**Ảnh có chứa văn bản, ký hiệu

Mô tả được tạo tự động**

**Đại học Bách khoa Hà Nội**

**Trường Công nghệ thông tin và truyền thông**

Mã học phần : IT 3103

Giảng viên : Trịnh Tuấn Đạt

BÁO CÁO BÀI TẬP LỚN MÔN

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

|  |  |
| --- | --- |
| Họ tên | MSSV |
| Lê Thế Anh | 20200018 |
| Phan Thế Anh | 20204941 |
| Lê Văn Bảo | 20205057 |
| Nguyễn Văn Chung | 20204945 |
| Nguyễn Duy Doanh | 20204948 |
| Hoàng Minh Dũng | 20205071 |

# **PHÂN CHIA CÔNG VIỆC**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Công việc** | **Đóng góp** |
| Nguyễn Duy Doanh - Leader | Dựng base, viết thuật toán cào dữ liệu, liên kết các trường dữ liệu với nhau, đổ dữ liệu vào giao diện. | 30 |
| Lê Thế Anh | Viết thuật toán cào dữ liệu, làm slide, báo cáo. | 20 |
| Phan Thế Anh | Viết thuật toán cào dữ liệu, thiết kế giao diện, vẽ biểu đồ lớp, biểu đồ class. | 20 |
| Lê Văn Bảo | Làm slide, báo cáo, tìm nguồn dữ liệu, hỗ trợ làm giao diện | 10 |
| Nguyễn Văn Chung | Làm side, báo cáo, tìm nguồn dữ liệu, hỗ trợ code | 10 |
| Hoàng Minh Dũng | Làm slide, báo cáo, hỗ trợ làm giao diện, tìm nguồn dữ liệu | 10 |

**MỤC LỤC**

[PHÂN CHIA CÔNG VIỆC 2](#_Toc126528090)

[I. Giới thiệu bài toán 4](#_Toc126528091)

[II. Thống kê dữ liệu đã thu thập 5](#_Toc126528092)

[**1.** **Số lượng dữ liệu** 5](#_Toc126528093)

[**2.** **Thuộc tính dữ liệu của thực thể** 5](#_Toc126528094)

[**3.** **Liên kết giữa các thực thể** 6](#_Toc126528095)

[**4.** **Nguồn dữ liệu** 6](#_Toc126528096)

[III. Biểu đồ UML 7](#_Toc126528098)

[**1.** **Biểu đồ phụ thuộc gói** 7](#_Toc126528099)

[**2.** **Biểu đồ lớp** 9](#_Toc126528100)

[IV. Giải thích thiết kế 18](#_Toc126528101)

[**1.** **Các gói (Package)** 18](#_Toc126528102)

[**2.** **Ý tưởng về các lớp** 20](#_Toc126528103)

[V. Các kĩ thuật lập trình hướng đối tượng đã sử dụng 22](#_Toc126528104)

[VI. Các công nghệ , thuật toán hay ho đã sử dụng 22](#_Toc126528105)

[VII. Giới thiệu về chương trình 23](#_Toc126528106)

[1. Cách xây dựng và chạy trương trình 23](#_Toc126528107)

[**2.** **Một vài hình ảnh về chương trình** 23](#_Toc126528108)

# **Giới thiệu bài toán**

Có rất nhiều các trang web cung cấp thông tin về lịch sử Việt Nam. Công việc cần thực hiện là thu thập dữ liệu về lịch sử Việt Nam ( các triều đại , nhân vật lịch sử ,các di tích, lễ hội , sự kiện lịch sử) và liên kết các dữ liệu này vào với nhau.

1. **Thống kê dữ liệu đã thu thập**
2. **Số lượng dữ liệu**

* Về các thời kì lịch sử: thu thập được 24 dữ liệu về thời kỳ từ kỉ Hồng Bàng đến thời điểm hiện tại.
* Về nhân vật lịch sử: thu thập được 1452 dữ liệu nhân vật
* Về di tích: thu thập được 128 dữ liệu về di tích lịch sử
* Về lễ hội văn hóa: thu thập được 52 dữ liệu lễ hội văn hóa nổi bật trong 1 năm.
* Về sự kiện lịch sử: thu thập được 371 dữ liệu sự kiện lịch sử từ kỉ Hồng Bàng đến năm 1975.

1. **Thuộc tính dữ liệu của thực thể**

* Về nhân vật(6 thuộc tính): tên nhân vật, tên cha, tên mẹ, năm sinh, năm mất, chức vụ, thời kỳ
* Về thời kỳ(4 thuộc tính): tên thời kỳ, thời gian bắt đầu, thời gian kết thúc, mô tả sơ lược.
* Về di tích(4 thuộc tính): tên di tích, địa điểm, mô tả về di tích, hình ảnh di tích.
* Về lễ hội(4 thuộc tính): tên lễ hội, thời gian tổ chức, địa điểm, mô tả lễ hội
* Về sự kiện(7 thuộc tính): tên sự kiện, thời điểm bắt đầu, thời điểm kết thúc, mô tả sự kiện, nhân vật liên quan, hình ảnh, tài liệu tham khảo.

1. **Liên kết giữa các thực thể**

* Trong một thời kỳ sẽ có nhiều nhân vật, và một nhân vật cũng có thể xuất hiện ở 1 hoặc hai thời kỳ kế tiếp nhau

Ví dụ: Triều đại nhà Lý có : Lý Công Uẩn, Lý Thái Tổ, Lý Thái Tông…

* Nhân vật có thể liên quan đến 1 hay nhiều sự kiện nào đó và sự kiện có thể có nhiều nhân vật liên quan.

Ví dụ: Sự kiện “Lý Công Uẩn lên ngôi, lập ra nhà Lý (1009)” có các nhân vật liên quan là: Lê Đại Hành, Lê Long Đĩnh, Lý Thái Tổ …

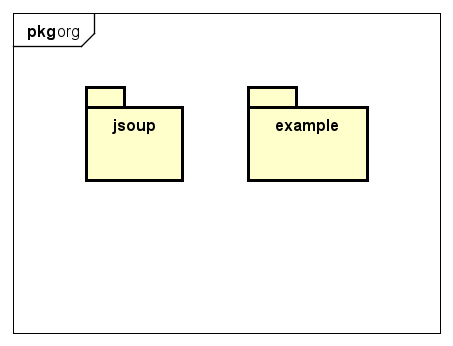
1. **Nguồn dữ liệu**

Dữ liệu được lấy từ các web “người kể sử (nguoikesu.com)” , “wikipedia (wikipedia.org)”, “thư viện lịch sử (thuvienlichsu.com)”, “vạn sử (vansu.vn)”.

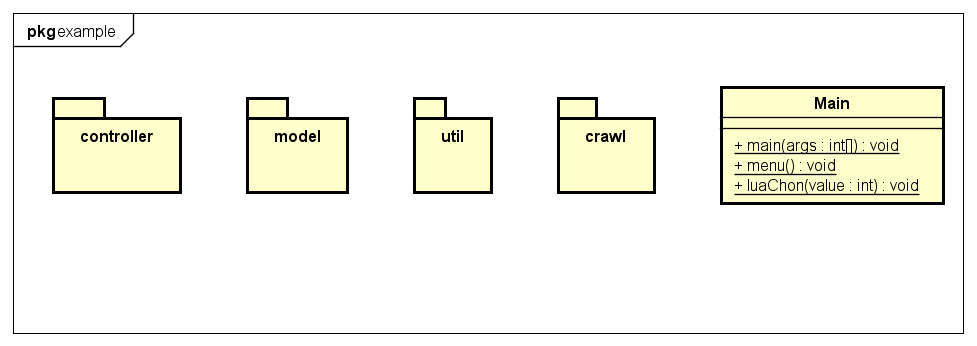
* Về Người kể sử (nguoikesu.com): Dữ liệu thu thập được từ trang web bao gồm các thông tin về: triều đại, nhân vật.
* Về wikipedia (wikipedia.org): Dữ liệu thu thập được từ trang web bao gồm: các di tích, lễ hội. Đồng thời bổ sung dữ liệu về các nhân vật có trong wikipedia API
* Về Thư viện lịch sử (thuvienlichsu.com): Dữ liệu thu thập được từ trang web là các sự kiện lịch sử
* Về Vạn sử (vansu.vn): Dữ liệu được thu thập từ trang web bổ sung thông tin về thời kỳ lịch sử cho Nhân vật

