

Dò bài cũ

- 1. Tính trừu tượng là gì? Và được thể hiện thông qua đâu?**
- 2. Các tính chất của tính trừu tượng**
- 3. Bốn tính chất của OOP**

Tính trừu tượng (Abstraction)

- Là khả năng **ẩn các chi tiết** xử lý, chỉ **hiển thị tính năng/kết quả** (tên phương thức).
- Chỉ quan tâm đến **kết quả** đạt được, **không** quan tâm cách thực hiện/implement.
- Trong **Java** tính **trừu tượng** được thể hiện qua **abstract class** và **interface**.

Các tính chất của abstract

- **Không thể tạo** được **đối tượng**.
- **Lớp abstract** có thể **chứa thuộc tính** và **method bình thường**.
- Lớp **abstract** vẫn có thể **extend** từ lớp **abstract** khác. (tiện cho việc phân cấp kế thừa)
- Một **lớp chứa method abstract** thì bắt buộc lớp đó phải là **lớp abstract** (ngược lại thì không).
- **Abstract không thể** đi chung với **final**.

Các tính chất của interface

- **Không thể tạo** được **đối tượng**.
- Các **method** toàn bộ là **public abstract** (cần lưu ý đối với java 8+).
- Các **trường** trong interface đều là **public, static** và **final**.
- **Không** tạo được **constructor**.
- Có thể **extend** từ **một** nhiều hoặc **nhiều interface** khác.

NỘI DUNG CHÍNH

1

Java Collection Framework

2

Phân biệt List, Queue, Set

3

Trình bày về ArrayList và LinkedList

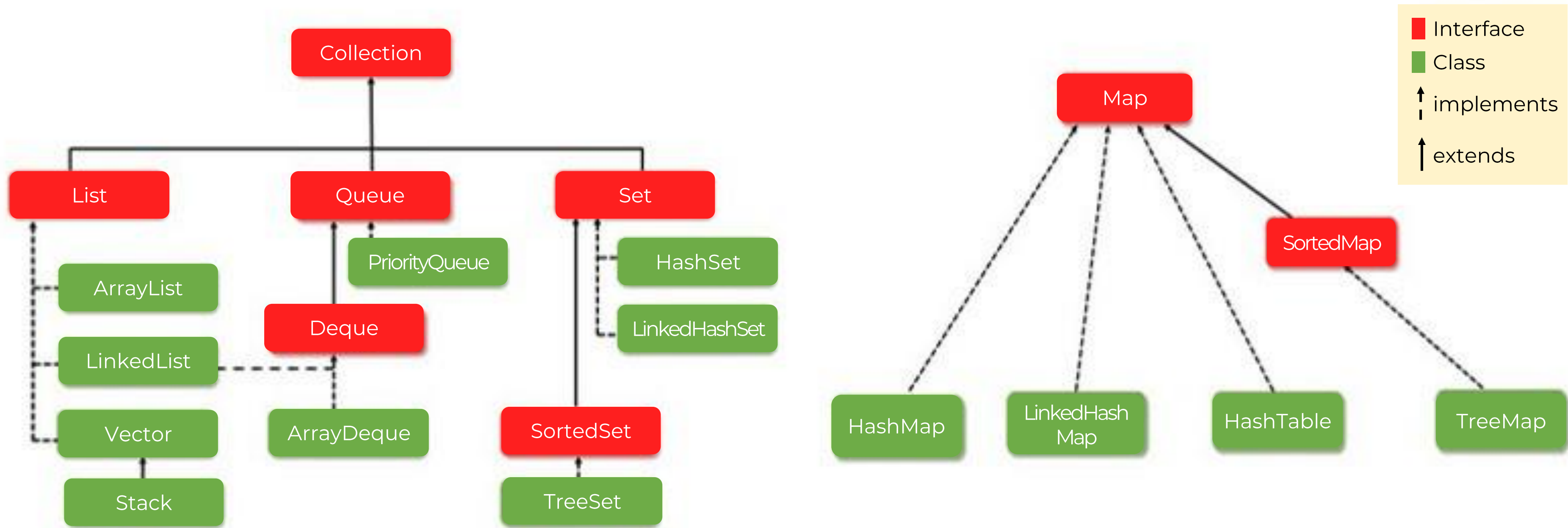
4

LinkedList

Java Collection Framework

- Là một **tập hợp** các **lớp** và **interface**
- Sử dụng để **lưu trữ** và **xử lý dữ liệu** theo các cấu trúc dữ liệu phổ biến như danh sách, hàng đợi

