ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

BÀI TẬP LỚN

HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

*(Mã học phần: IT3103)*

**Nhóm thực hiện : Nhóm 8**

**Lớp : 135287**

**Giảng viên hướng dẫn : GV.Trịnh Tuấn Đạt**

***Hà Nội, tháng 2 năm 2023***

Mục Lục

I. Mục tiêu đề tài.

1.1. Đề tài……………………………………………………………………4

1.2. Xác định yêu cầu chức năng của chương trình…………………………5

1.3. Nhóm sinh viên thực hiện đề tài………………………………………..5

II. Phân tích thiết kế

2.1. Xác định đối tượng của sản phẩm………………………………………5

2.1.1. Khách hàng, người sử dụng sản phẩm………………………………..5

2.1.2. Người quản trị………………………………………………………..5

2.2. Xác định nhiệm vụ……………………………………………………...5

2.2.1. Thu thập dữ liệu………………………………………………………6

2.2.2. Thiết kế……………………………………………………………….6

2.2.2.1. Biểu đồ UML………………………………………………………..6

III. Công nghệ sử dụng……………………………………………………..18

IV. Thống kê dữ liệu

4.1. Số lượng dữ liệu:....................................................................................18

4.2. Thuộc tính dữ liệu……………………………………………………..19

4.3. Số lượng liên kết giữa các thực thể……………………………………20

4.4. Thống kê nguồn dữ liệu………………………………………………..21

V. Giải thích thiết kế

5.1. Thiết kế thu thập dữ liệu……………………………………………….23

5.2. Thiết kế các thực thể…………………………………………………..26

5.3. Thiết kế giao diện……………………………………………………...27

5.4. Thiết kế lưu dữ liệu……………………………………………………28

VI. Các kĩ năng lập trình hướng đối tượng đã dùng

6.1. Đóng gói (Encapsulation).......................................................................28

6.2. Kế thừa (Inheritance)..............................................................................28

6.3. Đa hình (Polymorphism)........................................................................28

6.4. Trừu tượng (Abstraction).......................................................................29

6.5. Lập trình tổng quát (Generic).................................................................29

6.6. Ứng dụng nguyên lý SOLID…………………………………………..29

VII. Hướng dẫn sử dụng……………………………………………………30

# **Phân công công việc và đánh giá kết quả các thành viên trong nhóm**

| **Tên thành viên** | **Công việc thực hiện** | **Mức độ đóng góp** |
| --- | --- | --- |
| Trần Tiến Ngọc 20205009 | Thu thập dữ liệu triều đại, code phân tích dữ liệu triều lễ hội và triều đại, thiết kế giao diện | 16% |
| Nguyễn Quang Trường 20200653 | Thu thập dữ liệu di tích, code phân tích dữ liệu sự kiện lịch sử và di tích, thiết kế giao diện, thiết kế UML | 16% |
| Hoàng Danh Quân 20205112 | Thu thập dữ liệu triều đại và di tích, viết báo cáo | 18% |
| Đào Xuân Minh 20205100 | Thu thập dữ liệu nhân vật lịch sử, code phân tích dữ liệu triều lễ hội, triều đại và nhân vật lịch sử, tổng hợp code, làm slide | 18% |
| Nguyễn Lê Minh 20205002 | Thu thập dữ liệu lễ hội, code phân tích dữ liệu triều lễ hội và triều đại, viết báo cáo | 14% |
| Văn Đăng Huy 20205086 | Thu thập dữ liệu nhân vật lịch sử, vua, các lễ hội, code phân tích nhân vật lịch sử, vua, các lễ hội, xây dựng thiết kế phần thu thập dữ liệu, viết báo cáo. | 18% |

**1. Xác định yêu cầu đề tài**

**1.1. Đề tài**

Có rất nhiều các trang web cung cấp thông tin về lịch sử Việt Nam (<https://nguoikesu.com/>, Wikipedia, …). Cần tìm các trang web này và thu thập **tự động** dữ liệu về lịch sử Việt Nam và liên kết các dữ liệu này lại với nhau

Các thực thể cần thu thập bao gồm:

* Các triều đại lịch sử Việt Nam (thời Tiền Sử, Hồng Bàng, An Dương Vương, Bắc Thuộc lần I, …)
* Các nhân vật lịch sử Việt Nam (Ví dụ, các vị vua Việt Nam có thể thu thập tại <https://vi.wikipedia.org/wiki/Vua_Vi%E1%BB%87t_Nam>)
* Các địa điểm/di tích lịch sử Việt Nam
* Các lễ hội văn hóa Việt Nam
* Các sự kiện lịch sử Việt Nam

Mỗi thực thể cần có định danh, có các thuộc tính, và cần được liên kết với nhau.

Lưu ý:

* Việc thu thập dữ liệu phải là tự động
* Phải thống nhất cách đặt tên các thuộc tính cho mỗi loại thực thể
* Lượng dữ liệu thu thập được càng nhiều càng được đánh giá cao
* Liên kết được dữ liệu thu thập từ nhiều nguồn sẽ được đánh giá cao

Dữ liệu thu thập được cần được lưu trữ dưới dạng JSON hoặc CSV. Sau đó cung cấp chức năng tìm kiếm và hiển thị thông tin cho người dùng.

**1.2. Xác định yêu cầu chức năng của chương trình**

- Thực hiện xây dựng thực thể: Xây dựng các thuộc tính và phương thức mỗi thực thể

- Thực hiện thu thập dữ liệu: Thu thập dữ liệu từ nhiều website và liên kết với nhau. Đồng thời cần đảm bảo chương trình dễ bổ sung thêm dữ liệu.

- Thực hiện xây dựng giao diện tìm kiếm: Giao diện đẹp, hiển thị thông tin đầy đủ, rõ ràng. Đồng thời cung cấp tính năng tìm kiếm dữ liệu.

**1.3. Nhóm sinh viên thực hiện đề tài**

1. Trần Tiến Ngọc 20205009

2. Nguyễn Quang Trường 20200653

3. Hoàng Danh Quân 20205112

4. Nguyễn Lê Minh 20205002

5. Văn Đăng Huy 20205086

6. Đào Xuân Minh 20205100

**2. Phân tích thiết kế**

**2.1. Xác định đối tượng của sản phẩm**

**2.1.1. Khách hàng, người sử dụng sản phẩm**

Người dùng thực hiện tìm kiếm trong thanh tìm kiếm.

