

Bài 4.7: Phương thức đệ quy

- ✓ Định nghĩa đệ quy
- ✓ Đặc điểm của đệ quy
- ✓ Ý nghĩa sử dụng
- ✓ Ví dụ và bài tập thực hành

Định nghĩa

- ✓ Một phương thức đệ quy là phương thức gọi lại chính nó trực tiếp hoặc gián tiếp qua các phương thức khác.
- ✓ Suy ra, một cách làm gọi là đệ quy nếu trong các bước thực hiện để giải quyết vấn đề lại chứa chính bản thân nó.
- ✓ Đệ quy là chủ đề phức tạp, khóa học này chỉ tiếp cận đệ quy ở góc độ đơn giản nhất.
- ✓ Chi tiết hơn về đệ quy bạn có thể tìm hiểu thêm trên internet hoặc trong các khóa học cấu trúc dữ liệu và giải thuật.

Đặc điểm của đệ quy

- ✓ Chỉ áp dụng với các phương thức.
- ✓ Các phương thức đệ quy thường chỉ biết giải quyết vấn đề ở trường hợp cơ bản nhất.
- ✓ Trường hợp cơ bản đó ta gọi là điểm dừng, trường hợp cơ sở.
- ✓ Với các lời gọi đệ quy trùng với trường hợp này, phương thức chỉ return giá trị điểm dừng tương ứng.
- ✓ Khi phương thức đệ quy được gọi ở các trường hợp phức tạp hơn, vấn đề sẽ được chia làm 2 phần:
 - ✓ Phần 1 phương thức biết cách xử lý.
 - ✓ Phần 2 phương thức không biết cách xử lý.
- ✓ Ở phần thứ 2, vấn đề sẽ được giải quyết bằng cách gọi lại chính nó. Do vấn đề vẫn là vấn đề cũ nhưng quy mô nhỏ hơn.
- ✓ Hành động gọi lại chính nó gọi là lời gọi đệ quy, bước đệ quy.

Đặc điểm của đệ quy

- ✓ Khi thực hiện lời gọi đệ quy, chương trình sẽ phải chờ đợi. Tới khi tất cả các lời gọi đệ quy kế tiếp thực hiện xong và trả về kết quả hoặc kết thúc.
- ✓ Để đảm bảo phương thức đệ quy có thể kết thúc được, sau mỗi lời gọi đệ quy, vấn đề đang giải quyết phải trở nên nhỏ dần và tiến đến trường hợp cơ sở.
- ✓ Đệ quy cũng xảy ra lặp vô hạn, tiêu tốn tài nguyên CPU, bộ nhớ. Do đó cần hết sức cẩn trọng khi sử dụng.

Ý nghĩa sử dụng

- ✓ Thường dùng để giải quyết các vấn đề có tính truy hồi.
- ✓ Vấn đề nào có thể giải bằng đệ quy thì có thể giải bằng vòng lặp.
- ✓ Ưu điểm của đệ quy là ngắn gọn, tường minh.
- ✓ Nhược điểm là tốn tài nguyên: CPU, bộ nhớ, thời gian.
- ✓ Nếu có thể khử đệ quy bằng vòng lặp, hãy thực hiện điều đó.
- ✓ Luôn đảm bảo rằng vấn đề phải trở nên nhỏ hơn sau mỗi lời gọi đệ quy để tránh đệ quy vô hạn.