Java Collection Framework



List, Queue, Set

▪ List

- Là một **danh sách** có thứ tự của các phần tử.
- Các phần tử trong List **có thể trùng lặp**

▪ Queue

- Là một **hàng đợi** (danh sách theo thứ tự đợi)
- Thêm và lấy theo quy tắc "**First-In-First-Out**" (FIFO).

▪ Set

- Là một **tập hợp** các phần tử **không có thứ tự**
- **Không** chứa các phần tử **trùng lặp**

ArrayList và LinkedList

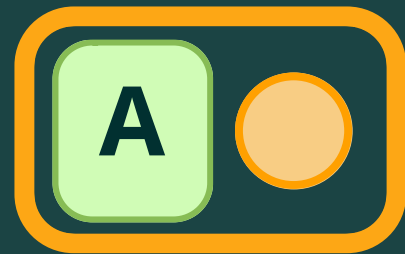
■ ArrayList

- Một mảng có thể **co/giãn** được vùng nhớ
- Đặc điểm
 - **Truy cập: Nhanh**
 - **Thêm/xóa: Chậm**

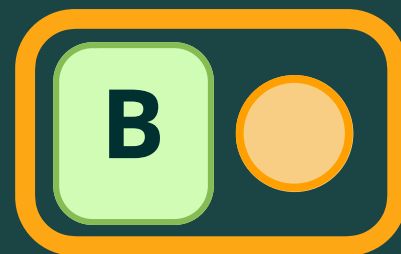
■ LinkedList

- Sử dụng một **danh sách liên kết** để lưu trữ
- Đặc điểm
 - **Truy cập: Chậm**
 - **Thêm/xóa: Nhanh**

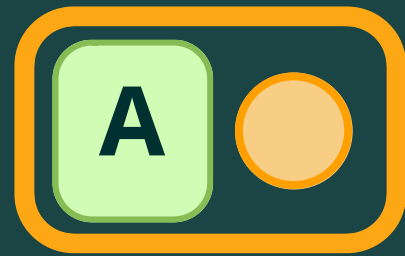
LinkedList



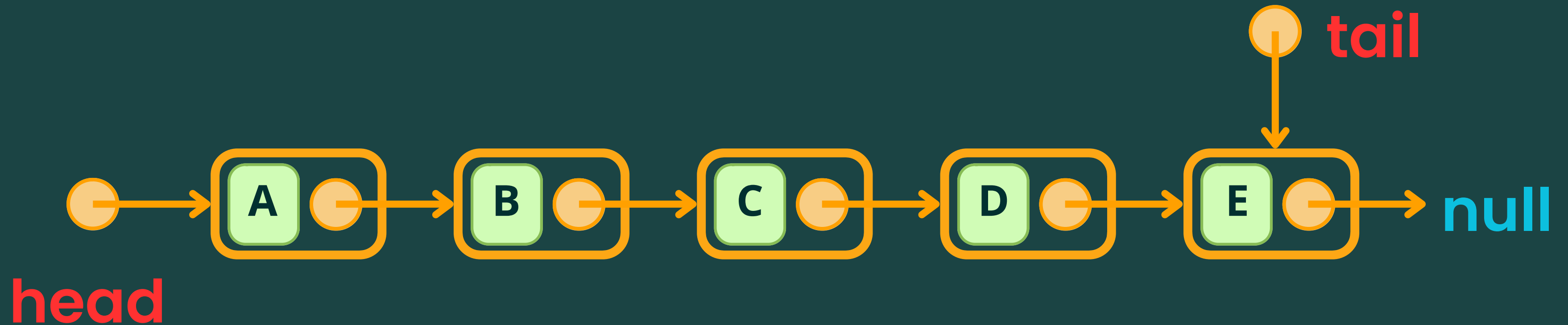
- **Node** (Nút) gồm 2 thành phần:
 - Biến để **lưu giá trị**
 - Con trỏ để **tham chiếu tới node khác**



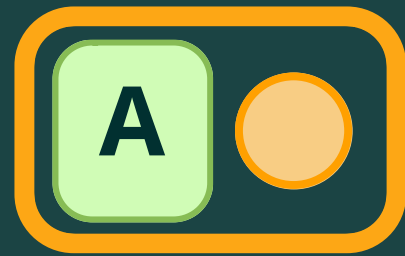
Danh sách liên kết đơn



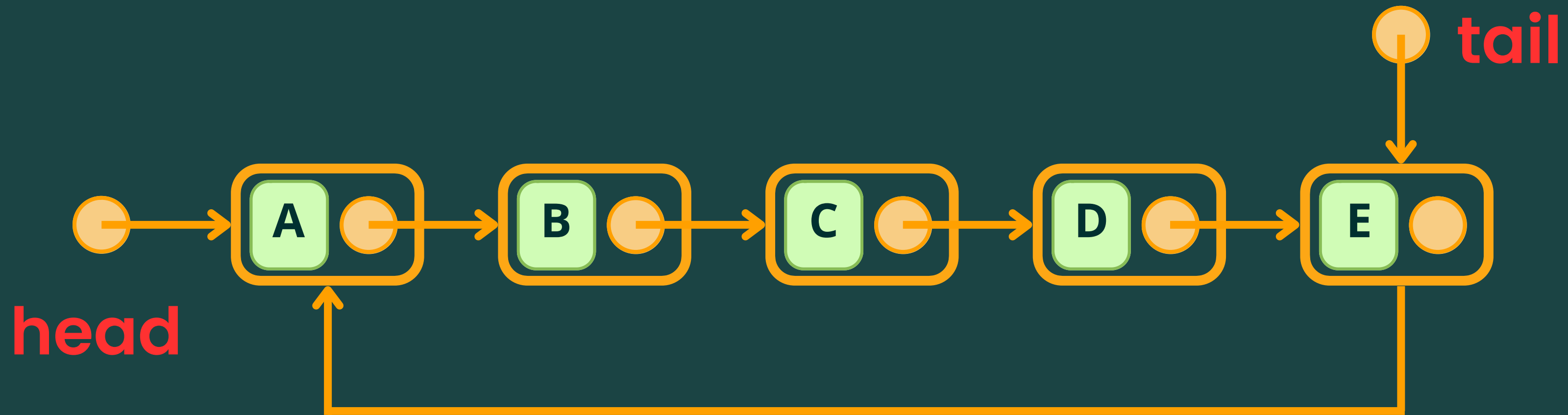
- **Node** (Nút) gồm 2 thành phần:
 - Biến để **lưu giá trị**
 - Con trỏ để **tham chiếu tới node khác**



Danh sách liên kết đơn vòng



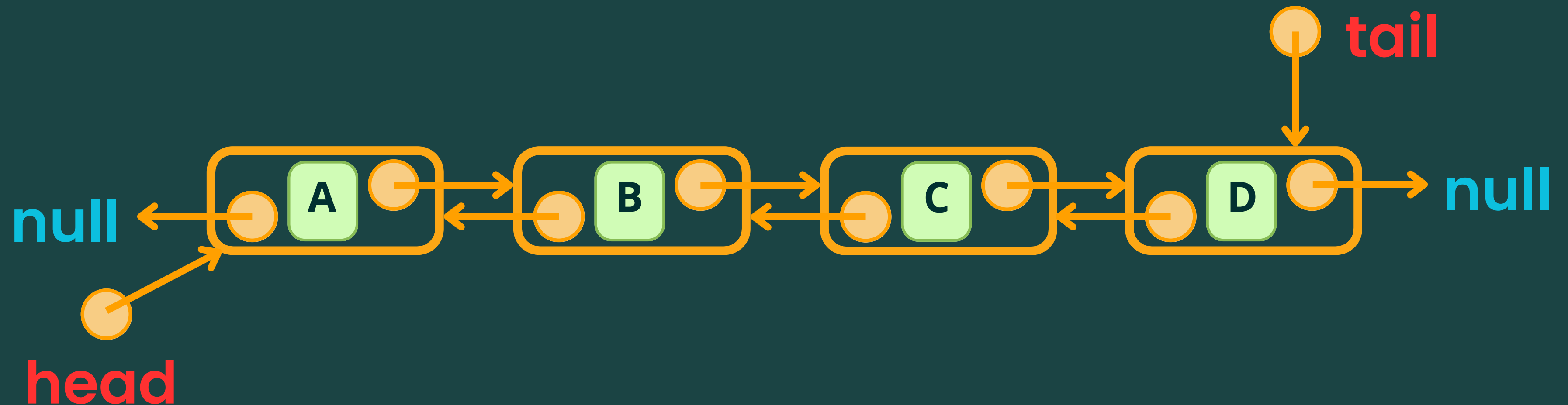
- **Node** (Nút) gồm 2 thành phần:
 - Biến để **lưu giá trị**
 - Con trỏ để **tham chiếu tới node khác**



