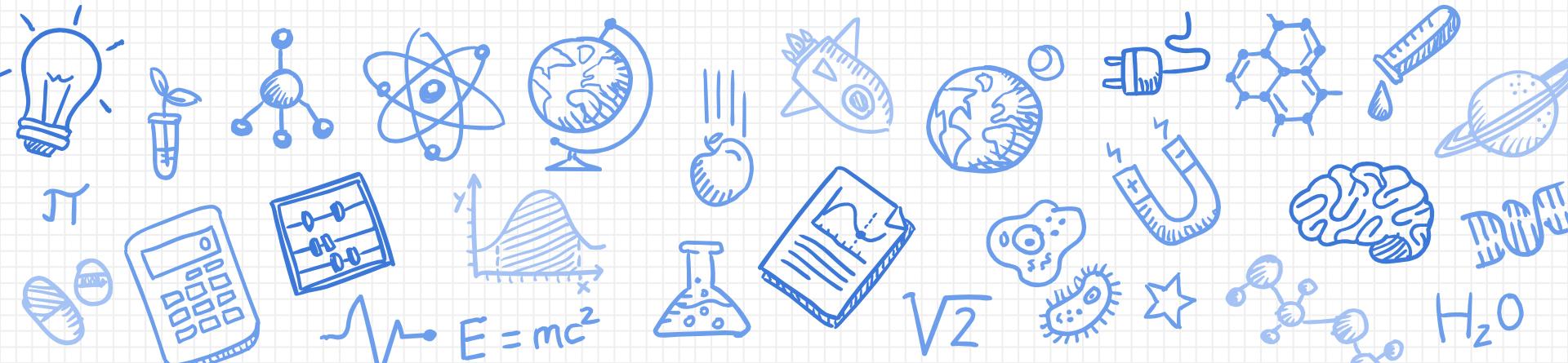


Bài 2.10: Lớp LinkedList

<https://braniumacademy.net>





Nội dung bài học

- ❑ Tổng quan
 - ❑ Các phương thức thường dùng và mô tả
 - ❑ Ví dụ minh họa

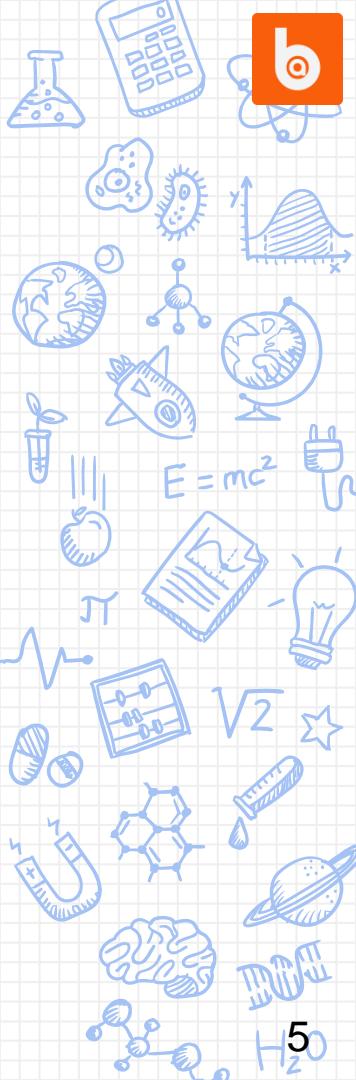
Tổng quan

- ArrayList là mảng có thể tự động co dãn kích thước. Nó triển khai interface List, trong gói java.util.
- Sử dụng để lưu trữ các phần tử của kiểu T nào đó, cho phép cả giá trị null.
- Có phương thức cho phép quản lý kích thước của mảng sử dụng để lưu trữ các phần tử bên trong ArrayList.
- Khi thêm/ xóa phần tử khỏi ArrayList thì kích thước của nó tự động co dãn cho phù hợp.
- ArrayList không đồng bộ. Là thành viên của Collections framework.



