**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ TT & TRUYỀN THÔNG**



**BÁO CÁO KẾT QUẢ**

**MÔN BẢO TRÌ PHẦN MỀM**

**ĐỀ TÀI**

**Tìm hiểu công cụ đảo ngược dùng trong bảo trì phần mềm**

**NHÓM 1**

Lương Đức Duy

Ngô Minh Phương

Hồ Hữu Nhân

Trần Thanh Điền

**Cần Thơ 8/2014**

Mục lục

[Kế hoạch thực hiện iii](#_Toc397006332)

[1 Giới thiệu iv](#_Toc397006333)

[2 Lịch sử ra đời iv](#_Toc397006334)

[3 Các đặc điểm chính iv](#_Toc397006335)

[3.1 Chuyển code sang sơ đồ UML iv](#_Toc397006336)

[3.1.1 Class diagram iv](#_Toc397006337)

[3.1.2 Sequnce diagram iv](#_Toc397006338)

[3.2 Chuyển code sang Flowchart iv](#_Toc397006339)

# Kế hoạch thực hiện

|  |  |
| --- | --- |
| **Thời gian** | **Nội dung** |
| Tuần 1 – Tuần 4 | Các thành viên về tìm hiểu 1 công cụ đảo ngược.  Họp nhóm các thành viên báo cáo và phản biện. |
| Tuần 5 | Thống nhất chọn công cụ theo hướng IDE tích hợp plugin.  Tiếp tục nghiên cứu và viết báo cáo dựa trên công cụ.  Lương Đức Duy, Ngô Minh Phương:   * Nghiên cứu plugin eUML 2 dùng để chuyển code về Sequence Diagram và Class diagram. * Mức độ hoàn thành: 100%.   Trần Thanh Điền, Hồ Hữu Nhân:   * Nghiên cứu plugin CodeRocket dùng để chuyển code về Flowchart. * Mức độ hoàn thành: 100%. |

# Giới thiệu

Công cụ đảo ngược có vai trò chiết xuất thông tin về kiến thức hoặc thiết kế của phần mềm. Công cụ đảo ngược giúp nhóm bảo trì hiểu rõ được phần mềm mình cần bảo trì theo nhiều mục tiêu khác nhau. Công cụ đảo ngược có thể trích xuất mô hình thiết kế UML thiết kế cho phần mềm đang cần bảo trì, có thể chuyển các đoạn mã thành sơ đồ giải thuật, hiểu rõ hơn cơ sở dữ liệu, v.v Mục tiêu cuối cùng của công cụ đảo ngược hiểu rõ, hiểu nhanh được phần mềm mình cần bảo trì. Đặc biệt trong các chương trình, hệ thống phức tạp, hoặc trong trường hợp tài liệu của sản phẩm đã lỗi thời hoặc bị mất.

Nhóm chọn nghiên cứu theo hướng IDE, cụ thể là Eclipse tích hợp các plugin để thực hiện quá trình đảo ngược.

Lý do:

* Eclipse miễn phí.
* Eclipse là IDE lập trình mã nguồn mở rất phổ biến hiện nay.
* Eclispe bản thân là một IDE nên lập trình viên có thể vừa đảo ngược phần mềm, vừa có thể tiếp tục bảo trì phần mềm trực tiếp tren Eclipse.
* Do tính mở nên có rất nhiều công ty đã tùy chỉnh Eclipse theo những công cụ nhất định, tuy nhiên do phát triển nên từ Eclipse các công cụ khác đều thừa hưởng được kho plugin mà Eclipse hỗ trợ.
* Do Eclipse bản thân cũng là công cụ mã nguồn mở nên có khả năng tùy chỉnh rất cao, chúng ta có thể tự viết plugin cho Eclipse.
* Eclipse hỗ trợ tốt Java và các Framework hỗ trợ Java.
* Eclipse được phát triển bởi tổ chức Eclipse và được sự hỗ trợ từ cộng đồng mạng rất lớn.

Bài báo cáo sẽ nói về lịch sử của Eclipse và giới thiệu plugin tích hợp và Eclipse giúp Eclipse trở thành công cụ đảo ngược mạnh mẽ.

# Lịch sử ra đời

Eclipse ban đầu được phát triển như là dự án Smart Canada, phục vụ cho dòng máy tính IBM VisualAge. Trong tháng mười một năm 2001, một tập đoàn được thành lập, có hội đồng quản trị quản lý để tiếp tục phát triển Eclipse như là một phần mềm mã nguồn mở. Người ta ước tính rằng IBM đã đầu tư gần đến $40 triệu để phát triển Eclipse vào thời điểm đó. Các thành viên ban đầu trong hội đồng gồm có: Borland, IBM, Merant, QNX Software Systems, Rational Software, Red Hat, SuSE, TogetherSoft và WebGain. Vào cuối năm 2003, số thành viên trong hội đồng đã tăng lên hơn 80. Tháng một 2004, tổ chức Eclipse được thành lập.