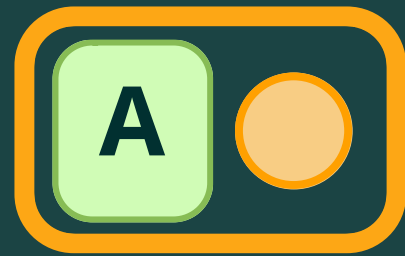
Danh sách liên kết đôi



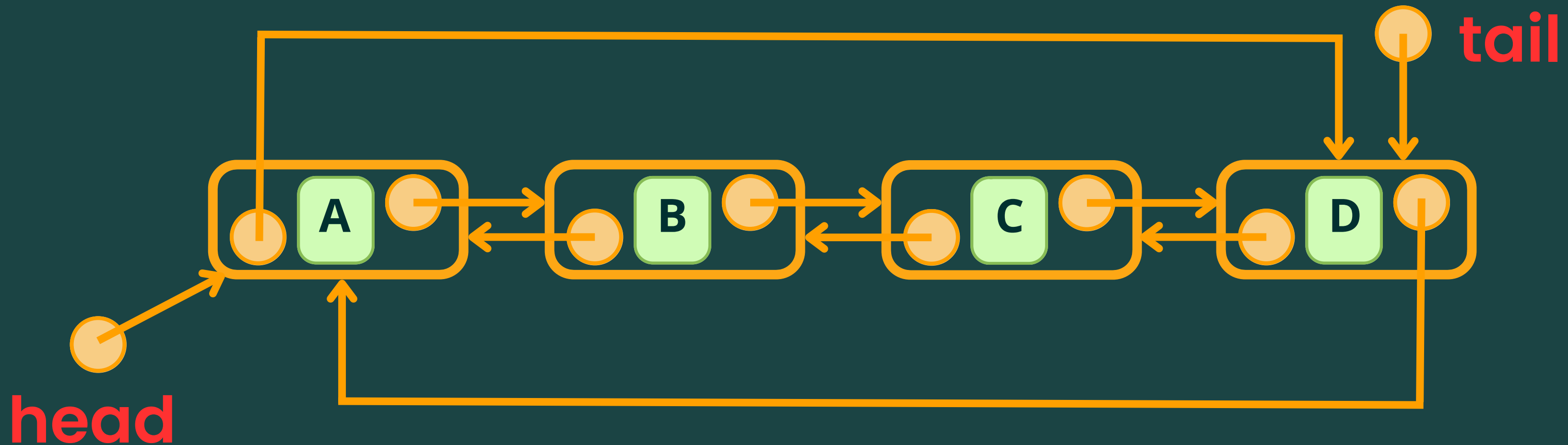
- **Node** (Nút) gồm 2 thành phần:
 - Biến để **lưu giá trị**
 - Con trỏ để **tham chiếu tới node khác**



Danh sách liên kết đôi vòng



- **Node** (Nút) gồm 2 thành phần:
 - Biến để **lưu giá trị**
 - Con trỏ để **tham chiếu tới node khác**





DATABASE

**THANK
YOU**