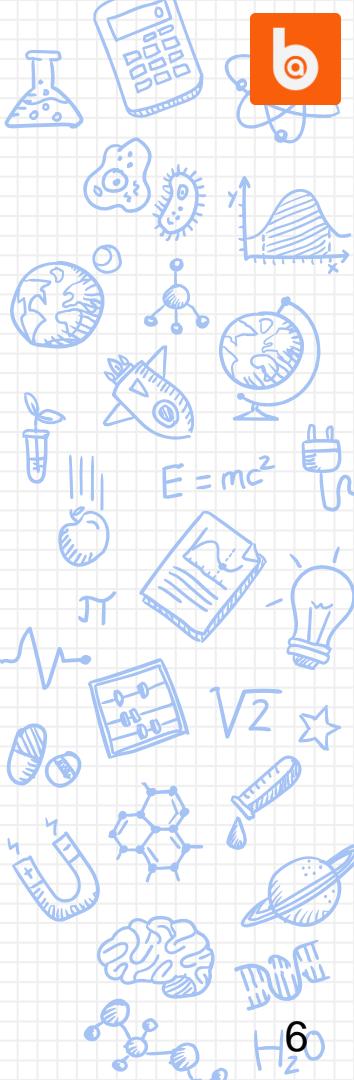
Tổng quan

- ❑ **Khả năng lưu trữ**, tức capacity của ArrayList: kích thước hiện thời của mảng bên trong ArrayList.
 - ❑ **Kích thước**, tức size của ArrayList: số lượng phần tử hiện tại đang lưu trữ trong ArrayList.
 - ❑ Kích thước của ArrayList luôn \leq khả năng lưu trữ của nó.



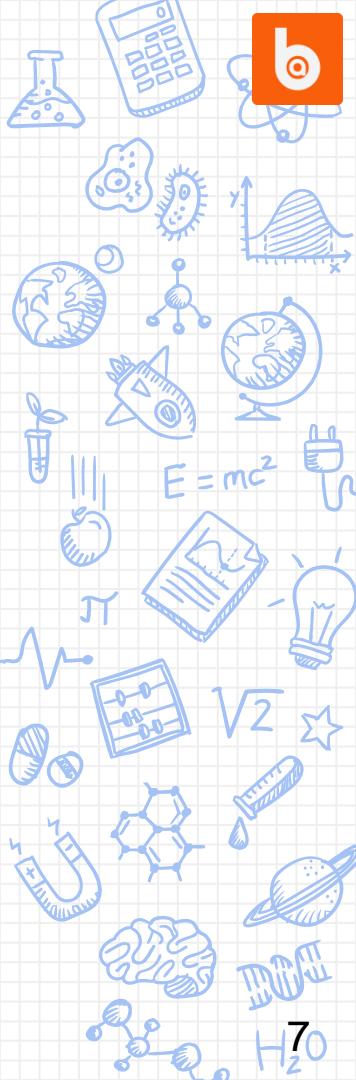
Các phương thức và mô tả

Phương thức	Mô tả
<code>ArrayList()</code>	Hàm tạo một danh sách rỗng với capacity = 10.
<code>ArrayList(Collection<? Extends E> c)</code>	Tạo danh sách từ một collection truyền vào theo đúng thứ tự các phần tử hiện có của c.
<code>ArrayList(int capacity)</code>	Tạo danh sách rỗng với khả năng lưu trữ cho trước.
<code>boolean add(E e)</code>	Thêm phần tử e vào cuối danh sách.
<code>void add(int index, E e)</code>	Chèn phần tử e vào vị trí index trong danh sách.
<code>boolean addAll(Collection<? Extends E> c)</code>	Thêm tất cả các phần tử trong collection c vào cuối danh sách hiện tại. Theo đúng trật tự trong c.
<code>boolean addAll(int index, Collection<? Extends E> c)</code>	Chèn tất cả các phần tử trong collection c vào vị trí index của danh sách hiện thời.
<code>void clear()</code>	Xóa toàn bộ các phần tử trong danh sách hiện tại.
<code>Object clone()</code>	Trả về bản sao của danh sách hiện tại.
<code>boolean contains(Object o)</code>	Kiểm tra xem phần tử o có trong danh sách không.
<code>void ensureCapacity(int minCapacity)</code>	Tăng khả năng lưu trữ của danh sách hiện thời nếu cần thiết để chắc chắn rằng nó có thể chứa ít nhất minCapacity phần tử.
<code>E get(int index)</code>	Trả về phần tử tại vị trí index trong danh sách.