Lịch sử phát hành:

Từ năm 2006, tổ chức Eclipse đã phối hợp phát hành các phiên bản vào tháng 6 hằng năm. Cụ thể:

|  |  |  |  |
| --- | --- | --- | --- |
| Version Name | Date | Platform Verion | Projects |
| Callisto | 30/06/2006 | 3.2 | Callisto |
| Europa | 29/06/2007 | 3.3 | Europa |
| Ganymede | 25/06/2008 | 3.4 | Ganymede |
| Galileo | 24/06/2009 | 3.5 | Galileo |
| Helios | 23/06/2010 | 3.6 | Helios |
| Indigo | 22/06/2011 | 3.7 | Indigo |
| Juno | 27/06/2012 | 3.8 và 4.2 | Juno |
| Kepler | 26/06/2013 | 4.3 | Kepler |
| Luna | 25/06/2014 | 4.4 | Luna |
| Mars | 24/06/2015 (dự kiến) | 4.5 | Mars |

# Các đặc điểm chính

## Chuyển code sang sơ đồ UML

### Class diagram

Có nhiều tùy chọn để chuyển code sang sơ đồ class diagram:

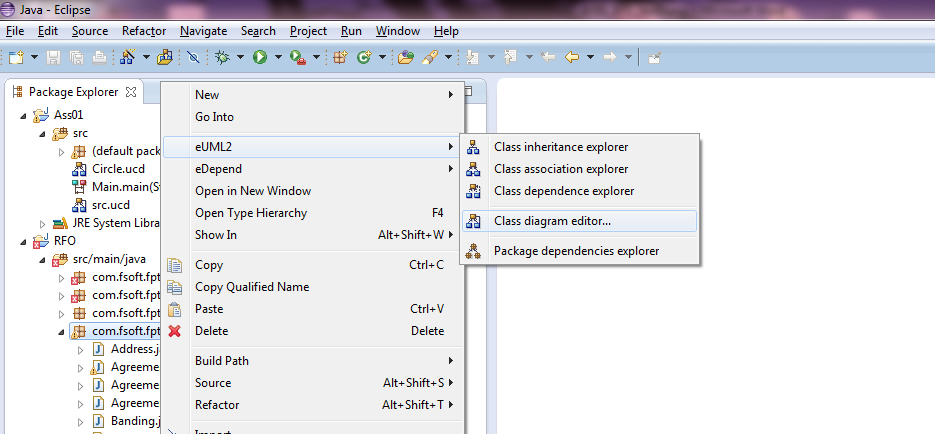
**Cách 1:** Chuyển nguyên một package sang sơ đồ class diagram

1. Chọn package
2. Edit 🡪 eUML2.
3. Lúc này sẽ có xuất hiện nhiều tùy chọn:

* Class inheritance explorer
* Class association explorer
* Class dependence explorer
* Class diagream editor…

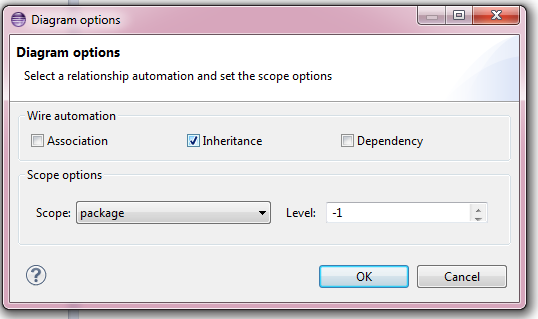
Tùy vào chức năng mà bạn chọn thì công cụ sẽ chuyển tất cả các class trong packge đó sang sơ đồ class diagram.

**Ví dụ 1:**

****

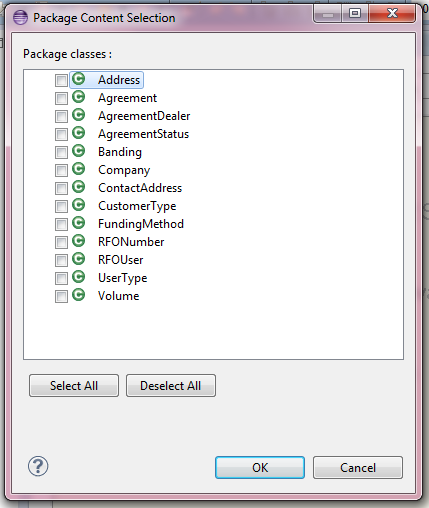
Lúc này bạn sẽ chọn một trong những chức năng mà công cụ cung cấp. Ví dụ này chúng ta sẽ chọn Class diagram editor…

Sau đó sẽ xuất hiện khung “Diagram options”.



