**Quy tắc Java cơ bản**

**1. Quy tắc đặt tên trong Java**

Thực ra những quy tắc này chúng ta tự đưa ra để dễ dàng phát triển dự án. Ví dụ khi một dự án có nhiều lập trình viên tham gia thì cần phải đưa ra một chuẩn (quy tắc) để đặt tên biến, tên hàm .... để khi nhìn vào các lập trình viên có thể hiểu code của nhau.

Trước tiên chúng ta tìm hiểu về quy tắc chung khi đặt tên biến, tên hằng, tên package, tên class, tên interface ...

**1.1. Quy tắc chung.**

* Lập trình viên nên khai báo tên có ý nghĩa và thể hiện được mục đích của file/ biến/ phương thức/... đó.
* Tên khai báo không nên dài quá 20 ký tự hoặc có thể ít hơn nhưng phải đảm bảo đầy đủ về mặt ý nghĩa của nó, và tên cũngkhông được đặt quá ngắn, trừ khi đó là tên tạm (ví dụ như: a, i, j,...).
* Tránh đặt những tên tương tự nhau. Ví dụ như, hai biến có tên là persistentObject và persistentObjects không nên được sử dụng trong một Class vì sẽ dễ gây ra nhầm lẫn trong quá trình viết code.
* Tránh đặt tên gây khó hiểu, dễ nhầm lẫn về mặt ý nghĩa.
* Tên chứatừ viết tắt cũng nên được hạn chế sử dụng trừ khi từ viết tắt đó phổ biến và được nhiều người biết đến. Ví dụ như: bạn không được đặt tên biến là diemTB mà nên đặt tên là diemTrungBinh. nhưng bạn hoàn toàn có thể đặt tên là fileHTML thay vì fileHypertextMarkupLanguage vì tên này quá dài và từ HTML cũng là một từ khá phổ biến, ít nhất là trong giới lập trình viên chúng ta.
* Tránh kết hợp nhiều ngôn ngữ khác nhau (Tiếng Anh + Tiếng Việt + ...), chẳng hạn như addSinhVien, addLop,...
* Không trùng với các "từ khóa" (chúng ta sẽ tìm hiểu ở phần tiếp theo).
* Không được bắt đầu bằng số, ví dụ như: 123sinhVien.
* Tên phải đượcbắt đầu bằng một chữ cái, hoặc các ký tự như $, \_, ...
* Không được chứa khoảng trắng, các ký tự toán học. Nếu tên bao gồm nhiều từ thì phân cách nhau bằng dấu \_.
* Trong Java có phân biệt chữ hoa chữ thường. Ví dụ, như hocSinh sẽ khác với hocsinh.

**1.2. Đặt tên biến.**

* Tên biến phải tuân theo quy tắc chung ở trên được đặt theo**quy tắc lạc đà** (Camel Case): đó là chữ cái đầu tiên của từ đầu tiên phải viết thường và chữ cái đầu tiên của các từ tiếp theo phải viết hoa, ví dụ: diemTrungBinh.
* Ngoài ra, trong một số trường hợp, tên biến cần phải thể hiện rõ kiểu dữ liệu của biến đó. Ví dụ: biến có kiểu là List thì nên đặt tên là studentList, biến có kiểu là Set thì nên đặt tên là studentSet, biến có kiểu là Map thì nên đặt tên là studentMap, biến có kiểu là Array thì nên đặt tên là studentArray,...

**1.3. Đặt tên hằng số (Constant).**

Tên hằng số phải tuân theo quy tắc chung ở trên và phải được viết hoa (ví dụ PI). Nếu tên hằng số có từ hai từ trở lên thì phải có dấu \_ ngăn cách giữa các từ, ví dụ: SO\_FIBONACCI.

**1.4. Đặt tên phương thức (Method).**

* Tên phương thức phải tuân theo quy tắc chung ở trên.
* Chữ cái đầu tiên của từ đầu tiên trong tên phương thức phải viết thường và là một động từ, còn chữ cái đầu tiên của các từ tiếp theo phải viết hoa (giống quy tắc đặt tên biến). Ví dụ: tinhDiemTrungBinh, themNhanVien,...

**1.5. Đặt tên Class và Interface.**

* Tên Class và Interface phải tuân theo quy tắc chung ở trên và chữ cái đầu tiên của mỗi từ phải được viết hoa. Ví dụ: Class SinhVien.
* Tên Class nên có thêm những từ có hậu tố phía sau để thể hiện rõ hơn mục đích của Class đó, chẳng hạn như DivZeroException (chúng ta sẽ tìm hiểu rõ hơn về **Exception**ở những bài sau).
* Tên Interface nên có thêm chữ I đằng trước. Ví dụ: IFrame.
* Tên lớp dẫn xuất nên có từ Abstract làm tiền tố, ví dụ: Class AbstractStudent (chúng ta sẽ tìm hiểu về lớp dẫn xuất ở những bài sau).

