

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



REPORT PROJECT

Object-Oriented Programming

Project title: Viet Nam History

Professor : Trịnh Tuấn Đạt
Subject : IT3100E
Class : 141182

Group: 02

Students: Hoàng Duy Anh - 20214944

Lê Đức Dũng - 20214952

Cao Gia Khánh - 20214962

Nguyễn Phương Thảo - 20214973

1 Work Breakdown Description

Name	Student ID	Job Description	Contribution
Hoàng Duy Anh	20214944	Events Crawler, Figures Crawler	25%
Lê Đức Dũng	20214952	Slide, Report, GUI, Object Design	25%
Cao Gia Khánh	20214962	UML Diagram, GUI, Dynasty Crawler	25%
Nguyễn Phương Thảo	20214973	Report, Festival Crawler, Relic Crawler	25%

2 Data Analysis

2.1 Data counts

Dynasty: 33

Event: 302

Festival: 249

Figure: 2612

King: 328

Relic: 5714

Total Data : 9238

2.2 Data Types

- Character and King will have the same 3 attributes: ten, namSinh, namMat
- Character (Nhân vật): queQuan, danToc, namNhapNgu, ghiChu
- King (Vua): articleLink, mieuHieu, thuyHieu, nienHieu, tenHuy, theThu, namTriVi
- Relic (Di tích): name, location, type, rank, desc, characters, kings, dynasties
- Festival (Lễ hội): tenLeHoi, thoiGian, diaDiem, character, noiDung
- Event (Sự kiện): ten, thoiGian, diaDiem, nienDai, nhanVatLienQuan
- Dynasty (Triều đại): startYear, endYear, name, kings, capital, founder

2.3 Liên kết dữ liệu

Total entities link:

- King - Character (Vua - Nhân vật)
- Relic - King (Di tích - Vua)
- Relic - Character (Di tích - Nhân vật)
- Relic - Dynasty (Di tích - Triều Đại)
- Festival - Character (Lễ hội - Nhân vật)
- Event - Dynasty (Sự kiện - Triều đại)
- Event - Character (Sự kiện - Nhân vật)
- Dynasty - King (Triều đại - Vua)

3 UML Diagram

3.1 Package Dependency Diagram

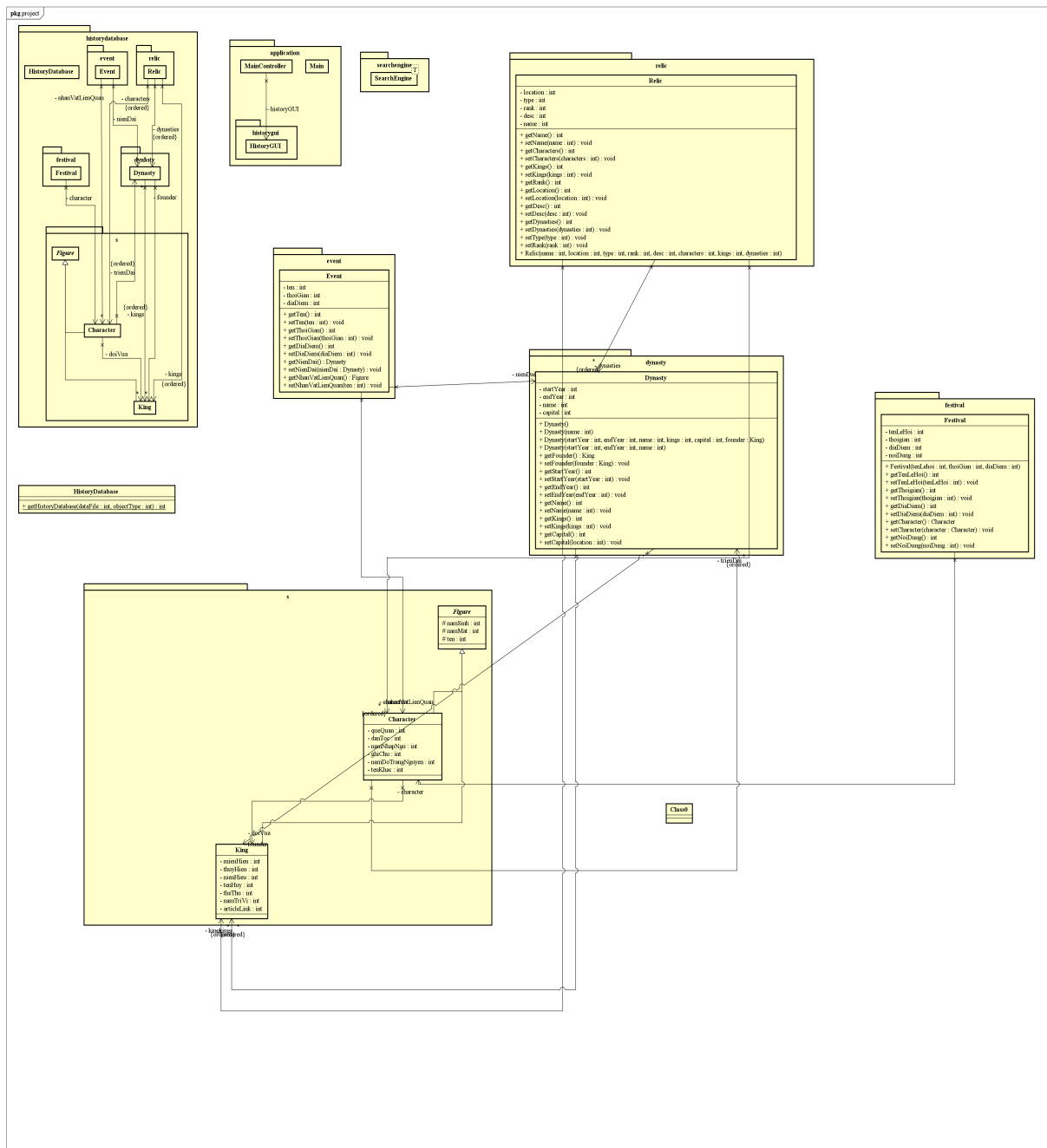


Figure 1: Package Dependency Diagram

3.2 Usecase Diagram

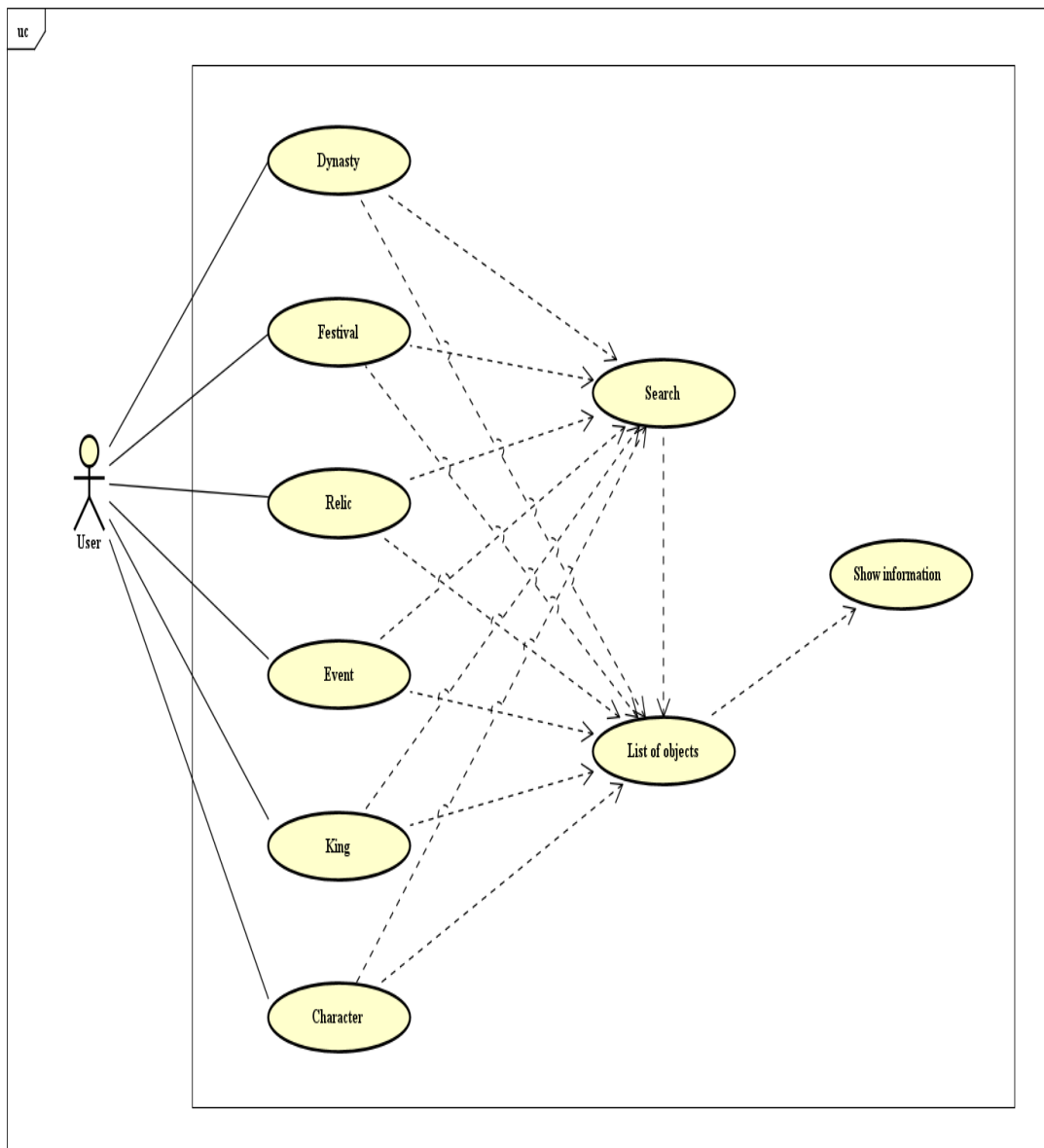


Figure 2: Usecase Diagram

3.3 Database Diagram

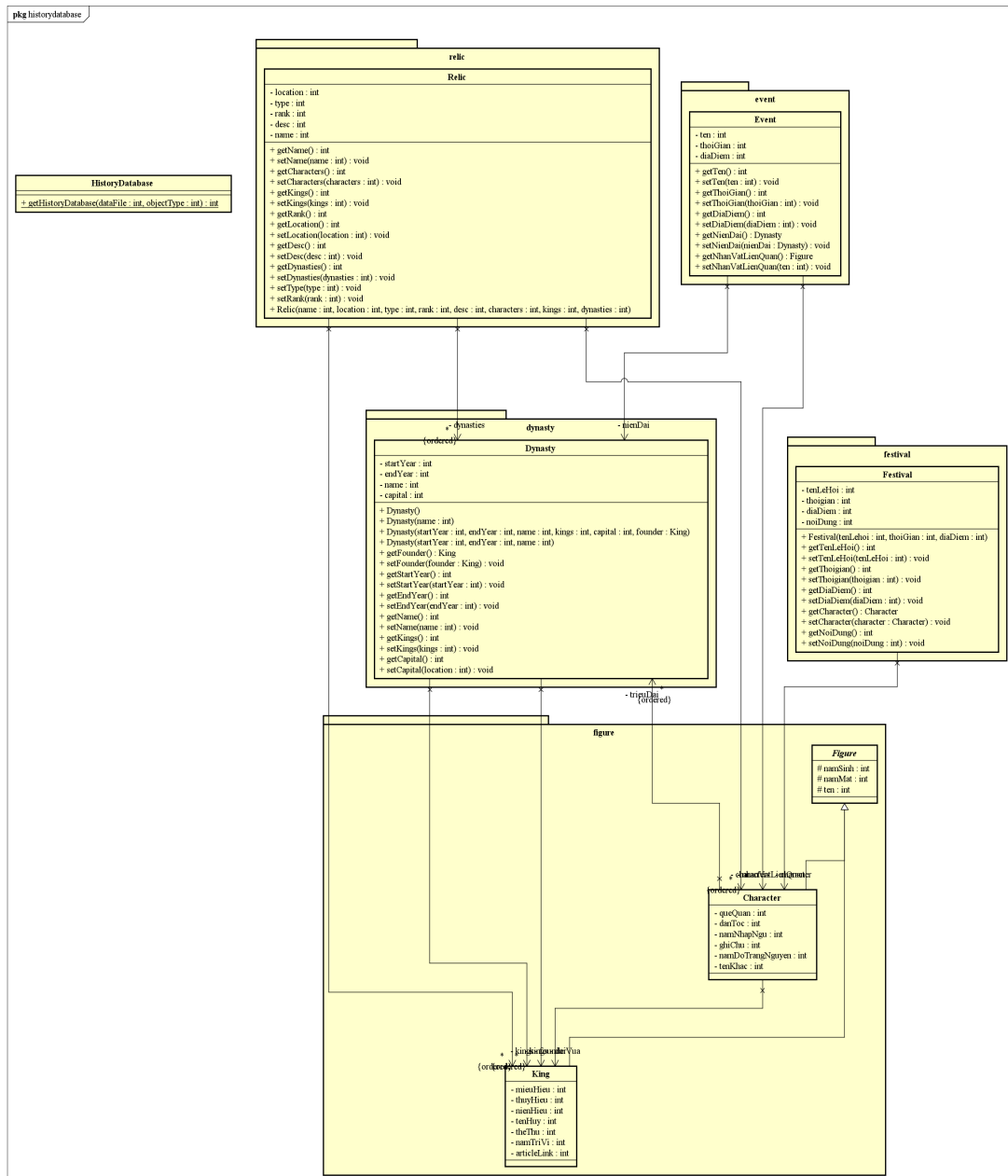


Figure 3: Design Database

4 Design Analysis

4.1 Package Dependency Diagram Analysis

- Package historydatabase có chứa 5 package nhỏ tượng trưng cho 5 objects: event, relic, dynasty, festival và figure cộng thêm với lớp Historydatabase có chứa các data
- Package application có chứa historyGUI và lớp MainController và lớp Main để người dùng có thể tương tác
- Package searchengine có chứa lớp SearchEngine phục vụ cho việc tìm kiếm thông tin 1 cách dễ dàng hơn

4.2 UseCase Diagram Analysis

- Người dùng tương tác với 6 objects Dynasty, Festival, Relic, Event, King, Character(King và Character có thể gộp chung vào thành Figure) thông qua Search hoặc có thể tìm kiếm trực tiếp bằng danh sách các thông tin được liệt kê