2. **Biểu đồ UML**
3. **Biểu đồ phụ thuộc gói**

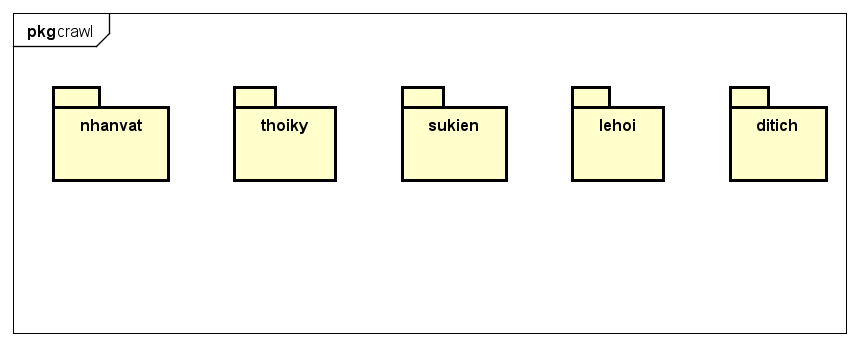
Có hai package lớn, đó là Jsoup và Example.Trong đó Joup là package di Astah tự tạo ra trong quá trình import class

****

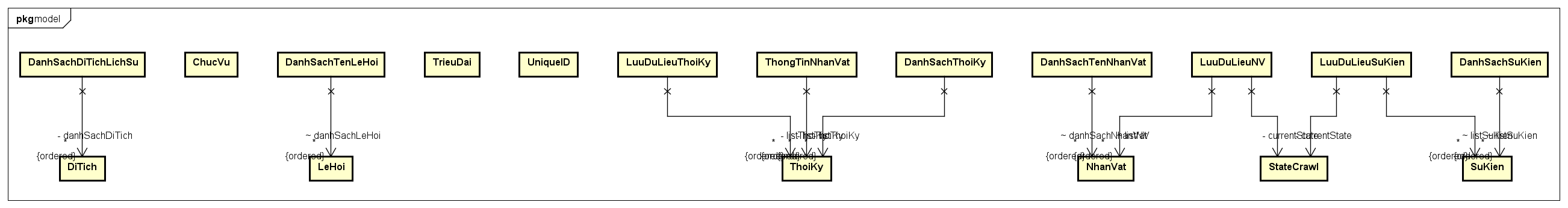
Trong package Example bao gồm 4 package lớn: controller, model, util, crawl



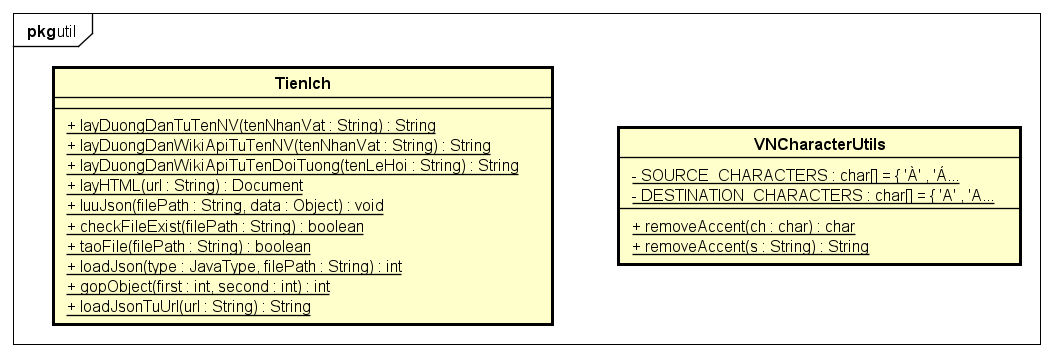
* Package Crawl chứa các package khác nhau nhằm mục đích thu thập dữ liệu cho từng đối tượng ở các trang web khác nhau. Package này mục địch chính dùng để cào dữ liệu từ các trang web



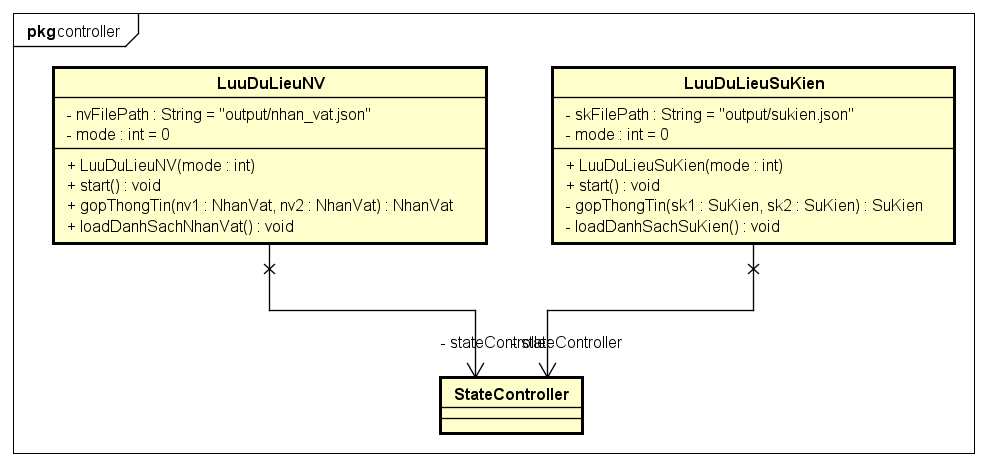
* Package model chứa các lớp lưu trữ và khai báo thuộc tính của đối tượng



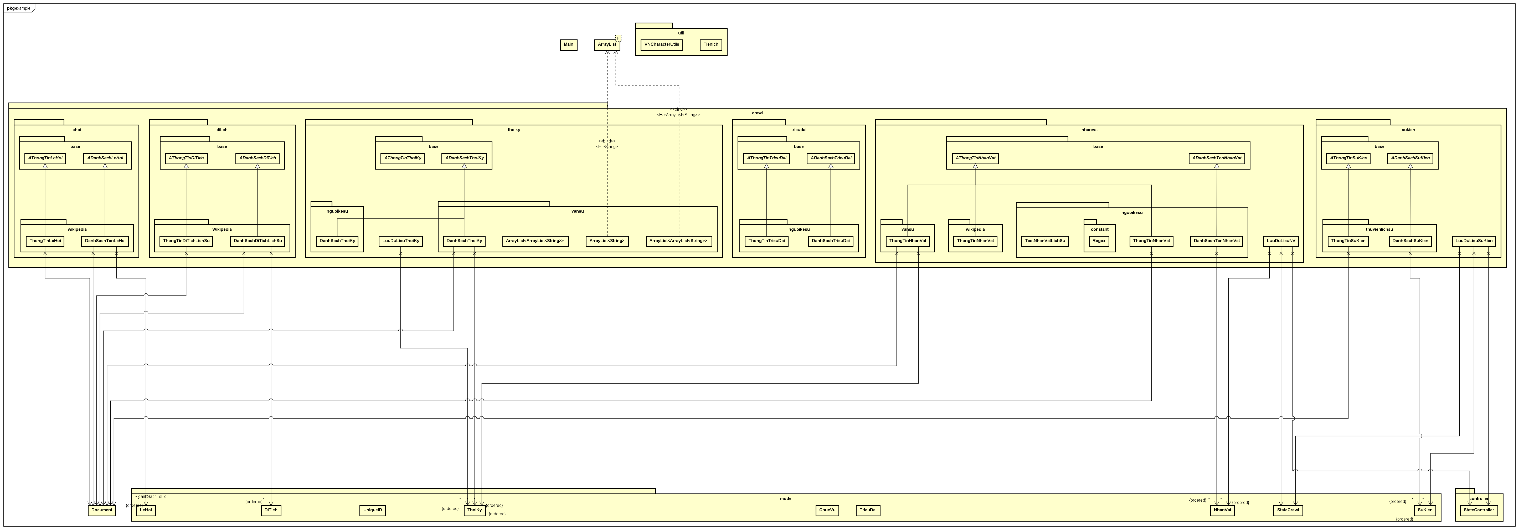
* Package Util chứa các tiện ích dùng chung như: Lấy đường dẫn từ tên đối tượng, tạo, lưu và check file Json, xóa bỏ dấu tiếng Việt,…