Ví dụ minh họa số 1

✓ Ví dụ 1: tính $n!$

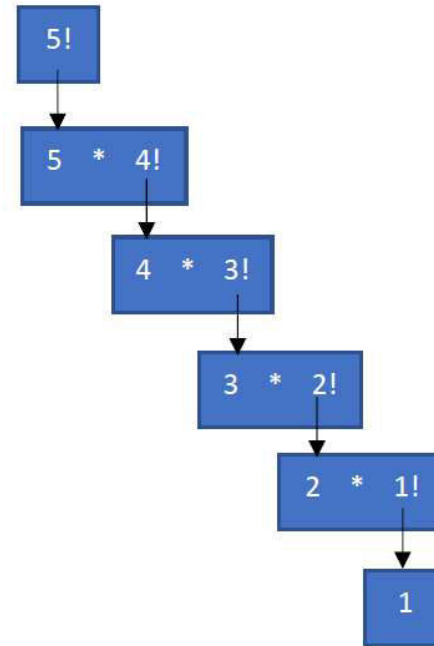
✓ $5! = 5 * 4 * 3 * 2 * 1$

✓ $5! = 5 * (4 * 3 * 2 * 1)$

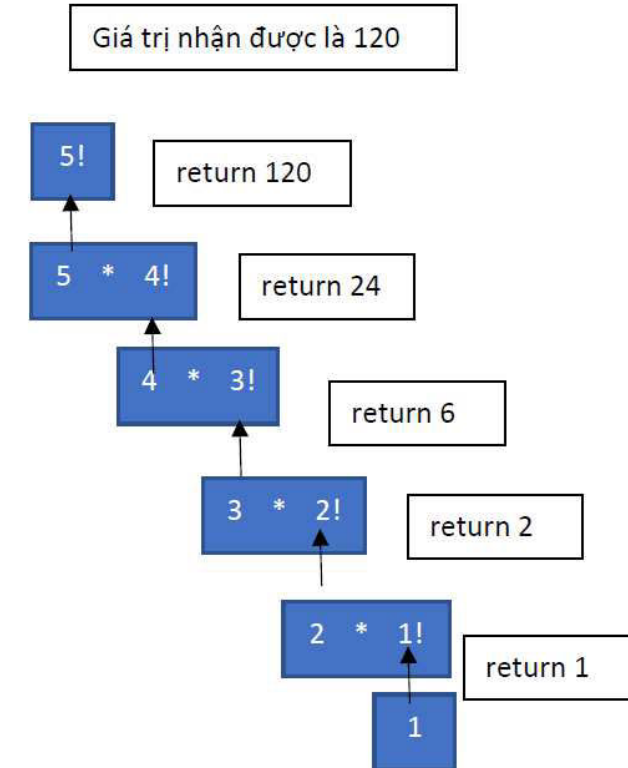
✓ $5! = 5 * 4!$

✓ Vậy ta có $n! = n * (n-1)!$

Lời gọi:

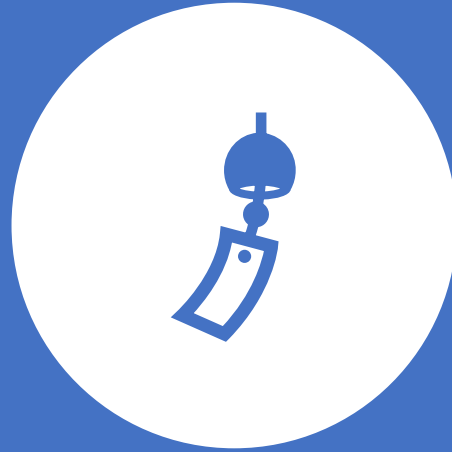


Giá trị trả về từ mỗi lời gọi đệ quy



Ví dụ minh họa số 2

- ✓ Ví dụ 2: tính tổng $S = 1 + 2 + \dots + n$:
 - ✓ $S = n + n-1 + \dots + 2 + 1$
 - ✓ $S = n + (n-1 + \dots + 2 + 1)$
 - ✓ $S = (n + n-1) + (n-2 + \dots + 2 + 1)$
- ✓ Vậy $S = n + S_{n-1} = (n + n-1) + S_{n-2}$. Trong đó S_{n-1} là tổng từ 1 đến $n-1$. Tương tự cho các S_k khác.



Nội dung tiếp theo

Các cấu trúc dữ liệu cơ bản