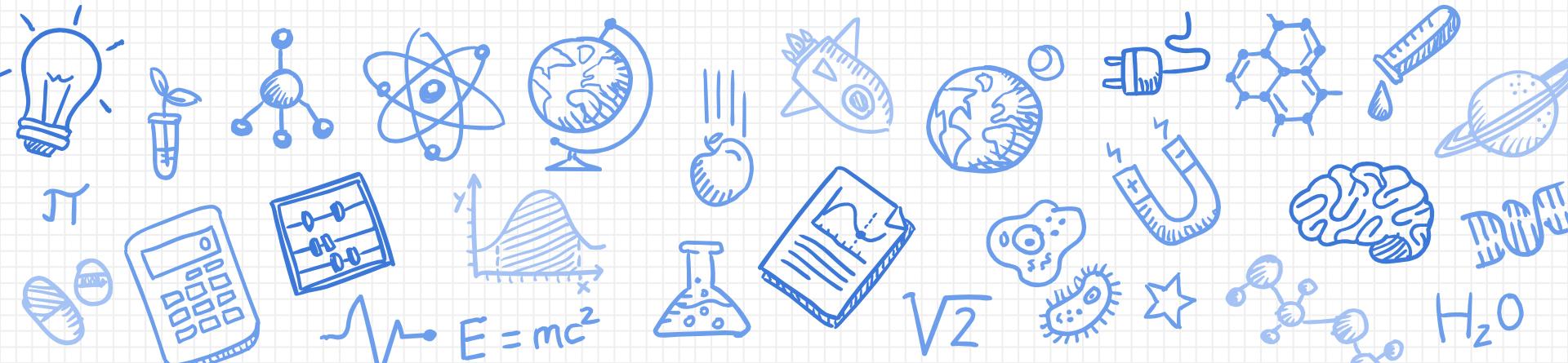
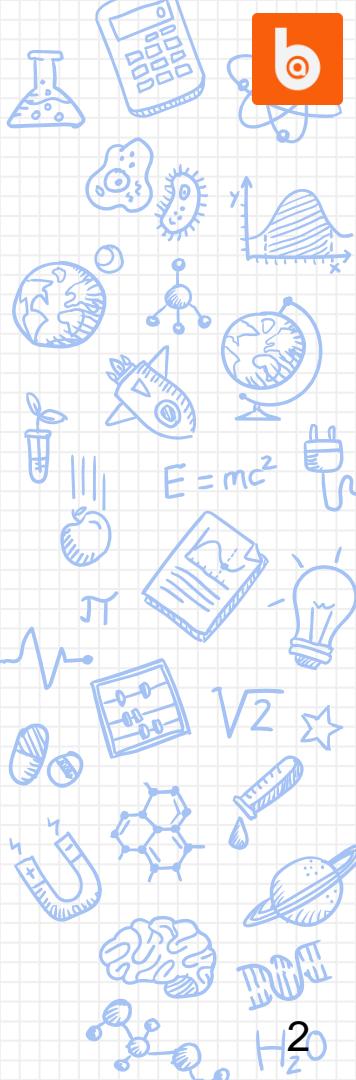


Các khái niệm mở đầu

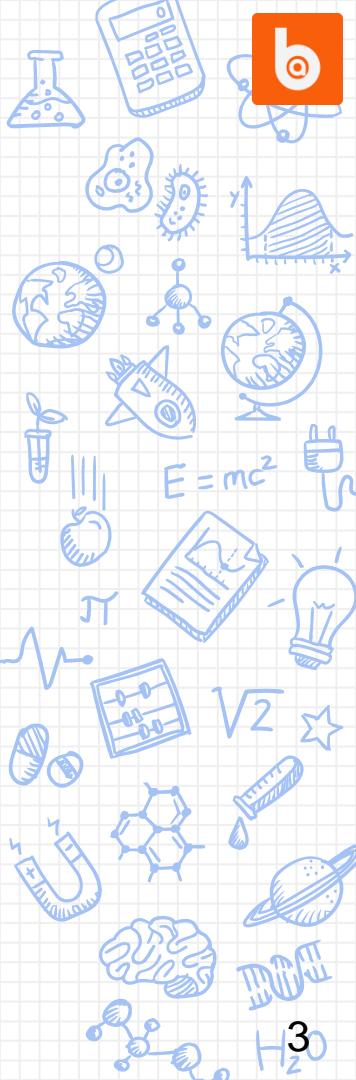
<https://braniumacademy.net>





Nội dung bài học

- Cấu trúc dữ liệu là gì?
- Sử dụng cấu trúc dữ liệu có lợi ích gì?
- Có những cấu trúc dữ liệu nào?
- Giải thuật là gì?
- Có những loại giải thuật nào?
- Mã giả là gì?
- Độ phức tạp thuật toán là gì?