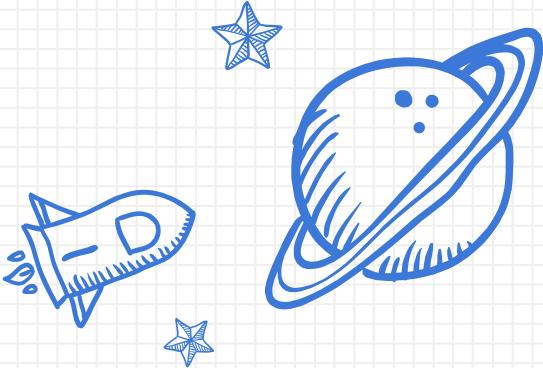
Các phương thức và mô tả

<code>int indexOf(Object o)</code>	Trả về vị trí xuất hiện đầu tiên của o trong danh sách hiện tại. Nếu không tồn tại o, trả về -1.
<code>int lastIndexOf(Object o)</code>	Trả về vị trí xuất hiện cuối cùng của o trong danh sách. Trả về -1 nếu không tìm thấy.
<code>boolean isEmpty()</code>	Trả về true nếu danh sách rỗng và ngược lại.
<code>Iterator<E> iterator()</code>	Trả về iterator dùng để duyệt các phần tử trong danh sách hiện thời.
<code>ListIterator<E> listIterator()</code>	Trả về một list iterator để duyệt các phần tử trong danh sách hiện thời.
<code>ListIterator<E> listIterator(int index)</code>	Trả về một list iterator để duyệt các phần tử trong danh sách hiện thời bắt đầu từ vị trí hiện thời.
<code>E remove(int index)</code>	Xóa phần tử ở vị trí index.
<code>boolean remove(Object o)</code>	Xóa phần tử o đầu tiên xuất hiện trong danh sách.
<code>boolean removeAll(Collection<?> c)</code>	Xóa tất cả các phần tử có mặt trong c khỏi danh sách hiện tại.
<code>boolean removeIf(Predicate<? super E> filter)</code>	Xóa tất cả các phần tử trong danh sách hiện thời thỏa mãn điều kiện được chỉ ra trong tham số.
<code>protected void removeRange(int fromIndex, int toIndex)</code>	Xóa các phần tử có chỉ số bắt đầu từ fromIndex đến trước toIndex.



Các phương thức và mô tả

<code>void replaceAll(UnaryOperator<E> op)</code>	Thay thế phần tử trong danh sách với kết quả sau khi áp dụng toán tử op với phần tử đó.
<code>boolean retainAll(Collection<?> c)</code>	Giữ lại chỉ các phần tử trong danh sách hiện thời mà có mặt trong có mặt trong collection c.
<code>E set(int index, E element)</code>	Thay thế phần tử tại vị trí index bởi giá trị element.
<code>void sort(Comparator<? Super E> c)</code>	Sắp xếp danh sách theo trật tự được chỉ ra bởi Comparator c.
<code>int size()</code>	Trả về số lượng các phần tử hiện có trong danh sách.
<code>List<E> subList(int fromIndex, int toIndex)</code>	Trả về một list của các phần tử bắt đầu từ vị trí fromIndex đến trước vị trí toIndex.
<code>Object[] toArray()</code>	Trả về mảng tất cả các đối tượng trong danh sách theo thứ tự đầu đến cuối.
<code>T[] toArray(T[] a)</code>	Trả về một mảng tất cả các phần tử trong danh sách ở kiểu T.
<code>void trimToSize()</code>	Cắt giảm khả năng lưu trữ của danh sách về kích thước hiện thời của nó.



Tiếp theo

Lớp Vector trong Java