Bài 34: Lớp ArrayList

- ✓ Bản chất của ArrayList
- ✓ Tạo, thêm mới phần tử
- ✓ Cập nhật ArrayList
- ✓ Duyệt ArrayList
- ✓ Một số phương thức khác
- ✓ Bài tập thực hành

Bản chất ArrayList

- Nhược điểm của mảng là số lượng phần tử luôn cố định nên không thể mở rộng hay thu hẹp khi cần
- Kế thừa những ưu điểm của mảng và bổ sung thêm khả năng quản lý các phần tử ưu việt hơn, ArrayList ra đời
- ArrayList là một phần của collection framework trong Java. Nó nằm trong gói java.util
- ArrayList cung cấp khả năng lưu trữ mảng động trong Java

Bản chất ArrayList

- ArrayList có thể chậm hơn mảng thông thường nhưng có thêm nhiều tính năng hữu ích mà mảng thường không có
- ArrayList dùng để lưu trữ các đối tượng cùng kiểu
- Kích thước của ArrayList tự động tăng giảm phù hợp khi ta thêm mới hoặc xóa bỏ phần tử của nó
- Nếu muốn lưu trữ kiểu nguyên thủy với ArrayList ta sử dụng kiểu lớp bao của chúng
- Có thể sử dụng chỉ số mảng để truy cập phần tử trong ArrayList

Tạo mới ArrayList

- Để tạo mới đối tượng của ArrayList ta dùng 1 trong 3 phương thức khởi tạo sau:

Phương thức	Mô tả
ArrayList()	Tạo một đối tượng ArrayList rỗng với khả năng chứa 10 phần tử
ArrayList(Collection c)	Tạo một đối tượng ArrayList với khởi đầu là một tập các phần tử trong một collection c nào đó. Nếu tham số là null ta sẽ nhận NullPointerException
ArrayList(int capacity)	Tạo một đối tượng ArrayList rỗng với khả năng lưu trữ khởi đầu bằng số phần tử trong tham số

Ví dụ:

```
ArrayList<Order> orders = new ArrayList<>();
ArrayList<Order> orders1 = new ArrayList<>(40);
ArrayList<Order> orders2 = new ArrayList<>(orders);
```

Thêm mới phần tử vào ArrayList

➢ Ta dùng phương thức add để thêm mới phần tử:

Phương thức	Mô tả
add(object)	Thêm mới phần tử object cùng kiểu với kiểu của ArrayList vào cuối danh sách.
add(index, object)	Thêm mới một đối tượng object cùng kiểu với kiểu của ArrayList vào vị trí chỉ số index trong danh sách. Các phần tử phía sau chỉ số đó sẽ tự động được lùi về sau 1 vị trí. Lưu ý rằng index phải nằm trong đoạn [0, size-1] với size là kích thước hay số phần tử thực tế hiện thời của ArrayList
addAll(Collection c)	Thêm toàn bộ các phần tử của collection c vào cuối danh sách hiện thời.
addAll(index, Collection c)	Chèn toàn bộ các phần tử của collection c vào vị trí index của danh sách hiện thời. Lưu ý index phải hợp lệ, tức thuộc khoảng [0, size-1].

Ví dụ:

```
ArrayList<String> friends = new ArrayList<>();
friends.add("Nam");
friends.add("Nhung");
friends.add("Hưng");
friends.add("Linh");
friends.add(0, "Hương"); // thêm phần tử vào vị trí 0

for (int i = 0; i < friends.size(); i++) {
    System.out.print(friends.get(i) + " ");
}
```

Ghi nhớ

- Khi ta tạo mới một đối tượng với keyword new thì đối tượng sẽ được cấp phát một vùng nhớ trong bộ nhớ heap.
- Quá trình mở rộng ArrayList khi nó đã đầy và ta tiếp tục thêm phần tử mới vào:
  - Tạo một đối tượng ArrayList mới tại vùng nhớ lớn hơn trong heap
  - Copy các phần tử của ArrayList cũ vào vùng nhớ mới
  - Thêm phần tử mới vào danh sách đã tạo
  - Xóa bỏ vùng nhớ heap cũ không còn sử dụng

Thay đổi giá trị phần tử ArrayList

➢Để thực hiện ta dùng phương thức set():

Phương thức	Mô tả
set(index, newObject)	Thay thế phần tử tại vị trí index trong ArrayList hiện thời bằng phần tử newObject. Trong đó newObject phải đảm bảo cùng kiểu với kiểu của ArrayList. Lưu ý rằng index phải nằm trong đoạn [0, size-1] với size là kích thước hay số phần tử thực tế hiện thời của ArrayList

Ví dụ:

```
ArrayList<String> friends = new ArrayList<>();
friends.add("Nam");
friends.add("Nhung");
friends.add("Hưng");
friends.add("Linh");
friends.set(0, "Tony"); // thay phần tử tại vị trí 0 thành "Tony"
for (int i = 0; i < friends.size(); i++) {
    System.out.print(friends.get(i) + " ");
}
```

Kết quả:

```
Tony Nhung Hưng Linh
```

Loại bỏ phần tử khỏi ArrayList

- Để loại bỏ phần tử khỏi ArrayList ta dùng các phương thức sau:

Phương thức	Mô tả
remove(object)	Xóa phần tử object khỏi danh sách. Nếu phần tử này xuất hiện nhiều lần trong danh sách thì xóa phần tử đầu tiên tìm được.
remove(index)	Xóa bỏ phần tử tại chỉ số index khỏi danh sách hiện thời. Lưu ý index phải nằm trong đoạn [0, size-1] nếu không bạn sẽ gặp ngoại lệ.
clear()	Xóa bỏ toàn bộ các phần tử hiện có trong ArrayList
removeAll(Collection c)	Xóa bỏ toàn bộ các phần tử chứa trong collection cho trong tham số.
removeRange(fromIndex, toIndex)	Xóa bỏ các phần tử trong danh sách mà chỉ số của các phần tử này nằm trong khoảng [fromIndex, toIndex). Lưu ý các index phải hợp lệ.

Ví dụ:

```
ArrayList<String> friends = new ArrayList<>();
friends.add("Nam");
friends.add("Nhung");
friends.add("Hưng");
friends.add("Linh");
friends.remove("Nam"); // xóa bỏ phần tử có giá trị "Nam"
friends.remove(1);     // xóa bỏ phần tử tại vị trí 1
```

Duyệt ArrayList

- Để duyệt ArrayList ta dùng for thường hoặc foreach:

```
ArrayList<String> friends = new ArrayList<>();
friends.add("Nam");
friends.add("Nhung");
friends.add("Hưng");
friends.add("Linh");
// duyệt bằng for thường
for (int i = 0; i < friends.size(); i++) {
    System.out.print(friends.get(i) + " ");
}
System.out.println(); // in xuống dòng
// duyệt bằng foreach
for (var friend : friends) {
    System.out.print(friend + " ");
}
```

```
Nam Nhung Hưng Linh
Nam Nhung Hưng Linh
```

Một số phương thức khác

➢Sau đây là một số phương thức thường dùng khác:

Phương thức	Mô tả
size()	Trả về số lượng phần tử hiện thời trong danh sách. Tương đương với length của mảng.
indexOf(object)	Trả về chỉ số của lần xuất hiện đầu tiên của phần tử object trong danh sách. Nếu không tồn tại object này thì trả về -1.
lastIndexOf(object)	Trả về chỉ số của lần xuất hiện cuối cùng của phần tử object trong danh sách. Nếu không tồn tại object này thì trả về -1.
isEmpty()	Trả về true nếu danh sách rỗng và false trong trường hợp ngược lại.
clone()	Trả về một bản sao của danh sách hiện thời.
contains(object)	Trả về true nếu danh sách chứa phần tử cho trong tham số và false trong trường hợp ngược lại.
ensureCapacity(minCapacity)	Tăng khả năng chứa của danh sách hiện thời lên tối thiểu minCapacity phần tử. Sử dụng khi muốn chắc chắn rằng danh sách hiện thời có thể chứa ít nhất minCapacity phần tử.
subList(fromIndex, toIndex)	Trả về một danh sách các phần tử của ArrayList hiện thời có chỉ số trong khoảng [fromIndex, toIndex).
toArray()	Trả về một mảng các phần tử trong danh sách theo thứ tự xuất hiện từ đầu đến cuối.
toArray(T[] arr)	Trả về một mảng các phần tử trong danh sách ở kiểu T[] theo thứ tự xuất hiện từ đầu đến cuối.

Nội dung tiếp theo

Các gói trong Java