**2.1.2. Người quản trị**

Người quản trị thực hiện cập nhật dữ liệu thường xuyên

**2.2. Xác định nhiệm vụ**

**2.2.1. Thu thập dữ liệu**

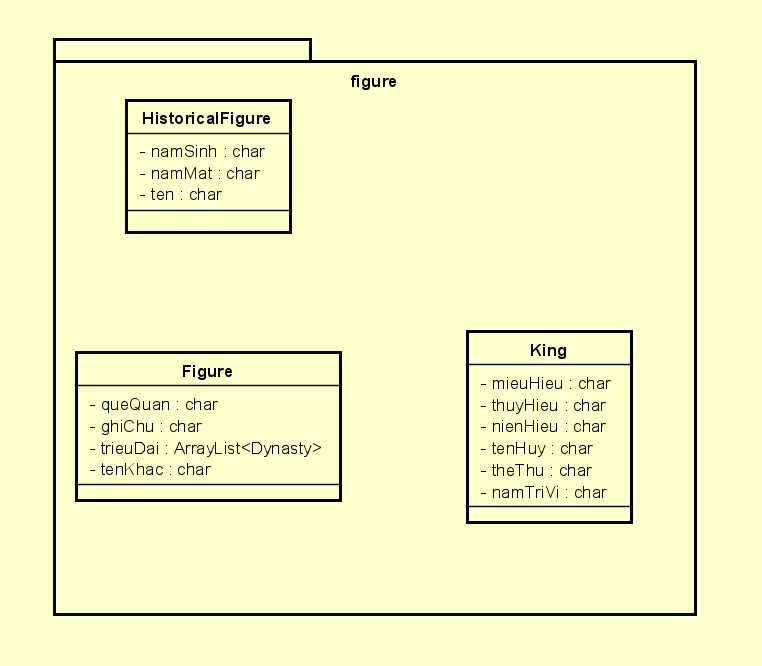
Thu thập dữ liệu từ nhiều website

**2.2.2. Thiết kế**

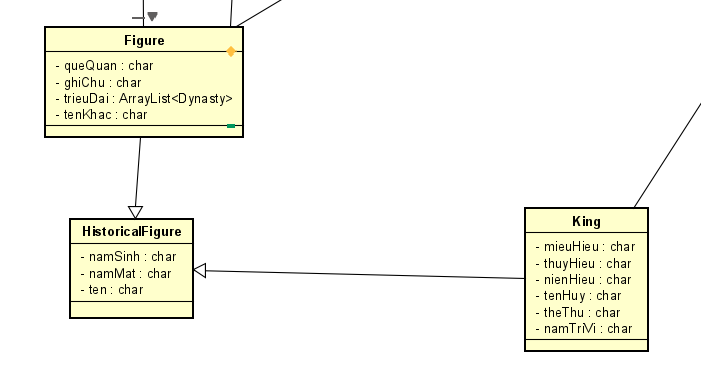
**2.2.2.1. Biểu đồ UML**

1. Package Object

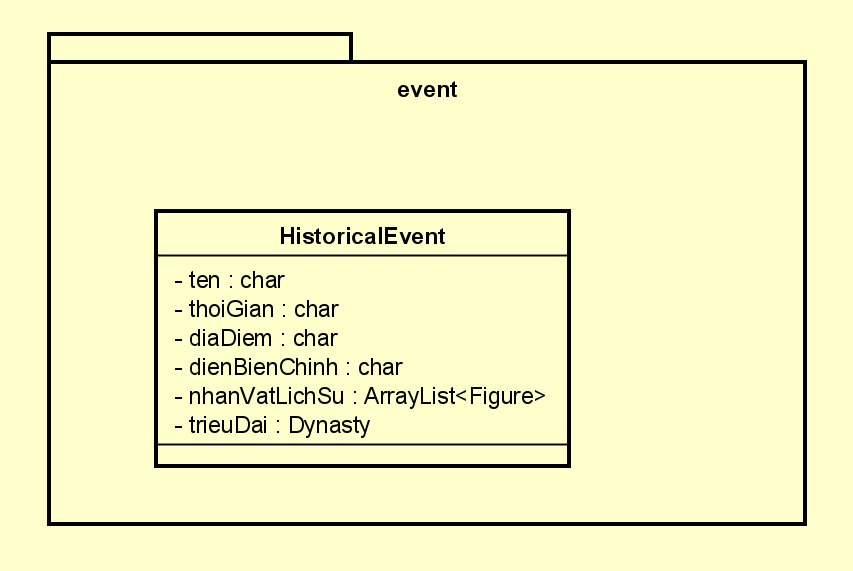
1.1. Package object.figure



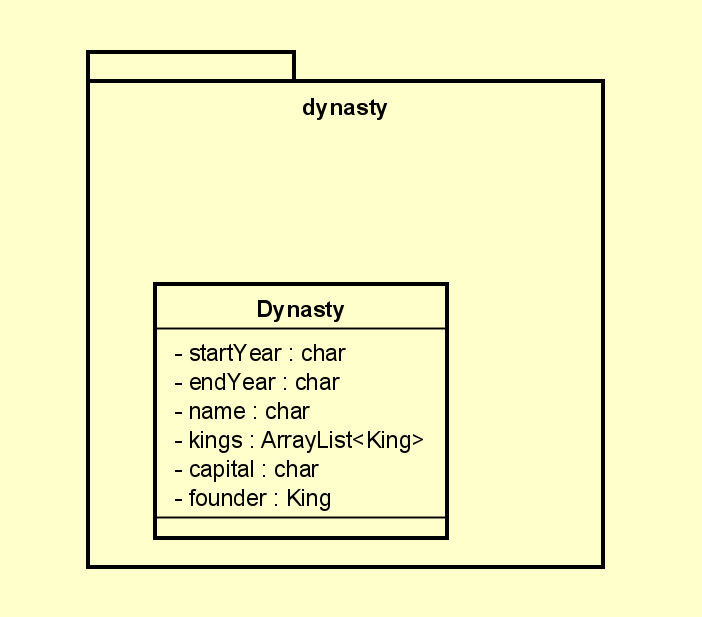
Biểu đồ lớp



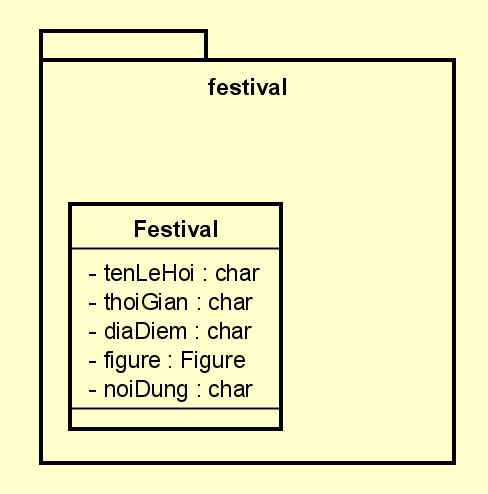
1.2. Package object.event



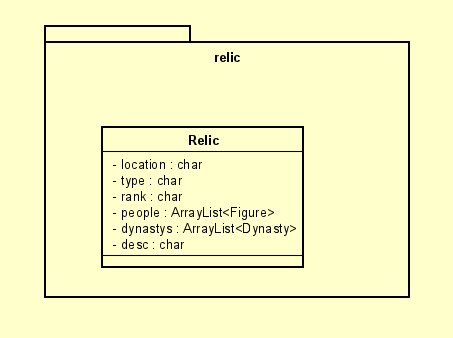
1.3. Package object.dynasty



1.4. Package object.festival

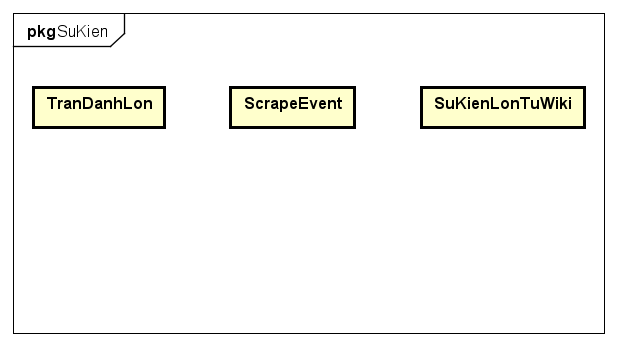


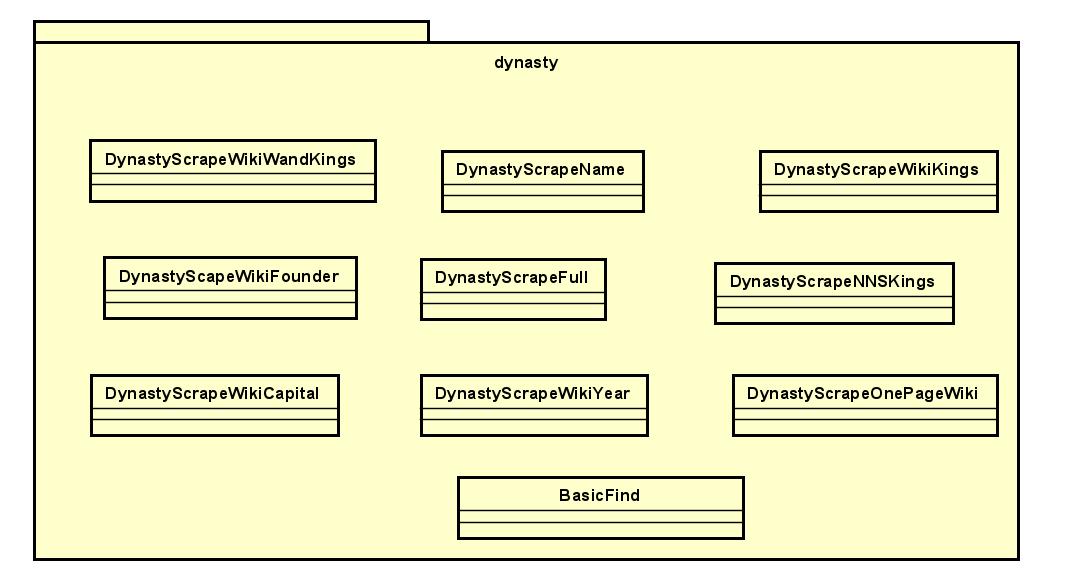
1.5. Package object.relic



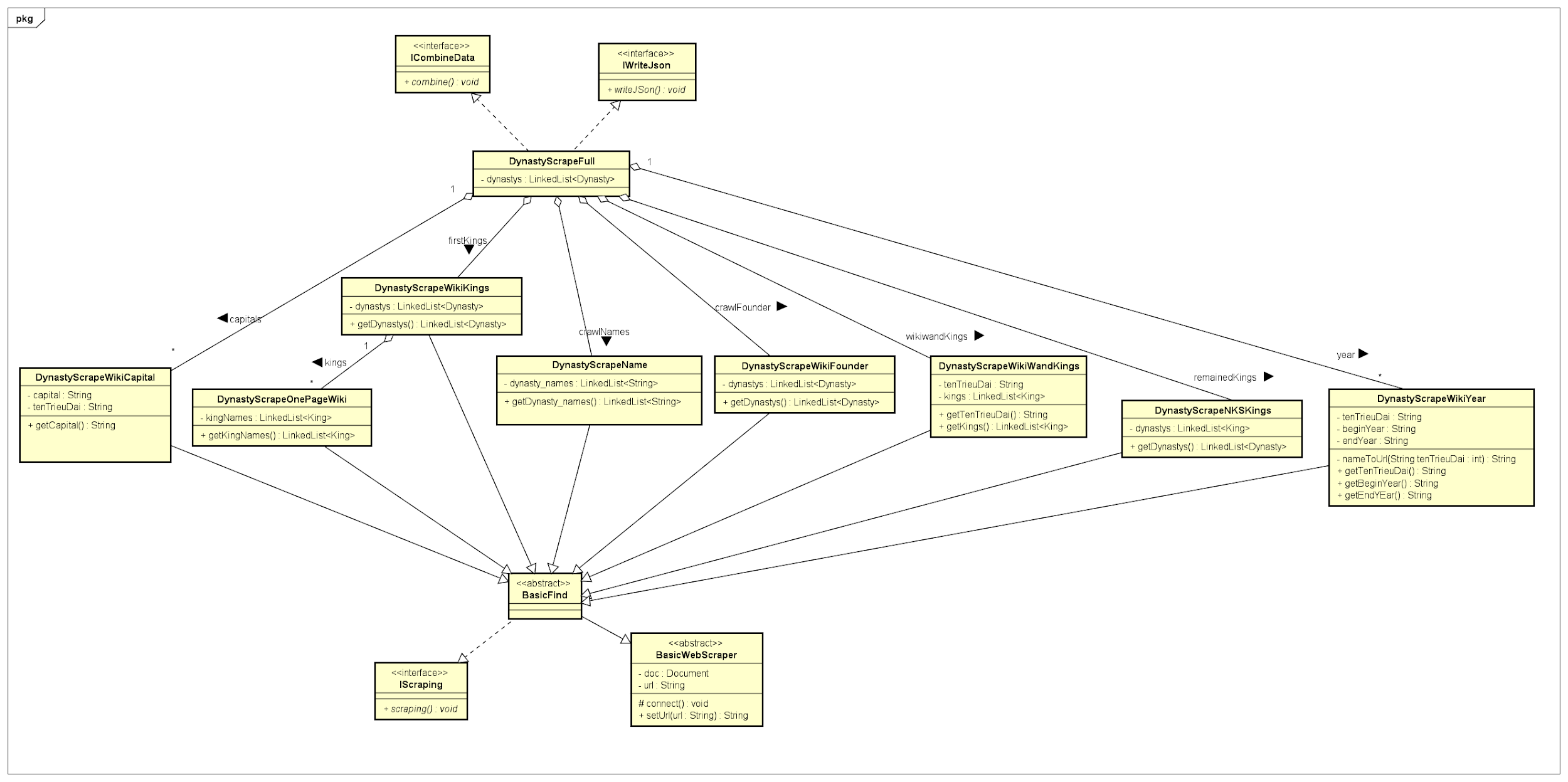
2. Package webcrawler

2.1. Package webcrawler.dynasty

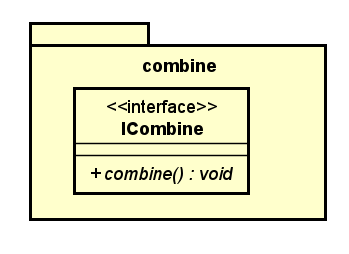




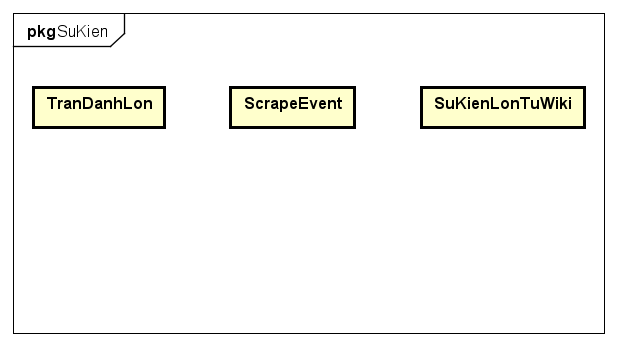
Biểu đồ lớp

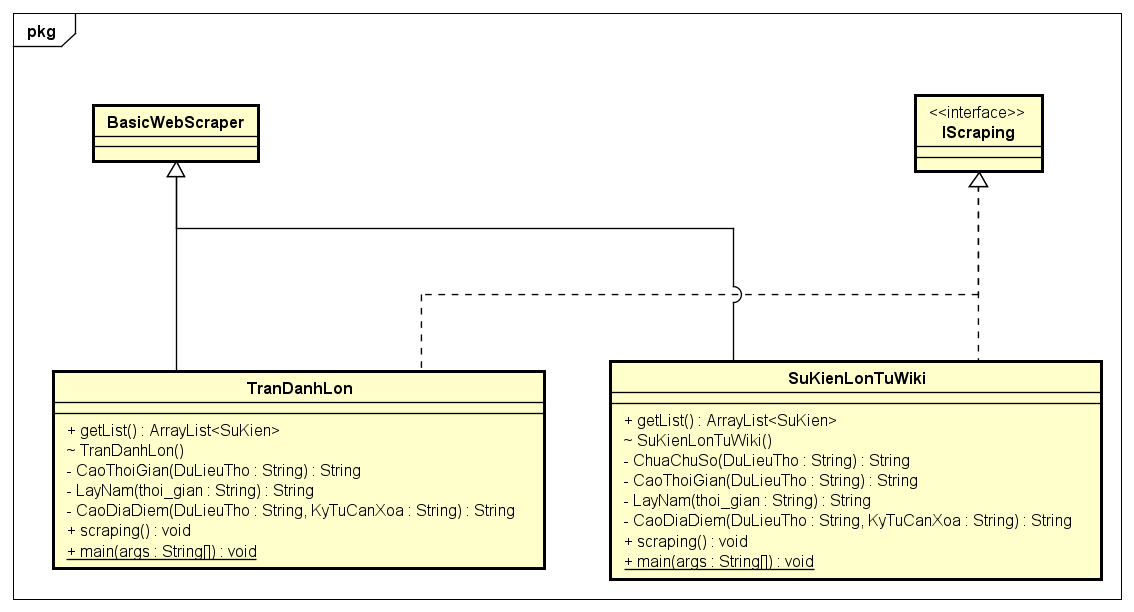


