

BÀI 14**THỰC HÀNH VỀ THƯ VIỆN CÁC HÀM TỰ ĐỊNH NGHĨA**

Học xong bài này, em sẽ:

- ✓ Tạo được một thư viện nhỏ.
- ✓ Viết được chương trình sử dụng thư viện vừa tạo ra.

Nhiệm vụ 1. Tổ chức thư viện các hàm người lập trình tự viết

Yêu cầu:

Tạo được thư viện **myLib** gồm một số hàm thực hiện các thuật toán sắp xếp, tìm kiếm do ta tự viết trong các bài học trước.

Hướng dẫn thực hiện:

- Tạo thư mục dự án **myPrj** chứa thư mục con **myLib** là thư viện các hàm ta tự viết.
- Trong thư mục **myLib**, tạo một tệp rỗng, có tên “_init_.py”. Nếu có tệp này, Python biết đây sẽ là một gói chứa một số tệp mã nguồn.
- Trong thư mục **myLib**, tạo hai tệp “mySort.py” và “mySearch.py”. Sao chép mã lệnh của các hàm thực hiện sắp xếp, tìm kiếm mà ta đã viết thành công vào hai tệp tương ứng. Mỗi hàm bắt đầu từ câu lệnh **def** định nghĩa hàm đó cho đến hết toàn bộ cả hàm.
- Thử sử dụng **myLib** như một thư viện: Viết tệp chương trình “demoLib.py” bắt đầu với 2 dòng lệnh **import** khai báo sử dụng thư viện. Tham khảo mã lệnh trong *Hình 1*; Chú ý thay dấu “...” bằng một dãy số và dấu “?” bằng một số cụ thể.



```
File Edit Format Run Options Window Help
1 from myLib import mySort
2 from myLib import mySearch
3 dayso = [...] #dayso là một dãy số
4 x =? #x là một số
5 mySort.sxNoibot(dayso)
6 print(dayso)
7 print(mySearch.tkNhiPhan(dayso, x))
```

Hình 1. Minh họa cách sử dụng thư viện do người lập trình tạo ra

Nhiệm vụ 2. Sử dụng thư viện vừa tạo ra

Yêu cầu:

Sử dụng thư viện vừa tạo ra để viết phiên bản mới cho chương trình chính của Bài tập lớn.

Hướng dẫn thực hiện:

Mở văn bản chương trình sản phẩm SP#1; làm các việc sau:

- Bổ sung hai dòng khai báo sử dụng thư viện **myLib**.
- rà soát từ đầu văn bản chương trình và cắt bỏ phần mã nguồn của các hàm đã có trong thư viện **myLib**.
- Nếu phát hiện còn hàm ta tự viết để thực hiện sắp xếp, tìm kiếm được sử dụng trong chương trình mà chưa có trong thư viện **myLib** thì cắt dán mã nguồn vào **myLib**.
- Chạy thử chương trình.



Chỉnh sửa các chi tiết để có thể áp dụng chương trình phân tích kết quả học tập theo các quy định của trường em.

BÀI TÌM HIỂU THÊM

KHAI THÁC THƯ VIỆN CỦA PYTHON

Python có nhiều mô đun. Bộ cài đặt Python cho nền tảng Windows chỉ tích hợp sẵn các mô đun tiêu chuẩn được viết bằng ngôn ngữ C. Ngoài các thư viện tiêu chuẩn, còn nhiều gói khác bên ngoài, các mô đun lẻ và các khung phát triển ứng dụng. Với người lập trình, cần phân biệt ba loại mô đun sau:

1) Các hàm tích hợp sẵn dùng ngay, không cần khai báo **import** ở đầu chương trình. Python 3 có gần 70 hàm có thể dùng ngay. Ta đã quen với một số trong đó như: **print()**, **input()**, **len()**, **sum()**, **max()**, **min()**, ... hay các hàm chuyển đổi kiểu dữ liệu số **int()**, **float()** chuyển đổi kiểu dữ liệu có cấu trúc **str()**, **list()**, **tuple()**, **set()**.

2) Các mô đun tiêu chuẩn đi kèm bản cài đặt Python, chỉ cần khai báo sử dụng bằng câu lệnh **import** là có thể dùng các hàm có trong thư viện. Hãy mở cửa sổ trình thông dịch (Python shell), thử chạy lệnh **help('modules')** và xem kết quả.

3) Các mô đun khác có thể gọi chung là các mô đun ngoài phải cài đặt gói bổ sung, sau đó mới có thể dùng và cách dùng giống như một mô đun tích hợp sẵn.

Một số thư viện ngoài chưa tích hợp sẵn là: **Matplotlib** để vẽ đồ thị hàm số, vẽ các biểu đồ; **PyGame** để làm việc với dữ liệu đồ họa, âm thanh trong trò chơi điện tử; **Sqlite3** để làm việc với CSDL qua ngôn ngữ truy vấn SQL.