4.3 Class Diagram

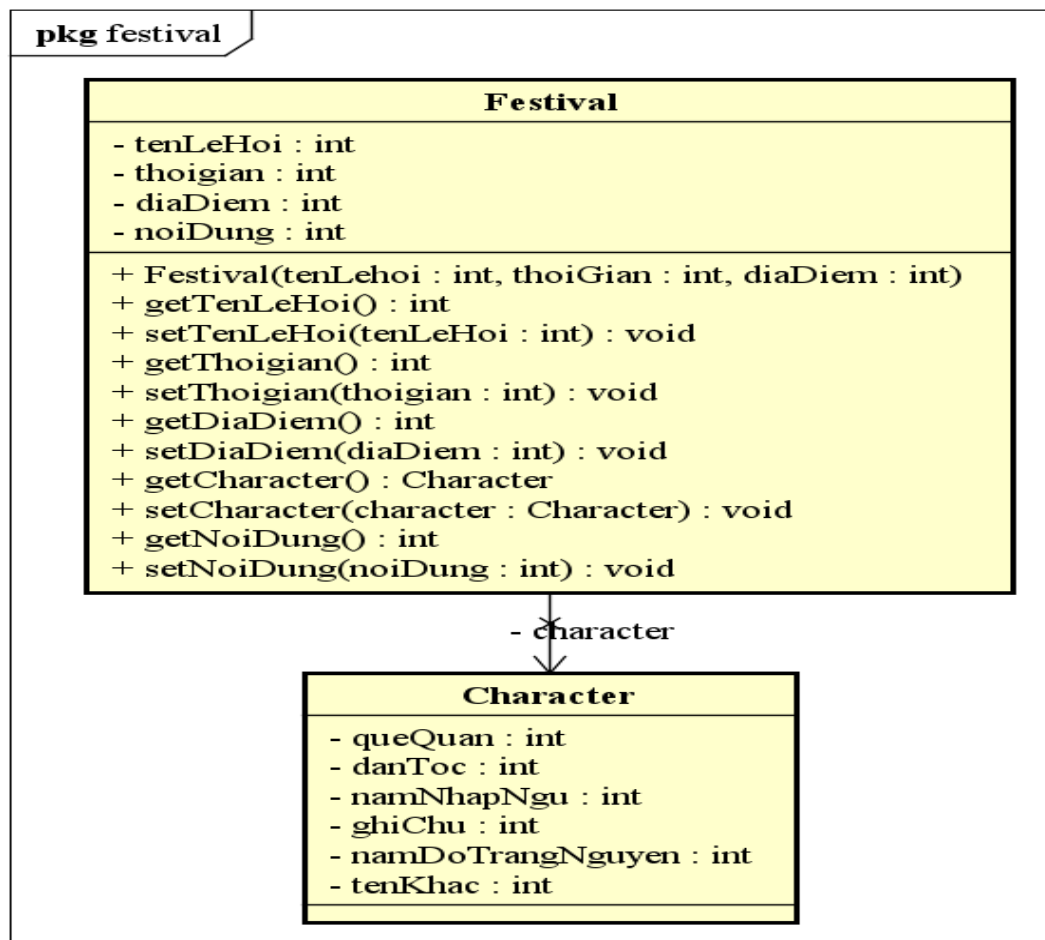


Figure 4: Link Festival - Character

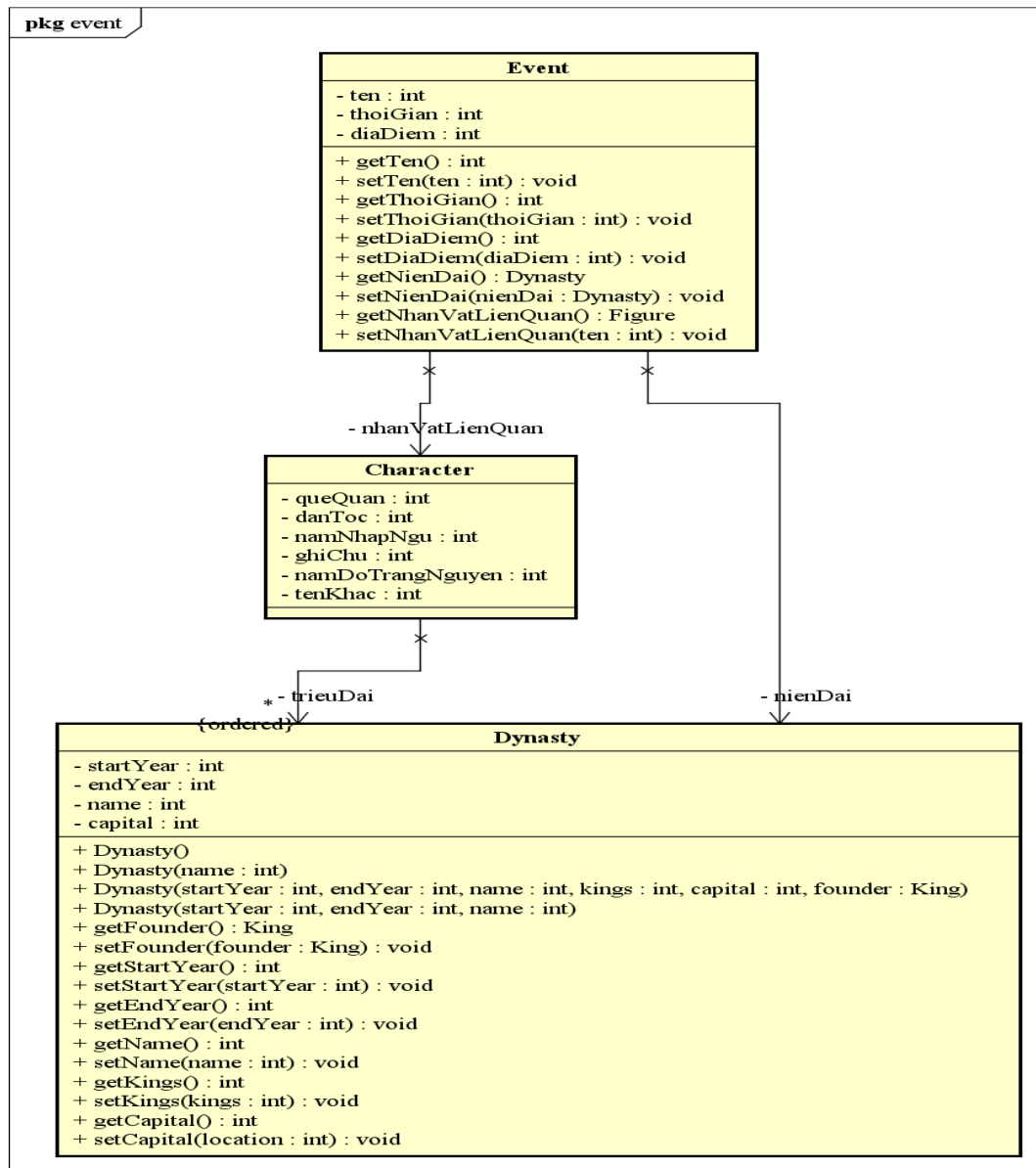


Figure 5: Link Event - Character - Dynasty

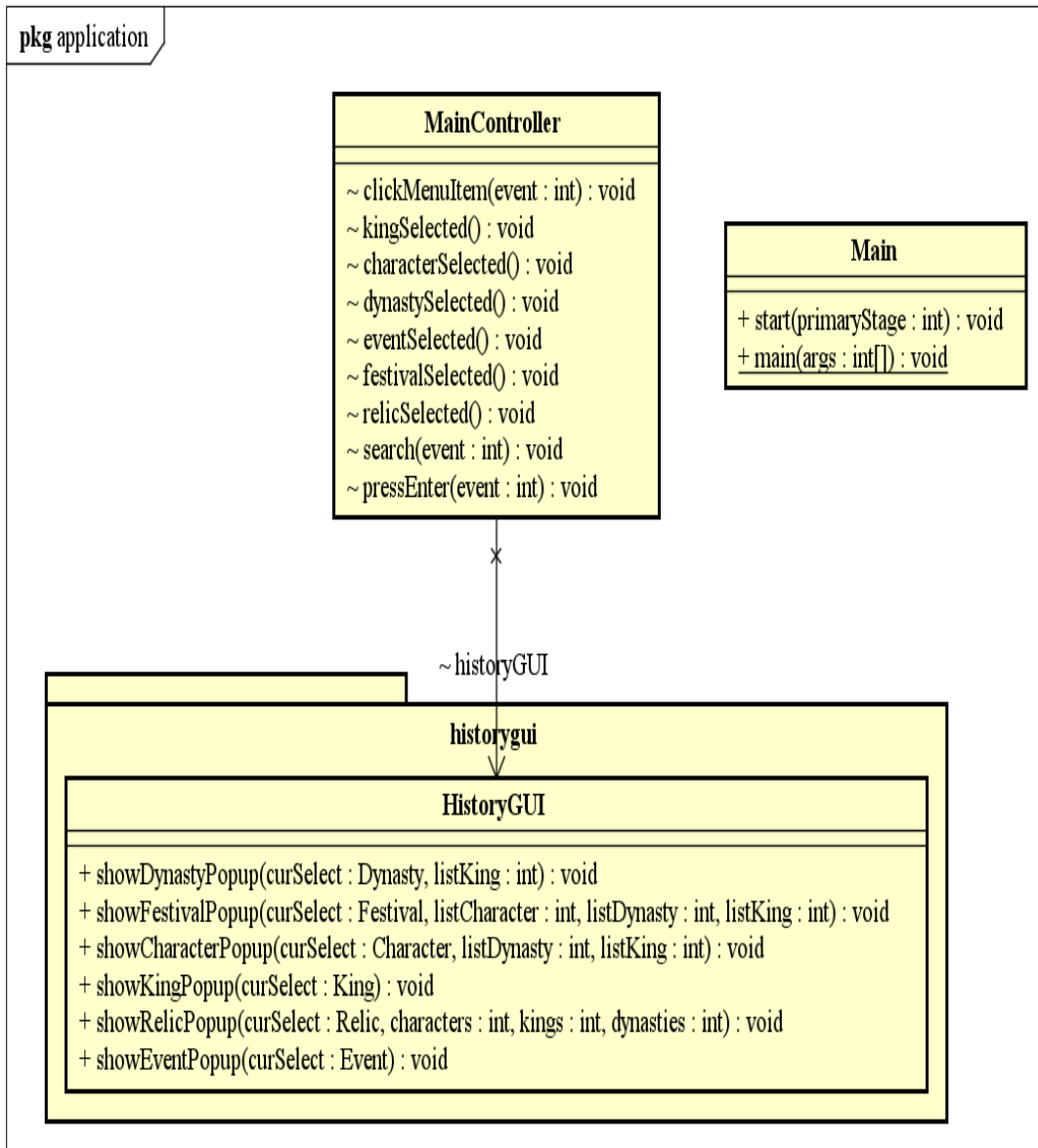


Figure 6: Link Event - Character - Dynasty

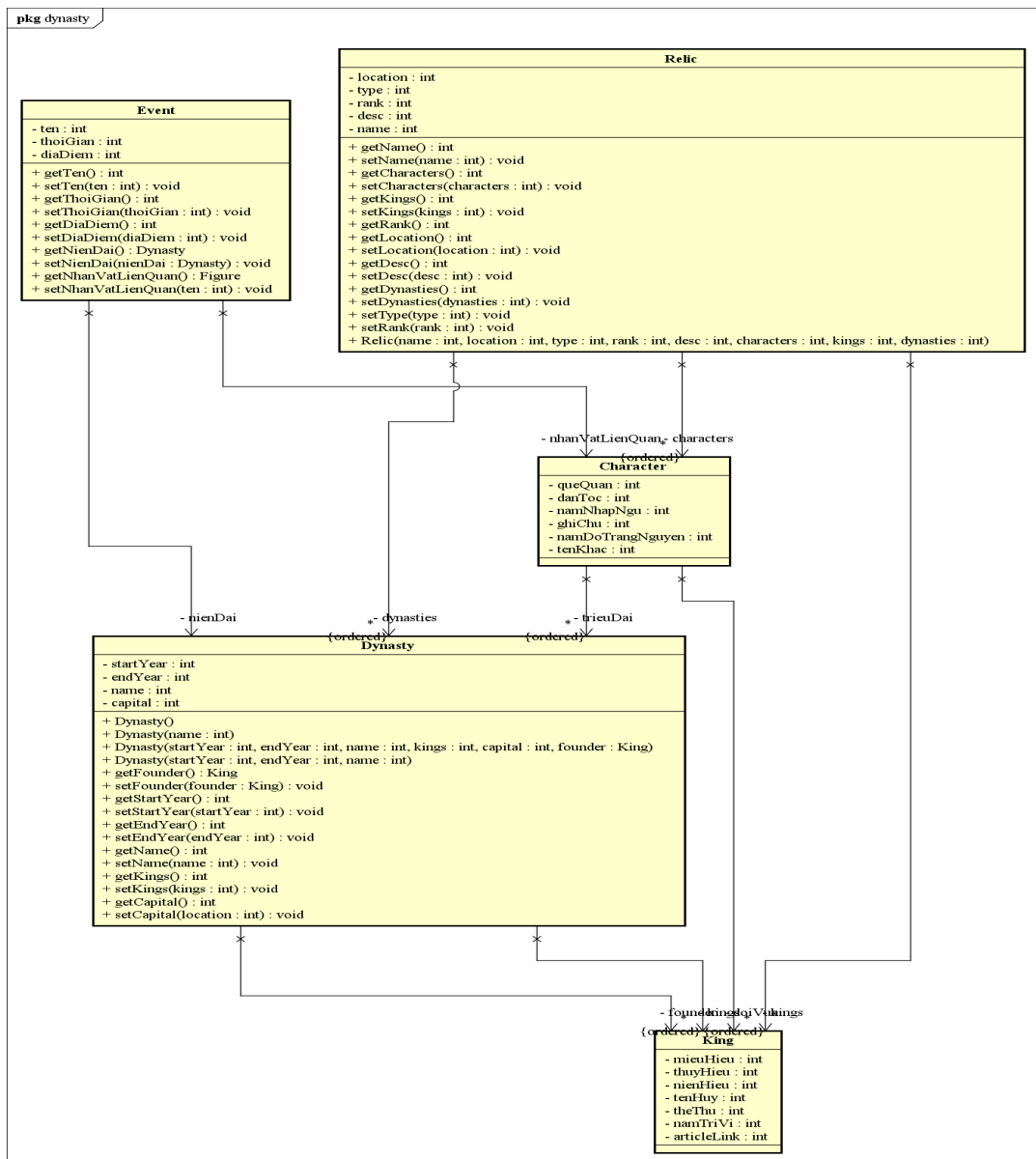


Figure 7: Link Event - Relic - Dynasty - Character - King

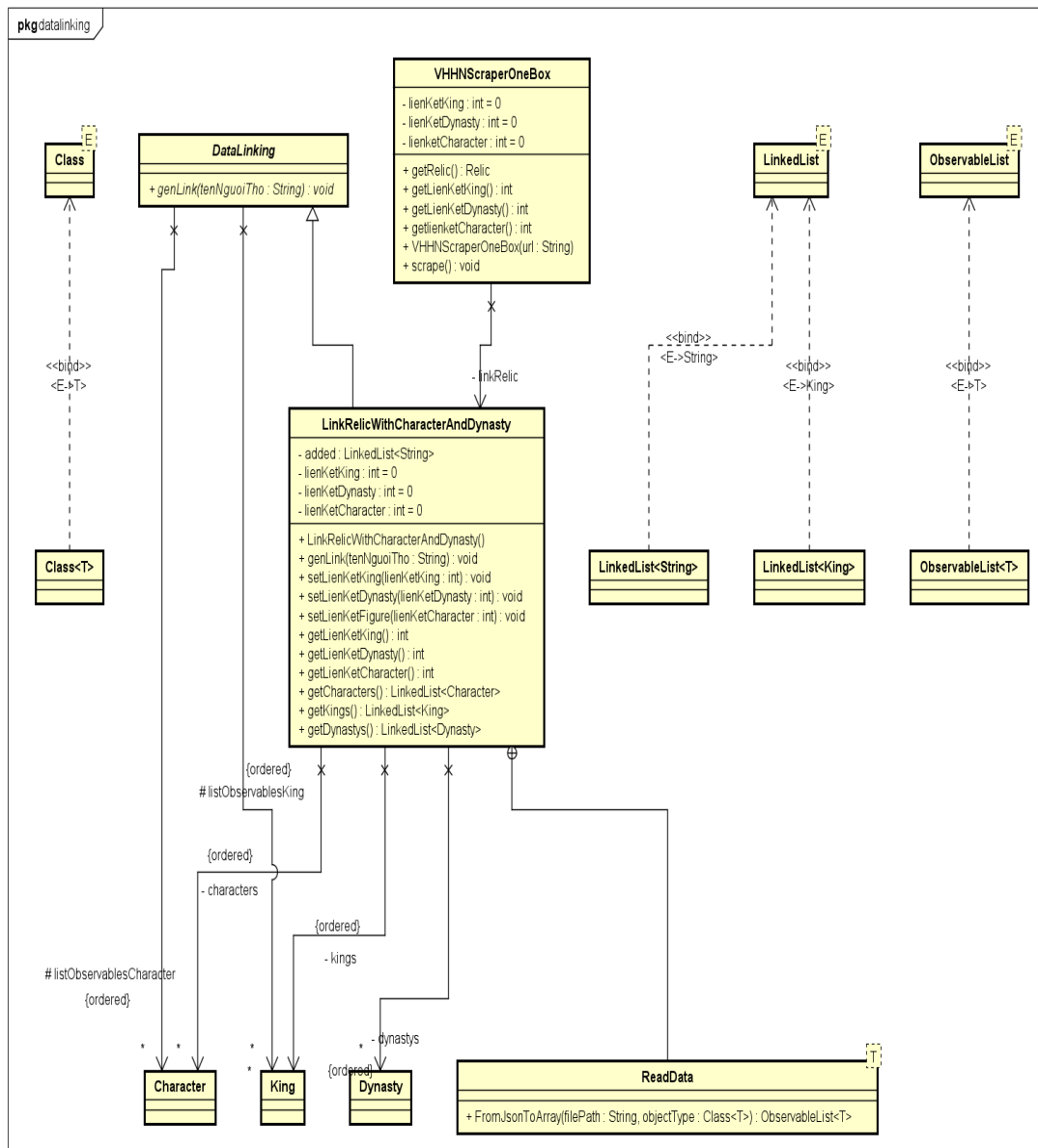


Figure 8: Crawler

5 OOP Techniques

5.1 Abstraction

Tính trừu tượng sử dụng lập trình Interface và Abstract class, nêu cụ thể ở lớp crawler, lưu file crawler vào json và lớp figure

5.2 Encapsulation

Tính đóng gói: mỗi lớp đều có thuộc tính private, chỉ được truy cập thông qua phương thức getter() và sửa đổi thông qua phương thức setter(). Việc đóng gói như vậy tránh những truy cập bất hợp pháp hoặc sẽ tiện lợi hơn khi sửa đổi code. Ví dụ như việc nếu để thuộc tính Name là public, khi muốn sử dụng ta phải đi tìm tất cả những chỗ có thuộc tính này và sửa lại, trong khi đó với phương thức setter(), ta chỉ cần truyền vào Name mình muốn sửa.

5.3 Inheritance

Tính kế thừa: được thể hiện rõ nhất trong việc xây dựng class cho các thực thể Character, King... Vì chúng đều có chung thuộc tính đó là tên (name), năm sinh (namSinh), năm mất (namMat) . Tính kế thừa cũng được thể hiện qua các class Crawler như CharacterCrawler, DynastyCrawler, WikipediaScraper... kế thừa từ class BaseWebCrawler với các thuộc tính url, doc do tất cả các class Crawler đều cần có url để kết nối và có Document để làm việc với file json. Các class Crawler nhỏ hơn của từng trang web cũng kế thừa từ class cha của chúng, ví dụ với Triều đại: DynastyCrawler là lớp cha có các phương thức connect() và start(), các lớp con như NguoiKeSuDynastyCrawler hay WikiDynastyCrawler sẽ kế thừa các phương thức đó và thực thi vì chúng đều cần phải kết nối và ghi file json.

5.4 Polymorphism

Tính đa hình: Các hàm crawler được override và hàm lưu file json

6 Libraries

- Maven
- Java Coding Convention
- Astah
- Jsoup
- Javafx
- Gson
- Git, Github