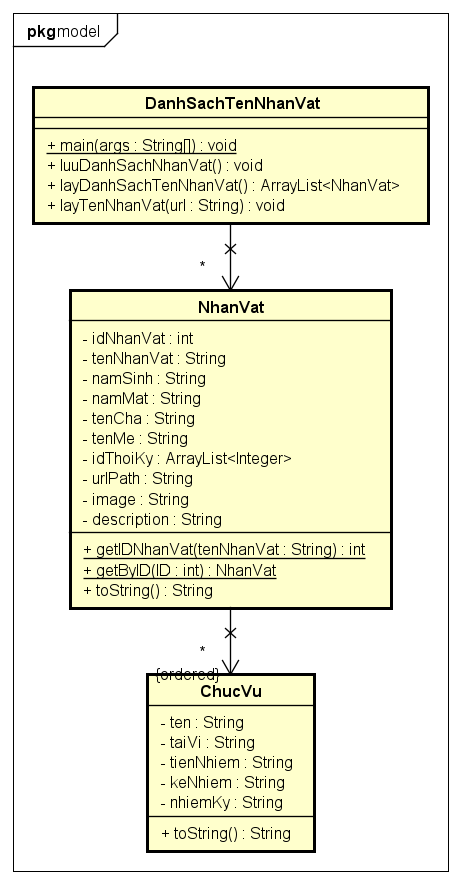
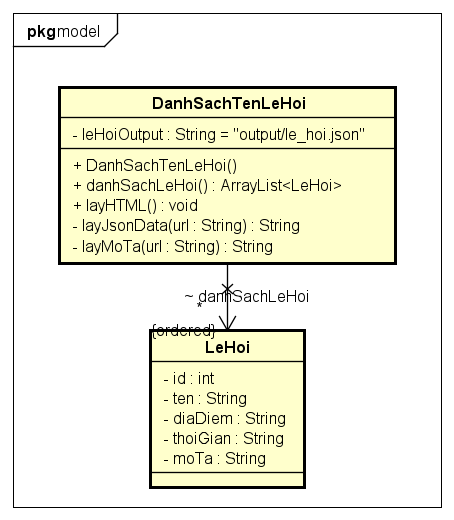
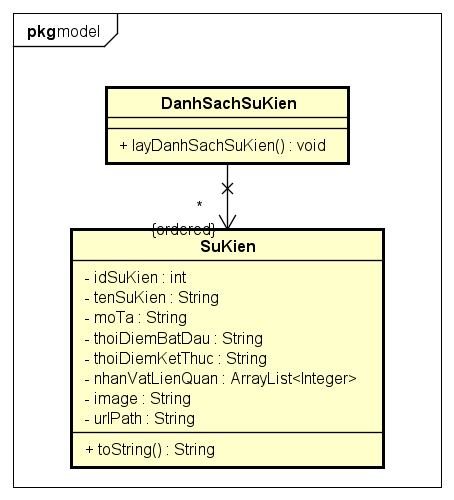
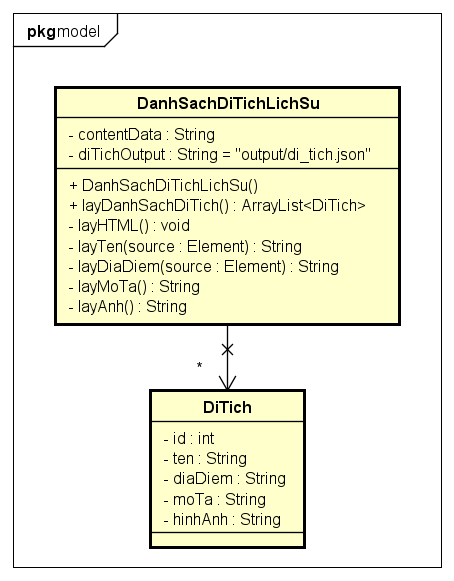
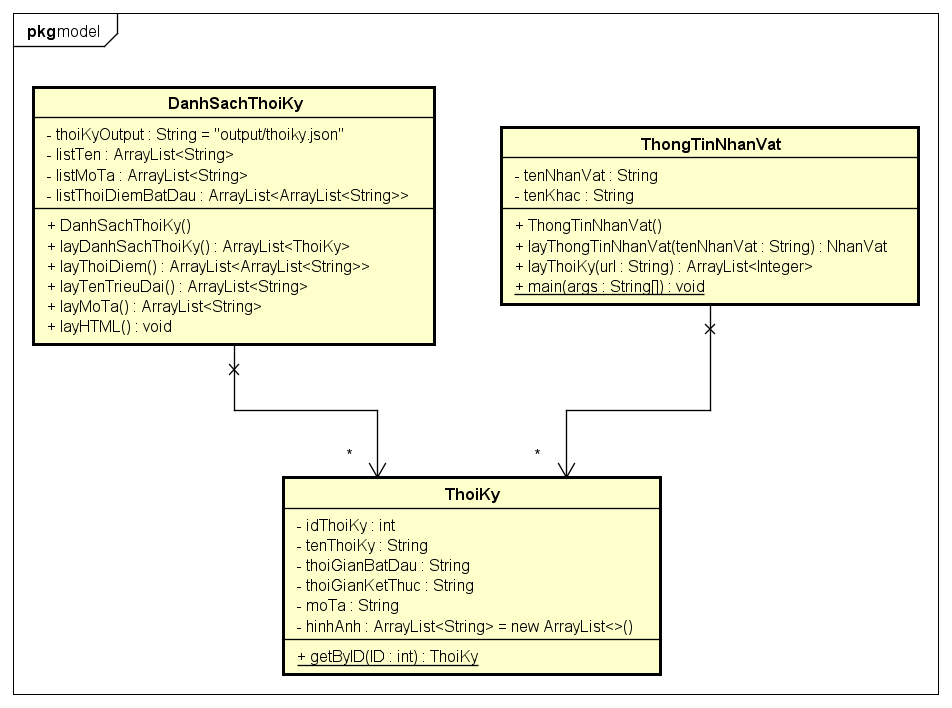
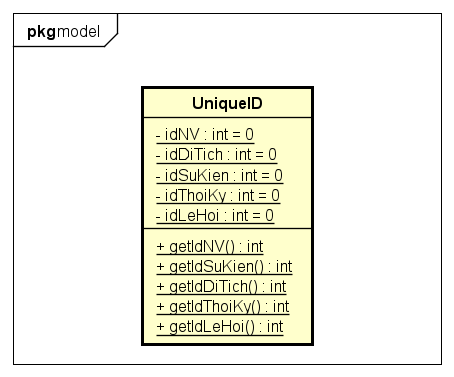
* Package Controller bao gồm các lớp có chức năng kiểm soát, điều khiển việc cào dữ liệu, lưu trạng thái của quá trình cào dữ liệu



1. **Biểu đồ lớp**

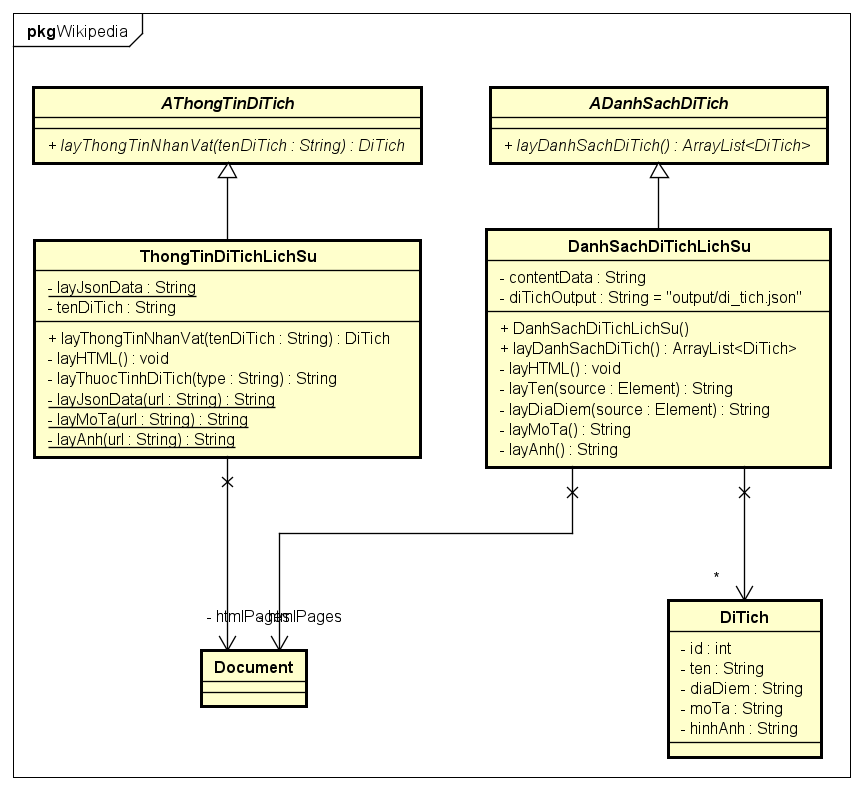


1. Package model bao gồm các class lưu trữ các thuộc tính của nhân vật.

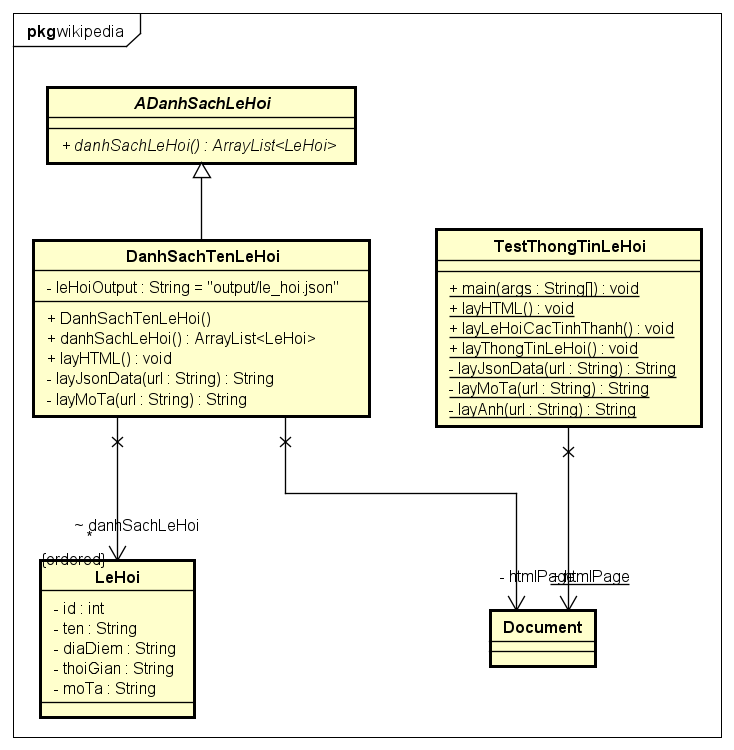
* **Class NhanVat - Class LeHoi**
* ******Class DiTich - Class SuKien**
* ******Class UniqueId - Class ThoiKy**

1. **Package** [**crawl**](https://github.com/hust-oop-project/crawl/tree/master/src/main/java/org/example/crawl) : chứa các package khác nhau nhằm mục đích thu thập dữ liệu cho từng đối tượng ở các trang web khác nhau. Package này mục đích chính dùng để cào dữ liệu từ các trang web

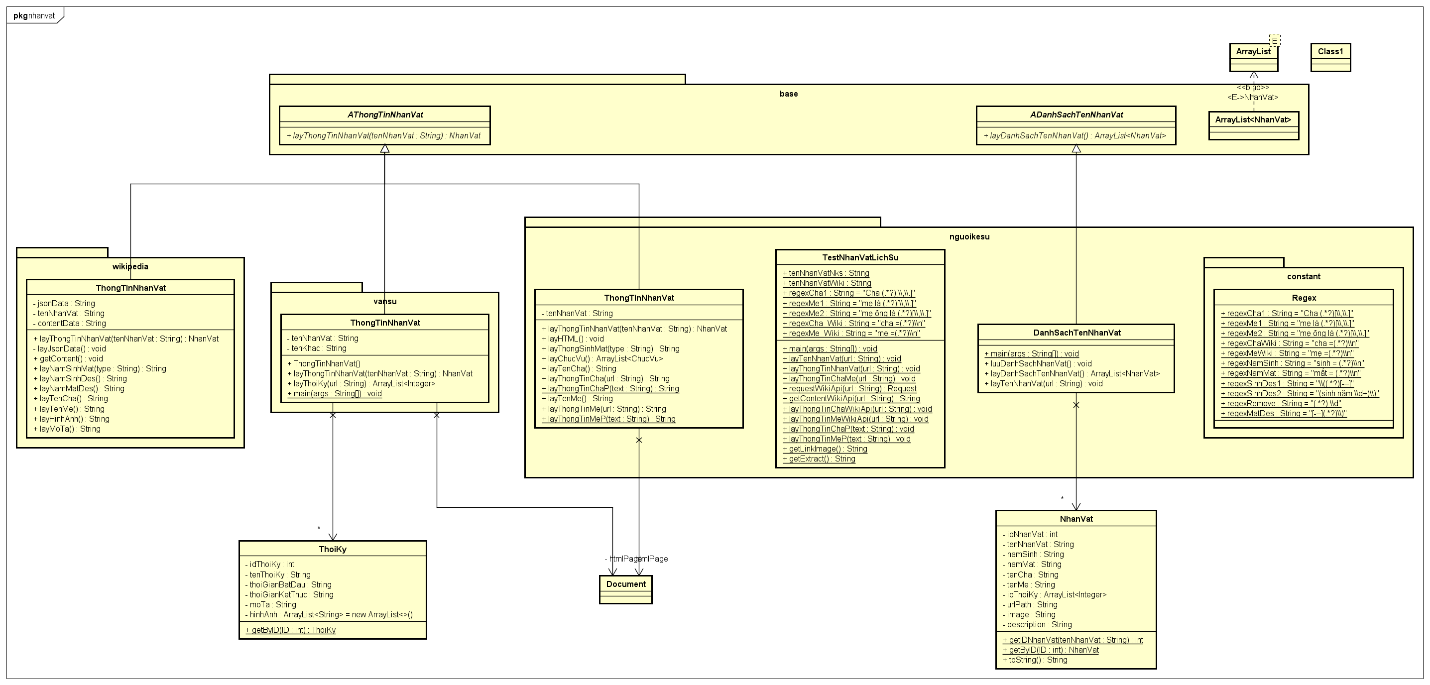
* **Package crawl/ditich**



* Trong class ThongTinDiTichLichSu có các phương thức như layJsonData() có nhiệm vụ gọi API từ trang web wikipedia lấy thông tin đối tượng như layMoTa() (Lấy mô Tả), layAnh() (lấy ảnh của di tích). Phương thức layHTML() có nhiệm vụ connect với link url trong dữ liệu, từ đó sử dụng Java Jsoup để tiến hành lấy dữ liệu trong các thẻ HTML.
* Đối với class DanhSachDiTichLichSu(): class này có cấu trúc tương tự như ThongTinDiTichLichSu và nó có nhiệm vụ lấy toàn bộ danh sách các di tích và lưu vào 1 ArrayList và xuất chúng ra file Json .
* **Package crawl/lehoi**



* **Package crawl/nhanvat**



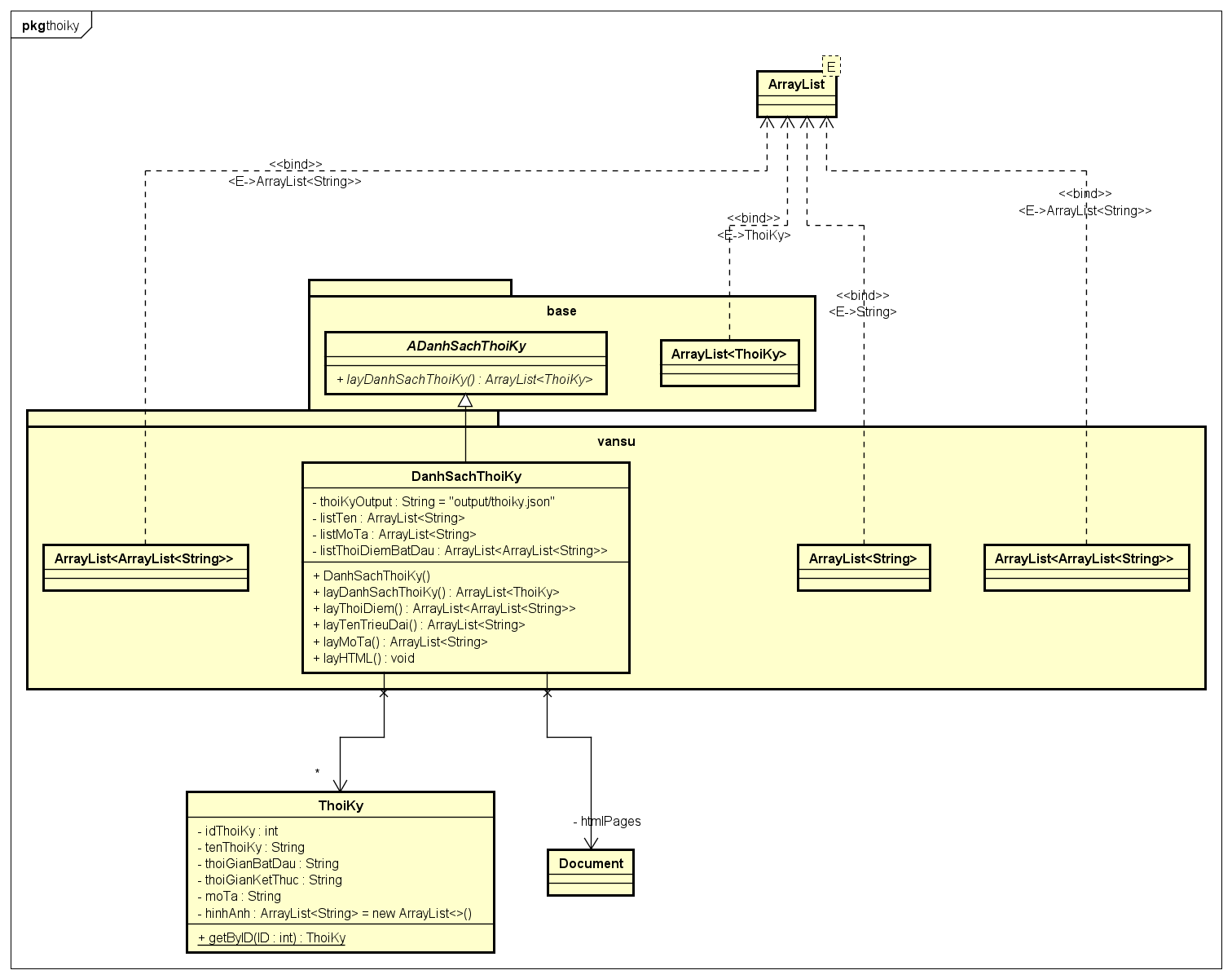
* Trong package nhanvat có 4 package chính: Base, nguoikesu, Wikipedia, vansu.