2.2. Package webcrawler.combine

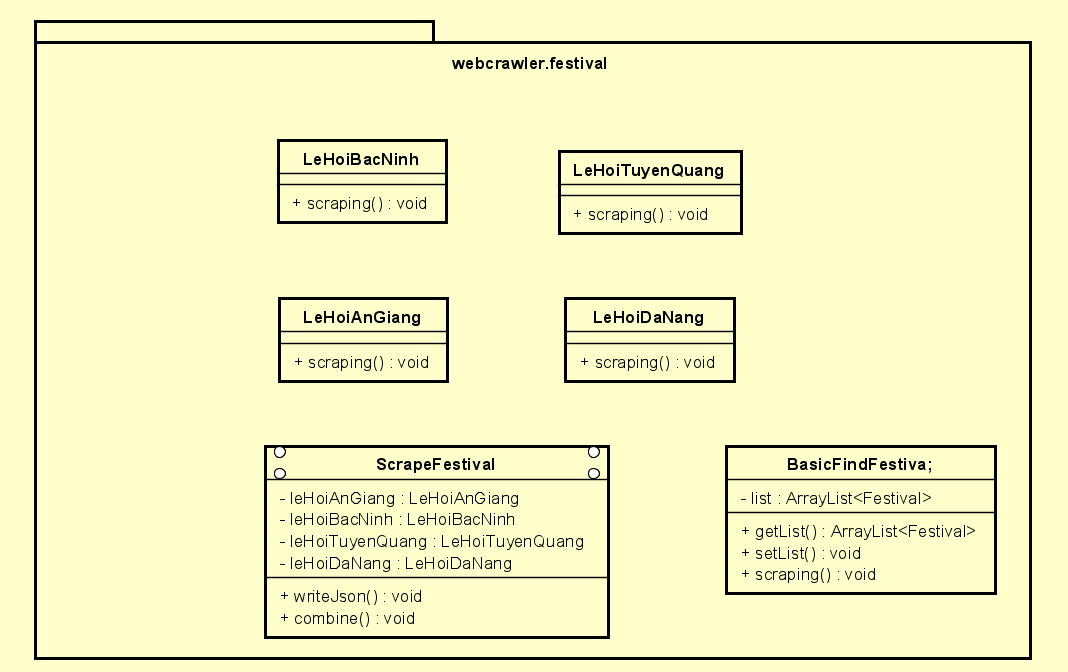


2.3. Package webcrawler.event

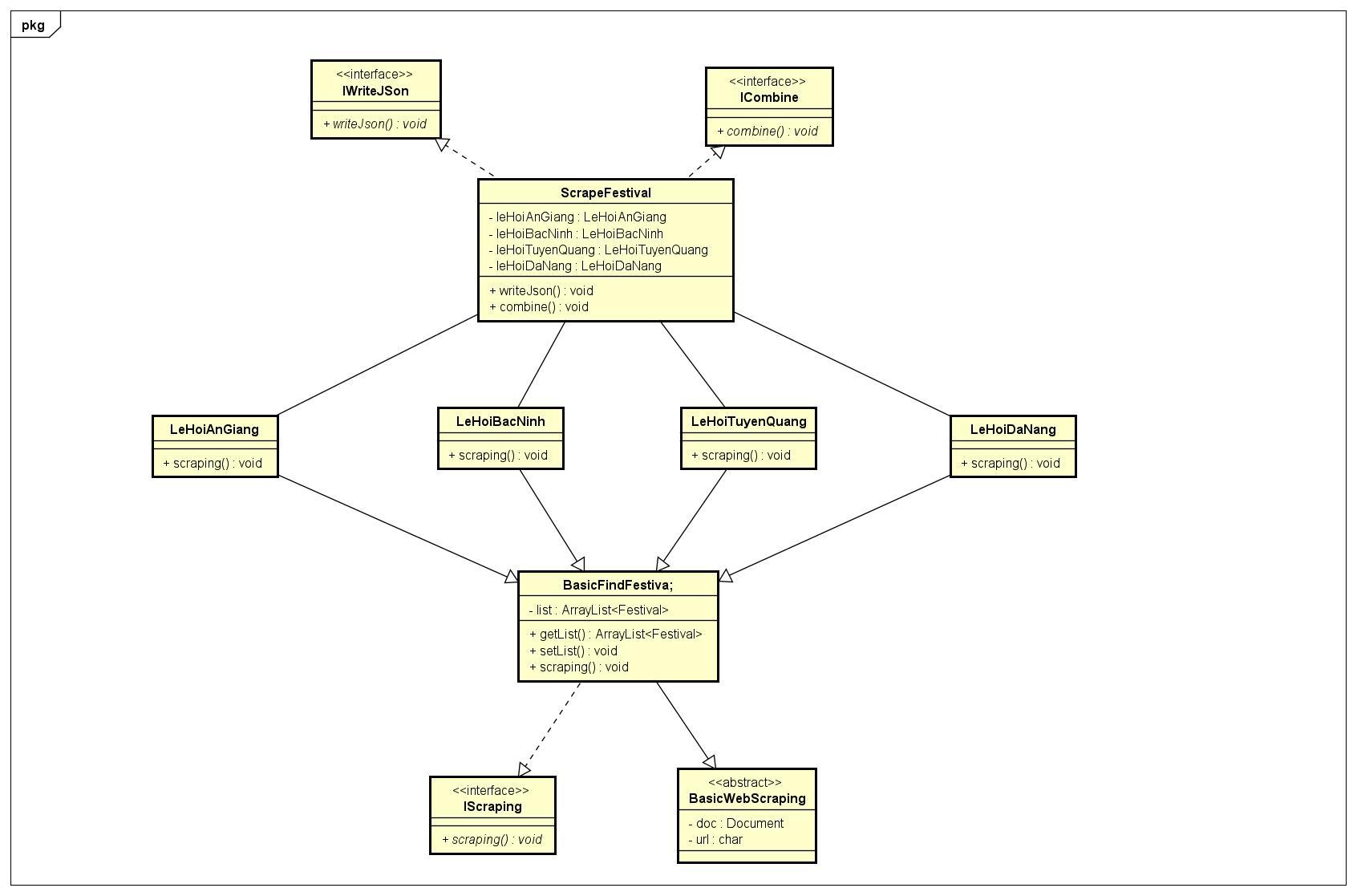


Biểu đồ lớp

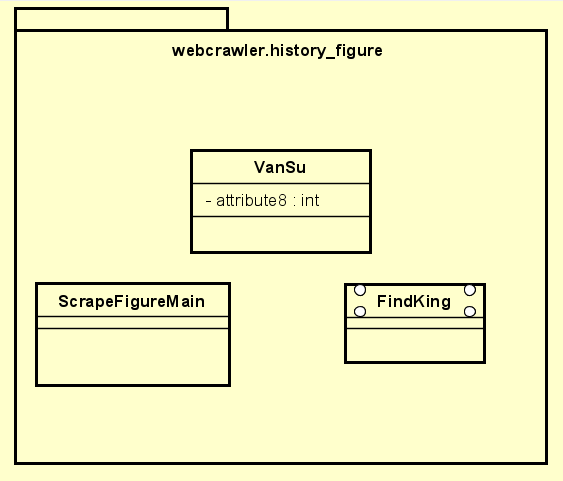
2.4. Package webcrawler.festival



Biểu đồ lớp

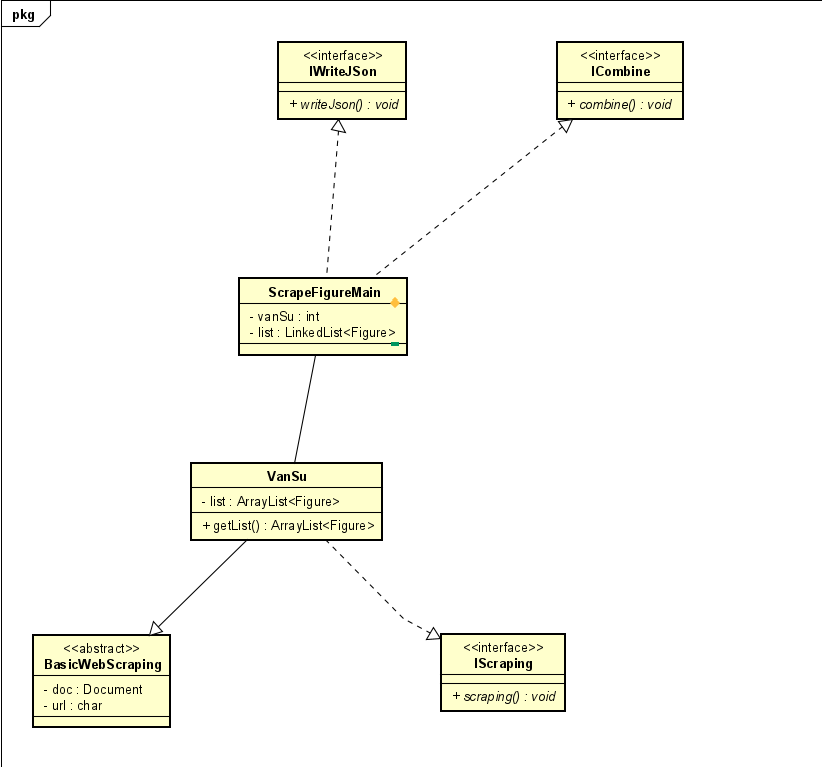


2.5. Package webcrawler.history\_figures

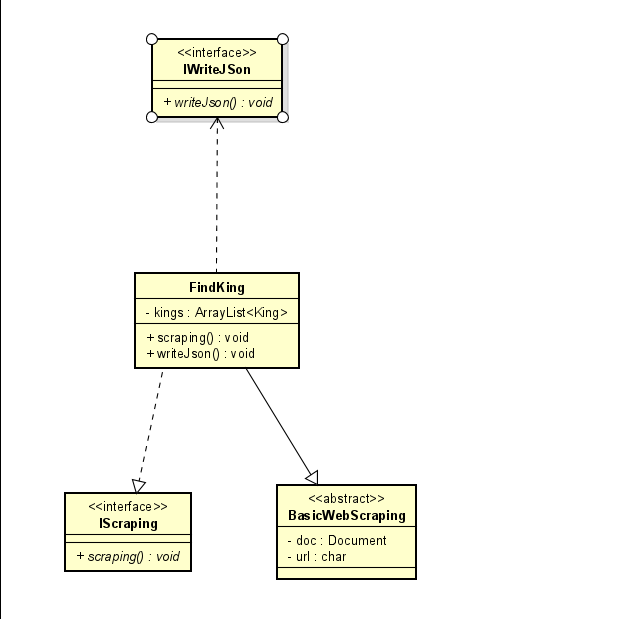


Biểu đồ lớp trong package

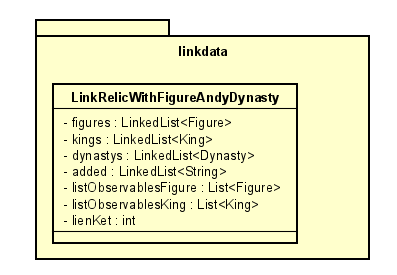
Class ScrapingFigureMain



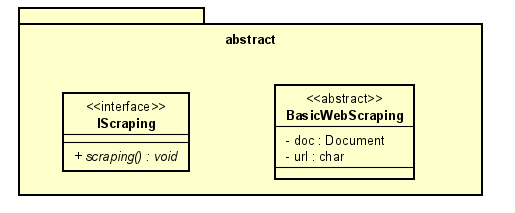
Class FindKing



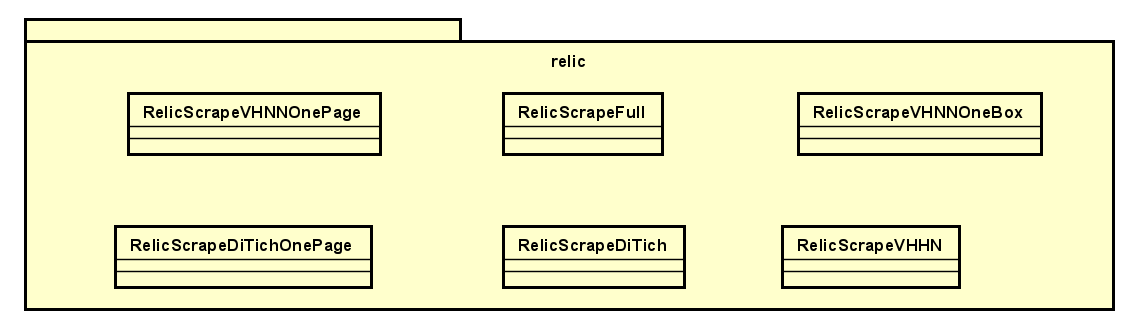
2.6. Package webcrawler.linkdata



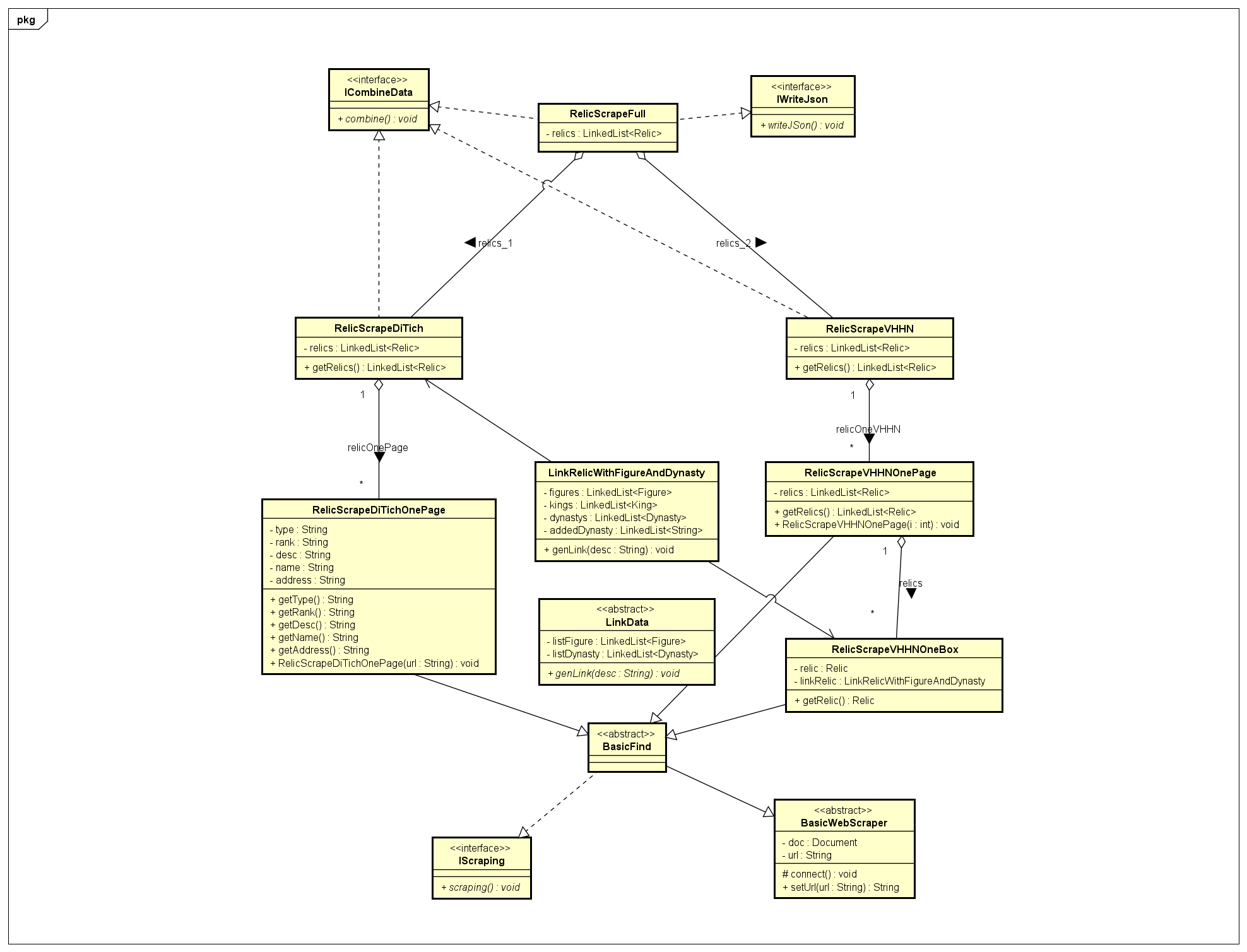
2.7. Package webcrawler.parent



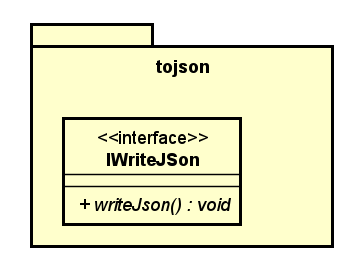
2.8. Package webcrawler.relic



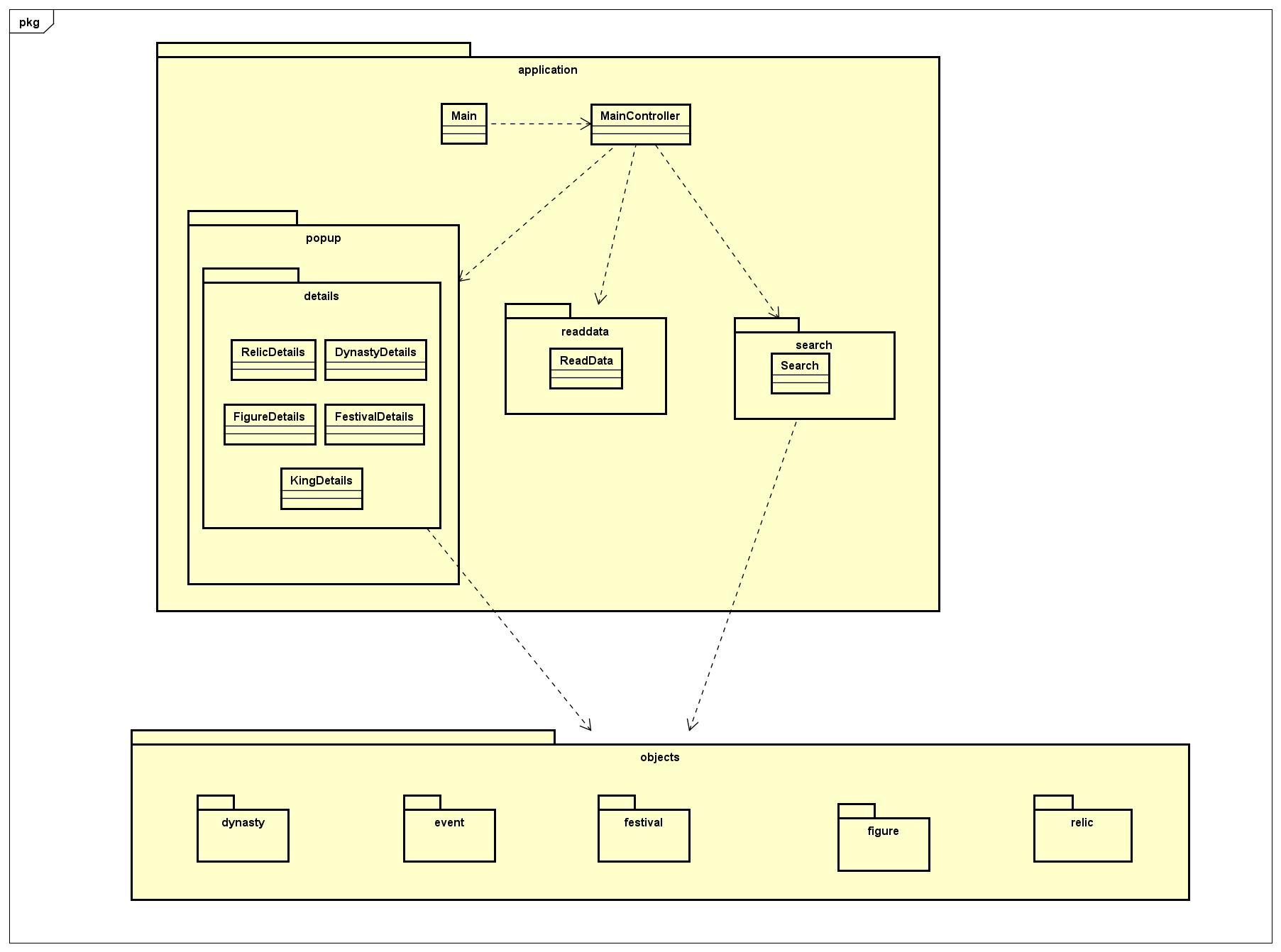
Biểu đồ lớp



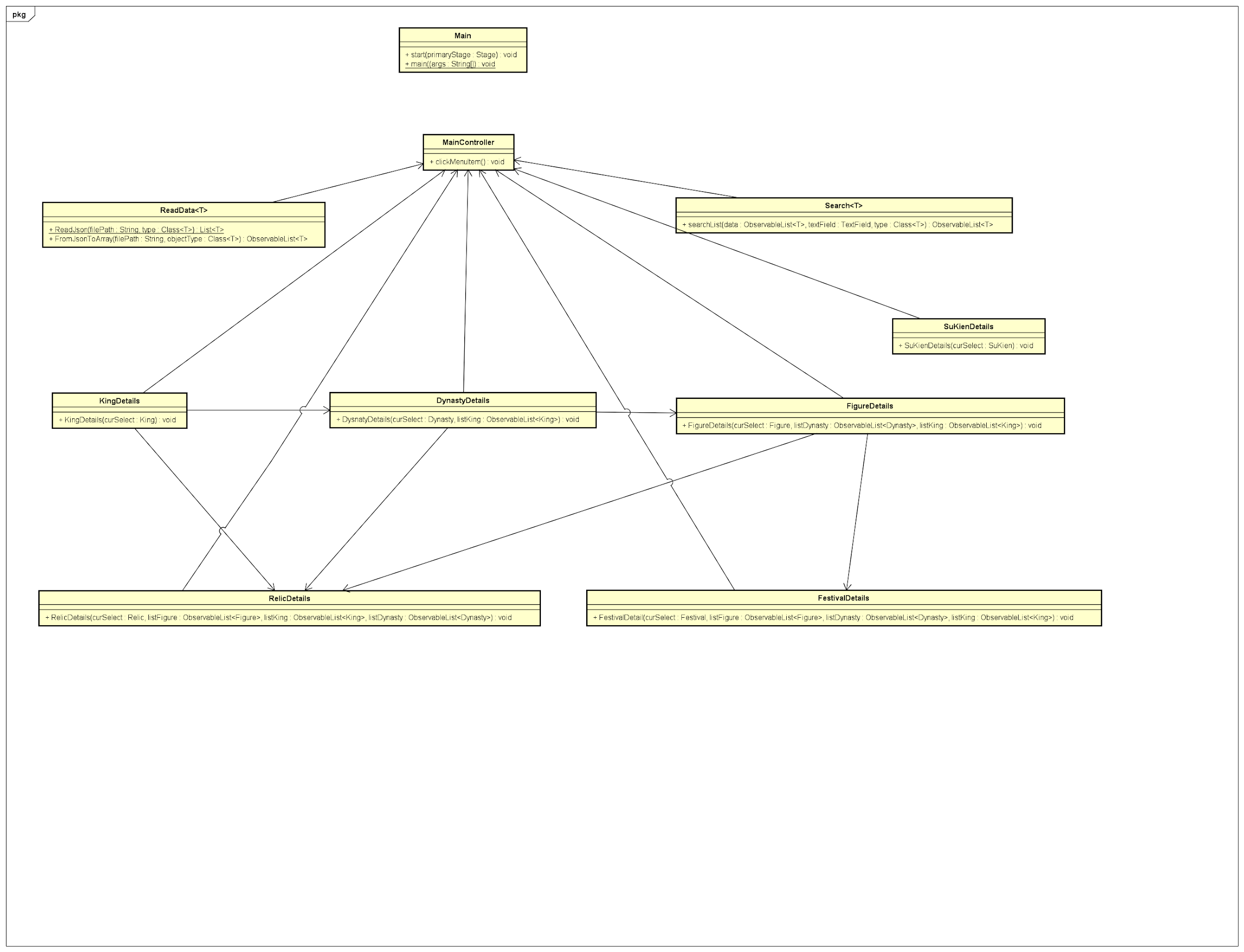
2.9. Package webcrawler.tojson:



3. Package applicaition



Biểu đồ lớp



