

**ĐẠI HỌC KHOA HỌC TỰ NHIÊN
THÀNH PHỐ HỒ CHÍ MINH**



GIỚI THIỆU VỀ ECLIPSE

Phạm Thế Bảo

2005

Phần I: GIỚI THIỆU ECLIPSE

A. Eclipse là gì ?

Eclipse là phần mềm miễn phí, được các nhà phát triển sử dụng để xây dựng những ứng dụng J2EE, sử dụng Eclipse nhà phát triển có thể tích hợp với nhiều công cụ hỗ trợ khác để có được một bộ công cụ hoàn chỉnh mà không cần dùng đến phần mềm riêng nào khác. Eclipse SDK bao gồm 3 phần chính: **Platform, Java Development Toolkit (JDT), Plug-in Development Environment (PDE)**. Với JDT, Eclipse được xem như là một môi trường hỗ trợ phát triển Java mạnh mẽ. PDE hỗ trợ việc mở rộng Eclipse, tích hợp các **Plug-in** vào Eclipse Platform. Eclipse Platform là nền tảng của toàn bộ phần mềm Eclipse, mục đích của nó là cung cấp những dịch vụ cần thiết cho việc tích hợp những bộ công cụ phát triển phần mềm khác dưới dạng Plug-in, bản thân JDT cũng có thể được coi như là một Plug-in làm cho Eclipse như là một **Java IDE (Integrated Development Environment)**.

B. Kiến trúc Eclipse:

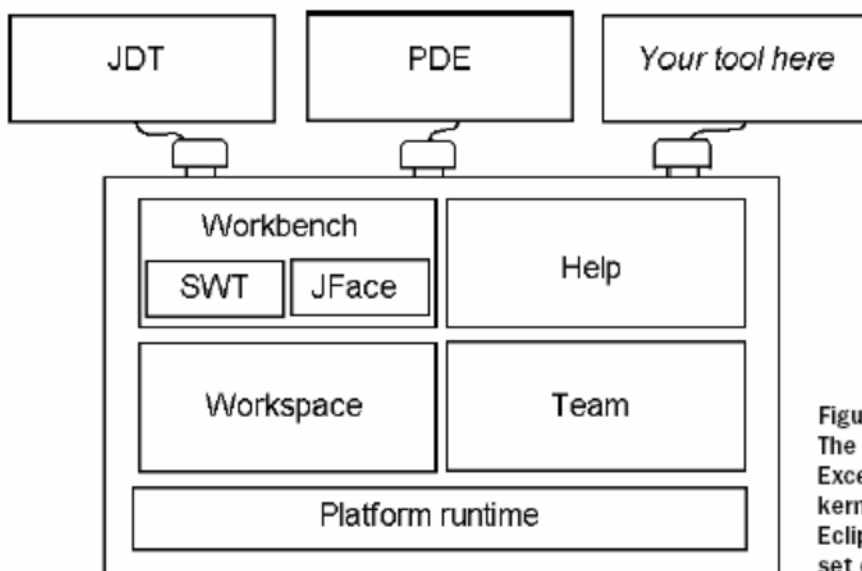


Figure 1.1
The Eclipse architecture.
Except for a small runtime kernel, everything in Eclipse is a plug-in or a set of related plug-ins.

1. The Platform runtime :

Công việc chính của **Platform runtime** là phát xem plug-in nào đang có trong thư mục plug-in của Eclipse. Mỗi Plug-in đều có 1 tập tin Manifest liệt kê những kết nối mà plug-in cần. Plug-in chỉ được tải vào Eclipse mỗi khi thực sự cần thiết để giảm lượng tài nguyên yêu cầu và thời gian khởi tạo.

2. The workspace :

- Workspace chịu trách nhiệm quản lý tài nguyên người dùng được tổ chức dưới dạng **Project**. Mỗi Project là một thư mục con trong thư mục Workspace.
- Workspace bảo quản cấp thấp lịch sử những sự thay đổi tài nguyên, tránh thất thoát tài nguyên người dùng.
- Workspace đồng thời chịu trách nhiệm thông báo những công cụ cần thiết cho việc thay đổi tài nguyên.

