* Hàm: đưa code vào hàm và dùng tên hàm để giải thích mục đích của nó
* Good routine names: 171
* How long a routine can be: 173
* How to use routine parameter: 174
* **Nguyên tắc đặt tên hàm, phương thức:**
  + **Mô tả chức năng của hàm đó**: Nếu 1 hàm tính số liệu báo cáo và xuất ra 1 file, thì tên ComputeReport() là chưa đủ. Tên ComputeReportAndOutputFile() thì đầy đủ hơn nhưng lại quá dài. Giải pháp là xây dựng 1 hàm chỉ làm 1 chức năng, không làm quá nhiều chức năng trong 1 hàm. Nếu cần thì vẫn phải đặt tên dài để mô tả hàm đầy đủ
  + **Không dùng những từ vô nghĩa, mơ hồ:** Những từ như HandleCalculation(), PerformServices(), ProcessInput(), DealWithOutput() không chỉ rõ được chức năng cụ thể của hàm mà chỉ mang nghĩa chung chung. Tên hàm mơ hồ cũng có thể là do chức năng của hàm cũng không rõ ràng. Trong trường hợp đó, cần thiết kế lại hàm với mục tiêu rõ rang hơn
  + **Không phân biệt các tên hàm bằng các con số:** các tên như FormatOutput, FormatOutput1, FormatOutput2 không cho ta biết được sự khác nhau giữa các hàm
  + **Sử dụng giá trị trả về:** Dùng giá trị trả về để đặt tên cho hàm, như cos(), book.CurrentColor(), …. Đối với procedure thì dùng các động từ PrintFile(), Info.check()
  + **Sử dụng những cặp từ trái nghĩa hay đi chung:** add/remove, begin/end, create/destroy, get/set, open/close, show/hide.VD: addBook(), removeBook(),…
  + **Sử dụng qui ước cho những hàm thông dụng:** như hàm lấy id của 1 class, nếu mỗi class viết mỗi kiểu như:

*employee.id.Get()  
dependent.GetId()  
supervisor()  
candidate.id()*

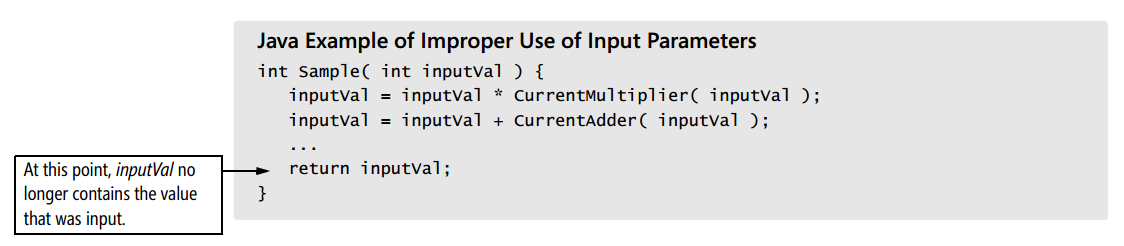
Khi có qui ước chung thì sẽ dễ code và dễ nhớ hơn

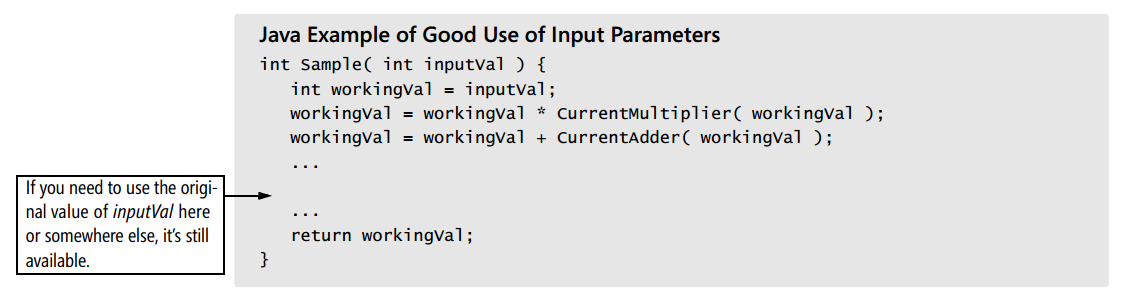
**Chiều dài của một hàm, phương thức**

* Thông thường, chiều dài tối đa của 1 hàm là khoảng 1 màn hình, từ 50 đến 150 dòng code
* Có nhiều ý kiến khác nhau về chiều dài của 1 hàm như:
  + Khi chiều dài 1 hàm tăng lên 200 dòng thì số lỗi giảm đi (Basili , Perricone, 1984)
  + Chiều dài hàm không liên quan đến lỗi (Shen et al. 1985)
  + Những hàm ngắn (ít hơn 143 dòng) có xác suất lỗi lớn hơn 23% so với các hàm dài hơn, nhưng việc sửa lỗi dễ hơn 2.4 lần