**3. Công nghệ sử dụng**

- Asta UML: Phần mềm Asta UML hỗ trợ người dùng vẽ các biểu đồ UML gói, biểu đồ UML lớp và thể hiện liên kết giữa các lớp.

- JavaFX: Sử dụng JavaFX để lập trình GUI cho chương trình. Các bộ công cụ và component trong JavaFX được sử dụng để tạo các component trong giao diện chính, xử lý dữ liệu từ file JSON và hiển thị cho người dùng.

- SceneBuilder: Sử dụng SceneBuilder để thiết kế GUI cho chương trình, bao gồm các component trên giao diện chính như thanh công cụ tìm kiếm, bảng thông tin, … Sau khi thiết kế xong, thành quả bao gồm file Main.fxml chứa các component và class MainController.java để điều khiển các component trong Main.fxml.

- Jsoup: Jsoup là thư viện cho phép kết nối với các website để thu thập dữ liệu, bao gồm các đường dẫn (URL), cấu trúc html của trang web và dữ liệu trên website, …

- Gson: Gson là thư viện cho phép chuyển đổi các đối tượng trong Java thành đối tượng Gson (JSONObject), và viết các đối tượng thành file .json.

**4. Thống kê dữ liệu:**

**4.1. Số lượng dữ liệu:**

Với các thực thể như trên, số lượng thông tin thực thể đã thu thập được như sau:

* King: 163 thực thể
* Figure: 2300 thực thể
* Relic: 4700 thực thể
* Dynasty: 33 thực thể
* SuKien: 302 thực thể

**4.2. Thuộc tính dữ liệu**

Với mỗi thực thể như trên, số lượng thuộc tính dữ liệu của mỗi thực thể bao gồm:

* King: 7 thuộc tính gồm:

+ paperURL: link bài viết về vị vua trên trang web

+ mieuHieu: Miếu hiệu

+ thuyHieu: Thúy hiệu

+ nienHieu: Niên hiệu

+ tenHuy: Tên húy

+ theThu: Thế thứ

+ namTriVi: Năm trị vì

+ ten: Tên vị vua

* Figure: 5 thuộc tính gồm

+ queQuan: Quê quán

+ ghiChu : Ghi chú về nhân vật

+ tenKhac: Tên khác

+ trieuDai: Triều đại của nhân vật

+ ten: Tên nhân vật

+ namSinh: Năm sinh

+ namMat : Năm mất.

* Relic: gồm 7 thuộc tính

+ name: Tên di tích lịch sử:

+ location: Địa điểm

+ type: Loại địa điểm như Đình, Chùa, …

+ rank: Danh hiệu của di tích

+ figures: Các nhân vật lịch sử liên quan

+ kings: Các vị vua liên quan

+ dynasty: Triều đại di tích được xây dựng

* Dynasty: 6 thuộc tính gồm

+ name: Tên triều đại

+ startYear: Năm bắt đầu triều đại

+ endYear: Năm kết thúc triều đại

+ kings: Tên các vị vua trong triều đại

+ capital: Thủ đô, nơi đóng đô

+ founder: Vị vua khai sinh triều đại

* SuKien: gồm 3 thuộc tính

+ ten: Tên sự kiện

+ thoi\_gian: Thời gian xảy ra sự kiện

+ dia\_diem: Địa điểm xảy ra sự kiện

* Festival: gồm 5 thuộc tính

+ tenLeHoi: Tên lễ hội

+ thoigian: Thời gian tổ chức lễ hội

+ diaDiem: Địa điểm tổ chức lễ hội

+ figure: Tên nhân vật được vinh danh tại lễ hội

+ noiDung: Nội dung, chủ đề lễ hội.

**4.3. Số lượng liên kết giữa các thực thể**

Có 8 liên kết giữa các thực thể bao gồm

- Figure - kế thừa - HistoricalFigure

- King - kế thừa - HistoricalFigure

- Festival - tôn vinh – Figure

- Relic - thờ - Figure

- HistoricalEvent - có - Figure

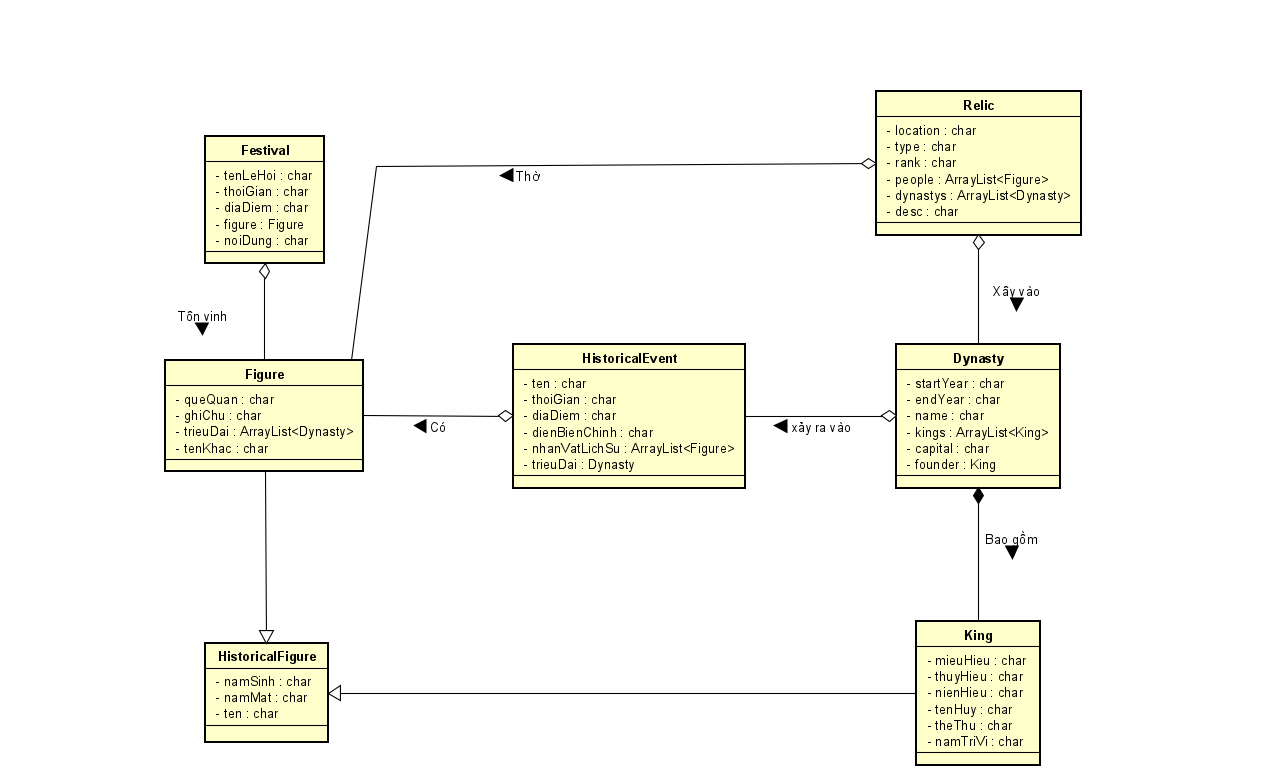
- HistoricalEvent - xảy ra vào - Dynasty