3. The Workbench :

Workbench là giao diện đồ họa người dùng của Eclipse, gồm có **Standard Widget Toolkit (SWT)** và **JFace**. Eclipse không hoàn toàn bắt buộc phải sử dụng SWT hay Jface để lập trình giao diện, bạn vẫn có thể sử dụng **AWT** hay **SWING** của Java thông qua việc cài đặt các Plug-ins.

4. Team support :

Trang bị hệ thống quản trị để quản lý dự án của người dùng : **Concurrent Versions System (CVS)**

5. Help :

Cung cấp hệ thống tài liệu mở rộng, có thể là định dạng HTML hay XML.

PHẦN II: LÀM QUEN VỚI ECLIPSE

A. Cài đặt Eclipse :

Giải nén tập tin Eclipse SDK vào thư mục mà bạn muốn cài đặt (Ví dụ : C:\Eclipse). Chép thư mục JRE của JDK vào thư mục con của thư mục Eclipse. Sau đó, chạy tập tin eclipse.exe để hoàn thành cài đặt. Chú ý, khi cài đặt Eclipse không đòi hỏi cài đặt, đăng ký vào hệ điều hành, sửa đổi biến môi trường hay yêu cầu boot lại máy.

B. Perspective :

Khi khởi động Eclipse, màn hình đầu tiên bạn thấy đó là Eclipse Workbench. Eclipse Workbench đưa ra khái niệm về Perspective, đó là những bối cảnh khác nhau của giao diện Eclipse hỗ trợ cho một công việc nhất định. Khi làm việc với Eclipse bạn sẽ luôn chuyển qua lại giữa các Perspective khác nhau. Ví dụ như khi soạn thảo chương trình, kiểm tra lỗi, biên dịch, sửa lỗi đều có mỗi Perspective khác nhau.

C. Tạo một chương trình Java :

Trước tiên, bạn phải tạo một Project mới:

1. Chọn File → New Project → Hộp thoại New Project mở ra có 3 chọn lựa ở bên cửa sổ trái: Java, Plug-in Development, và Simple. Chọn Java để tạo một Project Java. Sau đó chọn Java Project ở cửa sổ bên phải. Chú ý nếu bạn có cài đặt các Plug-ins phát triển Java thì sẽ được liệt kê hết tất cả ở phần chọn lựa con trong chọn lựa Java bên cửa sổ trái.
2. Click Next → Hộp thoại mới mở ra, yêu cầu đặt tên cho Project mới. Bạn hãy điền vào tên project là **Hello**.
3. Chọn Next → Hộp thoại kế tiếp cho phép bạn thay đổi các cấu hình Java. Chọn Finish để kết thúc.

Kế tiếp, bạn có thể tạo ra một chương trình Java:

1. Click phải chuột trên project **Hello**, chọn New → Class.
2. Hộp thoại mới mở ra, gồm có :
 - a. Source Folder : giữ nguyên, không thay đổi
 - b. Package : gõ vào **org.eclipseguide.hello**
 - c. Class Name : gõ vào **HelloWorld**
 - d. Ở phần Which Method Stubs Would You Like to Create?, chọn public static void main(String[] args)
3. Chọn Finish để kết thúc.