Trong đó package base chứa các class trừu tượng. Các package còn lại có tên lần lượt tương ứng với các trang web để lấy các thuộc tính của đối tượng, các class trong package này đều kế thừa các class tương ứng trong package base

Vì mỗi website có 1 cấu trúc khác nhau nên việc lấy đầy đủ thông tin của đối tượng ở 1 website duy nhất khá là khó khăn. Đó là lý do tại sao cần phải lấy dữ liệu ở nhiều web khác nhau để có thể đáp ứng một cách đầy đủ thông tin thuộc tính của các đối tượng.

Ở đây ta có thể thấy cả 3 package nguoikesu, Wikipedia, vansu đều có 3 class có tên giống nhau là ThongTinNhanVat, tuy nhiên chức năng của chúng ở từng package lại khác nhau. ở nguoikesu thì sẽ có chức năng lấy năm sinh năm mất, lấy chức vụ, lấy tên cha, tên mẹ. Tuy nhiên ở package vansu lại chỉ có chức năng tìm ID triều đại.

* **Package crawl/thoiky**

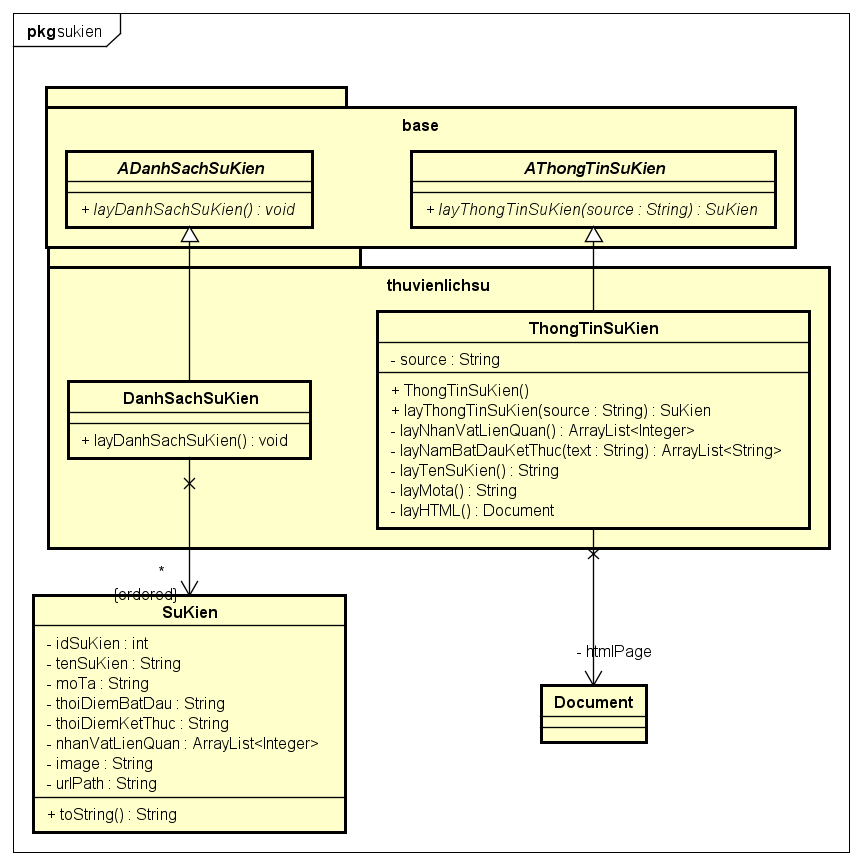


* Package crawl/thoiky cũng có cấu trúc tương tư như nhanvat hay lehoi.

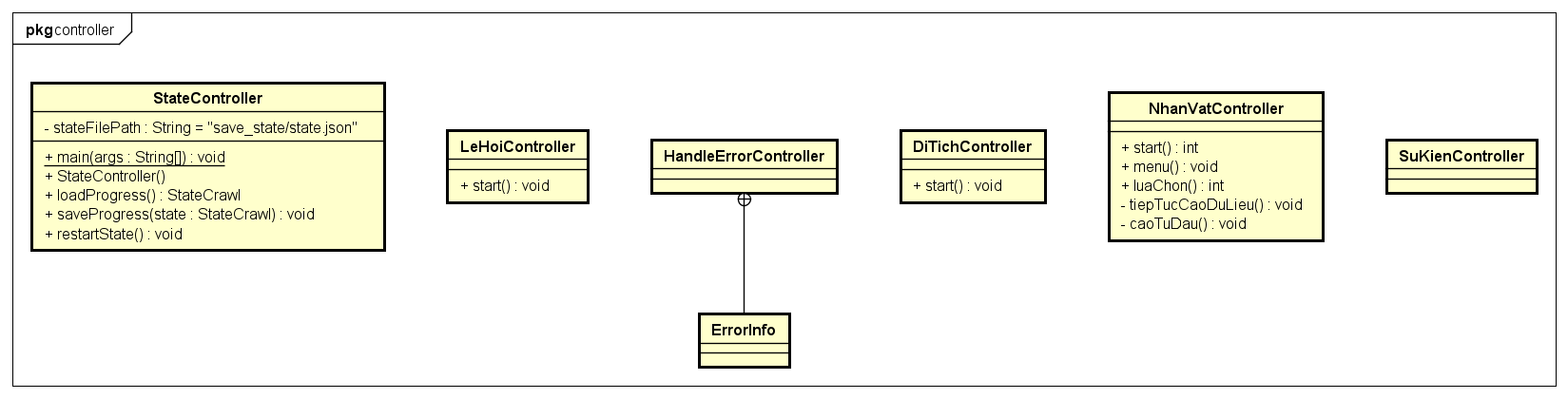
Có nhiều thời kỳ khác nhau, và trong mỗi thời kỳ cũng sẽ có nhiều triều đại.

Trong class DanhSachThoiKy có các phương thức layThoiDiem() để lấy khoảng thời gian thời kỳ đó tồn tại; phương thức layTenTrieuDai() để lấy tên các triều đại; Phương thức layMoTa() để lấy mô tả của triều đại; và cuối cùng, layHTML() để connect với link trang web cần cào dữ liệu

