

ArrayList



Java

Nhắc lại kiến thức

Interface	Abstract class
Không phải class	Là class
Không có hàm khởi tạo	Có hàm khởi tạo
Chỉ chứa abstract methods. Tuy nhiên, từ Java 8, có thể chứa các phương thức non-abstract bằng cách sử dụng default hoặc static	Một abstract class có thể chứa các phương thức abstract hoặc non-abstract
Interface chỉ chứa các biến static final	Abstract class có thể chứa các biến final , non-final , static và non-static
Abstract class có thể implements interface	Interface không thể implements abstract class
Interface được triển khai bằng cách sử dụng từ khóa implements	Abstract class được kế thừa bằng cách sử dụng extends
1 interface có thể extends 1 hoặc nhiều interface khác	1 abstract class có thể implements 1 hoặc nhiều interface

Class

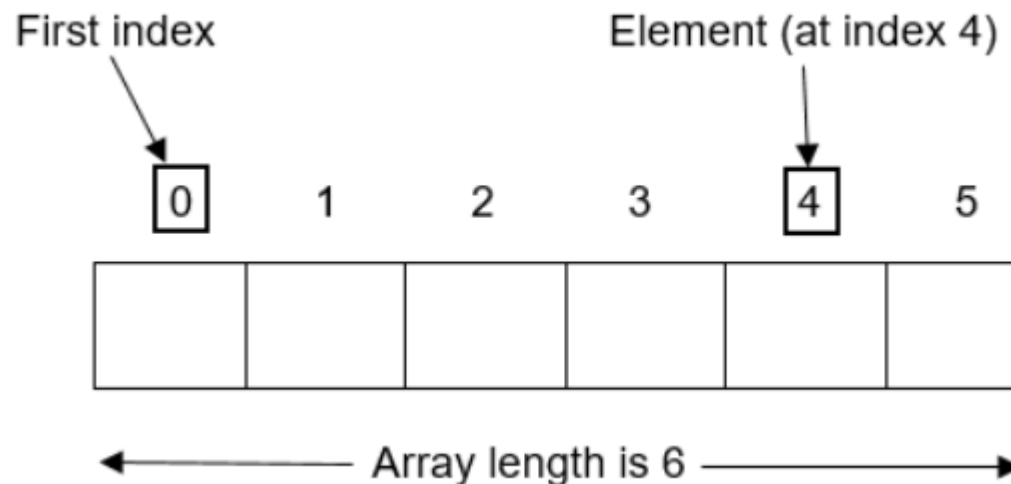
- Có hàm khởi tạo
- Chỉ có phương thức non-abstract
- Có thể khởi tạo đối tượng bằng từ khóa “new”

Abstract Class

- Có hàm khởi tạo
- Có phương thức abstract và non-abstract
- Không thể khởi tạo đối tượng trực tiếp bằng từ khóa “new” mà phải thông qua constructor của lớp con

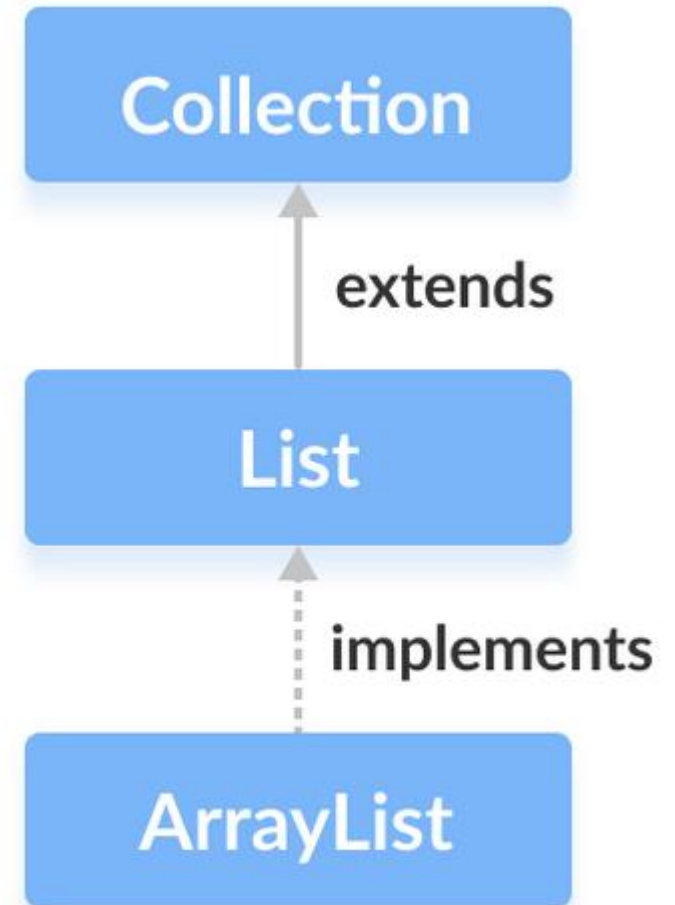
Nhắc lại về Array

- Khái niệm: là một tập hợp các phần tử có cùng kiểu dữ liệu, mỗi phần tử trong mảng được truy xuất thông qua các chỉ số của nó trong mảng
- Khai báo: `<type> [] tên_mảng = {...}`
hoặc `<type>[] tên_mảng = new type[kích_thước_mảng]`
- VD: `int[] intArr= {1, 2, 5, 3}` // Khởi tạo mảng có sẵn giá trị
`int[] intArr= new int[5]` // khởi tạo mảng chưa có gtri với kích thước mảng = 5
- Đặc điểm: Kích thước cố định, lưu các phần tử theo chỉ số (index), chỉ số của phần tử đầu tiên là 0



ArrayList

- Là một phần của Collection framework
- Cung cấp cho người dùng một mảng động (kích thước thay đổi)
- Cho phép lưu trữ và thao tác với một lượng lớn dữ liệu



ArrayList Và Array

Array

- Kích thước mảng cố định
- Có thể lưu phần tử trùng lặp
- Không thể thêm phần tử sau khi mảng đã đầy. Việc thêm và xóa phần tử ở giữa mảng gây nhiều khó khăn.

ArrayList

- Kích thước mảng có thể thay đổi
- Có thể lưu phần tử trùng lặp
- Thêm và xóa phần tử ở bất kì vị trí nào trong mảng một cách nhanh chóng.
- Có nhiều phương thức hỗ trợ các thao tác như removeAll-xóa tất cả, searching, iterations-duyet mảng, retainAll-sửa phần tử,...

Khởi tạo ArrayList

```
ArrayList<Kiểu dữ liệu> <Tên> = new ArrayList<Kiểu dữ liệu>();
```

```
ArrayList<Kiểu dữ liệu> <Tên> = new ArrayList<>();
```

```
List<Kiểu dữ liệu> <Tên> = new ArrayList<>();
```

Ví dụ:

```
ArrayList<String> cars = new ArrayList<String>();
```

```
ArrayList<Integer> myNumbers = new ArrayList<Integer>();
```

```
List<Integer> myNumbers = new ArrayList<Integer>();
```

Một số phương thức của ArrayList

- Thêm phần tử

```
ArrayList<String> animals = new ArrayList<>();  
// Add elements  
animals.add("Dog");  
animals.add("Cat");  
animals.add("Horse");  
System.out.println("ArrayList: " + animals);
```

ArrayList: [Dog, Cat, Horse]

- Truy cập phần tử trong ArrayList bằng hàm get()
`System.out.println(animals.get(1));`

-> Result: Cat

Một số phương thức của ArrayList

Khác với mảng, không thể khởi tạo ArrayList một cách trực tiếp. Tuy nhiên, chúng ta có thể sử dụng hàm `asList()` của class `Arrays` để làm điều đó

```
class Main {  
    public static void main(String[] args) {  
        // Creating an array list  
        ArrayList<String> animals = new ArrayList<>(Arrays.asList("Cat", "Cow", "Dog"));  
        System.out.println("ArrayList: " + animals);  
    }  
}
```

```
ArrayList: [Cat, Cow, Dog]
```

Một số phương thức của ArrayList

- Xóa phần tử trong arrayList

```
public static void main(String[] args) {  
    ArrayList<String> animals = new ArrayList<>();  
    // Add elements in the ArrayList  
    animals.add("Dog");  
    animals.add("Cat");  
    animals.add("Horse");  
    System.out.println("Initial ArrayList: " + animals);  
    // Remove the first element  
    animals.remove("Dog"); // hoặc animals.remove(0);  
    System.out.println("Final ArrayList: " + animals);  
}
```

Một số phương thức của ArrayList

`removeAll(Collection c)`

```
public static void main(String[] args) {  
    ArrayList<String> animals = new ArrayList<>();  
    // Add elements in the ArrayList  
    animals.add("Dog");  
    animals.add("Cat");  
    animals.add("Horse");  
    System.out.println("Initial ArrayList: " + animals);  
    // Remove all elements inside the collection  
    parameter from the list  
    animals.removeAll(animals);  
    System.out.println("Final ArrayList: " + animals); }
```

`clear()`

```
public static void main(String[] args) {  
    ArrayList<String> animals = new ArrayList<>();  
    // Add elements in the array list  
    animals.add("Dog");  
    animals.add("Cat");  
    animals.add("Horse");  
    System.out.println("Initial ArrayList: " + animals);  
    // Remove all elements from the list  
    animals.clear();  
    System.out.println("Final ArrayList: " + animals); }
```

Initial ArrayList: [Dog, Cat, Horse]
Final ArrayList: []

Update thông tin phần tử trong ArrayList

```
public static void main(String[] args) {  
  
    ArrayList<String> arrayList = new ArrayList<>();  
    arrayList.add("abc");  
    arrayList.add("def");  
    arrayList.set(0, "abc New");  
    System.out.println("arrayList: "+arrayList);  
}
```

arrayList: [abc New, def]

- Cập nhật giá trị của phần tử trong ArrayList bằng hàm:
set(index, element)

Sort()

```
public class CollectionsExample {  
    public static void main(String args[]) {  
        // create list  
        List<String> list = new ArrayList<String>();  
        // add elements to list  
        list.add("Java");  
        list.add("C");  
        list.add("C++");  
        list.add("PHP");  
        list.add("Python");  
        // sort list  
        Collections.sort(list);  
        // show list  
        for (String element : list) {  
            System.out.println(element);  
        }  
    }  
}
```



C
C++
Java
PHP
Python

Bài tập: Quản lý hộ dân

- Xây dựng lớp Hodan (hộ dân):
 - Số thành viên trong hộ (số người)
 - Số nhà của hộ dân đó. (Số nhà được gán cho mỗi hộ dân)
 - Thông tin về mỗi cá nhân trong hộ gia đình.
 - Với mỗi cá nhân, người ta quản lý các thông tin như: họ và tên, ngày sinh, nghề nghiệp.

YC:

- Thực hiện viết chức năng nhập thông tin cho hộ dân và các thành viên trong hộ dân đó
- Hiển thị thông tin chi tiết của hộ dân

Bài tập: Quản lý trung tâm đào tạo

- Trung tâm TechMaster gồm: 1b quản lý lớp học, 1 giảng viên và 1 lớp học
- Trong lớp học gồm: môn học và thông tin của “các” bạn học viên
- Mỗi bạn học viên gồm: id, tên, tuổi, học lực
- -> Hãy giúp thầy Cường tạo 1 chương trình quản lý trung tâm bằng việc :
 - Hiển thị thông tin chi tiết của trung tâm
 - Thực hiện thêm học viên vào lớp hiện tại
 - Thực hiện cập nhật thông tin học lực của 1b học viên dựa vào Id
 - Có 1 bạn xin nghỉ học, hãy giúp thầy xóa bạn đó khỏi lớp học

Bài tập: Quản lý hồ sơ học sinh

- Để quản lý hồ sơ học sinh của trường THPT, người ta cần quản lý những thông tin như
- sau:
 - - Các thông tin về: lớp, khoá học, kỳ học, và các thông tin cá nhân của mỗi học sinh.
 - - Với mỗi học sinh, các thông tin cá nhân cần quản lý gồm có: Họ và tên, ngày sinh, quê quán.
- 1. Hãy xây dựng lớp `Ngnoi` để quản lý các thông tin cá nhân của mỗi học sinh.
- 2. Xây dựng lớp `HocSinh` (học sinh) để quản lý các thông tin về mỗi học sinh.
- 3. Cài đặt chương trình thực hiện các công việc sau:
 - - Nhập vào một danh sách gồm n học sinh (n nhập từ bàn phím)
 - - Hiển thị ra màn hình tất cả những học sinh sinh năm 1985 và quê ở Thái Nguyên
 - - Hiển thị ra màn hình tất cả những học sinh của lớp 10A1

Một số hàm mở rộng

Phương thức	Mô tả
<code>lastIndexOf(Object o)</code>	Trả về chỉ mục trong danh sách với sự xuất hiện cuối cùng của phần tử được chỉ định, hoặc -1 nếu danh sách không chứa phần tử này.
<code>toArray()</code>	Trả về một mảng chứa tất cả các phần tử trong danh sách này theo đúng thứ tự.
<code>toArray(Object[] a)</code>	Trả về một mảng chứa tất cả các phần tử trong danh sách này theo đúng thứ tự.
<code>clone()</code>	Trả về một bản sao của ArrayList.
<code>clear()</code>	Xóa tất cả các phần tử từ danh sách này.
<code>trimToSize()</code>	Cắt dung lượng của thể hiện ArrayList này là kích thước danh sách hiện tại.
<code>contains(element)</code>	Kết quả trả về là true nếu tìm thấy element trong danh sách, ngược lại trả về false.

Phương thức	Mô tả
<code>add(Object o)</code>	Thêm phần tử được chỉ định vào cuối một danh sách.
<code>add(int index, Object element)</code>	Chèn phần tử element tại vị trí index vào danh sách.
<code>addAll(Collection c)</code>	Thêm tất cả các phần tử trong collection c vào cuối của danh sách, theo thứ tự chúng được trả về bởi bộ lặp iterator.
<code>addAll(int index, Collection c)</code>	Chèn tất cả các phần tử trong collection c vào danh sách, bắt đầu từ vị trí index.
<code>retainAll(Collection c)</code>	Xóa những phần tử không thuộc collection c ra khỏi danh sách.
<code>removeAll(Collection c)</code>	Xóa những phần tử thuộc collection c ra khỏi danh sách.
<code>indexOf(Object o)</code>	Trả về chỉ mục trong danh sách với sự xuất hiện đầu tiên của phần tử được chỉ định, hoặc -1 nếu danh sách không chứa phần tử này.