- Relic - xây vào - Dynasty

- Dynasty – bao gồm - King

Vậy có : 5 quan hệ Aggregation (kết tập), 1 quan hệ Composition (hợp thành) và 2 quan hệ Generalization (kế thừa)

Quan hệ giữa các thực thể được thể hiện qua sơ đồ sau



**4.4. Thống kê nguồn dữ liệu**

**Trang web thu thập dữ liệu của thực thể King:**

- Wikipedia, <https://vi.wikipedia.org/wiki/Vua_Vi%E1%BB%87t_Nam>:

**Trang web thu thập dữ liệu của thực thể Figure:**

- Vansu, <https://vansu.vn/viet-nam/viet-nam-nhan-vat/>

**Trang web thu thập dữ liệu của thực thể Festival(lễ hội):**

- Angiangtourist, <https://angiangtourist.vn/thoi-gian-va-dia-diem-to-chuc-cac-le-hoi-lon-o-an-giang/#Tr>

- [https://alltours.vn/bac-ninh/nhung-le-hoi-o-bac-ninh.html"-](https://alltours.vn/bac-ninh/nhung-le-hoi-o-bac-ninh.html%22-)

- <https://dulichkhampha24.com/le-hoi-o-da-nang.html>

- <https://alltours.vn/tuyen-quang/cac-le-hoi-o-tuyen-quang.html>

- <https://vi.wikipedia.org/wiki/L%E1%BB%85_h%E1%BB%99i_Vi%E1%BB%87t_Nam>

**Trang web thu thập dữ liệu của thực thể Event(sự kiện, trận đánh lớn,...):**

-

<https://vi.wikipedia.org/wiki/Ni%C3%AAn_bi%E1%BB%83u_l%E1%BB%8Bch_s%E1%BB%AD_Vi%E1%BB%87t_Nam>

- <https://vi.wikipedia.org/wiki/Danh_s%C3%A1ch_tr%E1%BA%ADn_%C4%91%C3%A1nh_trong_l%E1%BB%8Bch_s%E1%BB%AD_Vi%E1%BB%87t_Nam>

**Trang web thu thập dữ liệu các triều đại Việt Nam(Dynasty):**

-<https://nguoikesu.com/tu-lieu/bang-doi-chieu-cac-trieu-dai-viet-nam-va-cac-trieu-dai-trung-quoc>:

+ Dữ liệu về vua của 3 triều đại

-<https://vi.wikipedia.org/wiki/Vua_Vi%E1%BB%87t_Nam>:

+ Dữ liệu tên 32 triều đại/thời kì

- [https://vi.wikipedia.org/wiki/Nh%C3%A0\_Nguy%E1%BB%85n:](https://vi.wikipedia.org/wiki/Nh%C3%A0_Nguy%E1%BB%85n%20:)

+ Trang web wiki về nhà Nguyễn cào được dữ liệu về năm thành lập, năm kết thúc và các vị vua của Nhà Nguyễn, tương tự như vậy cho 8 triều đại ở trang [Triều đại – Wikipedia tiếng Việt](https://vi.wikipedia.org/wiki/Tri%E1%BB%81u_%C4%91%E1%BA%A1i)

-<https://vi.wikipedia.org/wiki/Th%E1%BB%A7_%C4%91%C3%B4_Vi%E1%BB%87t_Nam>

+ Thu thập được kinh đô của 33 triều đại

-

<https://www.wikiwand.com/vi/B%E1%BA%A3n_m%E1%BA%ABu:Danh_s%C3%A1ch_vua_v%C3%A0_ho%C3%A0ng_%C4%91%E1%BA%BF_Vi%E1%BB%87t_Nam>

+ Thu thập được dữ liệu về vua của 13 triều đại

**Trang web thu thập dữ liệu của thực thể Relic (Di tích lịch sử):**

-<http://ditich.vn/FrontEnd/DiTich/Form?do=&ItemId=>

+ Thu thập được 4275 dữ liệu di tích gồm tên, địa điểm, mô tả, tạo được tổng cộng 474 liên kết với triều đại, 463 liên kết với vua, 113 liên kết với các nhân vật lịch sử.

-<http://ditichlichsu-vanhoahanoi.com/category/2dtlsvh/page/>

+ Thu thập được 425 dữ liệu di tích gồm tên, địa điểm, mô tả, do mô tả ở trang này khá dài nên tạo được khá nhiều liên kết, tổng cộng 825 liên kết với triều đại, 891 liên kết với nhân vật lịch sử, 2422 liên kết với các vị vua.

**5. Giải thích thiết kế**

**5.1. Thiết kế thu thập dữ liệu**

Ý tưởng: Việc thu thập dữ liệu được thực hiện tự động, dữ liệu được cào theo từng thực thể. Mỗi thực thể sẽ có một package riêng, và mỗi website sẽ được thu thập dữ liệu trong một class riêng. Sau khi thu thập xong, các website sẽ được tổng hợp dữ liệu tại 1 class và ghi và file JSON.

- Phần thiết kế thu thập dữ liệu được nằm trong package webcrawler bao gồm các package sau:

a. Package webcrawler.abstract

Package chứa các class để từng class của các website kế thừa, bao gồm các class:

- BasicWebScraper: lớp abstract chứa thông tin url của website và thực hiện kết nối website

- IScaping: interface chứa phương thức scraping() để lọc dữ liệu của website, mỗi website sẽ override phương thức trên để lọc dữ liệu.

b. Package webcrawler.combine

Package chứa class để tổng hợp dữ liệu các website, gồm

- ICombineData: interface chứa phương thức combine() để tổng hợp của các website, mỗi class tổng hợp dữ liệu sẽ override phương thức trên để tổng hợp. Mỗi object của các class từng website sẽ gọi phương thức scraping() tại đây.

c. Package webcrawler.tojson

Package chứa class để xuất file JSON, gồm

- IWriteJson: interface chứa phương thức writeJson() để xuất file JSON, mỗi class tổng hợp dữ liệu sẽ override phương thức trên để xuất file JSON.

d. Package webcrawler.dynasty

Package chứa các class để thu thập dữ liệu các triều đại Việt Nam, gồm

- Mỗi website sẽ được làm thành 1 class kế thừa các class trong package webcrawler.abstract, thực hiện thu thập dữ liệu.

- DynastyScrapeFull: class tổng hợp dữ liệu từ các website kế thừa interface ICombineData và IWriteJson , mỗi object của 1 class website là thuộc tính của DynastyScrapeFull.

e. Package webcrawler.event

Package chứa các class để thu thập dữ liệu các sự kiện lịch sử, gồm

- Mỗi website sẽ được làm thành 1 class kế thừa các class trong package webcrawler.abstract, thực hiện thu thập dữ liệu.

- ScrapeEvent: class tổng hợp dữ liệu, kế thừa interface ICombineData và IWriteJson , mỗi object của 1 class website là thuộc tính của ScrapeEvent.

f. Package webcrawler.festival

Package chứa các class để thu thập dữ liệu các lễ hội, gồm

- BasicFindEvent: abstract class kế thừa các class trong package webcrawler.abstract, có thuộc tính là một ArrayList lưu các lễ hội.

Mỗi website sẽ được làm thành 1 class kế thừa class BasicFindEvent, thực hiện thu thập dữ liệu.

- FindFestival: class tổng hợp dữ liệu, kế thừa interface ICombineData và IWriteJson , mỗi object của 1 class website là thuộc tính của FindFestival.

g. Package webcrawler.history\_figure

Package chứa các class để thu thập dữ liệu các nhân vật, gồm

- Mỗi website sẽ được làm thành 1 class kế thừa các class trong package webcrawler.abstract, thực hiện thu thập dữ liệu.

- FindKing: kế thừa các class trong package webcrawler.abstract, thực hiện thu thập dữ liệu về các vị vua cho thực thể King.

- ScrapingFigureMain: thực hiện tổng hợp dữ liệu và xuất file JSON, kế thừa interface ICombineData và IWriteJson , mỗi object của 1 class website là thuộc tính của ScrapingFigureMain.

- VanSu: kế thừa các class trong package webcrawler.abstract, thực hiện thu thập dữ liệu về các nhân vật lịch sử khác cho thực thể Figure.

h. Package webcrawler.relic

Package chứa các class để thu thập dữ liệu các di tích lịch sử, gồm:

- Mỗi website sẽ được làm thành 1 class kế thừa các class trong package webcrawler.abstract, thực hiện thu thập dữ liệu.

- RelicScrapingDiTich: thực hiện tổng hợp dữ liệu và xuất file JSON, kế thừa interface ICombineData và IWriteJson , mỗi object của 1 class website là thuộc tính của RelicScrapingDitich.

i. Package webcrawler.linkdata

Package chứa các class để liên kết thuộc tính các thực thể, gồm:

- LinkRelicWithFigureAndDynasty: liên kết thực thể Figure và Dynasty với Relic.