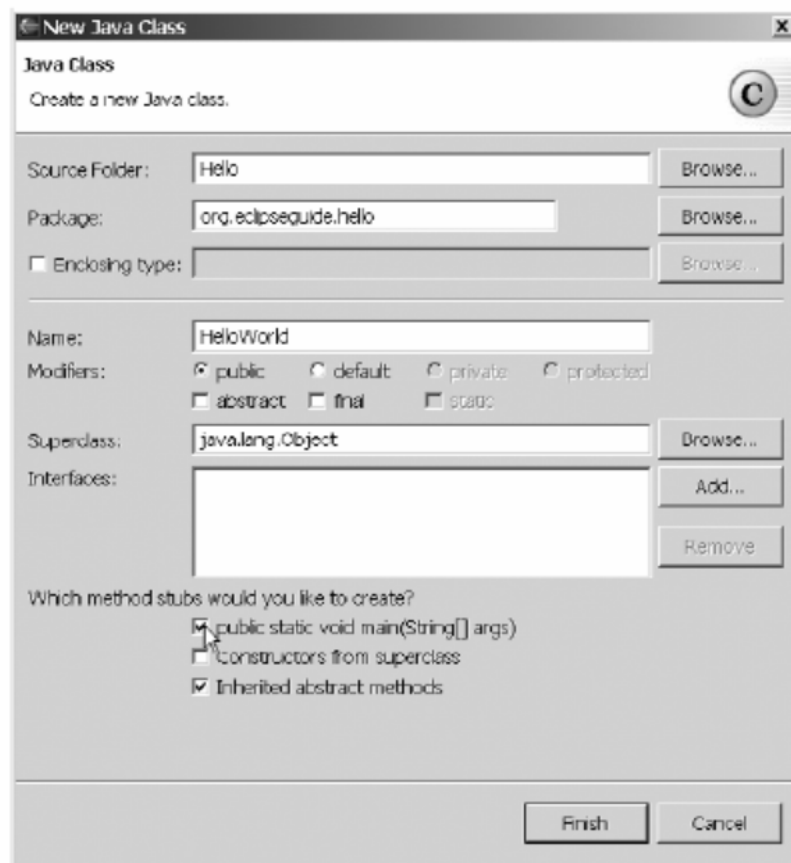


Figure 2.3
Creating the
HelloWorld class
using the New Java
Class wizard

Đoạn mã chương trình tự động được sinh ra, với các phương thức rỗng. Bạn phải tự điền thêm vào các chức năng mới cho các phương thức này, thêm dòng lệnh `System.out.println("Hello, world!")` vào hàm `main()` :

```
/*
 * Created on Feb 14, 2003
 *
 * To change this generated comment go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
package org.eclipseguide.hello;
/**
 * @author david
 */
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

D. Chạy chương trình Java:

1. Chọn chương trình Java bạn muốn chạy bên cửa sổ Package Explorer. Ở đây, sẽ chọn tập tin `HelloWorld.java`
2. Trên Menu chính, chọn `Run` → `Run As` → `Java Application`.
3. Cửa sổ Task sẽ thay đổi bằng cửa sổ Console, xuất ra kết quả chương trình như hình dưới đây:

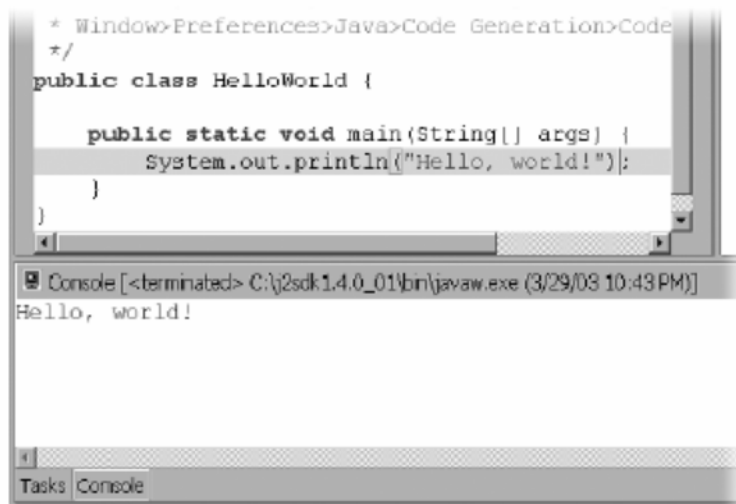


Figure 2.5
The Eclipse Console
view displays the
output from the
HelloWorld program.

E. Gỡ lỗi một chương trình Java (Debug):

Chúng ta sẽ debug chương trình HelloWorld.java. Trước tiên, bạn phải thay đổi chương trình HelloWorld.java theo chương trình dưới đây :

