

SAU BÀI HỌC NÀY EM SẼ:

- Biết và vận dụng được cách thiết kế chương trình theo mô đun cho một số bài toán cụ thể.
- Nhận biết được lợi ích của phương pháp thiết kế chương trình theo mô đun.



Em được giao việc quản lí cho cửa hàng bán thực phẩm của gia đình. Hằng ngày, em phải nhập danh sách các mặt hàng và doanh số bán hàng. Cuối ngày, em cần báo cáo ba mặt hàng có doanh số cao nhất và ba mặt hàng có doanh số thấp nhất trong ngày.

Các mặt hàng mới nhập sẽ được lưu trong tệp văn bản bao gồm nhiều dòng, mỗi dòng là tên mặt hàng và doanh số được ngăn cách bằng dấu phẩy. Tên tệp đầu vào là **Data.inp**.

Báo cáo cuối ngày là tệp văn bản "Data.out" gồm 6 dòng, ba dòng đầu tiên là tên ba mặt hàng có doanh số cao nhất, ba dòng cuối là tên ba mặt hàng có doanh số thấp nhất, cần có cả tên hàng và doanh số.

Data.inp	Data.out
Cà rốt, 1350	Gạo thường 23124
Khoai tây, 4400	Gạo ST25 19221
Hành tươi, 1367.5	Cam 9800
Bắp cải, 3400	Khoai lang 2100
Cà chua, 5609	Hành tươi 1367.5
Khoai lang, 2100	Cà rốt 1350
Gạo ST25, 19221	
Gạo thường, 23124	
Cam, 9800	
Chuối, 7823	

Em sẽ thiết kế chương trình như thế nào? Trao đổi với bạn về cách thiết kế chương trình sao cho hợp lí nhất.

1. THIẾT KẾ CHƯƠNG TRÌNH THEO MÔ ĐUN

Hoạt động 1 Thiết kế chương trình theo mô đun

Thực hiện các bước thiết kế giải bài toán trên theo phương pháp làm mịn dần, trao đổi và thảo luận để biết được cách thiết kế chương trình theo mô đun.



Bước 1. Thiết kế chung

Từ yêu cầu của bài toán, ta thấy có thể chia bài toán đã cho thành ba công việc chính, các công việc này tương đối độc lập với nhau.

1) Công việc nhập dữ liệu: Dữ liệu được nhập vào tệp `Data.inp` và được đọc để đưa vào chương trình.

2) Công việc xử lý dữ liệu: Các công việc chuẩn bị tính toán dữ liệu theo yêu cầu của bài toán.

3) Báo cáo, đưa dữ liệu ra theo yêu cầu.

Bước 2. Thiết lập công việc nhập dữ liệu

Thiết lập hàm `NhapDL(fin)` có tính năng đọc dữ liệu từ tệp `Data.inp` và đưa vào hai mảng `P` (mặt hàng) và `S` (doanh số). Hàm được mô tả tổng quát đọc dữ liệu từ tệp `fin`.

```

1 def NhapDL(fin):
2     f = open(fin, encoding="UTF-8")
3     P = []
4     S = []
5     for line in f:
6         A = line.split(",")
7         P.append(A[0])
8         S.append(float(A[1]))
9     f.close()
10    return P, S

```

Bước 3. Thiết lập công việc xử lý dữ liệu

Yêu cầu chính của báo cáo là lấy được thông tin của các mặt hàng có doanh số cao nhất và thấp nhất. Do vậy công việc xử lý dữ liệu chính là cần sắp xếp lại các mảng `P`, `S` theo thứ tự tăng dần của `S` (doanh số). Hàm `Sapxep(A,B)` được thiết kế tổng quát sẽ sắp xếp lại hai mảng `A`, `B` nhưng theo thứ tự tăng dần của `A`.

```

1 def Sapxep(A,B):
2     n = len(A)
3     for i in range(1,n):
4         j = i
5         while j > 0 and A[j] < A[j-1]:
6             A[j], A[j-1] = A[j-1], A[j]
7             B[j], B[j-1] = B[j-1], B[j]
8             j = j-1

```

Bước 4. Thiết lập báo cáo, đưa dữ liệu ra

Cuối cùng là chức năng đưa dữ liệu ra tệp `Data.out` sẽ được mô tả trong hàm `GhiDL(P,S,fout)`. Hàm này lấy dữ liệu từ các mảng `P`, `S` và đưa dữ liệu ra tệp `fout`.

```

1 def GhiDL(P,S,fout):
2     f = open(fout, "w", encoding="UTF-8")
3     n = len(P)
4     for i in range(n-1, n-4, -1):

```

```

5     print(P[i],S[i],file = f)
6     for i in range(2,-1,-1):
7         print(P[i],S[i],file = f)
8     f.close()

```

Sau các bước trên, chúng ta đã thực hiện xong việc thiết kế các công việc chính được đặt ra ban đầu. Mỗi công việc đó được viết thành một hàm riêng biệt, độc lập với nhau. Các hàm này được gọi là các mô đun con của chương trình.

Chương trình chính sử dụng các chương trình con trên được mô tả đơn giản như sau:

```

1 fin = "Data.inp"
2 fout = "Data.out"
3 P,S = NhapDL(fin)
4 Sapxep(S,P)
5 GhiDL(P,S,fout)

```

Phương pháp thiết kế chương trình như trên được gọi là thiết kế theo mô đun. Mỗi mô đun thường là các chức năng độc lập, riêng biệt theo yêu cầu của chương trình. Tùy thuộc vào từng bài toán và quan điểm thiết kế có thể tạo các mô đun này theo nhiều cách khác nhau.

Phương pháp thiết kế chương trình theo mô đun sẽ tách bài toán lớn thành các bài toán nhỏ hơn, hay thành các mô đun, tương đối độc lập với nhau, sau đó tiến hành thiết kế thuật toán và chương trình cho từng mô đun con. Mỗi mô đun có thể là một số hàm hoặc thủ tục độc lập. Chương trình chính là một bản ghép nối các hàm và thủ tục con.



1. Chương trình trên được thiết kế có bao nhiêu mô đun?
2. Các mô đun của chương trình trên có quan hệ với nhau như thế nào?

2. LỢI ÍCH CỦA PHƯƠNG PHÁP THIẾT KẾ THEO MÔ ĐUN

Hoạt động 2 Tìm hiểu lợi ích của phương pháp thiết kế theo mô đun

Với chương trình đã có trong Hoạt động 1, em sẽ làm gì nếu có các yêu cầu bổ sung như sau:

1. Yêu cầu thay đổi thông tin trong báo cáo: Ghi hai mặt hàng có doanh số cao nhất và bốn mặt hàng có doanh số thấp nhất.
2. Cập nhật, bổ sung các mặt hàng mới và doanh số trong ngày.
3. Yêu cầu làm thêm một báo cáo trong đó ghi doanh số bán trung bình trong ngày và danh sách các mặt hàng có doanh số lớn hơn doanh số trung bình này, kết quả đưa ra tệp [Data2.out](#).

Em có nhận xét gì về việc thực hiện các công việc bổ sung này?



Công việc bổ sung 1

Đây là công việc cần nâng cấp hàm GhiDL() và độc lập với các mô đun khác. Việc nâng cấp này rất đơn giản và được mô tả trong chương trình sau. Thay đổi chỉ ở hai lệnh tại dòng 4 và 6.

```

1 def GhiDL(P,S,fout):
2     f = open(fout,"w",encoding="UTF-8")
3     n = len(P)
4     for i in range(n-1,n-3,-1):
5         print(P[i],S[i],file = f)
6     for i in range(3,-1,-1):
7         print(P[i],S[i],file = f)
8     f.close()

```

So với mô đun gốc, bản nâng cấp này chỉ cần sửa hai dòng lệnh 4 và 6.

Công việc bổ sung 2

Công việc này rất đơn giản là mở tệp **Data.inp** và bổ sung thêm thông tin các mặt hàng mới và doanh số, không cần phải sửa chương trình.