Ở trong phần Wire automation chọn nhứng tùy chọn mà bạn muốn. Sau đó bấm OK.

Tiếp theo sẽ xuất hiện “Package Content Selection”. Hãy chọn những class mà chúng ta muốn chuyển. Sau đó tiếp tục bấm OK.



Lúc này công cụ sẽ tự chuyển các class mà ta chọn thành class diagram và các quan hệ giữa chúng (nếu có).

Đây là kết quả mà ta mong muốn.

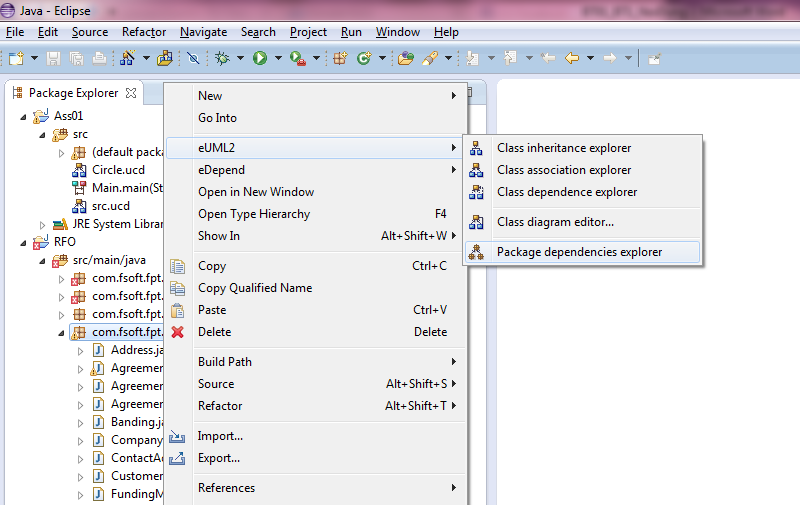


**Cách 2:** Chuyển code sang sờ đồ Class diagram thông qua Package diagram.

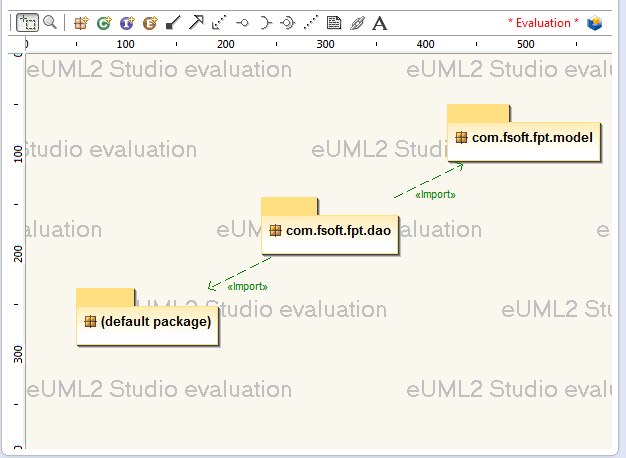
1. Chọn package mà bạn muốn.
2. Edit 🡪 eUML2.
3. Chọn chức năng package dependencies explorer.
4. Lúc này sẽ xuất hiện Package diagram và quan hệ giữa các package (nếu có).
5. Rồi bạn chỉ cần nhấn đúp vào package mà bạn muốn chuyển sang Class diagram và thực hiện theo hướng dẫn của cách 1.

**Ví dụ 2:**

Tương tự như ví dụ 1, tuy nhiên chúng ta sẽ chọn chức năng **Package dependencis explorer**.



Màn hình sẽ hiện ra Package diagram như sau:



Lúc này bạn sẽ nhấn đúp vào package mà bạn muốn và thực hiện tương tự như ví dụ 1 thì ta sẽ được kết quả như sau:



**Cách 3:** Chuyển trực tiếp

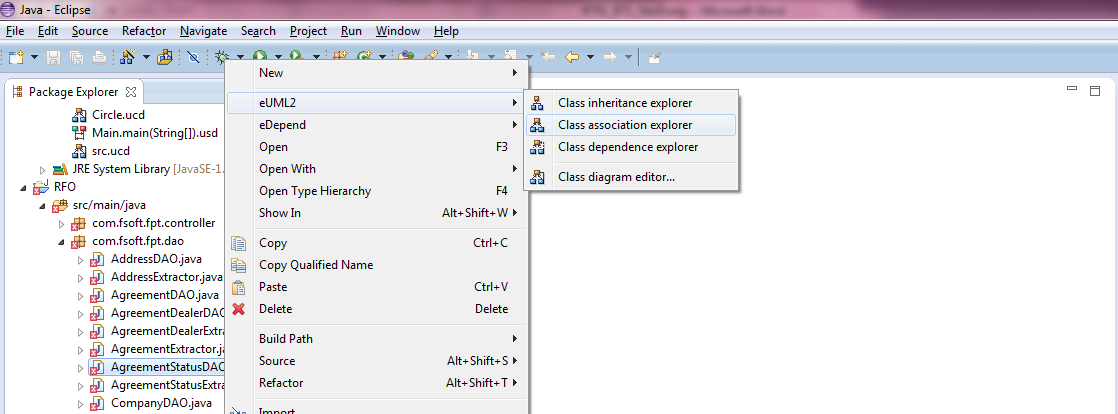
1. Chọn class mà bạn muốn chuyển sang class diagram.
2. Edit 🡪 eUML2.
3. Chọn một trong những tùy chọn sau:

* Class inheritance explorer
* Class association explorer
* Class dependence explorer
* Class diagream editor…

1. Xem kết quả.

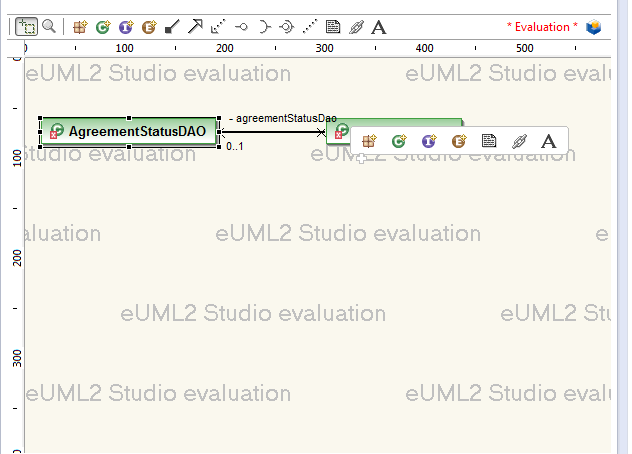
**Ví dụ:**

Chọn một class và nhấn chọn Edit 🡪 eUML2



Ở ví dụ này tôi sẽ chọn chức năng **Class association explorer**.

Và đây là kết quả.



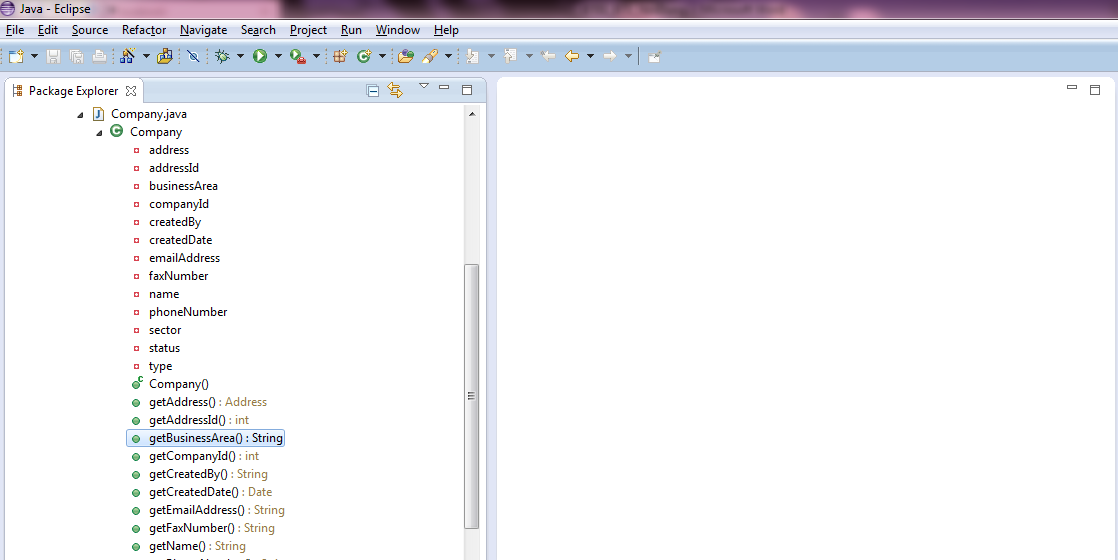
### Sequnce diagram

Các bước chuyển code sang sequence diagram:

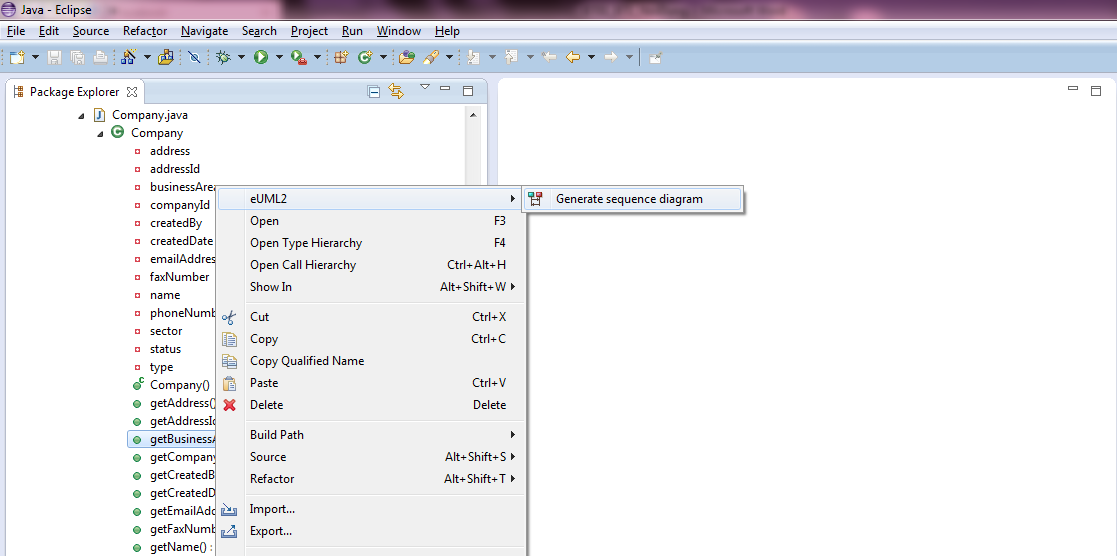
1. Chọn class muốn chuyển.
2. Bấm vào biểu tượng mở rộng 
3. Chọn method (phương thức, hàm) muốn chuyển.
4. Vào Edit 🡪 eUMl2.
5. Chọn Generate sequence diagram.
6. Nêu có hiện **Message filtering dialog** thì chọn Select all. Rồi bấm OK

**Ví dụ:**

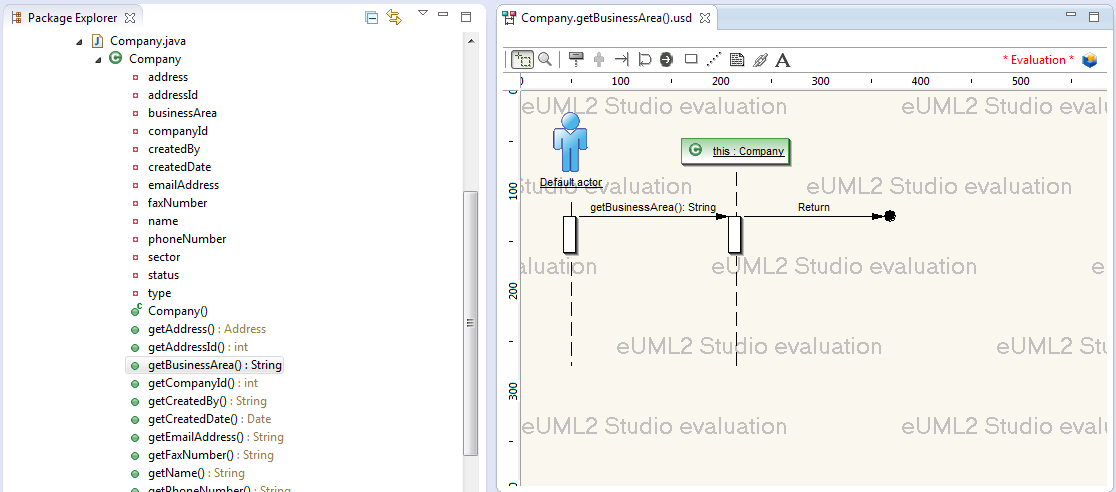
Đầu tiên vào chọn class và method mà mình muốn chuyển



Tiếp theo vào Edit 🡪 chọn eUML2 🡪 Generate sequence diagram.



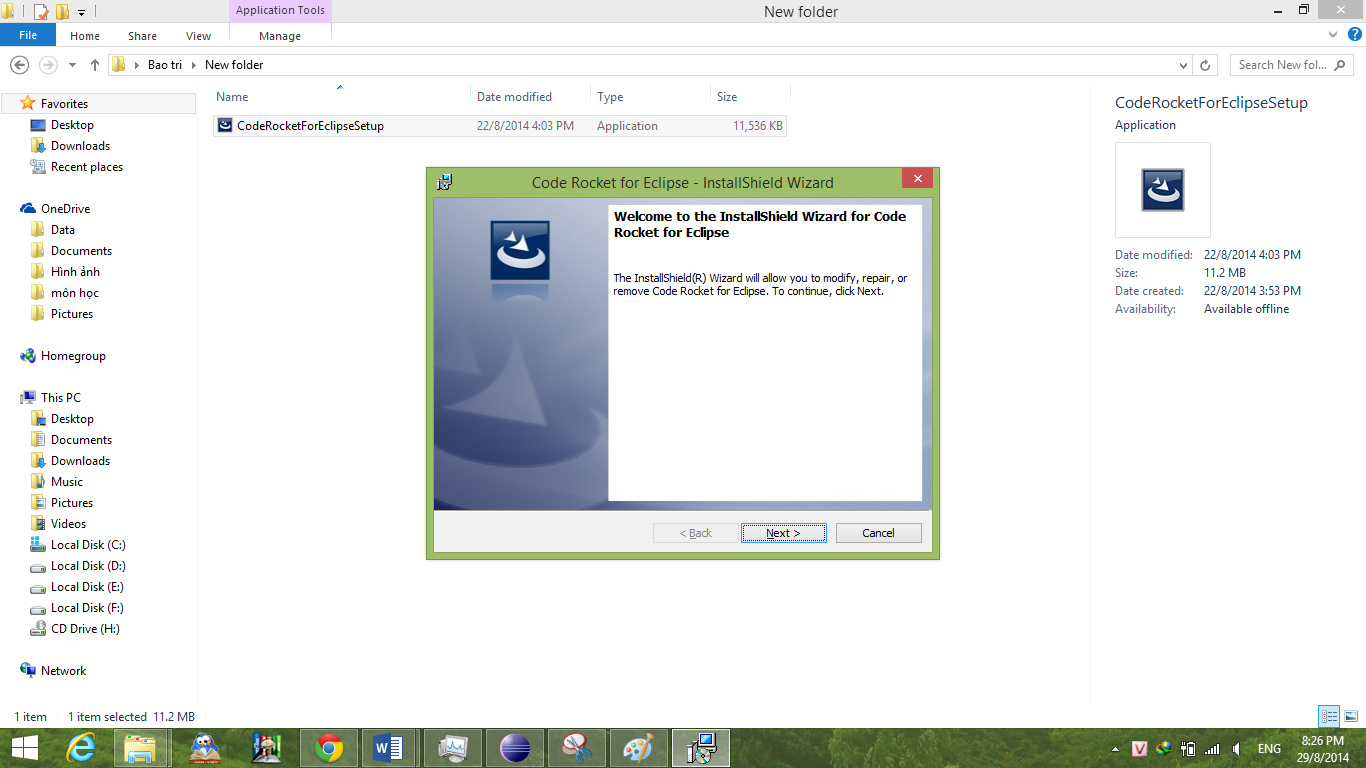
Xong. Kiểm tra lại kết quả



## Chuyển code sang Flowchart

* Cài đặt code rocket cho eclipse: file download bên dưới

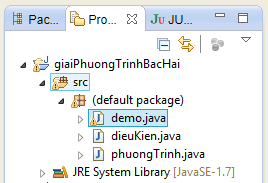
http://www.rapidqualitysystems.com/Products/Trial/CodeRocketForEclipse



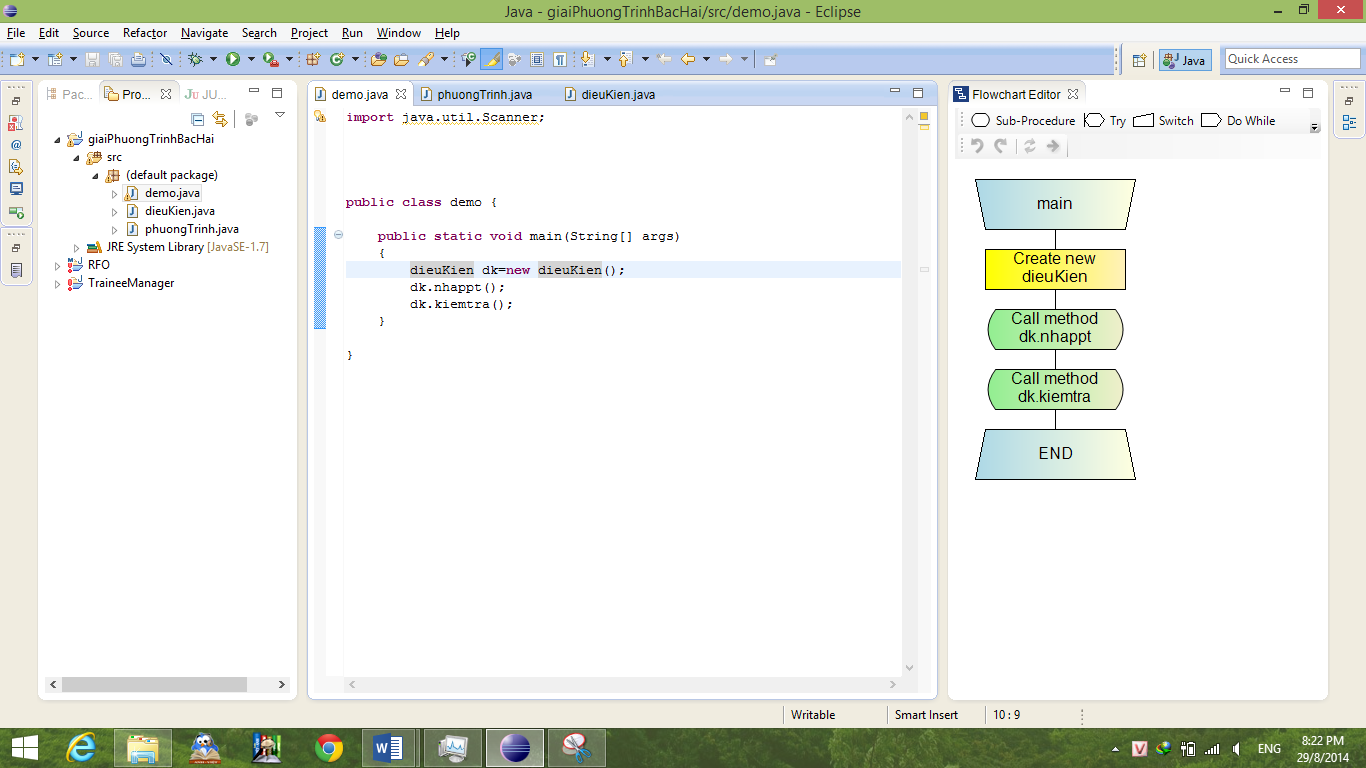
* Mở chương trình eclip.