```
public class HelloWorld {  
    public static void main(String[] args) {  
        say("Hello, world");  
    }  
    public static void say(String msg) {  
        for (int i = 0; i < 3; i++) {  
            System.out.println(msg);  
        }  
    }  
}
```

Khi debug một chương trình, bạn cần thiết lập các điểm ngắt cần thiết để kiểm tra lỗi, chương trình sẽ tạm dừng khi gặp các điểm ngắt này, cho phép bạn gỡ lỗi, nếu không có các điểm ngắt này chương trình sẽ tiếp tục thực hiện. Để tạo các điểm ngắt, bạn chọn dòng cần dừng lại, rồi double-click vào lề trái màu xám của cửa sổ soạn thảo phía trước dòng lệnh. Một dấu chấm màu xanh xuất hiện, cho biết đã kích hoạt điểm ngắt. Sau đó, bạn thực hiện các bước sau để debug chương trình:

1. Chọn chương trình cần debug.
2. Chọn Menu Run trên thanh Menu chính → chọn Debug As → Java Application
3. Eclipse tự động chuyển từ Java Perspective sang Debug Perspective.

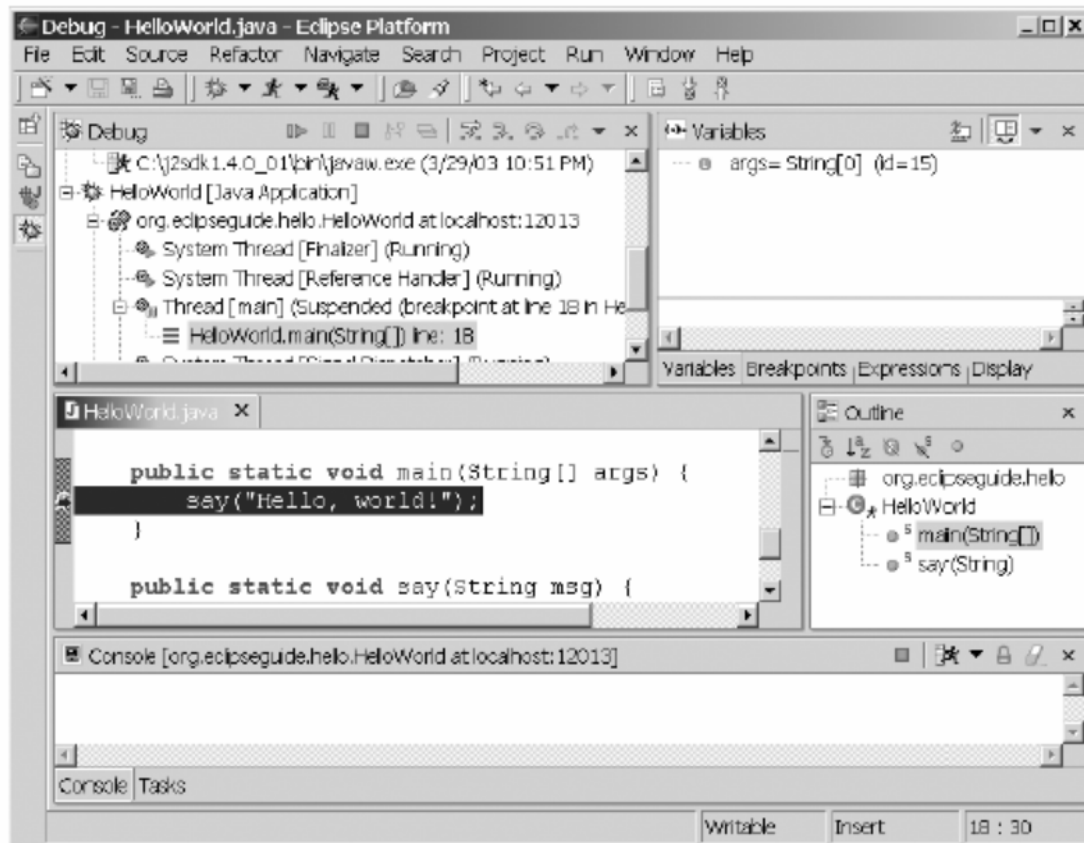


Figure 2.6 Debugging HelloWorld: Execution is suspended at the first breakpoint.

F. Các tính năng mới của Eclipse hỗ trợ cho việc soạn thảo chương trình Java:

1. **Code Completion (Hoàn chỉnh đoạn mã) :** Eclipse hỗ trợ bạn viết mã chương trình Java thêm chính xác, đồng thời giúp không cần phải nhớ nhiều về cú pháp câu lệnh. Mỗi khi bạn đặt dấu chấm hoặc ấn Ctrl + Space (kích hoạt Content Assistant), trình soạn thảo Java của Eclipse sẽ sô ra một danh sách các câu lệnh hỗ trợ tương ứng, đồng thời xuất hiện phần hướng dẫn sử dụng câu lệnh tương ứng bên cạnh.

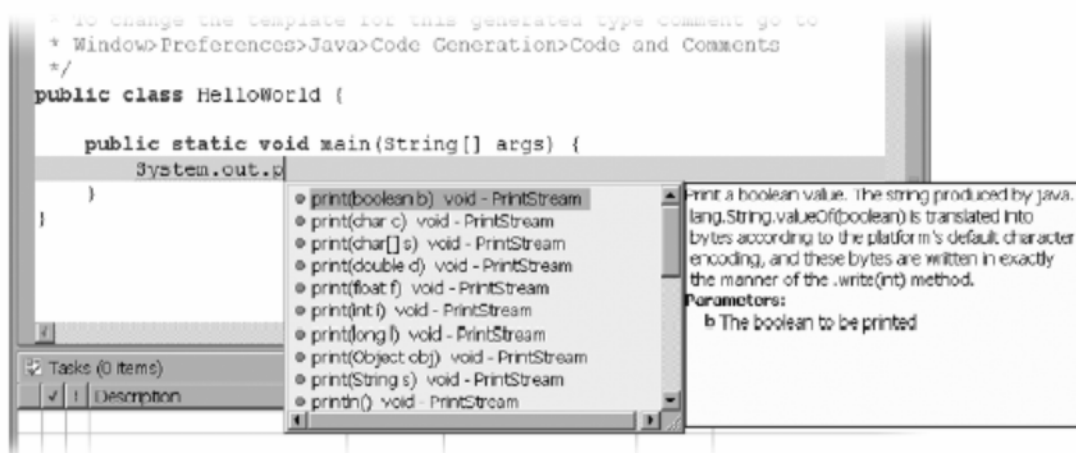


Figure 2.4 The Eclipse code assist feature displays a list of proposed methods and their Javadoc comments. Scroll or type the first letter (or more) to narrow the choice, and then press Enter to complete the code.

Chú ý: để sử dụng tính năng hỗ trợ phân hướng dẫn sử dụng, bạn cần phải chỉ cho Eclipse biết đường dẫn đặt Javadoc của JDK như sau :

- Trên thanh Menu chính, chọn Window → Preferences → ở danh sách cửa sổ bên trái, chọn Java → Installed JREs → phía cửa sổ bên phải chọn Standard VM, click nút Edit.
- Hộp thoại mới xuất hiện → bạn hãy thay đổi đường dẫn ở ô Javadoc URL theo đúng đường dẫn của Javadoc trong máy của bạn.