**Nguyên tắc truyền tham số cho hàm**

* Truyền tham số theo thứ tự input – modify – output
* Nếu 1 số hàm sử dụng các tham số giống nhau thì sắp xếp thứ tự các tham số ở các hàm đó giống nhau
* Sắp xếp những biến trạng thái, lỗi ở cuối cùng
* Hạn chế sử dụng tham số như biến cục bộ của hàm
  + Nếu cần dùng thì giá trị input vẫn còn
  + Tránh hiểu nhầm khi dùng lại tham số
  + Nếu sau này muốn chỉnh sửa hàm, dùng cái input đó ở 1 nơi nào đó trong hàm

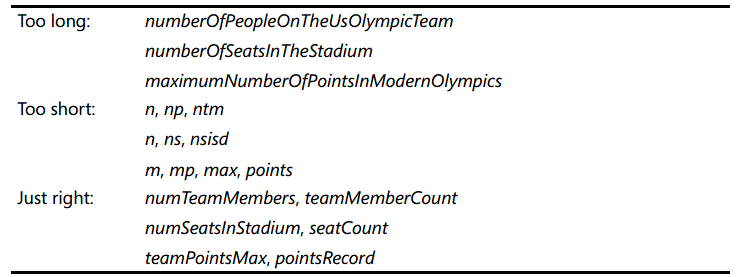




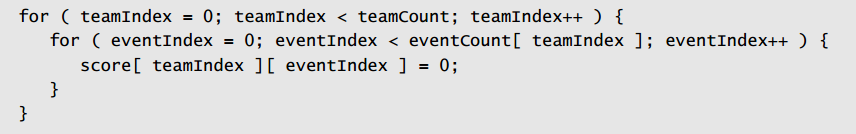
* Số tham số tối đa nên là 7: Nghiên cứu tâm lý chỉ ra rằng con người khó tập trung kiểm soát 7 thứ thông tin cùng 1 lúc (Miller 1956)

**Tên biến**

* Tên biến quá ngắn thì không thể hiện đủ ý nghĩa, trừ những biến dùng cho vòng lặp, hoặc chỉ dùng cho vài dòng code thì có thể đặt tên ngắn như i, j, k
* Tên biến quá dài thì khó dùng



* Biến vòng lặp: Thông thường ta hay dùng i, j, k, nhưng khi vòng lặp dài hoặc nhiều vòng lặp lồng nhau, thay vì đặt i, j, k thì nên đặt tên biến có ý nghĩa để dễ hiểu



* Biến trạng thái: là biến mô tả trạng thái của chương trình, quá trình, …
  + Không nên thêm từ flag trong tên biến trạng thái vì nó ko chỉ ra được biến làm gì
  + Tên biến như statusFlag = 0x08 không cho ta biết được thông tin gì về biến này trừ khi mình là người code hoặc có tài liệu ghi chú
  + Thay vào đó hãy đặt là: characterType = CONTROL\_CHARACTER
  + Đối với các biến trạng thái Boolean (true, false), thay vì đặt tên biến là status, ta có thể đặt các tên biến dễ hiểu hơn như: done, error, found, success
* Những trường hợp nên tránh:
  + Tên viết tắt
  + Tên biến trùng với từ khóa của ngôn ngữ lập trình
  + Tên biến tương tự nhau: fileNumber, fileIndex
  + Những biến khác nghĩa nhưng tên lại khá giống: Thay vì đặt clientRecs và clientReps, thì nên đặt clientRecords và clientReports
  + Tên biến phát âm giống nhau
  + Có số trong tên biến: total1, total2
  + Dùng những ký tự dễ nhìn nhầm: (1, l, I), (0, O), (, , .), (2, Z), (5, S), (G, 6). Vi dụ: 1970, 1 dấu phẩy đã được dùng trong 1 lệnh Fortran FORMAT, thay vì dùng dấu chấm, dẫn đến tính sai quỹ đạo của phi thuyền và làm mất đi phi thuyền trị giá 1,6 tỷ đôla (Weinberg 1983)