**5.2. Thiết kế các thực thể.**

Ý tưởng các thực thể được trừu tượng hóa thành các class.

a. Package objects.

Package lưu các thực thể cần thu thập dữ liệu.

b. Package objects.dynasty

Package lưu các thực thể Dynasty, là thực thể các triều đại Việt Nam.

- Dynasty: class lưu thuộc tính và phương thức của thực thể Dynasty

c. Package objects.dynasty

Package lưu các thực thể SuKien, là các sự kiện lịch sử Việt Nam.

- SuKien: class lưu thuộc tính và phương thức của thực thể SuKien

d. Package objects.festival

Package lưu các thực thể Festival, là các lễ hội Việt Nam.

- SuKien: class lưu thuộc tính và phương thức của thực thể Festival

c. Package objects.figure

Package lưu các thực thể Figure và King, là các nhân vật lịch sử Việt Nam.

- HistoricalFigure: abstract class, lưu thông tin tên năm sinh năm mất của nhân vật.

- Figure: kế thừa HistoricalFigure, gồm các thuộc tính của Figure

- King: kế thừa HistoricalFigure, gồm các thuộc tính của King

e. Package objects.relic

Package lưu thực thể Relic, là các di tích lịch sử Việt Nam.

- Relic: class lưu thuộc tính và phương thức của thực thể Relic

**5.3. Thiết kế giao diện**

Ý tưởng: Thiết kế giao diện phần mềm với JavaFX, các chức năng chia thành các package riêng

a. Package application

Package chứa các package con phục vụ các chức năng của phần giao diện, đồng thời chứa class chạy chương trình và class controller của JavaFX

- application.css: File css tạo giao diện phần mềm.

- Main: class chạy chính của phần mềm

- MainController: class controller điều khiển các component trong file Main.fxml.

b. Package application.popup.details

- Chứa giao diện hiện ra khi nhấn vào các trường thông tin hiện trên giao diện chính.

- Mỗi class phục vụ giao diện hiện ra của một thực thể.

- DynastyDetails: phục vụ giao diện hiện ra của thực thể Dynasty

- FestivalDetails: phục vụ giao diện hiện ra của thực thể Festival

- FigureDetails: phục vụ giao diện hiện ra của thực thể Figure

- KingDetails: phục vụ giao diện hiện ra của thực thể King

- RelicDetails: phục vụ giao diện hiện ra của thực thể Relic

- SuKienDetails: phục vụ giao diện hiện ra của thực thể SuKien

c. Package applicaition.readata.

Package có nhiệm vụ đọc file JSON, phục vụ cho việc hiển thị dữ liệu

- ReadData: class đọc thông tin từ file JSON, phục vụ cho class MainController

**5.4. Thiết kế lưu dữ liệu.**

Các file JSON lưu dữ liệu thu thập được lưu tại package data.

- dynasty.json: lưu dữ liệu triều đại.

- event.json: lưu dữ liệu sự kiện lịch sử.

- figureUpdate.json: lưu dữ liệu nhân vật lịch sử.

- king.json: lưu dữ liệu các vị vua

- relic.json: lưu dữ liệu di tích lịch sử

**6. Các kĩ năng lập trình hướng đối tượng đã dùng**

**6.1. Đóng gói (Encapsulation):**

Các thuộc tính được che dấu bằng kiểu dữ liệu private, cần sử dụng các phương thức public để sử dụng và thay đổi dữ liệu.

**Lợi ích:** Che giấu bảo mật dữ liệu.

**6.2. Kế thừa (Inheritance)**

- Trong phần thu thập dữ liệu, chương trình có các abstract class và interface, các website được tạo thành class riêng và kế thừa các abstract class và interface.

- Trong chương trình sử dụng kế thừa một cấp tại các package của các thực thể và kế thừa nhiều cấp tại package webcrawler.festival

- Trong nhiều class việc kết tập giúp code trở nên ngắn gọn sáng sủa hơn, ví dụ như khi cào dữ liệu về triều đại thì sẽ có 1 class DynastyScrapeFull gọi đến các thành phần, có thành phần cào tên, thành phần cào thời gian,…

**Lợi ích: Kế thừa giúp việc lập trình nhanh gọn.**

**6.3. Đa hình (Polymorphism)**

Tính đa hình trong chương trình thể hiện ở điểm sau:

- Trong các class của các thực thể có thực hiện nạp chồng phương thức (Method Overloading), cho phép khởi tạo object với nhiều kiểu tham số đầu vào.

- Khi các class kế thừa một interface, từng class cần thực hiện ghi đè phương thức (Method Overriding) của phương thức trong interface.

**Lợi ích:** Đa hình cho phép viết cùng một phương thức với nhiều kiểu tham số truyền vào. Ngoài ra, đa hình giúp tùy biến lại các phương thức kế thừa để phù hợp với từng đối tượng, ví dụ mỗi trang web cần thu thập dữ liệu theo cách khác nhau.

**6.4. Trừu tượng (Abstraction)**

- Tính trừu tượng thể hiện thông qua các abstract class và interface, và các lớp sau kế thừa các abstract class và interface.

Lợi ích: Tóm gọn các thuộc tính chung của các thực thể, giúp việc lập trình nhanh gọn.

**6.5. Lập trình tổng quát (Generic)**

Lập trình tổng quát thực hiện tại class ReadData, chỉ cần truyền loại thực thể vào object ReadData là có thể đọc được thông tin của thực thể đó trong file JSON.

**6.6. Ứng dụng nguyên lý SOLID**

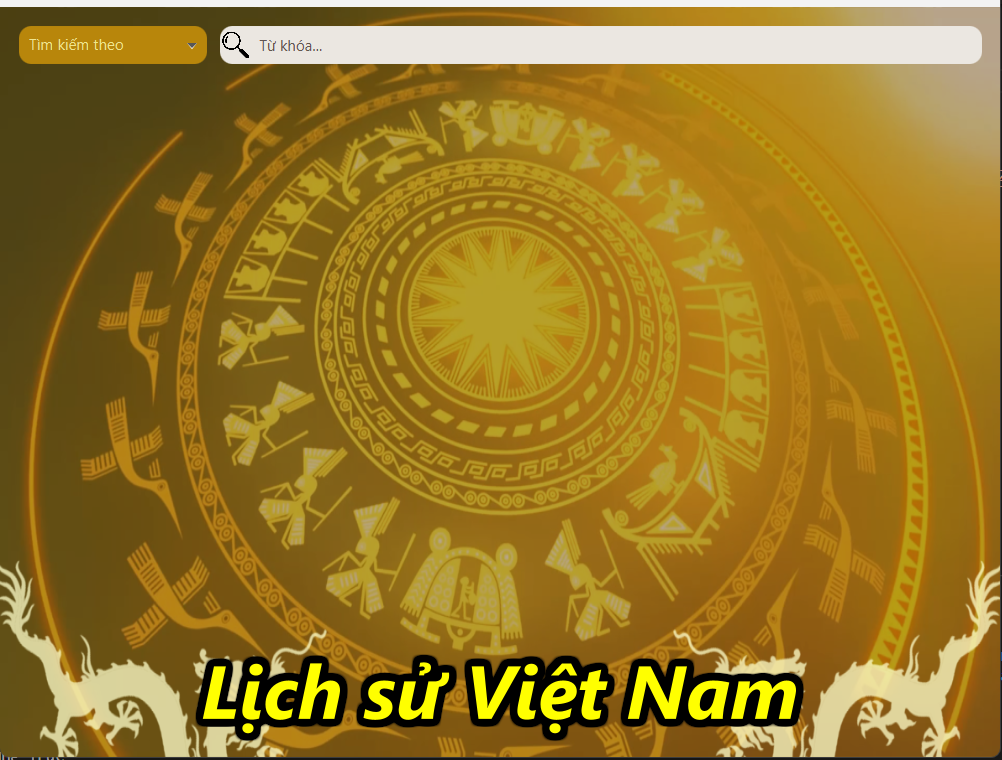
- Việc thiết kế các class chỉ đảm nhiệm 1 nhiệm vụ, ví dụ là RelicScrapeDiTich thì chỉ đảm nhiệm việc cào dữ liệu từ trang ditich.vn, mở rộng hơn nữa là các package cũng chỉ đảm nhận 1 nhiệm vụ chung, ví dụ như package webcrawler chỉ đảm nhiệm việc cào dữ liệu, package readdata chỉ đọc dữ liệu từ file json,…. tuân theo nguyên lí Single Responsibility.

- Do việc tạo ra nhiều Interface và abstract class nên khi muốn cào thêm dữ liệu từ trang mới thì chỉ cần tạo ra class mới và implement các phương thức, không phải chỉnh sửa các code cũ, class cũ tuân theo nguyên lý Open/Closed.

**7. Hướng dẫn sử dụng và hình ảnh demo**

Chỉ cần chạy file main trong class Main trong gói application là có thể chạy app. Sau đó thực hiện các chức năng đã ghi sẵn trên app

Ảnh demo chạy thử chương trình:



Hiển thị và tìm kiếm Vua:



Hiển thị và tìm kiếm nhân vật lịch sử:



Hiển thị và tìm kiếm lễ hội:



Hiển thị và tìm kiếm triều đại:



Hiển thị và tìm kiếm di tích:



Hiển thị và tìm kiếm sự kiện:



Cửa sổ thông tin chi tiết:

