3.3 Giới thiệu Hibernate và JDBC

3.3.1

3.3.2 Hibernate

* Lịch sử hình thành
  + Hibernate được nghĩ ra và bắt đầu phát triển vào năm 2001 bởi Gavin King nhằm thay thế cho phươn thức Entity Bean trong EJB2. (<http://www.hibernate.org/about/history>)
* Cách khái niệm liên quan (<http://www.devarticles.com/c/a/Java/Introducing-Hibernate/1/>)
  + Framwork:
    - Framwork là một khái niệm trong phát triển phần mềm. Dùng để chỉ các “cấu trúc hỗ trợ được định nghĩa”.
  + ORM Framwork (Ojbect Relational Mapping):
    - Là cơ chế giúp lập trình viên thao tác với database một các tự nhiên thông qua các đối tượng. Lập trình viên không phải quan tâm tới loại database đang sử dụng là gì.
    - Có những khái niệm cơ bản của ORM chúng ta nên nhắc tới
      * Identity (Định danh):
        + Để phân biệt một đối tượng với tất cả các đối tượng khác cùng kiểu. Mỗi đối tượng khi được khởi tạo đều được gán một Identity.
      * State (Trạng thái):
        + Trạng thái của đối tượng là giá trị được chứa bởi đối tượng đó. State là giá trị hiện hành được tập hợp chung với Identity
      * Behavior (Hành xử):
        + Behavior của một đối tượng bao gồm:

Một tập hợp các thao tác và phương thức mà đối tượng cung cấp. Còn được gọi là giao diện (Interface).

Những phản hồi của hệ thống cho người dùng gọi là những thông điệp.

Những thay đổi này tạo nên những đối tượng.

* + - * + Tất cả các cách giao tiếp với đối tượng(Object) phải thông qua giao diện (Interface). Trạng thái (State) của đối tượng (Object)
      * Type (Kiểu):
        + Các đối tượng (Object) có thể hỗ trợ giao diện (Interface) có những đặc tả khác nhau. Những đặc tả này là kiểu (Type) của giao tiếp, được bổ sung bởi đối tượng (Object). Những đối tượng (Object) khác nhau có thể cùng kiểu (Type) và chúng có thể tương tác với nhau thông qua cùng một giao diện (Interface)
      * Association (Tập hợp):
        + Một đối tượng (Object) của một kiểu (Type) có thể liên kết với kiểu khác. Liên kết này chính là Association. Nó giúp cho việc chuyển tải những đối tượng này với nhau.
      * Class (Lớp):
      * Encapsulation (Đóng gói):
      * Inheritance (Kế thừa):
  + Persistence Class: (http://www.devarticles.com/c/a/Java/Introducing-Hibernate/2/)
    - Persistence Class là một POJO (Plain Old Java Object) trong Spring Framwork (Java Bean) có những “getter” ,“setter” để truy cập thuộc tính của nó là những biến thực thể (Instance Variable). Persistence Class có các đặc điểm sau:
      * Thay thế hướng đối tượng cho bảng ở cơ sở dữ liệu.
      * Các thuộc tính của bảng trở thành những biến thực thể của persistence class.
      * Kiểu dữ liệu của các biến thực thể là miền (domain) của các thuộc tính.
      * Đối tượng persistence class thay thế cho hàng của bảng
  + Mapping File:
    - Mapping file là một file XML chứa ánh xạ (Mapping) chi tiết giữa Persistence Class và bảng nó thay thế. Các thành phần của Mapping File:
      * Hibernate Mapping:
        + Đây là thành phần gốc, bao gồm những thành phần khác.
      * Class:
        + Dùng để ánh xạ tên của bảng thành Persistence Class.
        + Tên thuộc tính được sử dụng để xác định tên lớp và thuộc tính bảng được sử dụng để xác định bảng mà lớp sẽ thay thế.
        + Ví dụ: Để ánh xạ một bảng tên ORDER thành môtj Persistence Class tên đầy đủ là com.someorg.persist

<class name=”com.someorg.persist.Order”table=”ORDER”>…</class>

* + - * Id:
        + Thành phần này được sử dụng để ánh xạ khoá chính cho bảng thành một biến thực thể (Instance Variable) của lớp.
        + Những cột con của id có thể ánh xạ thành những biến tương ứng
        + Thành phần generator được sử dụng để báo cho Hibernate biết lớp sẽ được sử dụng nằm ở đâu để tự động tạo id.
        + Ví dụ:

<id name="id" type="string" unsaved-value="null">  
        <column name="id" sql-type="char(32)" not-null="true"/>  
        <generator class="assigned"/>  
       </id>

Từ ví dụ ta thấy tên khoá chính là id mà nó đã được ánh xạ thành biến thực thể “id”. Kiểu dự liệu là “string” được ánh xạ cho cột “id” thành kiểu “char(32)”. Generator class là assigned nghĩa là khoá chính đã được gán.

* + - * Proprety:
        + Thành phần này thuộc về các cột con của nó. Ánh xạ những thuộc tính khác (hoặc cột khác) thành những biến thực thể của Persistence class
        + Ví dụ:

<proprety name = “name”>

<column name=”name” sql-type=”char(225)” not null=”true”/>

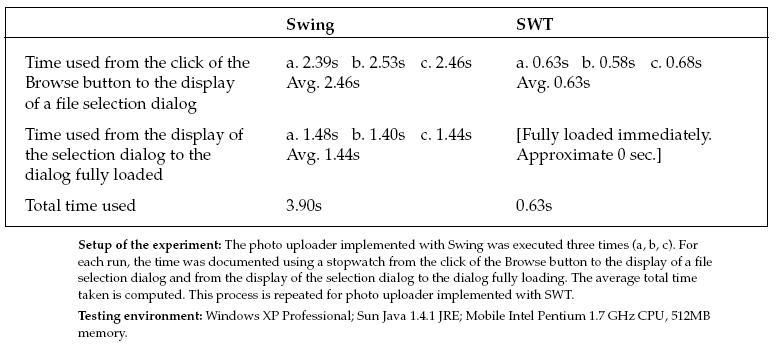
</proprety>

Thuộc tính name của proprety chứa tên của biến như một giá trị. Name của thành phần colunm chứa tên cột mà biến thực thể ánh xạ.

* + Hibernate Configuaration File:
    - File này có thể gọi con trỏ đến ứng dụng sử dụng Hibernate. Hầu hết các thành phần thông dụng sử dụng trong file XML là:
      * Proprety:
        + Thuộc tính
* Cơ chế làm việc
* Ưu khuyết điểm
  + Ưu điểm: (http://www.hibernate.org/about/why-hibernate)
    - Hibernate cho phép bạn phát triển thêm các lớp hướng đối tượng như:
      * Đa hình
      * Thừa kế
      * Liên kết
      * Các Framwork khác của JAVA
    - Hibernate không yêu cầu giao diện hoặc lớp cơ sở cho Persistent Class
    - Hiệu suất cao
    - Độ tin cậy
    - Khả năng mở rộng: Hibernate có thể tuỳ biến về khả năng mở rộng
    - Các ngôn ngữ truy vấn: Bao gồm những ngôn ngữ truy vấn cho Hibernate như:
      * HQL (Hibernate Query Language)
      * JPAQL (Java Persistence Query Language)
  + Khuyết điểm:

4. Phát Triển Ứng Dụng

* 1. SWT (<http://www.vogella.com/articles/SWT/article.html>)
     1. Giới thiệu về SWT
  + Ngoài AWT (Abstract Window Toolkit) của Java, một GUI (Graphic User Interface) Framwork cho mọi nền tảng trên Java. SWT (Standard Widget Toolkit) đã được IBM tạo ra để khắc phục những điểm yếu có trên AWT và Swing.
  + SWT - Standard Widget Toolkit là một gói công cụ mã nguồn mở, được phát triển bởi IBM, cung cấp cho các lập trình viên Java thêm một giải pháp để phát triển Giao diện đồ họa người dùng (GUI – Graphical User Interface). SWT có đầy đủ các GUI API đặc trưng
    1. Ưu nhược điểm của SWT
  + Ưu điểm:
    - SWT có thể truy cập vào các thành phần giao diện tự nhiên của từng nền tảng thông qua JNI ( Java Native Interface)
    - Tốc độ của SWT được cải thiện rất nhiều và tài nguyên bộ nhớ chiếm ít hơn so với Swing



* + Nhược điểm:
    - API không rõ ràng, tài liệu ít, không hệ thống và đầy đủ.
    - Các Widget không linh động, khi khởi tạo widget ta phải khai báo widget đó thuộc container nào đó
    - Widget còn ít so với Swing. Ví dụ như không thể đặt Background cho Button được
    1. Các thành phần cơ bản của SWT
  + Display, shell là các lớp thành phần quan trọng của SWT
    - Display chịu trách nhiệm quản lý những kích hoạt và điều khiển tương tác giữa các phần giao diện với nhau.
    - Shell là cửa sổ ứng dụng được quản lý bởi hệ điều hành.
  + Mỗi sự kiện SWT đều đòi hỏi có ít nhất một Display và một hoặc nhiều Shell.
    1. Vòng lặp
  + SWT không cung cấp vòng lặp cho những event riêng. Có nghĩa là người lập trình phải tự bắt đầu và kiểm tra event để cập nhật giao diện người dùng
  + Các vòng lặp sẽ thực thi phương thức readAndDispatch(). Nó sẽ thực hiện cho tới khi mainShell được đóng lại.
  + Ví dụ:

Display display = new Display();

Shell shell = new Shell(display);

Shell.open();

//

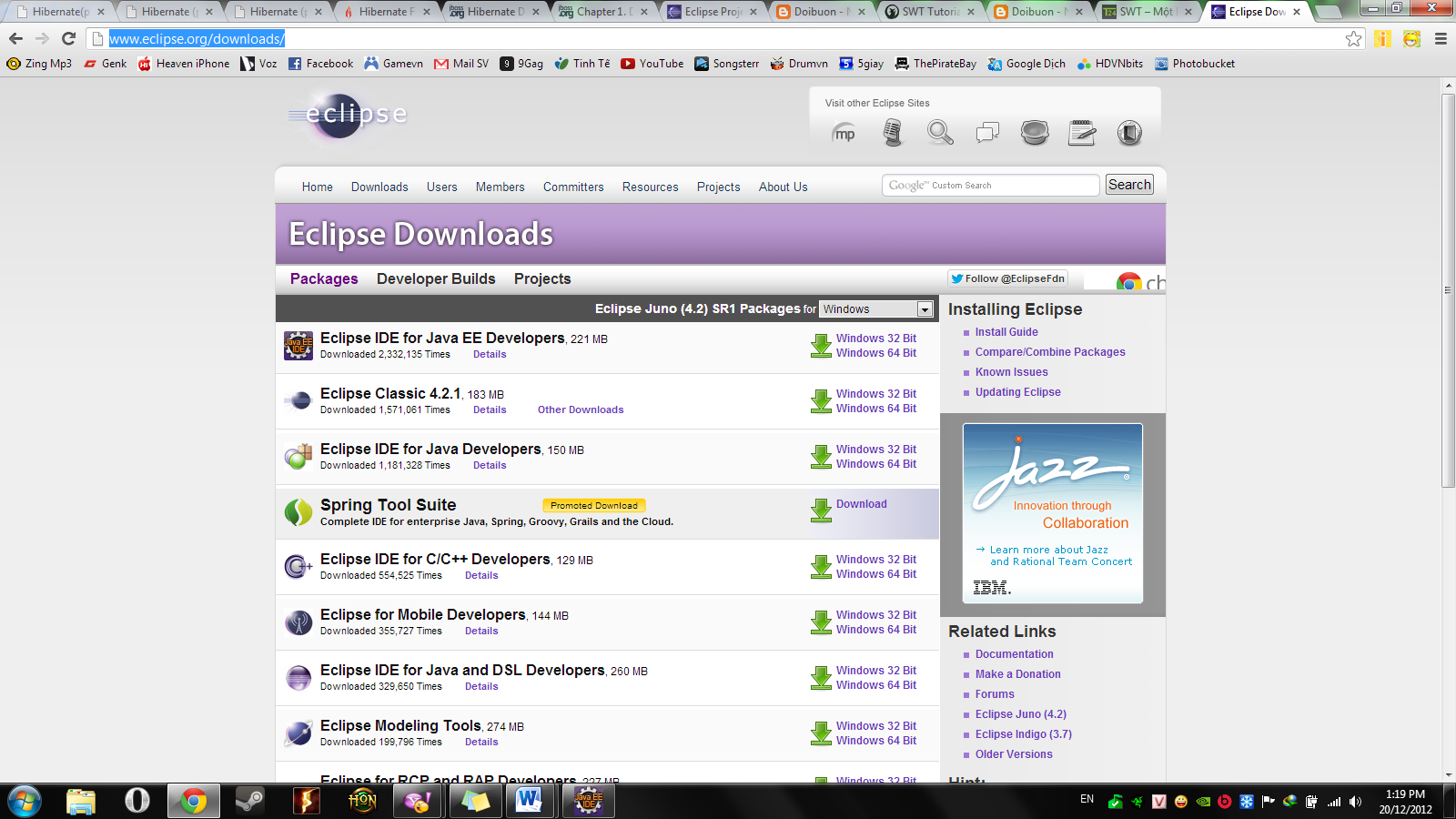
while (!shell.isDisposed()) {

if(!display.readAndDispatch())

display.sleep();

}

display.dispose();

* + 1. Gói SWT
  + SWT bao gồm vài gói chính:
    - org.eclipse.swt: bao gồm các lớp định nghĩa các constant và các exception được sử dụng cho các lớp SWT.
    - org.eclipse.swt.widgets: gồm các thành phần đồ hoạ cơ bản như interface và classes.
    - org.eclipse.swt.events: định nghĩa các event, listener mà các SWT sử dụng
    - org.eclipse.swt.dnd: gồm các lớp hỗ trợ việc kéo thả (drag-and-drop) các thành phần giao diện của SWT.
    - org.eclipse.swt.layout: gồm các lớp cung cấp việc sắp đặt tự động vị trí và kích thước cho các thành phần giao diện SWT
    - org.eclipse.swt.print: gồm các lớp hỗ trợ việc in ấn các thành phần giao diện SWT
    - org.eclipse.swt.graphic: cung cấp các point, rectangle, region, con trỏ, màu sắc, phông chữ,...
    1. Cài đặt
  + Tải eclip từ http://www.eclipse.org/downloads/
  + Tải thêm gói SWT.jar phù hợp với eclipse đã tải từ <http://download.eclipse.org/eclipse/downloads/drops/R-3.7.2-201202080800/index.php#SWT>



* + Sau đó thêm gói SWT.jar vào classpath bằng cách chọn Project -> Propreties -> JavaBuildPath -> Libraries - > Add Variable -> Eclipse Home - >Extend và chọn SWT.jar rồi nhấn Ok.