1 Bối cảnh mục đích:

* Sự tách rời giữa document và source code đã trở thành một vấn đề ở dự án cải tạo system trước đây
  + Khi cải tạo dự án thì cần đọc source code
    - => tốn thời gian
  + Gặp vấn đề lớn khi đọc source code ở những dự án mới phát triển và những dự án MainTain
    - Không được được code
    - Đọc được thì tốn rất nhiều thời gian để hiểu và để cải tạo và phát triển thêm
* => Tool AP(**Hiển thị hóa cấu trúc Application**) ra đời với mục đích:
  + Phân tích source code rồi hiển thị hóa cấu trúc
    - có thể giải quyết vấn đề không đọc được source code.
    - Giải quyết việc đọc nhưng tốn rất nhiều thời gian.
* C: \ ApStructureVisualizedTool \ Projects \ test \ pgm \ test \ src \ test \ Demo.java: 7: Lỗi cú pháp trên mã thông báo "sadasd", hãy xóa mã thông báo này
* 2021-04-13 10: 19: 11.262 LỖI Quá trình phân tích sẽ bị gián đoạn vì đã xảy ra lỗi biên dịch.
* 2021-04-13 10: 19: 11.262 LỖI org.eclipse.core.runtime.OperationCanceledException
* **Lỗi khi code không thể compile**

**Tác dụng:**

* Yêu cầu :Điều tra phạm vi ảnh hưởng khi sửa đổi một phương thức nhất định:
* Tác dụng:
  + Kiểm thử và thửu nghiệm phương thức gọi có thể bị ảnh hưởng khi sửa đổi một phương thức cụ thể
  + xác định tất cả các tính năng và phương pháp bạn có thể cần
  + kiểm tra tính gọi của các cột và trường DB
* yêu cầu :tạo tài liều bảo trì và giám chi phí bảo trì
  + Cách sử dụng : Duy trì các tài liệu bảo trì theo mã nguồn
* Tác dụng của công cụ: Các tài liệu bảo trì(Sơ đồ CRUD, nội dung xử lý phương pháp, sơ đồ chức năng) có thể được tạo từ mã nguồn mới nhất.
* >thời gian bảo trì được giảm xuống

Flow Chương trình:

**Phân tích phụ thuộc lớp:**

**/\***

Kết quả phụ thuộc lớp : Lớp nào, (lớp, trường,phương thức), lớp nào và những gì được tham chiếu ở định dạng CSV

Phân tích cú pháp thư mục hoặc thư mục chứa cá tệp java nguồn,Chỉ định một thư mục gốc hoặc mỗi thu mục chứa java (@ Nếu có nhiều mục tiêu phân cách chúng mời dấu cách )

Classpath& lt ; path&gt: class path bắt buộc phải biên dịch tệp nguồn sẽ được phân tích cú pháp(tìm kiếm jar theo đường dẫn được chỉ định)

Để tìm kiếm các jar trong thư mục được chỉ định, chỉ cần mọt đường dẫn đển thư mục bao gồm tất cả các jar

Encoding&lt en coding &gt:Mã hóa ký tự của tệp nguồn sẽ được phân tích.

includes &lt;package&gt :các gói bận muốn đứa vào mục tiêu phân tích trong số các gói được chỉ định

mặc định các gọi java và javax không bị loại trừ

Phiên bản của tệp nguồn sẽ được phân tich cú pháp(1.3,1.4etc) mặc định là 1.8

Đường dẫn tệp nguồn bổ sung cần thiết để biên dịch tệp được phân tích cú pháp.

Vì nó biên dịch từ tệp một, dường dãn tệp nguồn được thêm vào đường dẫn thười gian chạy java

Nếu có nhiều hơn 1 hãy kết nối chúng bởi dấu ;(@ không phải đường dẫn chính xác’ đường dẫn không phải là thư mục nguồn’)

“Thư mục mục tiêu phân tích hoặc nguồn ” được chỉ định bởi đối số sẽ tự động được đưa vaofgias trih tùy chọn này.

iewpath & lt; viewpath & gt; đường dẫn của thư mục trữ tệp JSP và tệp HTML để phân tích thông tin màn hình

Nếu một tệp được chỉ định và các thư mục con của nó sẽ được tìm kiếm, nếu có nhiều tệp hãy phân tách chúng bằng “;”.

Nếu không được chỉ định, thông tin màn hình sẽ không được phân tích.

Tùy thuộc vào giá trị mvctype , cần chỉ định tệp cấu hình hoặc đường dẫn của thư mục chứa tệp cấu hình

--mvctype & lt; mvctype & gt: chỉ định loại khuôn khổ lớp trình bày

Chuỗi ký tự có thể được chỉ định là một trong “servlet, structs 1, struts 1-maia, springmvc”.nếu không được khẳng định thì thông tin ánh xạ yêu cầu sẽ không được đọc.

[khi servlet được chỉ định]

Phân tích cú pháp các chú thích Web.xml và @WebServlet dựa treeb đặc tả JavaEE.

Khi đọc webxml hãy chỉ định đường dẫn thư mục chứa web.xml

[Khi struts1 được chỉ định]

\* Phân tích dựa trên tiền đề của tiêu chuẩn Struts1 (lệnh gọi Hành động # thực thi).

\* Đối với đường dẫn xem, cũng chỉ định đường dẫn thư mục chứa struts-config.xml.

\* Nếu không tìm thấy struts-config.xml, phân tích ánh xạ yêu cầu sẽ không được thực hiện.

\* [Khi struts1-maia được chỉ định]

\* Phân tích dựa trên sự tồn tại của DelegateActionProxy, là tiêu chuẩn để tích hợp Struts-Spring lên đến Maia2.6.

\* Ngoài "khi struts1 được chỉ định", hãy chỉ định đường dẫn thư mục chứa tệp định nghĩa Bean (thường là action.xml) của lớp Hành động.

\* [Khi springmvc được chỉ định]

\* Phân tích cú pháp chú thích @RequestMapping dựa trên thông số kỹ thuật Spring MVC.

\*

\* Trong cả hai trường hợp, các tìm kiếm đường xem bao gồm các thư mục con trong thư mục được chỉ định, vì vậy nếu có nhiều tệp cấu hình, hãy chỉ định thư mục gốc chứa tất cả các tệp.

**\*/**

System.*setProperty*("log4j2.disable.jmx", "true");

* Nếu log4j2.disable.jmx =true :// Các đối tượng cấu hình Log4j sẽ không có côn cụ với Mbeans và không thể được giám sát và quản lý từ xa.

Quá trình tìm kiếm tệp jar đưa vào classpath

Quá trình hoàn tất

***logger***.info("クラスパスに含めるjarファイルを検索します。");

long befJarCollect = System.*currentTimeMillis*();

String[] classpath = commandLine.hasOption(***OPTION\_NAME\_CLASSPATH***)

? commandLine.getOptionValue(***OPTION\_NAME\_CLASSPATH***).split(File.***pathSeparator***)

: new String[0];

List<String> jarFileList = new ArrayList<>();

// クラスパスに指定された値別に、フォルダ内のjarファイルを検索してListにつめる

for (String path : classpath) {

jarFileList.addAll(*collectFile*(path, ***JAR\_EXT***));

}

classpath = jarFileList.toArray(new String[0]);

long aftJarCollect = System.*currentTimeMillis*();

thông báo thời gian hoàn thành việc tìm kiếm File Jar.

Khai bóa và láy source path từ commanLine

String[] sourcepath = commandLine.hasOption(***OPTION\_NAME\_SOURCEPATH***)

? commandLine.getOptionValue(***OPTION\_NAME\_SOURCEPATH***).split(File.***pathSeparator***)

: new String[0];

Lấy endcoding và version nguồn

Trong CommandLine

Các gói đưa vào được phân cách nhau bở dấu “ ,”

Các gói được loài trừ phân tách bằng dấu”,”

Tìm kiếm tệp java đích

// 解析対象Javaソースコードを取得

***logger***.info("対象のjavaファイルを検索します。");

List<String> javaFileList = new ArrayList<>();

long befCollect = System.currentTimeMillis();

Thên mục tiêu phân tích đường dẫn nguồn

sourcepath = sourcepathList.toArray(new String[0]);

tìm kiếm tệp hva trong thư mục theo đối số và điền nó vào danh sách

javaFileList.addAll(*collectFile*(path, ***JAVA\_EXT***));

long aftCollect = System.*currentTimeMillis*();

***logger***.info("対象のjavaファイルを検索が完了しました。(" + (aftCollect - befCollect) / Constants.***MILLISEC\_FOR\_SEC*** + "秒)");

***logger***.info("");

**tạo bộ kiểm tra đặt tùy chọn:**

gọi đển Analyzer.execute

**Thực hiện Phân tích**

* Thực hiện phân tích phụ thuộc
* Kết quả phân tích được lưu trữ trong tệp DB(AnalysisData.DB)
* Trong quá trình phân tích nếu có một cú pháp nào đó mà công cụ không hỗ trợ , hãy xuất một bản ghi cảnh báo và tiếp tục xử lý
* Nếu một cú pháp không mong muốn, hãy ném một ngoại lệ và kết thúc quá trình
* @ trả về 0 nếu quá trình phân tích kết thúc bình thường
* @ trả về 1 nếu quá trình phân tihcs kết thúc không bình thường.

Gọi đến Phương thức Int Execute

* getConnection: jdbc:sqlite:C:\PlatformService\Source\AP\20200721\work\00\_database\AnalysisData.db
* truncateTable():
* 解析データをクリアします(Xóa dữ liệu phân tích)
* Dữ liệu phân tích đã được xóa.
* Bắt đầu phân tích
* Tổng số tệp: javaFileList.size()