gói (Encapsulation): Việc ràng buộc giữa code và data với nhau tạo thành một khối duy nhất được biết đến là đóng gói. Ví dụ: viên thuốc con nhộng được đóng gói với nhiều loại thuốc bên trong.

Một class trong java là một ví dụ về đóng gói. Java bean là một lớp được đóng gói hoàn toàn vì tất cả các dữ liệu thành viên là private.

* Trừu tượng (Abstraction): Trừu tượng là sự ẩn đi những chi tiết bên trong và hiển thị ra các chức năng, tính chất này gọi là trừu tượng. Ví dụ: khi gọi điện thoại chúng ta không biết xử lý nội bộ thế nào, khi đi xe máy cũng vậy, mà chúng ta chỉ biết đến các chức năng thông qua giao tiếp bên ngoài.

Trong java, chúng ta áp dụng tính chất trừu tượng bằng cách sử dụng abstract class và interface.

**1. Đối tượng (object)**

- Đối tượng là một thực thể có trạng thái và hành vi. Nó có thể mang tính vật lý hoặc logic.

- Nếu chúng ta xem xét thực tế chúng ta có thể tìm thấy nhiều đồ vật xung quanh chúng ta: cái bàn, con chó, con người, v.v… Tất cả các đối tượng này đều có thuộc tính và hành vi.

- Nếu chúng ta xem xét một con chó, thuộc tính của nó sẽ là – tên, giống, màu sắc, và các hành vi là: sủa, chạy, ăn, … Nếu bạn so sánh các đối tượng trong phần mềm với một đối tượng trong thế giới thực, chúng sẽ có đặc điểm rất giống nhau: thuộc tính đối tượng trong phần mềm được lưu trữ trong trường (field) và hành vi được lưu trữ trong phương thức (method).

**2. Lớp (Class**)

- Chúng ta có thể xem lớp như một khuôn mẫu (template) của đối tượng (Object). Trong đó bao gồm dữ liệu của đối tượng (fields hay properties) và các phương thức(methods) tác động lên thành phần dữ liệu đó gọi là các phương thức của lớp.

**3. Package**

- Một package (gói) trong java là một nhóm các kiểu tương tự của các lớp, giao diện và các package con .

- Package trong java có thể được phân loại theo hai hình thức, package được dựng sẵn và package do người dùng định nghĩa.

- Có rất nhiều package được dựng sẵn như java, lang, net, io, util, sql, …

**4. Constructor**

- Constructor trong java là một dạng đặc biệt của phương thức được sử dụng để khởi tạo các đối tượng.

- Java Constructor được gọi tại thời điểm tạo đối tượng. Nó khởi tạo các giá trị để cung cấp dữ liệu cho các đối tượng, đó là lý do tại sao nó được gọi là constructor.

- Khai báo của Constructor giống với khải báo của method (phương thức). Nó phải có **cùng tên với class** (lớp) và **không có giá trị trả về**.

- Có 2 kiểu của constructor:

+ Constructor mặc định (không có tham số truyền vào – **default constructors**).

+ Constructor tham số (**parameterized constructors**).

**5. Phạm vi truy cập (Access modifier)**

- Có hai loại modifier trong java: **access modifiers** và **non-access modifiers**.

- Các access modifiers trong java xác định độ truy cập (Phạm vi) vào dữ liệu của của các trường (field), phương thức (method), cấu tử (constructor) hoặc lớp (class).

- Có 4 kiểu của java access modifiers:

+ private.

+ (Mặc định).

+ protected.

+ public.

## Kết nối cơ sở dữ liệu với JDBC

**- JDBC (Java Database Connectivity)** là một API chuẩn dùng để tương tác với các loại cơ sở dữ liệu quan hệ (database relationship). JDBC bao gồm một tập hợp các class và các interface dùng cho ứng dụng Java có thể giao tiếp với các cơ sở dữ liệu (database) khác nhau.

[Diagram

Description automatically generated](https://gpcoder.com/wp-content/uploads/2019/09/java-jdbc-components.gif)

- JDBC có thể làm việc với bất kỳ cơ sở dữ liệu ([**MySQL**](https://www.mysql.com/), [**PostgreSQL**](https://www.postgresql.org/), [**Oracle**](https://www.oracle.com/), [**SQL Server**](https://www.microsoft.com/en-us/sql-server/sql-server-downloads), …) dựa vào một **Driver** được cung cấp.

- JDBC API bao gồm hai package chính:

* **java.sql** : là một phần của Java standard.
* **javax.sql** : là một phần của Java enterprise.

- Các thành phần của JDBC:

[Diagram

Description automatically generated](https://gpcoder.com/wp-content/uploads/2019/09/jdbc-components-interaction.png)

* **DriverManager**: là một class quản lý danh sách các Driver (database drivers). Các yêu cầu kết nối từ ứng dụng Java sẽ được class này tìm kiếm Driver phù hợp đầu tiên để thiết lập kết nối với cơ sở dữ liệu.
* **Driver**: là một interface dùng để xử lý các giao tiếp với cơ sở dữ liệu. Thông thường ứng dụng Java sẽ không giao tiếp trực tiếp với class này mà thông qua DriverManager.
* **Connection** : là một interface cung cấp tất cả các method cần thiết cho việc giao tiếp với database. Interface này chứa nhiều phương thức đa dạng để tạo kết nối với một Database. Tất cả các thông tin giao tiếp với cơ sở dữ liệu chỉ có thể thông qua đối tượng Connection. Một Connection đại diện cho một phiên (session) làm việc với cơ sở dữ liệu.
* **Statement** : là một interface cho phép gửi các câu lệnh SQL tới Database. Ngoài ra, một số Interface kết thừa từ nó cung thêm các tham số để thực thi các thủ tục đã được lưu trữ (stored procedure).
* **ResultSet**: đại diện cho tập hợp các bản ghi (record) có được sau khi thực hiện truy vấn (query).
* **SQLException** : class này xử lý bất cứ lỗi nào xuất hiện trong khi làm việc với Database.

## Java Swing

**- Java Swing** là cách gọi rút gọn khi người ta nhắc đến Swing của [**Java Foundation**](https://en.wikipedia.org/wiki/Java_Foundation_Classes) (JFC). Nó là bộ công cụ GUI mà Sun Microsystems phát triển để xây dựng các ứng dụng tối ưu dùng cho window (bao gồm các thành phần như nút, thanh cuộn,…).

- [**Swing**](https://en.wikipedia.org/wiki/Swing_(Java)) được xây dựng trên AWT API và hoàn toàn được viết bằng Java. Tuy nhiên, nó lại khác với AWT ở chỗ bộ công cụ này thuộc loại nền tảng độc lập, bao gồm các thành phần nhẹ và phức tạp hơn AWT.

- Các gói javax.swing bao gồm các lớp cho Java Swing API như JMenu, JButton, JTextField, JRadioButton, JColorChooser,…

- Việc xây dựng ứng dụng sẽ trở nên dễ dàng hơn với **Java Swing** vì chúng ta có các bộ công cụ GUI giúp đỡ công việc.

- Swing được chính thức phát hành vào tháng 3/1998. Nó đi kèm với thư viện Swing 1.0 với hơn 250 lớp, 80 giao tiếp.

- Hiện nay con số này đã được tăng lên, ở phiên bản Swing 1.4 có 451 lớp và 85 giao tiếp.

- Các bản phát hành Java 2 (SDK 1.2 và các phiên bản mới hơn) đều bao gồm Swing trong runtime environment.

**- Các component có trong Java Swing:**

1. **JLabel**

- Lớp JLabel có thể hiển thị hoặc text, hoặc hình ảnh hoặc cả hai. Các nội dung của Label được gán bởi thiết lập căn chỉnh ngang và dọc trong khu vực hiển thị của nó. Theo mặc định, các label được căn chỉnh theo chiều dọc trong khu vực hiển thị. Theo mặc định, text-only label là căn chỉnh theo cạnh, image-only label là căn chỉnh theo chiều ngang.

- Constructor của JLabel:

+ JLabel(): Tạo một instance của JLabel, không có hình ảnh, và với một chuỗi trống cho title.

+ JLabel(Icon image): Tạo một instance của JLabel với hình ảnh đã cho.

+ JLabel(Icon image, int horizontalAlignment): Tạo một instance của JLabel với hình ảnh và căn chỉnh ngang đã cho.

+ JLabel(String text): Tạo một instance của JLabel với text đã cho.

+ JLabel(String text, Icon icon, int horizontalAlignment): Tạo một instance của JLabel với text, hình ảnh, và căn chỉnh ngang đã cho.

+ JLabel(String text, int horizontalAlignment): Tạo một instance của JLabel với text và căn chỉnh ngang đã cho.

2. **JCheckBox**

- Lớp JCheckBox là một trình triển khai của một checkbox, là một item mà có thể được lựa chọn (selected) hoặc không được lựa chọn (unselected), và hiển thị trạng thái của nó tới người dùng.

- Các Constructor của JCheckBox:

+ JCheckBox(): Tạo một unselected checkbox ban đầu không có text và icon.

+ JCheckBox(Action a): Tạo một checkbox, với các thuộc tính được lấy từ Action đã cho.

+ JCheckBox(Icon icon): Tạo một unselected checkbox với một icon.

+ JCheckBox(Icon icon, boolean selected): Tạo một checkbox với một icon và xác định rằng ban đầu nó là selected hoặc không .

+ JCheckBox(String text): Tạo một unselected checkbox ban đầu với text.

+ JCheckBox(String text, boolean selected): Tạo một checkbox với text và xác định rằng ban đầu nó là selected hoặc không.

+ JCheckBox(String text, Icon icon): Tạo một unselected checkbox ban đầu với text và icon đã cho.

+ JCheckBox(String text, Icon icon, boolean selected): Tạo một checkbox với text và icon, và xác định rằng ban đầu nó là selected hoặc không.

3. **JButton**

- Lớp JButton được sử dụng để tạo một nút button mà có trình triển khai là độc lập nền tảng. Thành phần này có một label và tạo một sự kiện (event) khi được nhấn. Nó cũng có thể có Image.

- Các Construtor của JButton:

+ JButton(): Tạo một button mà không thiết lập text hoặc icon.

+ JButton(Action a): Tạo một button tại đây các thuộc tính được nhận từ Action đã cung cấp.

+ JButton(Icon icon): Tạo một button với một icon.

+ JButton(String text): Tạo một button với text.

+ JButton(String text, Icon icon): Tạo một button với text ban đầu và một icon.

4. **JTextField**

- Lớp JTextField là một thành phần cho phép sửa đổi một dòng text đơn.

- Các Constructor của JTextField:

+ JTextField(): Xây dựng một TextField mới.

+ JTextField(Document doc, String text, int columns): Xây dựng một JTextField mới mà sử dụng mô hình lưu trữ text đã cho và số cột đã cho.

+ JTextField(int columns): Xây dựng một TextField mới và trống với số cột đã cho.

+ JTextField(String text): Xây dựng một TextField mới được khởi tạo với text đã cho.

+ JTextField(String text, int columns): Xây dựng một TextField mới được khởi tạo với text và các cột đã cho.

5. **JCombobox**

- Lớp JComboBox là một thành phần mà kết hợp một button, một trường có thể chỉnh sửa và một drop-down list. Tại một thời điểm chỉ có một item có thể được lựa chọn từ list.

- Các Constructor của JCombobox:

+ JComboBox(): Tạo một JComboBox với data model mặc định.

+ JComboBox(Object[] items): Tạo một JComboBox mà chứa các phần tử trong mảng đã cho.

+ JComboBox(Vector<?> items): Tạo một JComboBox mà chứa các phần tử trong Vector đã cho.

6. **JTable**

- Lớp JTable được sử dụng để hiển thị dữ liệu trên các ô của bảng hai chiều. Các constructor được sử dụng phổ biến của lớp JTable là:

+ JTable(): Tạo một bảng với các ô trống.

+ JTable(Object[][] rows, Object[] columns): Tạo một bảng với dữ liệu đã cho.

**- Ngoài ra trong Java Swing còn rất nhiều component khác, ở đây em chỉ liệt kê những component có sử dụng trong đồ án cuối kỳ.**

**Ứng dụng của Java Swing:**

**- Java Swing** được dùng để hỗ trợ tạo giao diện đồ hoạ người dùng (với Java).

Bộ công cụ này cung cấp các bộ điều khiển nâng cao như thanh trượt, colorpicker, Tree, TabbedPane và bảng điều khiển,..

- Swing có những đặc điểm:

* Độc lập với thiết bị
* Có thể tuỳ chỉnh, mở rộng
* Khá nhẹ
* Có thể cấu hình

- Ngoài ra bạn cũng có thể tùy chỉnh các điều khiển xoay một cách dễ dàng mà không ảnh hưởng đến các thành phần khác.

# Giới thiệu hệ thống

- Hệ thống quản lý showroom bán xe, hệ thống cho phép người dùng quản lý nhập, xuất thông tin của các loại xe mà cửa hàng kinh doanh, nhập xuất thông tin của nhân viên, quản lý thông tin, tài khoản của nhân viên, tính lương của nhân viên, tạo hoá đơn mua bán, quản lý thông tin của khách hàng.

## Thiết kế đối tượng và chức năng:

+ Đối tượng cơ sở dữ liệu:

1. Nhân viên (nhanvien)

+ id int

+ cmnd varchar(255)

+ ten nvarchar(255)

+ diachi nvarchar(255)

+ chucvu nvarchar(255)

+ sdt nvarchar(255)

+ luong int

2. Tài khoản (account)

+ username varchar(255)

+ password varchar(255)

+ usertype varchar(255)

3. Xe (xe)

+ id int

+ model varchar(255)

+ mauxe nvarchar(255)

+ thuonghieu nvarchar(255)

+ loai nvarchar(255)

+ phankhoi int

+ xuatxu nvarchar(255)

+ trangthai nvarchar(255) “ĐÃ BÁN” “CHƯA BÁN”

+ gia int

4. Khách hàng (khachhang)

+ id int

+ cmnd varchar(255)

+ ten nvarchar(255)

+ diachi nvarchar(255)

+ sdt varchar(255)

5. Hoá đơn (bill)

+ idbill int

+ idsale int

+ idkhachhang int

+ tonghoadon int

6. Chi tiết hoá đơn (billdetail)

+ idbill int

+ idxe int

+ gia int

## Chức năng theo đối tượng

1. Nhân viên (nhanvien)

+ Nhập thông tin cho từng nhân viên.

+ Xuất thông tin cho từng nhân viên.

+ Xoá thông tin cho từng nhân viên.

+ Chỉnh sửa thông tin cho từng nhân viên.

+ Tạo tài khoản cho nhân viên.

+ Đổi mật khẩu cho tài khoản.

+ Tính lương.

+ Sale có khả năng tạo bill và được hưởng hoa hồng từ tổng bill.

2. Xe (xe)

+ Nhập thông tin cho từng xe.

+ Xuất thông tin cho từng xe.

+ Xoá thông tin cho xe.

+ Chỉnh sửa thông tin cho từng xe.

4. Khách hàng (khachhang)

+ Nhập thông tin cho từng khách hàng.

+ Xuất thông tin cho từng khách hàng.

+ Xoá thông tin khách hàng.

+ Chỉnh sửa thông tin cho từng nhân viên.

5. Hoá đơn (bill)

+ Tạo hoá đơn buôn bán.

+ Tính tổng giá trị các hoá đơn.

+ Xem hoá đơn theo từng khách hàng.

## Miêu tả chi tiết chức năng

1. Đăng nhập.

- Người dùng nhập user, password vào trường thông tin tương ứng trên giao diện và chọn quyền tương ứng với 2 quyền sau là Quản Lý và Sales, hệ thống sẽ kiểm tra xem tài khoản và quyền của user có giống như trong CSDL hay không. Dựa vào quyền user đã chọn sẽ hạn chế hoặc không hạn chế một vài chức năng của form chức năng của chương trình.

- Quản lý và Sales sẽ dùng chung 1 giao diện tuy nhiên một vài button trong giao diện sẽ bị disable tùy thuộc vào quyền của người dùng.

2. Đổi mật khẩu

- Người dùng có thể đổi mật khẩu của tài khoản thông qua form đổi mật khẩu. Việc đổi mật khẩu yêu cầu người dùng nhập mật khẩu hiện tại, mật khẩu mới và nhập lại mật khẩu mới thêm một lần nữa.

3. Quản lý tài khoản

- Quản lý có thể quản lý tất cả các tài khoản trong hệ thống và có khả năng tạo thêm tài khoản quyền quản lý ở form quản lý tài khoản này.

- Việc quản lý bao gồm xem cái thông tin của tài khoản username, password, usertype.

- Có thể thay đổi password của bất kỳ tài khoản nào.

4. Quản lý thông tin nhân viên

- Ở form quản lý thông tin nhân viên sẽ hiển thị danh sách nhân viên của cửa hàng.

- Quản lý có thể thêm, sửa, xóa thông tin nhân viên.

- Các hành động thêm và xóa sẽ đồng thời thêm và xóa tài khoản tương ứng của nhân viên có trong CSDL.

5. Quản lý thông tin các xe đang bán ở cửa hàng.

- Ở form quản lý các xe, người quản lý có thể thêm thông tin các loại xe vào hệ thống, sửa và xóa các thông tin ấy.

- Nhân viên không có khả năng thêm, sửa và xóa nhưng có thể tìm kiếm xe theo phân loại xe 2 bánh hay 4 bánh hay theo hãng sản xuất (Có khả năng xem danh sách xe đang bán ở cửa hàng.

6. Quản lý khách hàng

- Quản lý có khả năng thêm, sửa, xóa thông tin của khách hàng, tìm kiếm khách hàng theo tên.

- Nhân viên có các chức năng như quản lý ngoại trừ khả năng sửa và xóa.

- Có thể xem lịch sử mua hàng của từng khách hàng.

7. Quản lý hóa đơn.

- Quản lý có khả năng tạo, xóa hóa đơn. Hóa đơn có thể tìm kiếm theo tên khách hàng.

- Sales chỉ có khả năng tạo hóa đơn và xem danh sách hóa đơn.

- Sau khi sales đã hoàn thành một giao dịch (đã tạo hóa đơn) thì sẽ được hưởng 5% hoa hồng theo tổng giá trị hóa đơn ấy.

- Sau khi tạo hóa đơn thì sẽ cập nhật lại trạng thái của xe đã bán để cho các sales khác biết được sản phẩm nào đã được bán.

# Thiết kế cơ sở dữ liệu

## Sơ đồ chức năng

Diagram, table

Description automatically generated with medium confidence

## Sơ đồ lớp

Diagram

Description automatically generated

## Thiết kế cơ sở dữ liệu

Diagram

Description automatically generated

# Kết quả

-

# TÀI LIỆU THAM KHẢO

1. Fiex Marketing (06/06/2021), *Code Swing Java là gì? Hướng dẫn tự học lập trình Java Swing cơ bản*: <https://ironhackvietnam.edu.vn/java-swing/>

2. *Lập trình hướng đối tượng(OOPs) trong java*: https://viettuts.vn/java/tong-quan-java-oops

3. *Lập trình hướng đối tượng (OOPs) trong java*: <https://topdev.vn/blog/lap-trinh-huong-doi-tuong-oops-trong-java/>

4. *Hướng dẫn kết nối cơ sở dữ liệu với Java JDBC*: <https://topdev.vn/blog/huong-dan-ket-noi-co-so-du-lieu-voi-java-jdbc/#:~:text=JDBC%20(Java%20Database%20Connectivity)%20l%C3%A0,li%E1%BB%87u%20(database)%20kh%C3%A1c%20nhau>.

5. *Component trong Java Swing - Lập trình Java cơ bản*: <https://vncoder.vn/bai-hoc/component-trong-java-swing-144>