**1.6. Đặt tên Package.**

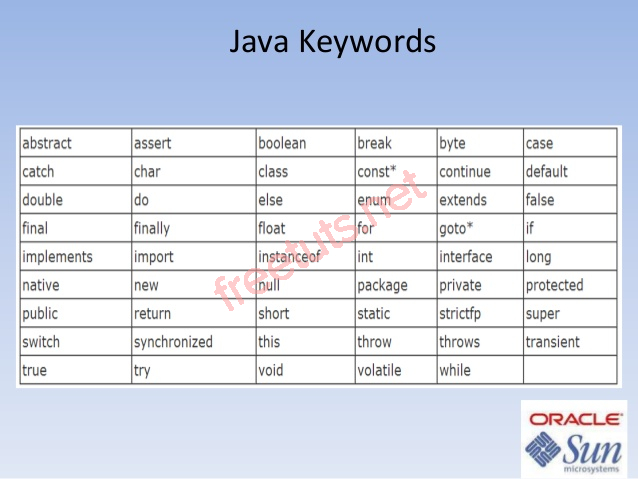
Tên**Package**phải tuân theo **quy tắc chung**ở trên và phải**viết thường**.

**1.7. Đặt tên Project.**

Tên **Project**phải tuân theo **quy tắc chung** ở trên và **chữ cái đầu tiên của mỗi từ phải viết hoa**.

**2. Các từ khóa quan trọng trong Java.**

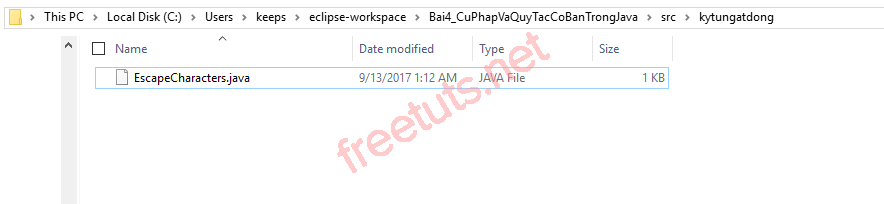
Dưới đây là bảng liệt kê **các từ khóa quan trọng**trong Java. Chúng ta sẽ tìm hiểu ý nghĩa, công dụng của các từ khóa này trong những bài sau.



1. **Một số ký tự ngắt văn bản (escape character) phổ biến trong Java.**

| **Kí tự** | **Mô tả** |
| --- | --- |
| \t | Chèn vào một khoảng trắng trong đoạn văn bản. |
| \b | Xóa lùi 1 ký tự trước \b trong đoạn văn bản. Có bao nhiêu ký tự \b thì sẽ xóa bấy nhiêu ký tự (chỉ hoạt động khi có chuỗi tồn tại phía trước và phía sau \b). |
| \n | Chèn một dòng mới trong đoạn văn bản. |
| \r | Thay thế các ký tự trước \r bằng số các ký tự phía sau \r |
| \' | Chèn vào dấu nháy đơn trong đoạn văn bản. |
| \" | Chèn vào dấu nháy kép trong đoạn văn bản. |
| \\ | Chèn vào dấu \ trong đoạn văn bản. |

**Lưu ý**: Các ký tự như \b và\r khi chạy bằng công cụ Eclipse đôi khi sẽ làm phát sinh lỗi. Nên trong ví dụ dưới đây , tôi sẽ sử dụng **Terminal**để biên dịch chương trình. Để tìm hiểu về cách sử dụng Terminal để biên dich, các bạn quan sát cấu trúc thư mục chứa Project của tôi như sau:

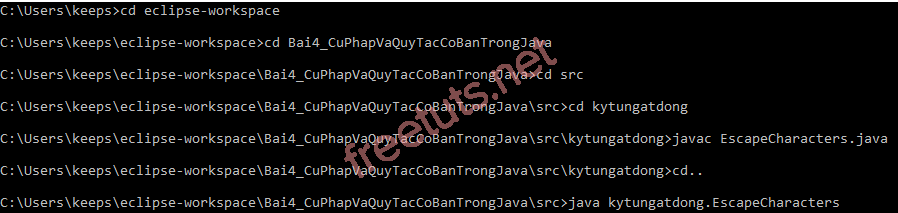


Nội dung **EscapeCharacters.java**:

**Ký tự ngắt dòng trong Java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | package kytungatdong;    public class EscapeCharacters {        public static void main(String[] args) {          System.out.println("Tab: " + "Welcome\tto Freetuts!");          System.out.println("Backspace: " + "Welcome to\b Freetuts!");          System.out.println("New line: " + "Welcome\nto Freetuts!");          System.out.println("Welcome to\rFreetuts!"); // /r          System.out.println("Single quote: " + "Welcome to\'Freetuts!");          System.out.println("Double quote: " + "Welcome to\''Freetuts!");          System.out.println("Backslash: " + "Welcome to\\Freetuts!");      }    } |

Để biên dịch chương trình, chúng ta gõ như sau:



Lưu ý, sau khi gõ dòng lệnh javac EscapeCharacters.java, các bạn phải gõ cd.. (dòng lệnh số 6) rồi sau đó mới gõ tiếp dòng java kytungatdong.EscapeCharacters nếu không thì trong một số trường hợp sẽ dẫn đến lỗi như sau: "Error: Could not find or load main class EscapeCharacters"

Kết quả sau khi biên dịch chương trình:

