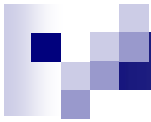


Lập trình hướng đối tượng

Khái niệm



Nội dung

- Lịch sử phát triển của kỹ thuật lập trình
- Hạn chế của kỹ thuật lập trình truyền thống
- Khái niệm lập trình hướng đối tượng
 - Đóng gói / Che dấu thông tin



Tài liệu tham khảo

- *Thinking in Java*, chapter 1, 2
- *Java how to program*, chapter 8



Mục tiêu của kỹ sư phần mềm

- Tạo ra sản phẩm tốt một cách có hiệu quả
- Nắm bắt được công nghệ
-



Phần mềm ngày càng lớn

- Một số hệ Unix chứa khoảng 4M dòng lệnh
- MS Windows chứa hàng chục triệu dòng lệnh
- Người dùng ngày càng đòi hỏi nhiều chức năng, đặc biệt là chức năng *thông minh*
- Phần mềm luôn cần được sửa đổi



Vì vậy

- Cần kiểm soát chi phí
 - Chi phí phát triển
 - Chi phí bảo trì
- Giải pháp chính là ***sử dụng lại***
 - Giảm chi phí và thời gian phát triển
 - Nâng cao chất lượng



Để sử dụng lại (mã nguồn)

- Cần dễ hiểu
- Được coi là chính xác
- Có giao diện rõ ràng
- *Không yêu cầu thay đổi khi sử dụng trong chương trình mới*



Các phương pháp lập trình

- Lập trình không có cấu trúc
- Lập trình có cấu trúc (lập trình thủ tục)
- Lập trình chức năng
- Lập trình logic
- Lập trình hướng đối tượng



Lập trình không có cấu trúc (non-structured programming)

- Là phương pháp xuất hiện đầu tiên
 - các ngôn ngữ như Assembly, Basic
 - sử dụng các biến tổng thể
 - lạm dụng lệnh GOTO
- Các nhược điểm
 - khó hiểu, khó bảo trì, hầu như không thể sử dụng lại
 - chất lượng kém
 - chi phí cao
 - không thể phát triển các ứng dụng lớn



Ví dụ

```
10    k = 1
20    gosub 100
30    if y > 120 goto 60
40    k = k+1
50    goto 20
60    print k, y
70    stop
100   y = 3*k*k + 7*k-3
110   return
```