Công việc bổ sung 3

Công việc này mới và độc lập hoàn toàn với các công việc khác của bài toán, nên có thể tách thành một hàm (mô đun) độc lập, có thể giao cho một nhóm khác thực hiện. Hàm mới sẽ đặt tên là BC2() và có nội dung đơn giản như sau:

```

1 def BC2(P,S,fout):
2     f = open(fout,"w",encoding="UTF-8")
3     n = len(P)
4     average = sum(S)/n
5     print("Doanh số trung bình:",average,file = f)
6     for i in range(n):
7         if S[i] > average:
8             print(P[i],S[i],file = f)
9     f.close()

```

Trong chương trình chính cần bổ sung lệnh sau đây để thực hiện báo cáo mới này.

```
BC2(P,S,"Data2.out")
```

Thiết kế thuật toán và chương trình theo mô đun có các ưu điểm sau:

- Chương trình ngắn gọn, sáng sửa, dễ hiểu.
- Các mô đun được thiết lập một lần và sử dụng nhiều lần.
- Dễ dàng nâng cấp, thay đổi, chỉnh sửa mà không mất công sửa lại toàn bộ chương trình.
- Dễ dàng bổ sung các mô đun mới.
- Có thể chia sẻ trong môi trường làm việc nhóm, ví dụ phân công mỗi người một công việc độc lập.



1. Phân loại các công việc bổ sung trên vào ba loại sau:

- Công việc mới hoàn toàn.
- Công việc nâng cấp một mô đun cũ.
- Công việc không liên quan đến thuật toán và lập trình.

2. Công việc sau đây, nếu có, sẽ thuộc nhóm công việc nào?

Nhập một giá trị số nào đó, ví dụ K, cần tìm trong danh sách các mặt hàng có doanh số xấp xỉ K (hơn kém nhau không quá hằng số $C = 1$).



LUYỆN TẬP

1. Nếu công việc bổ sung 3 có thêm yêu cầu in ra số lượng mặt hàng đã bán trong ngày thì cần thêm hay sửa lệnh nào của chương trình BC2().
2. Viết thêm một chương trình cho công việc bổ sung 4 như sau: Cần in ra danh sách 1/3 số mặt hàng có doanh số thấp nhất trong ngày.



VẬN DỤNG

1. Thiết lập chương trình cho công việc thường làm vào cuối giờ bán hàng: Cho trước số K (một doanh số giả định), cần tìm ra mặt hàng có doanh số nhỏ hơn K nhưng gần với K nhất. Bài toán này có thể sử dụng thuật toán tìm kiếm nào để giải?
2. Thiết kế thuật toán và chương trình theo mô đun cho bài toán sau.

Một công ty du lịch có n địa điểm tham quan được đánh số theo thứ tự 0, 1, 2, ..., $n - 1$. Công ty này luôn tổ chức các tour du lịch đi lần lượt từ vị trí 0, 1, 2, ... và kết thúc tại vị trí cuối cùng $n - 1$. Để thuận tiện cho việc quảng bá du lịch công ty đã lấy ý kiến khách hàng đánh giá bằng điểm số cho từng địa điểm du lịch trên, các đánh giá có thể là các số dương, 0 hoặc số âm bất kì. Số lớn hơn 0 biểu thị đánh giá tốt, số nhỏ hơn 0 biểu thị đánh giá xấu về địa điểm đó. Mỗi khách hàng sẽ gửi lên công ty du lịch bảng đánh giá của mình, được biểu thị bằng một dãy n số, ví dụ như sau:

1, -3, 4, 10, 0, -5, -8, 2, -1, 7, 2

Công ty du lịch hứa sẽ tổ chức một tour riêng cho mỗi khách hàng, bảo đảm sự hài lòng cao nhất của khách hàng. Tour du lịch riêng của khách hàng sẽ là một dãy các vị trí liên tục các địa điểm, ví dụ từ vị trí i đến j, tức là xuất phát từ i, khách hàng sẽ lần lượt đi qua các vị trí i, $i + 1$, $i + 2$, ... và kết thúc tại j. Công ty du lịch bảo đảm rằng tổng các đánh giá của khách hàng trên tour riêng của mình là lớn nhất.

Em hãy giúp công ty du lịch thiết lập tour du lịch tối ưu cho khách hàng nếu biết trước các đánh giá của khách hàng đó.