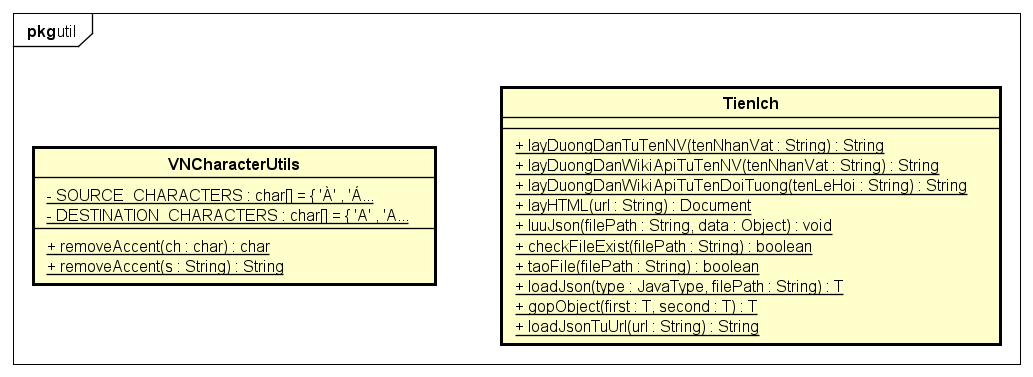
* **Package crawl/sukien**

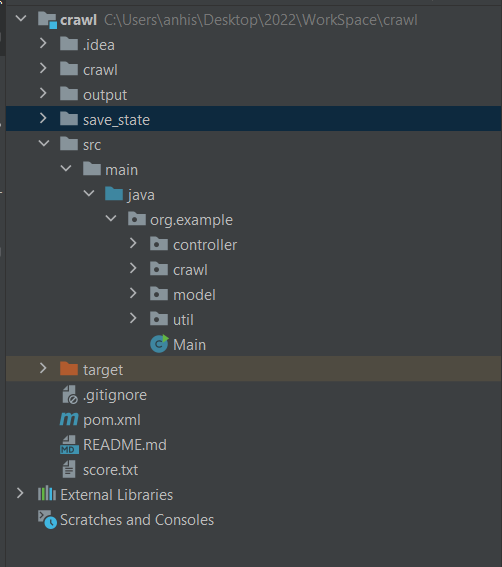


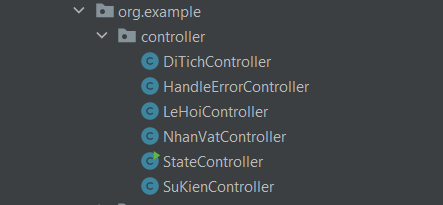
1. **Package Controller**

****

1. **Package Util**



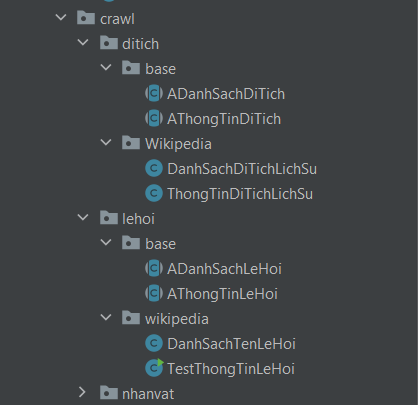
1. **Giải thích thiết kế**
2. **Các gói (Package)**

* [src](https://github.com/hust-oop-project/crawl/tree/master/src)/[main](https://github.com/hust-oop-project/crawl/tree/master/src/main)/[java](https://github.com/hust-oop-project/crawl/tree/master/src/main/java)/[org](https://github.com/hust-oop-project/crawl/tree/master/src/main/java/org)/**example**/[controller](https://github.com/hust-oop-project/crawl/tree/master/src/main/java/org/example/controller" \o "controller): Bao gồm các lớp có chức năng kiểm xoát và điều khiển việc cào dữ liệu của các package trong crawl
* [src](https://github.com/hust-oop-project/crawl/tree/master/src)/[main](https://github.com/hust-oop-project/crawl/tree/master/src/main)/[java](https://github.com/hust-oop-project/crawl/tree/master/src/main/java)/[org](https://github.com/hust-oop-project/crawl/tree/master/src/main/java/org)/**example**/[crawl](https://github.com/hust-oop-project/crawl/tree/master/src/main/java/org/example/crawl" \o "crawl) : chứa các package khác nhau nhằm mục đích thu thập dữ liệu cho từng đối tượng ở các trang web khác nhau. Package này mục đích chính dùng để cào dữ liệu từ các trang web

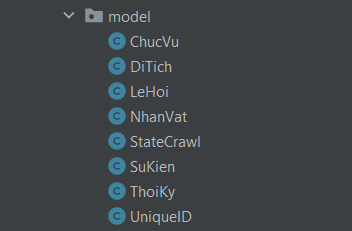
Ảnh có chứa văn bản

Mô tả được tạo tự động

* Trong [src](https://github.com/hust-oop-project/crawl/tree/master/src)/[main](https://github.com/hust-oop-project/crawl/tree/master/src/main)/[java](https://github.com/hust-oop-project/crawl/tree/master/src/main/java)/[org](https://github.com/hust-oop-project/crawl/tree/master/src/main/java/org)/**example**/[crawl](https://github.com/hust-oop-project/crawl/tree/master/src/main/java/org/example/crawl" \o "crawl) có chứa các package có tên trùng với tên các đối tượng cần cào dữ liệu từ các trang web. Cấu trúc của các package con tương đồng nhau. Đều bao gồm 1 package base chứa các class abstract để các class trong package còn lại kế thừa chúng.Các package tiếp theo được đặt tên theo các trang web mà sử dụng để cào dữ liệu. Ví dụ package Wikipedia cho biết dữ liệu được lấy từ trang web Wikipedia.



* [src](https://github.com/hust-oop-project/crawl/tree/master/src)/[main](https://github.com/hust-oop-project/crawl/tree/master/src/main)/[java](https://github.com/hust-oop-project/crawl/tree/master/src/main/java)/[org](https://github.com/hust-oop-project/crawl/tree/master/src/main/java/org)/**example**/[model](https://github.com/hust-oop-project/crawl/tree/master/src/main/java/org/example/model" \o "model) : chứa các lớp lưu trữ và khai báo thuộc tính của đối tượng



* [src](https://github.com/hust-oop-project/crawl/tree/master/src)/[main](https://github.com/hust-oop-project/crawl/tree/master/src/main)/[java](https://github.com/hust-oop-project/crawl/tree/master/src/main/java)/[org](https://github.com/hust-oop-project/crawl/tree/master/src/main/java/org)/**example**/[util](https://github.com/hust-oop-project/crawl/tree/master/src/main/java/org/example/util" \o "util) : chứa các lớp tiện ích như : Lấy đường dẫn từ tên đối tượng, call API, tạo và check file Json

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. **Ý tưởng về các lớp**

* Về các lớp thu thập dữ liệu :

Đối với các lớp thu thập dữ sẽ nằm trong hai package chính đó là : modal và crawl

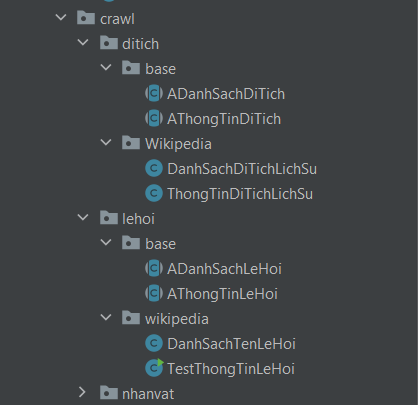
* Các class trong lớp modal chứa thuộc tính của các đối tượng cũng như phương thức toString nhằm ghi thông tin vào file Json. Ví dụ:

Ảnh có chứa văn bản

Mô tả được tạo tự độngẢnh có chứa văn bản

Mô tả được tạo tự động

* Các class trong package crawl có chức năng lấy thông tin nhân vật từ các trang web, bao gồm các đoạn code , các thư viện hỗ trợ cho việc cào dữ liệu



# **Các kĩ thuật lập trình hướng đối tượng đã sử dụng**

* Về tính kế thừa: Các Class trong package có tên là 1 website ví dụ: wikipedia, nguoikesu, vansu, đều kế thừa các class trong package Base ứng với mỗi đối tượng.
* Về tính trừu tượng và bao đóng: Các class trong package base đều là các lớp trừu tượng. Các thuộc tính của đối tượng thành phần trong một lớp đều được lưu dưới dạng private, để thực hiện các thay đổi với biến thì phải gọi bằng các setter, getter của biến. Các hàm chỉ được gọi trong nội bộ class được lưu dưới dạng private, còn lại được lưu dưới dạng public để thuận lợi cho quá trình thực hiện chương trình
* Chương trình quản lý dự án và các thư viện mở (dependence) – Maven

# **Các công nghệ , thuật toán hay ho đã sử dụng**

* Chương trình quản lý dự án và các thư viện mở (dependence) - Maven
* Ngôn ngữ lập trình chủ yếu: Java
* Thư viện đồ họa: JavaFx
* Công cụ xử lý chuỗi nâng cao thông qua các biểu thức chính quy – Regular Expression
* Call API của Wikipedia để truy xuất thông tin về các thuộc tính của đối tượng ( call API) bằng cách sử dụng các thư viện chuẩn: Json, Jsoup, OkHttpClient, JsonNode.

# **Giới thiệu về chương trình**

1. Cách xây dựng và chạy trương trình

Nguyên tắc xây dựng chung :

* Với các đối tượng có lượng dữ liệu cần thu thập lớn, nhiều trường, phức tạp:

- Trước tiên ta cần lấy danh sách(tên) các đối tượng cần cào, tạo đối tượng và lưu vào file json.

- Sau đó duyệt qua danh sách này để tiến hành thu thập dữ liệu cho từng đối tượng.Dữ liệu thu thập đươjc từ nhiều nguồn sẽ qua một bước gộp dữ liệu để được đối tượng hoàn chỉnh.

- Quá trình liên kết dữ liệu sẽ được dựa vào tên của các đối tượng(liên kết nhân vật với sự kiện) hoặc một khoảng thời gian nào đó(liên kết thời kì lịch sử với nhân vật).

* Với các đối tượng có lượng dữ liệu nhỏ, ít thuộc tính thì chúng ta sẽ thu thập toàn bộ và lưu lại sau khi quá trình thu thập kết thúc.

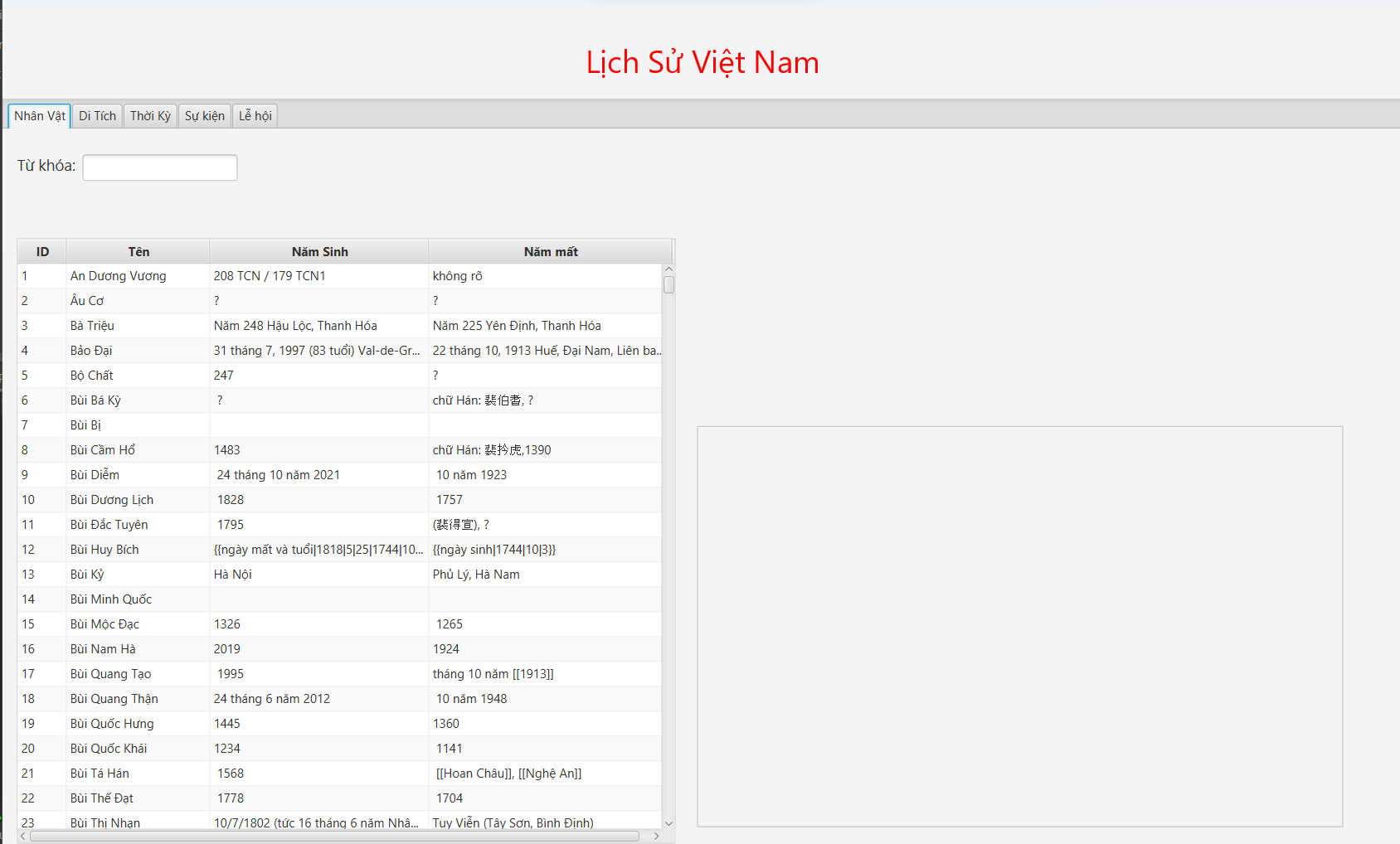
Quá trình thu thập dữ liệu có lưu lại trạng thái chính cách làm là lưu lại index của đối tượng đó trong mảng để tiếp tục nếu quá trình thu thập xảy ra lỗi.

Chạy chương trình :

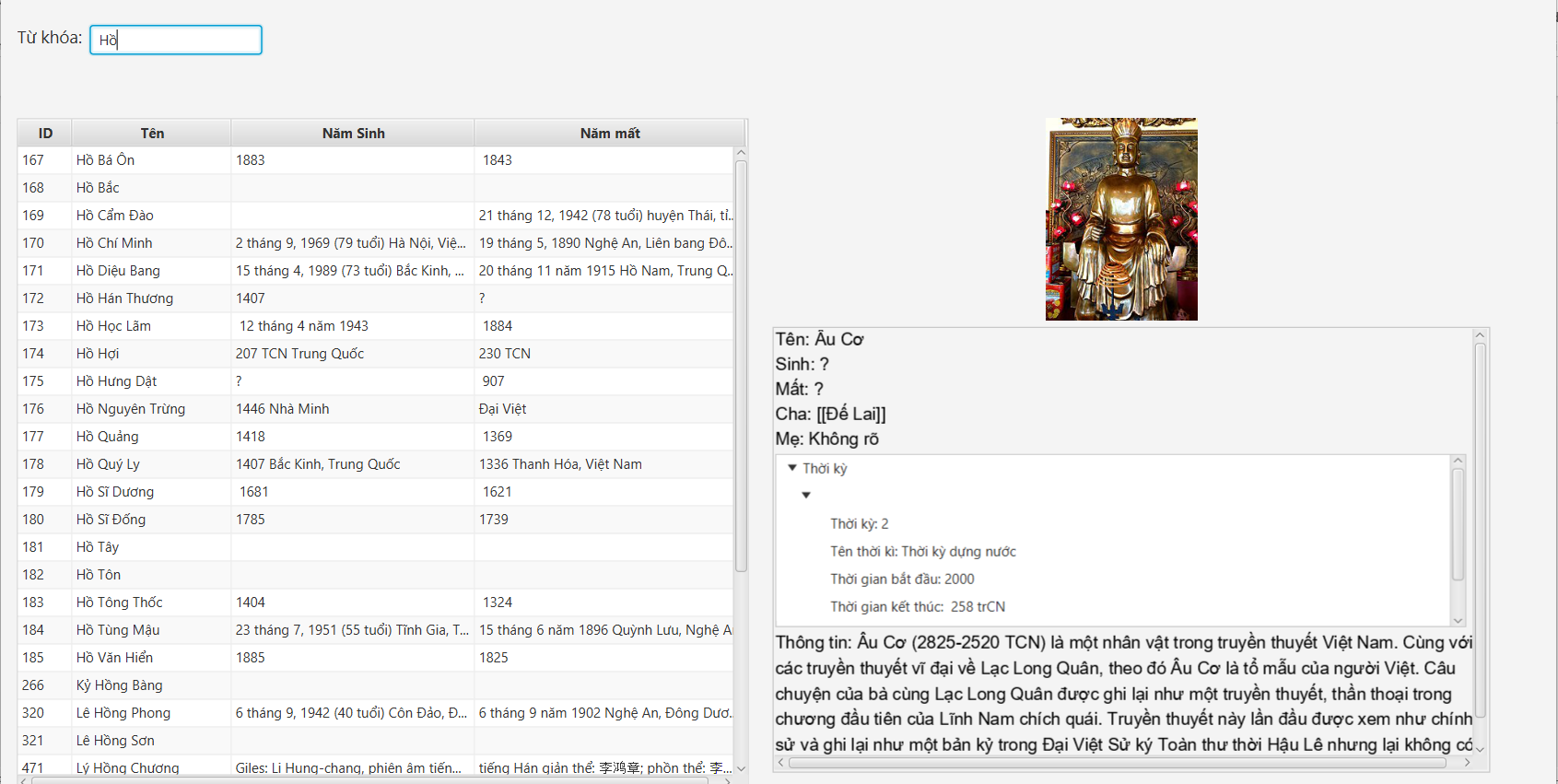
* Run Class Main.
* Chạy từ 1 đến 6 để cào và lưu dữ liệu.
* Mở gui/Painter.java để mở giao diện.

1. **Một vài hình ảnh về chương trình**

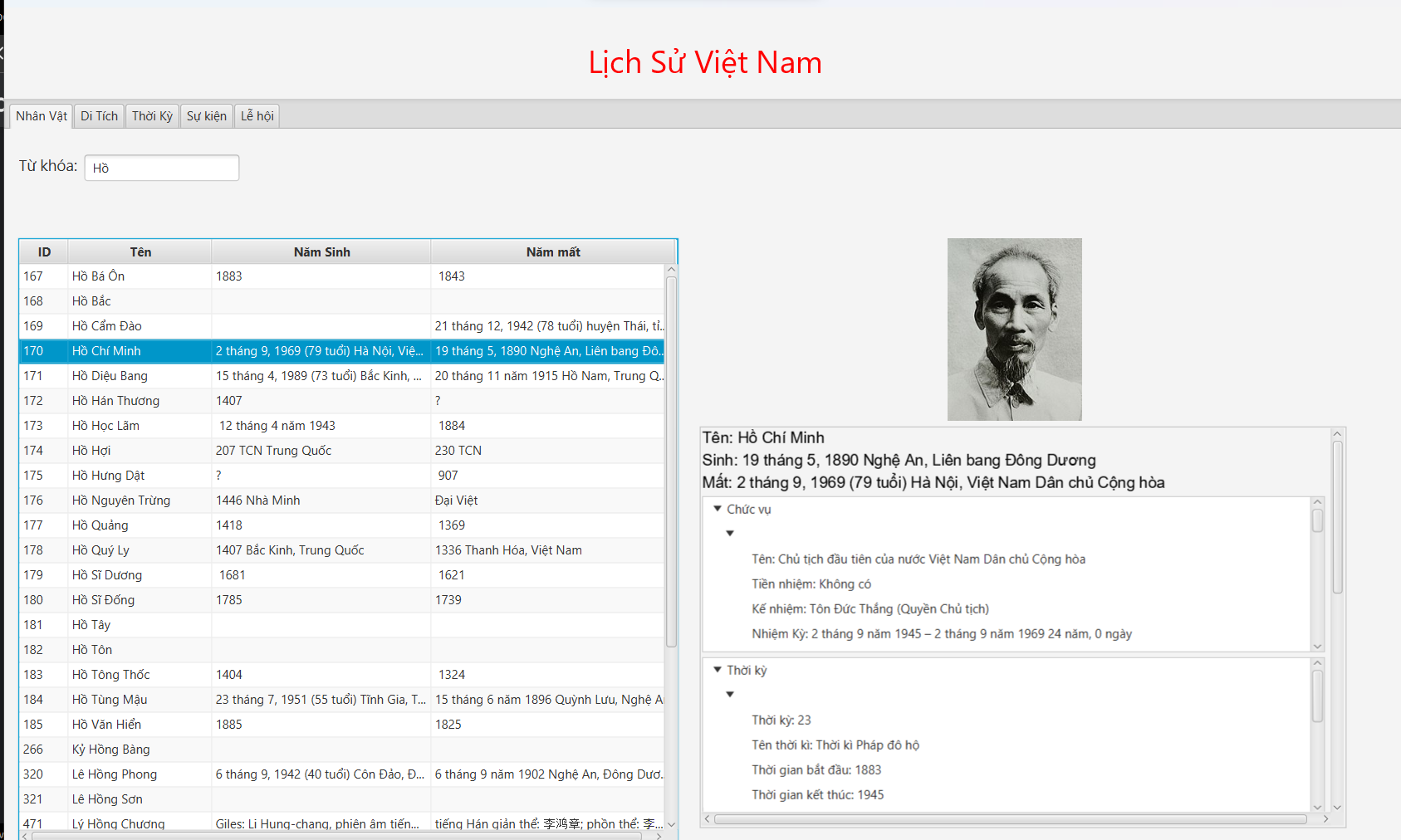
* Giao diện Nhân Vật

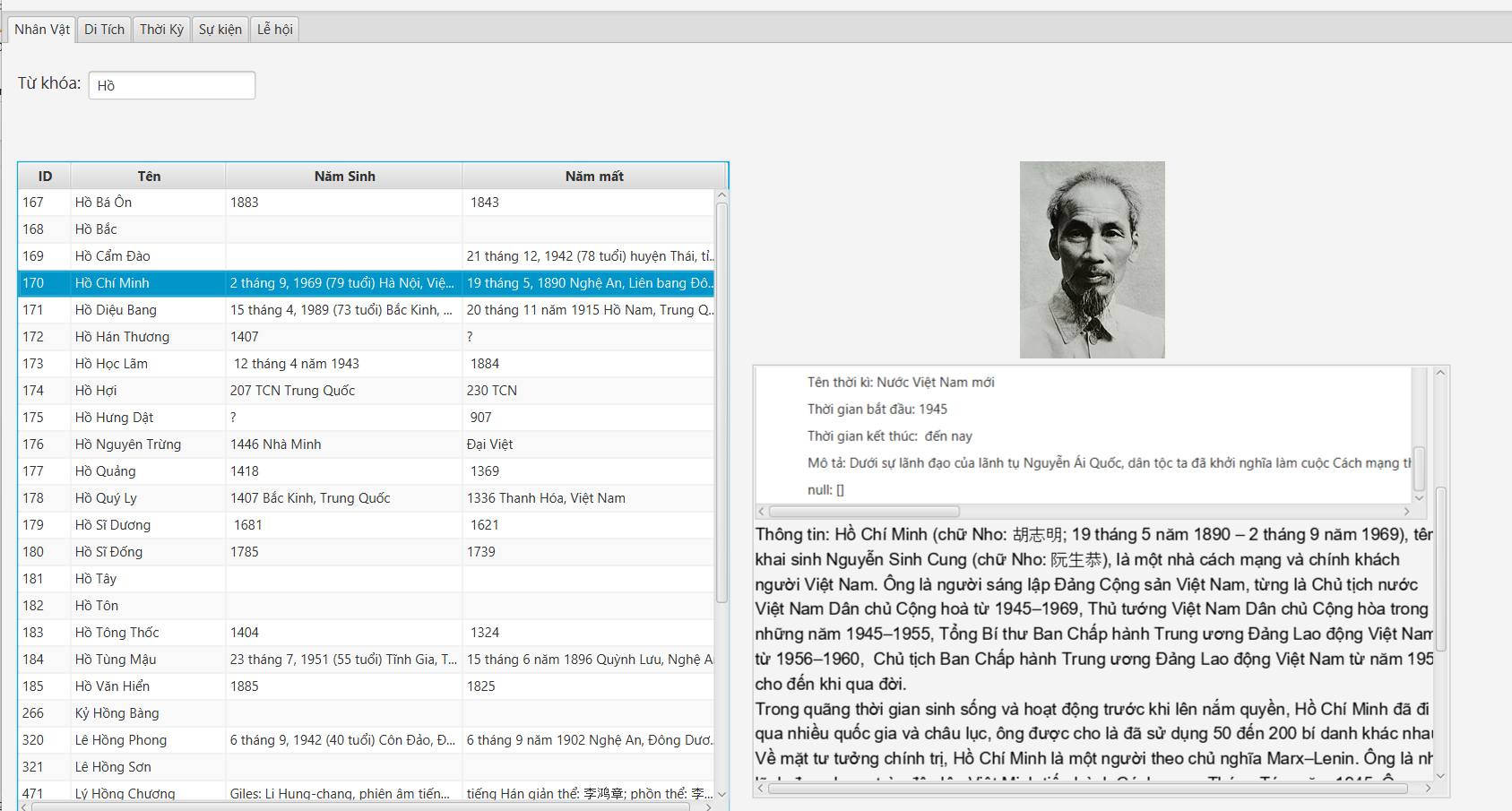


* Khi tìm kiếm một nhân vật, ta chỉ cần nhập tên nhân vật vào ô tìm kiếm, một list các gợi ý sẽ hiện ra giúp cho bạn dễ dàng và nhanh chóng chọn lựa chính xác đáp án của mình hơn.



Ví dụ khi ta chọn Hồ Chí Minh, chương trình sẽ hiện ra đầy đủ thông tin của nhân vật như ảnh dưới đây:





Khi muốn tìm kiếm các thông tin về các đối tượng khác, chúng ta chỉ cần kick chuột vào option tương ứng trên thanh công cụ và nhập từ khóa muốn tìm kiếm, tương tự như phần nhân vật ở trên.