2. **Quick Fix (Sửa lỗi nhanh):** Mỗi khi bạn gõ vào một câu lệnh mà có vấn đề về lỗi thì Eclipse ngay lập tức thông báo lỗi, khi đó ngay phía trước vị trí dòng lệnh xuất hiện một ký hiệu hình bóng đèn sáng, click vào ký hiệu này (hoặc đặt con trỏ chuột ngay dòng lệnh lỗi, nhấn Ctrl+1) Eclipse sẽ đề nghị bạn một danh sách các phương pháp khắc phục lỗi tương ứng, double-click vào biện pháp thích hợp Eclipse sẽ hỗ trợ bạn sửa lỗi một cách nhanh chóng.
3. **Refactor:** Trong Project của bạn nếu có việc các lớp sử dụng kế thừa hoặc cài đặt từ các lớp khác, khi đổi tên một lớp sẽ ảnh hưởng đến toàn bộ Project, bắt buộc bạn phải tra lại toàn bộ Project để thay đổi. Eclipse đã giúp bạn làm việc này một cách nhanh chóng nhờ tính năng Refactor, Eclipse sẽ tự động cập nhật toàn bộ Project cho phù hợp với tên mới, bạn chỉ cần thực hiện theo các bước sau : nhấn phải chuột trên tên tập tin cần đổi, chọn Refactor → Rename. Ngoài ra, Refactor sẽ giúp bạn rút trích ra được lớp giao diện (Interface) từ các lớp dựng sẵn và Eclipse sẽ tự động cài đặt Interface trên các lớp có sử dụng giao diện này, tương tự như trên bạn chọn Extract Interface trong Refactor và tiếp tục thực hiện theo các hộp thoại xuất hiện kế tiếp.
4. **Local History:** giúp bạn so sánh giữa phiên bản mới và phiên bản cũ của tập tin bạn đang làm việc:
 - Nhấp phải trên tập tin trong cửa sổ Package Explorer.
 - Chọn Compare With → Local History.Bạn có thể cho phép Eclipse thay đổi mã chương trình thành trở lại phiên bản cũ bằng cách tương tự, thay vì chọn Compare With, bạn sẽ chọn Replace With.

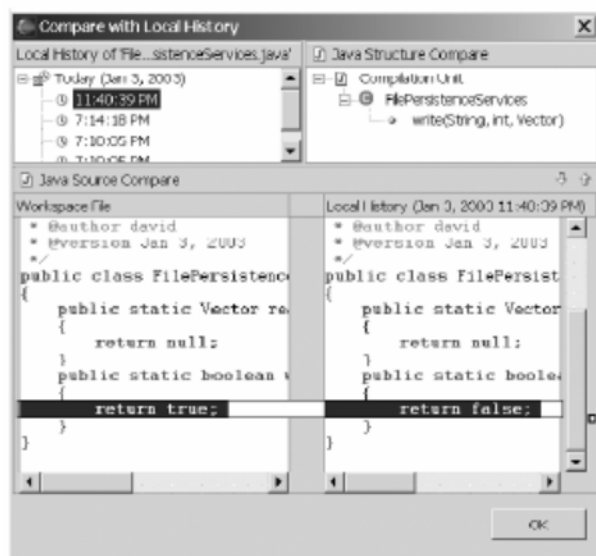


Figure 3.6
Comparing the current
code with a previous
version in the local history

5. **Java Scrapbook pages (Trang Java rời đính kèm):** Khi viết chương trình Java, đôi khi bạn có một ý tưởng mới mà bạn không chắc nó sẽ hoạt động tốt, bạn chỉ muốn thử trước mà không muốn ảnh hưởng đến chương trình bạn đang làm việc. Eclipse sẽ giúp bạn viết một chương trình nhỏ rời ra bằng cách sử dụng trình soạn thảo đơn giản được thực thi dưới dạng command prompt. Đó là Java Scrapbook Pages. Bạn có thể gõ vào Scrapbook Pages các câu lệnh Java và thực thi nó mà không cần khai báo lớp (class) hay phương thức (method). Để tạo một trang Scrapbook, bạn phải chuyển sang bối cảnh soạn thảo Java (Java Perspective), rồi thực hiện các bước sau :
- Nhấp phải chuột vào tên Project (ở đây là **HelloWorld** project).
 - Chọn New → Scrapbook Page → Hộp thoại xuất hiện yêu cầu nhập vào tên tập tin : **Test**.
 - Sau đó, gõ vào một vài dòng lệnh Java để thử :

```
for(int i = 1; i < 10; i++)  
{  
    HelloWorld.say(Integer.toString(i));  
}
```
 - Thực thi đoạn mã này như sau : nhấp phải chuột trên màn hình soạn thảo, chọn Set Imports trong Menu phụ.
 - Hộp thoại Java Snippet Imports xuất hiện, nhấn vào nút Add Packages.
 - Hộp thoại mới xuất hiện yêu cầu bạn chọn gói được sử dụng : **org.eclipseguide.hello**. Nhấn OK.
 - Bây giờ, bạn có thể thực thi đoạn mã bằng cách quét chuột chọn toàn bộ đoạn mã → nhấp phải chuột → Execute. Như bất kì chương trình Java bình thường, kết quả sẽ được xuất ra trong ô Console View.

CHÚ Ý: khi sử dụng với **StringTokenizer**, bạn phải khai báo gói tương ứng trong **java.util.***, bằng cách sau : trong hộp thoại Java Snippet Imports, nhấn vào nút Add Types và gõ vào **StringTokenizer**, Eclipse sẽ tự động tìm thấy gói tương ứng **java.util.StringTokenizer**.

G. Thay đổi cấu hình và thông số cài đặt, bao gồm kiểu định dạng mã và thiết lập biến đường dẫn môi trường:

1. Câu chú thích Javadoc:

Đây là đoạn văn bản xuất hiện ở phần đầu khi bạn tạo ra một lớp mới. Bạn có thể thay đổi nó theo các bước sau đây:

- Chọn WindowPreferences → Java → Code Generation
- Nhấn chuột vào tab Code and Comment.
- Chọn Code → New Java files → click vào nút Edit.

- Thay đổi lại như sau :

```
/* ${file_name}
 * Created on ${date}
 */
${package_declaration}
${typecomment}
${type_declaration}
```

- Click OK trong hộp thoại Edit Template.
- Bằng cách tương tự bạn có thể thay đổi TypeComment trong Code → Types.

2. Định dạng lại đoạn chương trình: để đồng bộ hoá toàn bộ đoạn mã chương trình theo một chuẩn định dạng nhất định, bạn click phải chuột trên màn hình soạn thảo → chọn Format. Bạn có thể thay đổi lại chuẩn định dạng này :

- Chọn Preferences → Java → Code Formatter.
- Ở phía bên cửa sổ Options, chọn tab New Lines.
- Đánh dấu kiểm chọn vào Insert a New Line Before an Opening Brace (Cho phép chèn thêm một dòng trước khi mở ngoặc {).

3. Mẫu đoạn mã chung: Eclipse cho phép ta có thể tạo ra những mẫu đoạn mã chung, giúp ta dễ dàng sử dụng nhiều lần mà không cần phải gõ lại toàn bộ như trước.

- Windows → Preferences → Java → Editor → Templates.
- Nhấn vào nút New.
- Hộp thoại New Templates xuất hiện : đặt tên cho mẫu sử dụng là **sop** và **Shortcut for System.out.println()** vào ô mô tả (Description).
- Điền mẫu vào Template: `System.out.println("${cursor}") ;`.
- Nhấn OK để đóng hộp thoại Template.
- Nhấn OK để đóng Preferences.

Cursor chỉ ra rằng vị trí con trỏ chuột được đặt vào đó khi chèn đoạn lệnh vào. Khi sử dụng, bạn chỉ cần gõ vào **sop** và ấn Ctrl+Space thì câu lệnh `System.out.println("${cursor}")` xuất hiện.

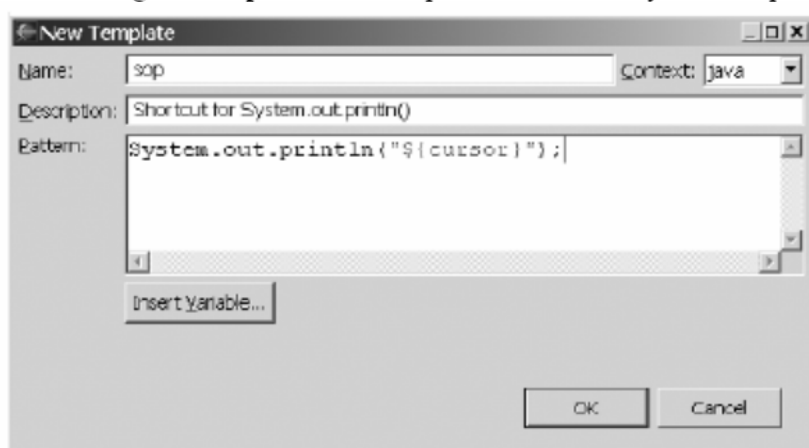


Figure 2.7 Creating a shortcut for `System.out.println()` using a Java editor template