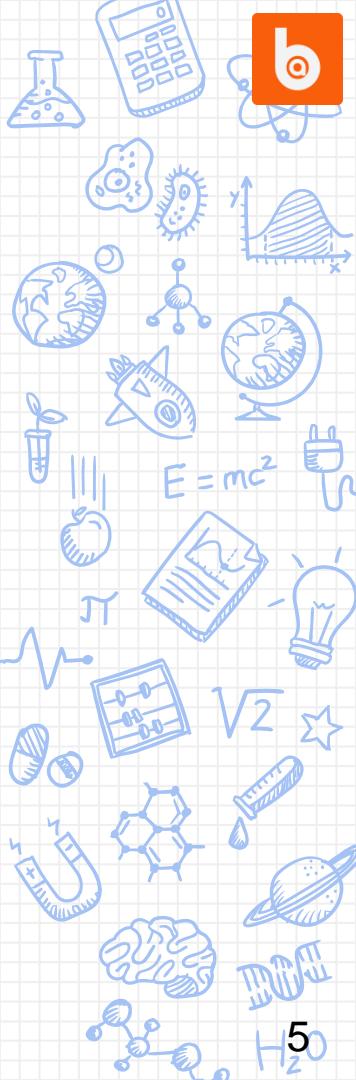
Cấu trúc dữ liệu là gì?

- Cấu trúc dữ liệu là cách thức tổ chức, lưu trữ dữ liệu để việc sử dụng nó đạt hiệu quả cao nhất.
- Tên cấu trúc dữ liệu chỉ ra cách thức nó tổ chức dữ liệu trong bộ nhớ. Có nhiều cách tổ chức dữ liệu khác nhau như: mảng, cây nhị phân...
- Cấu trúc dữ liệu không phải là bất cứ ngôn ngữ lập trình nào mà nó là tập hợp các thuật toán từ đó ta có thể sử dụng trong bất cứ ngôn ngữ nào nhằm xây dựng cấu trúc dữ liệu trong bộ nhớ.



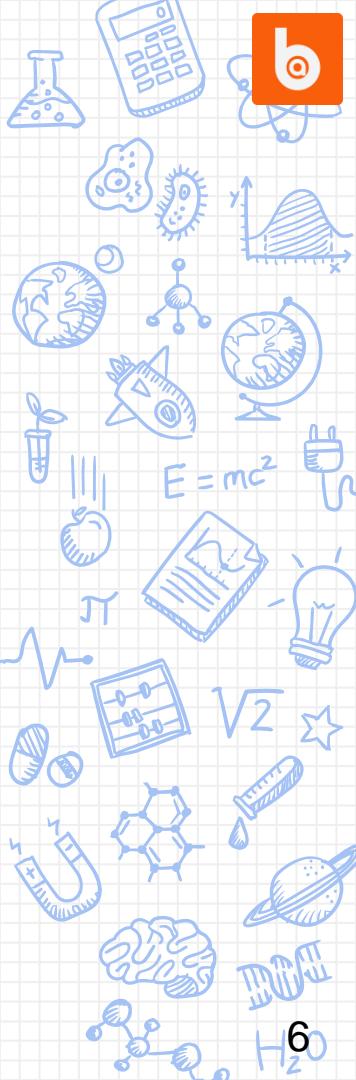
Cấu trúc dữ liệu là gì?

- ❑ Abstract data types là tập hợp các quy tắc để tổ chức dữ liệu.
 - ❑ Có hai kiểu cấu trúc dữ liệu: nguyên thủy và dẫn xuất.



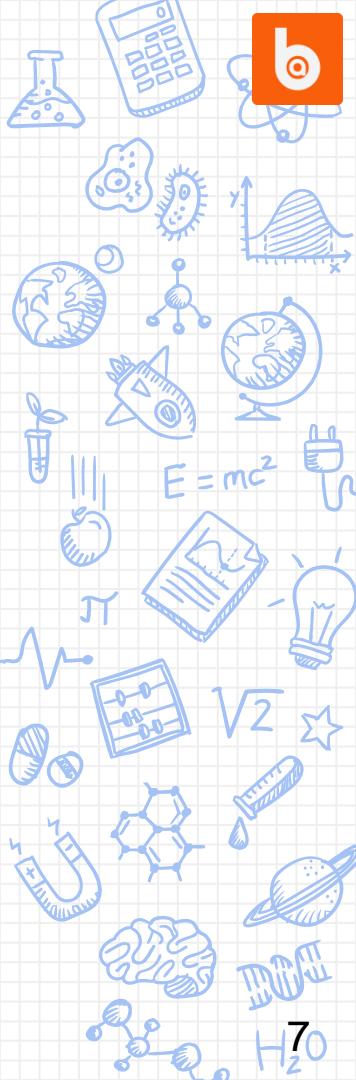
Các loại cấu trúc dữ liệu?

- Kiểu cấu trúc dữ liệu nguyên thủy là các kiểu dữ liệu nguyên thủy: int, float, double, char, ... chúng có thể lưu trữ các giá trị đơn.
- Kiểu cấu trúc dữ liệu dẫn xuất gồm cấu trúc dữ liệu tuyến tính và phi tuyến tính.
- Kiểu tuyến tính: mảng, danh sách liên kết, stack, queue.
- Kiểu phi tuyến tính: cây, bảng băm, đồ thị...



Lợi ích của cấu trúc dữ liệu?

- Tính hiệu quả:** nếu chọn đúng cấu trúc dữ liệu để triển khai, nó sẽ làm cho chương trình rất hiệu quả về không gian và thời gian.
- Tính tái sử dụng:** cấu trúc dữ liệu cho phép tái sử dụng, do đó nhiều chương trình có thể sử dụng cùng một cấu trúc dữ liệu.
- Tính trừu tượng:** các cấu trúc dữ liệu chứa cấp độ trừu tượng do vậy bên ngoài không thể biết hoạt động ở bên trong cấu trúc dữ liệu mà chỉ nhìn thấy giao diện public ra bên ngoài.



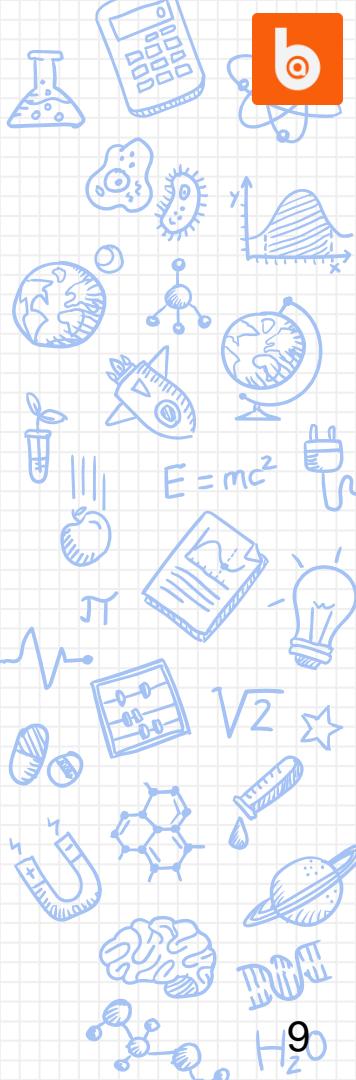
Giải thuật là gì?

- ❑ Theo thuật ngữ lập trình máy tính, thuật toán là một tập các hướng dẫn được xác định rõ ràng để giải quyết một vấn đề cụ thể. Nó cần một tập hợp các đầu vào và tạo ra một đầu ra mong muốn.
- ❑ Đặc điểm của một thuật toán tốt:
 - ✓ Đầu vào và đầu ra phải được xác định chính xác.
 - ✓ Các bước trong thuật toán phải rõ ràng, mạch lạc.
 - ✓ Thuật toán nên là cách giải quyết hiệu quả nhất trong số các cách cùng được sử dụng để giải quyết một vấn đề.



Giải thuật là gì?

- ❑ Đặc điểm của một thuật toán tốt:
 - ✓ Một thuật toán không nên chứa các đoạn mã máy tính. Thay vào đó nên được viết theo cách sao cho có thể sử dụng được trong các ngôn ngữ lập trình khác nhau.



Ví dụ thuật toán cộng hai số

Bước 1: Bắt đầu.

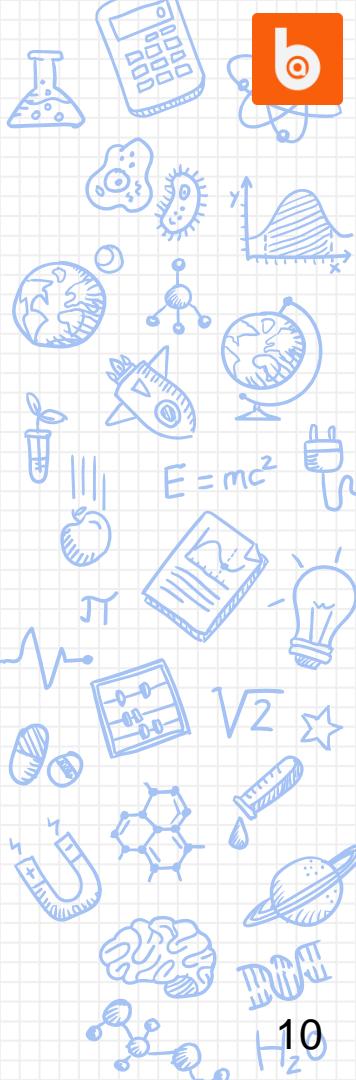
Bước 2: Khai báo biến num1, num2 và sum.

Bước 3: Đọc vào giá trị cho num1, num2.

Bước 4: Cộng num1 và num2 sau đó gán kết quả cho sum
sum <- num1 + num2.

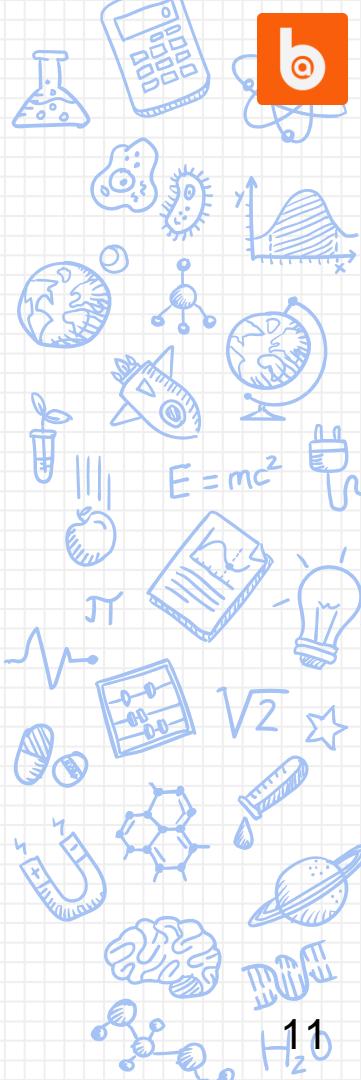
Bước 5: Hiện kết quả lên màn hình.

Bước 6: Kết thúc.



Có những loại giải thuật nào?

- Thuật toán Brute Force.
- Thuật toán đệ quy.
- Thuật toán chia để trị.
- Thuật toán quy hoạch động.
- Thuật toán tham lam.
- Thuật toán quay lui.

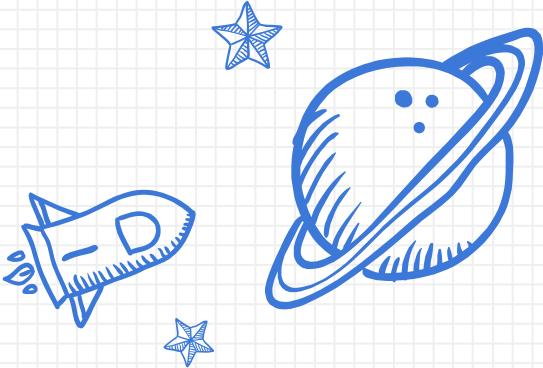


Mã giả là gì?

- ❑ Mã giả là bản mô tả đơn giản về các bước cần thực hiện trong một thuật toán hoặc chương trình bằng ngôn ngữ tự nhiên.
- ❑ Mục đích của việc sử dụng mã giả là để mọi người dễ hiểu hơn so với mã thật của ngôn ngữ lập trình.
- ❑ Mã giả không chạy được, chỉ dùng để con người đọc hiểu.
- ❑ Sử dụng mã giả giúp cho lập trình viên chỉ tập trung vào phần thuật toán của quá trình lập trình.

Độ phức tạp thuật toán là gì?

- ❑ Là thước đo lượng thời gian mà một thuật toán cần để xử lý xong một bài toán với đầu vào kích thước n.
- ❑ Độ phức tạp thuật toán được tính theo tiêm cận khi n tiến đến vô cùng.
- ❑ Phân tích độ phức tạp thuật toán hữu ích khi so sánh các thuật toán với nhau hoặc khi tìm kiếm các cải tiến của thuật toán.
- ❑ Độ phức tạp của thuật toán còn được gọi là độ phức tạp hoặc thời gian chạy.



Tiếp theo

Mảng 1 chiều