# Mục lục

Mnc Inc 1

H¢éNG DAN SS DUNG ECLIPSE 2

1. Các thao tác co bán 2
	1. [Create và import project 2](#_TOC_250000)
	2. BuiId, run chvong trình 2
	3. Debug chvong trình 3
	4. Ðóng gói chvong trình 3
2. M®t so thao tác khác 3
	1. Tao ra fiIe Javadoc mô tá cho fiIe source code 3
	2. Chon compiIer tùy thích 5
	3. Tùy chính hieu th… LineNumber và Ivu fiIe UTF−8 5
	4. Tùy chính tn b®ng chèn bracket và semicoIon 6
	5. Chèn thv vi¾n trnc tiep và project 6
	6. Sú dnng các UserLibrary 7
3. Các thao tác tang toc b® viet code 8
	1. Sú dnng chúc nang code tempIate 8
	2. Sú dnng chúc nang quick fix 9
	3. Các phím tat thvòng dùng 10

**HƯỚNG DẪN SỬ DỤNG ECLIPSE**

1. **Các thao tác cơ bản**

## *Create và import project*

* Tạo project mới: Vào menu **File** chọn **New** / **Project**, chọn loại project mà bạn muốn tạo.
* Import project có sẵn: Vào menu **File** chọn **Import**. Trong hộp thoại **Import**, chọn mục **General / Existing project into workspace**, click **Next** trỏ đường dẫn đến project cần import, click **Finish**

## *Build, run chương trình*

Click vào nút **Run** trên thanh công cụ, chọn **Run**…

Khi hộp thoại run mở ra ta chọn **New Java Application**. Sau khi chọn project xong thì chọn **Run** để thực thi chương trình.


## *Debug chương trình*

Để có thể debug chương trình ta cần đặt beakpoint tại một vị trí nào đó trong source code cần debug, rồi vào menu **Run** chọn **Debug**… (sau đó cấu hình như trên). Lưu ý là có một số trình bật firewall (Bit Defender) khi đó chương trình sẽ không thể debug được.

## *Đóng gói chương trình*

* Chọn một project cần export trong cây explore rồi nhấn chuột phải chọn export.
* Trong cửa sổ Export ta chọn java/JAR file, chọn next.
* Trong cửa sổ Jar Export ta chọn đường dẫn cho file output.jar, chọn next.
* Tiếp tục chọn next, trong textbox MainClass ta cần chọn file class chứa hàm main trong

chương trình của ta.

* Chọn Finish để hoàn tất

# Một số thao tác khác

## *Tạo ra file Javadoc mô tả cho file source code*

Khi viết các phương thức trong eclipse, nếu bạn tuân theo các quy tắc đặt comment cho phương thức thì bạn sẽ có thể tạo ra các file doc mô tả về các phương thức, các lớp mà bạn viết, các mô tả này bao gồm mô tả về chức năng của hàm, đối số input, output, giá trị trả về và các Exception mà phương thức sẽ throws.

Để phát sinh tự động các comment đó ta đặt con trỏ chuột tại vị trí tên hàm ( public void myMethod(…) ), nhấn tổ hợp phím Alt+Shift+J chương trình sẽ phát sinh tự động các comment đó. Việc còn lại của bạn là chỉ cần viết các thong tin mô tả cần thiết. Lưu ý là nếu bạn viết các thong tin đó bằng tiếng việt sử dụng bộ gõ Unikey thì cần phải sử dụng bảng mã Unicode dựng sẵn.

Đây là quy tắc comment:

1. /\*\*

* + 1. \* Các mô tả cho phương thức
		2. \* @param iValue : mô tả cho parameter
		3. \* @param dValue : mô tả cho parameter
		4. \* @return mô tả giá trị trả về 6. \*/
1. public String myMethod(int iValue, double dValue) {
2. return ”";

9. }

Sau khi tạo ra các comment ghi chú cho các phương thức theo quy định, bạn đã có thể tạo

ra các file javadoc mô tả về lớp mà bạn vừa viết.

Vào menu project, chọn generate javadoc…

Bạn chọn **Next** để tùy chỉnh thêm một số cấu hình khác như chèn thêm một số file doc

của các thư viện mà bạn sử dụng; hoặc click Finish để tạo ra file javadoc.

## *Chọn compiler tùy thích*

Vào menu Window, chọn Preferences.


## *Tùy chỉnh hiểu thị LineNumber và lưu file UTF-8*

Để hiển thị các dòng trong eclipse ta vào menu Preference. Mặc định eclipse sử dụng encode CP1252 nên ta không thể soạn thảo được các kí tự tiếng việt có dấu. Do vậy muốn lưu trữ file source theo dạng UTF-8 ta làm như sau:


## *Tùy chỉnh tự động chèn bracket và semicolon*

Trong khi code, bạn phải làm một số thao tác như di chuyển con trỏ xuống cuối dòng rồi ‘;’ hoặc mở ‘{‘, như vậy tốc độ thao tác code của ta phần nào bị giảm xuống. Để khắc phục chức năng này ta có thể bật chức năng tự động chèn bracket và semicolon. Sau khi bật chức năng này trong khi code ta có thể chèn ‘;’ hoặc ‘{‘ bất cứ nơi nào trong dòng code.


## *Chèn thư viện trực tiếp và project*

Bạn có một bộ thư viện class java, bạn vừa down một bộ java library trên mạng,… tất cả chúng đều được nén trong file JAR, vấn đề là làm sao sử dụng nó trong các project? Cách sử dụng rất đơn giản, bạn chỉ việc chép trực tiếp file thư viện đó vào project của mình và thực hiện thêm một số thao tác để Eclispe hiểu các thư viện đó trong project của bạn.

Sau khi đã chép thư viện vào, nhấn chuột phải vào project đó và chọn properties:

Ghi chú: cấu hình cái library sử dụng trong project được lưu trong file .classpath

## *Sử dụng các UserLibrary*

Bạn có một tập các thư viện thường xuyên sử dụng, mỗi lần sài bạn phải thừc hiện các thao tác như trên… thật mất thời gian, có cách nào để khắc phục việc này hay không? Câu trả lời chính là sử dụng UserLibrary. Tạo ra UserLibrary:

Add UserLibrary vào project: nhấn chuột phải vào project trong cây explore, chọn build path, user library; sau đó sẽ mở ra hộp thoại Add Library ( hoặc vào properties của project ) chọn library cần add vào và click finish

# Các thao tác tăng tốc độ viết code

## *Sử dụng chức năng code template*

Trong khi lập trình có những đoạn code bạn phải viết nhiều lần, hoặc có những đoạn code quá dài, để giảm bớt thời gian viết code eclipse cung cấp cho ta một chức năng rất mạnh đó là code template.

Tạo code template:


## *Sử dụng chức năng quick fix*

Một chức năng khác rất mạnh trong việc hỗ trợ lập trình đó là chức năng quick fix. Với chức năng này bạn có thể chỉ sửa rất nhanh các lỗi trong khi lập trình. Ngoài ra nó còn giúp ta tạo ra các đoạn lệnh khai báo biến, khai báo hàm, tạo biến input cho hàm, các local variable,… và còn rất nhiều cái khác.

Để gọi chức năng quick fix ta đặt con trỏ chuột trên dòng chứa lỗi, nhấn Ctrl+1 để hiện ra

khung chỉnh sửa.

Ví dụ về sử dụng quick fix để tạo khai báo biến.


## *Các phím tắt thường dùng*

* Run: Ctrl+F11,
* Debug: F11, F6: step over, F7: step into, Ctrl+R: goto breakpoint
* Ctrl+PGUP, PGDOWN chuyển các tab
* Alt+’-’ mở menu context của các TabEditor
* Ctrl+Shift+J tạo comment cho các hàm( comment có các annotation )
* Ctrl+O mở bảng để goto một biến hoặc một hàm
* Alt+Shift+Z bao 1 khối bằng một đoạn lệnh như try, catch, while
* Ctrl+Shift+M import 1 package
* Ctrl+/ comment 1 dòng lệnh
* Ctrl+Shift+P go to next bracket
* Ctrl+Alt+H mở hàm nào gọi hàm tại vị trí chuột
* Alt+T – X exact method
* Alt+T – S encapsulation field
* Ctrl+Q go to last edit