**CÁCH THIẾT KẾ CÁC CHƯƠNG TRÌNH ĐẦU TIÊN**

**1. Bước thiết kế 1: Xác định mục tiêu của bạn**

Để viết một chương trình thành công, trước tiên bạn cần phải xác định mục tiêu của mình là gì. Lý tưởng nhất là bạn có thể nêu điều này trong một hoặc hai câu. Nó thường hữu ích khi thể hiện điều này như một kết quả mà người dùng phải đối mặt.

* Cho phép người dùng sắp xếp danh sách tên và số điện thoại liên lạc.
* Tạo danh sách khuyến nghị cổ phiếu cho các cổ phiếu có tổ tức cao
* Lập mô hình mất bao lâu để một quả bóng rơi từ tháp rơi xuống đất.

Mặc dù bước này có vẻ hiển nhiên, nhưng nó cũng rất quan trọng. Điều tồi tệ nhất bạn có thể làm là viết một chuowg trình không thực sự làm những gì bạn (hoặc xếp của bạn) muốn!

**2. Bước thiết kế 2: Xác định yêu cầu**

Mặc dù việc xác định vấn đề giúp bạn xác định được kết quả mong muốn nhưng nó vẫn còn mơ hồ. Bước tiếp theo là suy nghĩ về các yêu cầu.

Yêu cầu là một từ hoa mỹ cho cả những ràng buộc và giải pháp của bạn cần tuân theo (ví dụ: ngân sách, dòng thời gian, không gian, bộ nhớ v.v.), cũng như các khả năng mà chương trình phải thể hiện để đáp ứng nhu cầu của người dùng. Ví dụ:

* Số điện thoại nên được lưu lại để có thể gọi lại sau này.
* Các khuyến nghị cổ phiếu nên tận dụng dữ liệu giá lịch sử
* Người dùng có thể nhập chiều cao của tháp
* Chương trình sẽ đưa ra kết quả trong vòng 10 giây kể từ khi người dùng gửi yêu cầu của họ

**3. Thiết kế bước 3: Xác định các công cụ, mục tiêu và kế hoạch dự phòng của bạn.**

Khi bạn là một lặp trình viên có kinh nghiệm, có nhiều bước khác thường diễn ra tại thời điểm này, bao gồm:

* Xác định kiến trúc mục tiêu và / hoặc hệ điều hành mà chương trình của bạn sẽ chạy
* Xác định bộ công cụ bạn sẽ sử dụng
* Xác định xem bạn sẽ viết chương trình của mình một mình hay là một phần của nhóm.
* Xác định chiến lược thử nghiệm / phản hồi / phát hành của bạn
* Xác định cách bạn sẽ sao lưu code của mình.

Tuy nhiên, là một lập trình viên mới, câu trả lời cho những câu hỏi này thường rất đơn giản: Bạn đang viết một chương trình để sử dụng riêng, một mình, trên hệ thống của riêng bạn, sử dụng IDE bạn đã mua hoặc tải xuống và code của bạn có thể không được ai sử dụng nhưng bạn. Điều này làm cho mọi thứ trở nên dễ dàng.

Điều đó nói rằng, nếu bạn định làm việc trên bất kỳ thứ gì có độ phức tạp không tầm thường, bạn nên có kế hoạch sao lưu code của mình. Chỉ nén hoặ sao lưu thư mục vào một vị trí khác trên máy của bạn là không đù (mặ dù điều này tốt hơn là không có gì). Nếu hệ thống của bạn gặp sự cố, bạn sẽ mất mọi thứ. Một chiến lược sao lưu tốt bao gồm việc xóa hoàn toàn bản sao code khỏi hệ thống của bạn. Có rất nhiều cách dễ dàng để thực hiện việc này: Zip nó và gửi nó qua email cho chính bạn, sao chép nó vào Dropbox hoặc một dịch vụ đám mây khác, FTP nó vào một máy khác, sao chép nó vào một máy khác trên mạng cục bộ của bạn hoặc sử dụng hệ thống kiểm soát phiên bản ở trên một máy khác hoặc trong đám mây (ví dụ: github). Hệ thống kiểm soát phiên bản có thêm lợi thế là không chỉ có thể khôi phục các tệp của bạn mà còn có thể khôi phục chúng về phiên bản trước đó.

**4. Bước thiết kế 4: Chia các bài toán khó thành các bài toán dễ**

**5. Bước thiết kế 5: Tìm ra chuỗi sự kiện**

Tại thời điểm này, chúng ta đã sẵn sàng triển khai code.