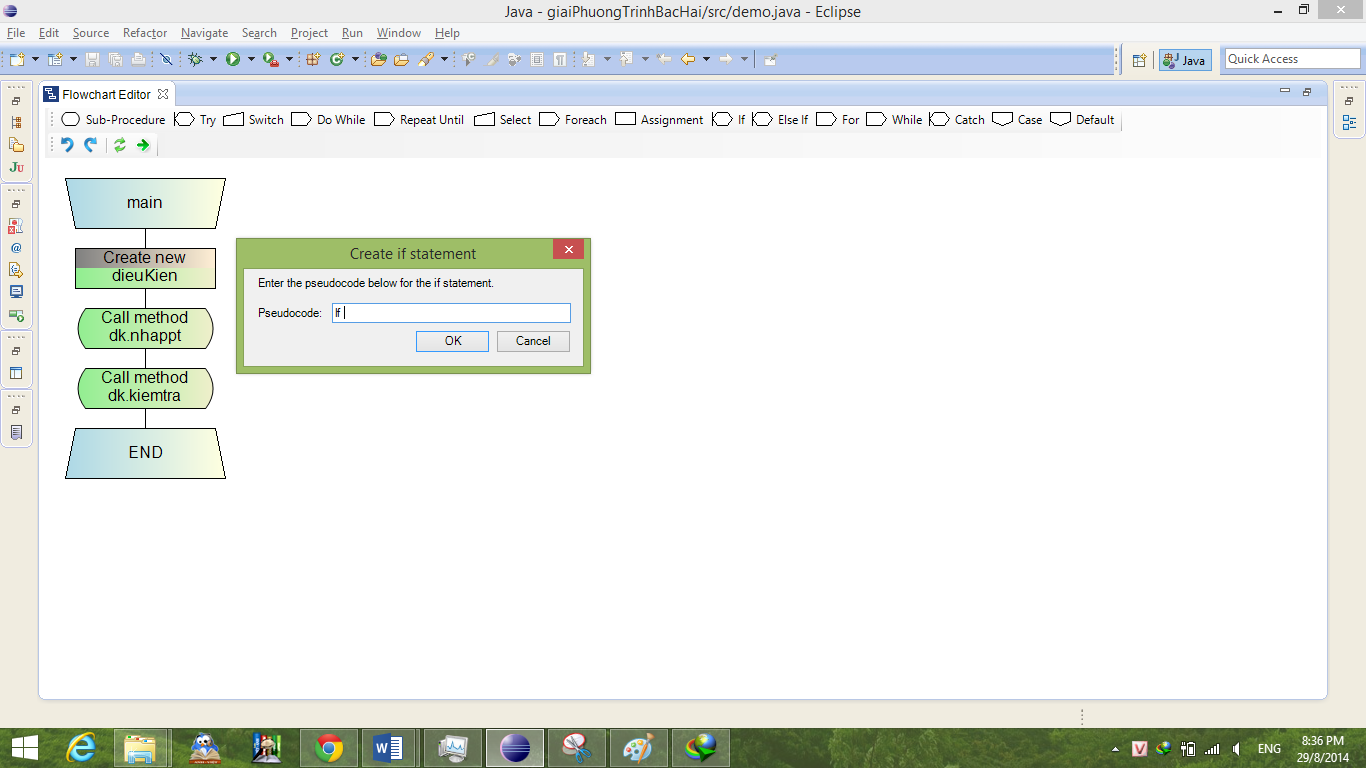
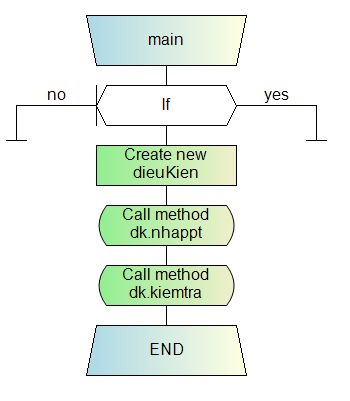
* Mở một project bất kỳ. VD: mở một project giải phương trình bậc hai.



* Ở mục Flowchart Editor: xem kết quả code đã được chuyển sang dạng sơ đồ, đồng thời code cũng được đồng bộ với sơ đồ. Chức năng “High light” cho phép người dùng quan sát code lẫn sơ đồ rỏ ràng hơn.

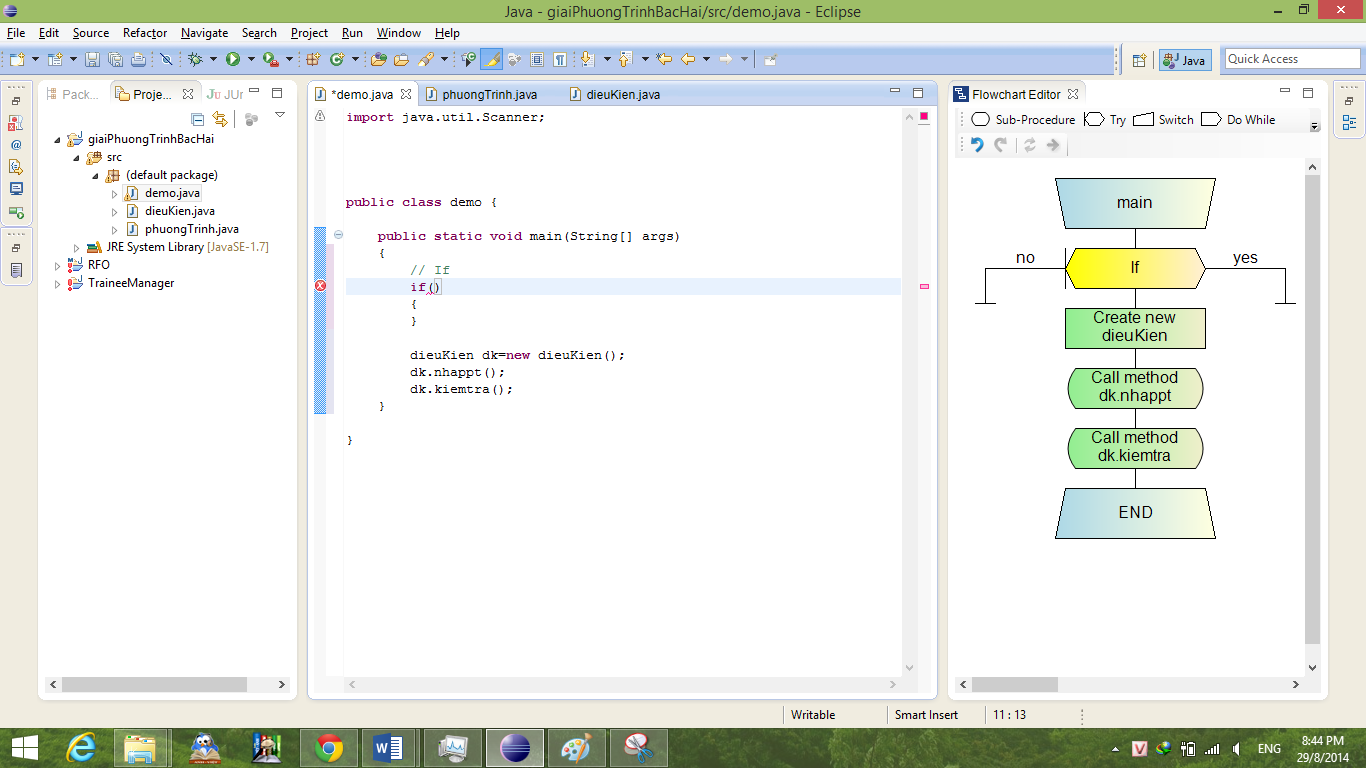


* Ở mục Flowchart Editor: hổ trợ kéo thả



Commit khi muốn đồng bộ với code

* Code sau khi đã đồng bộ với sơ đồ vừa kéo thả khi nãy



* Xuất các tài liệu dạng word hoặc html:

