

# Bài 08

## COLLECTION

Company: DEVPRO VIỆT NAM

Website: [devpro.edu.vn](http://devpro.edu.vn)

Design by Minh An

## Nội dung

- Giới thiệu
- Một số class thông dụng
  - ArrayList
  - LinkedList
  - HashMap
  - ...

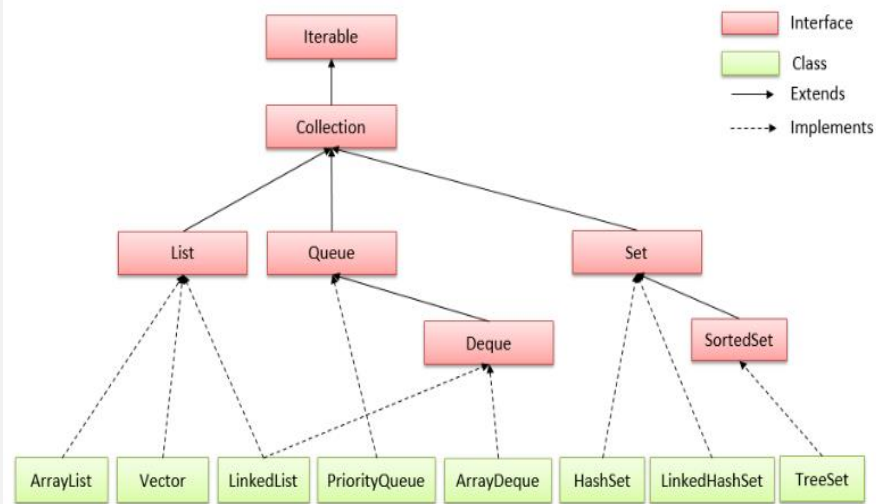
Design by Minh An

## Collection trong java

- Collection cung cấp một bộ các class để làm việc với một nhóm nhiều các đối tượng (danh sách).
- Các lớp này cung cấp các phương thức để xử lý danh sách: thêm mới, sửa , xóa, chèn thêm, thay thế, sắp xếp, tìm kiếm, ...
- Collection khắc phục nhược điểm của mảng là cố định số lượng các phần tử mảng khi chương trình đang được thực thi.

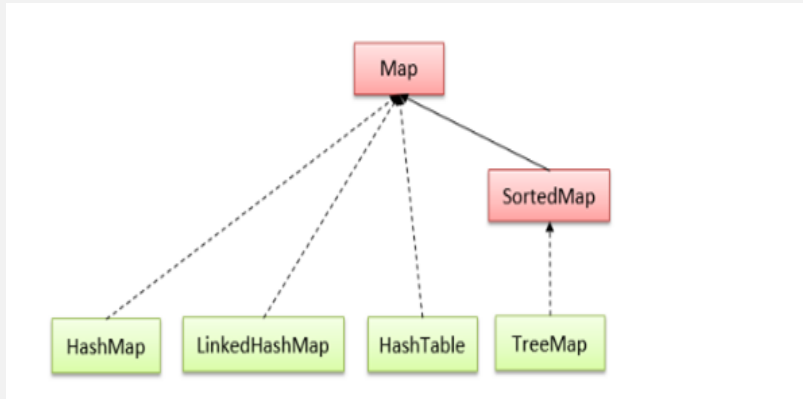
Design by Minh An

## Collection



Design by Minh An

## Collection



Design by Minh An

## ArrayList class

- ArrayList hỗ trợ mảng động (có thể thêm hay bớt số phần tử khi chương trình đang thực thi).
- Khởi tạo (Constructor)

```
ArrayList[<ClassName>] list =  
    new ArrayList([<ClassName>]);
```

- Ví dụ:

```
ArrayList list_1 = new ArrayList();  
ArrayList<Person> list =  
    new ArrayList(<Person>);
```

Design by Minh An

## Một số phương thức xử lý ArrayList

**list.size():** trả về số lượng phần tử của list

**list.add(obj):** thêm đối tượng obj vào cuối list

**list.add(i, obj):** chèn đối tượng obj vào vị trí có chỉ số i trong list

**list.addAll(collection):** Thêm collection vào cuối list

**list.addAll(i, collection):** Chèn collection vào list tại vị trí chỉ số i

**list.clear():** Xóa mọi phần tử trong list

**list.clone():** Trả về một đối tượng là nhân bản của list

**list.contains(obj):** Trả về true nếu obj thuộc list, ngược lại trả về false

**list.get(i):** Trả về phần tử tại vị trí chỉ số i

Design by Minh An

## Một số phương thức xử lý ArrayList



**list.indexOf(obj):** Trả về chỉ số tại vị trí đầu tiên chứa obj, nếu không có obj trong list trả về - 1

**list.lastIndexOf(obj):** Trả về chỉ số tại vị trí cuối cùng chứa obj, nếu không có obj trong list trả về - 1

**list.remove(i):** Xóa phần tử tại vị trí chỉ số i

**list.set(i, obj):** Thay thế phần tử tại vị trí chỉ số i bằng obj.

Design by Minh An

## Duyệt ArrayList



- Sử dụng for (thường) với list.size() và list.get(i)
- Sử dụng for each
- Sử dụng Iterator

```
Iterator<Emp> itor = list.iterator();  
while (itor.hasNext()) {  
    itor.next().display();  
}
```

Design by Minh An

## Sort ArrayList by Collection.sort()



- Sắp danh sách nhân sự theo lương

```
public static sort(ArrayList<Emp> list) {  
    Collection.sort(list, new Comparator<Emp>() {  
        @Override  
        public int compare(Emp e1, Emp e2) {  
            if (e1.getSalary() > e2.getSalary()) {  
                return 1;  
            }  
            else if (e1.getSalary() == e2.getSalary()) {  
                return 0;  
            }  
            else {  
                return -1;  
            }  
        }  
    });  
}
```

Design by Minh An

## Ví dụ



Tạo danh sách học viên sử dụng ArrayList, gọi các phương thức của ArrayList xử lý danh sách học viên, mỗi học viên gồm các thông tin: Mã học viên, họ tên, giới tính, tên khóa học tham gia.

Design by Minh An

## LinkedList trong java



- Package: **java.util.LinkedList**
- LinkedList là một cấu trúc dữ liệu được sử dụng để lưu trữ danh sách tuyến tính gọi là danh sách liên kết.
- LinkedList được sử dụng khi số lượng phần tử trong danh sách cần lưu trữ lớn.
- Khởi tạo LinkedList: tương tự ArrayList.
- Ngoài các phương thức tương tự như ArrayList thì LinkedList có thêm các phương thức:

```
l1list.addFirst(obj) ;  
l1list.addLast(obj) ;  
l1list.getFirst() ;  
l1list.getLast() ;  
l1list.removeFirst() ;  
l1list.removeLast()
```

Design by Minh An

## Duyệt LinkedList



- Sử dụng for (thường) với list.size() và list.get(i)
- Sử dụng for each
- Sử dụng Iterator

```
Iterator<Students> itor = llist.iterator();  
while (itor.hasNext()) {  
    itor.next().display();  
}
```

Design by Minh An

## Sort LinkedList by Collection.sort()



- Sắp danh sách học viên theo tuổi

```
public static sort(LinkedList<Students> list) {  
    Collection.sort(list, new Comparator<Students>() {  
        @Override  
        public int compare(Emp e1, Emp e2) {  
            if (e1.getAge() > e2.getAge()) {  
                return 1;  
            }  
            else if (e1.getAge() == e2.getAge()) {  
                return 0;  
            }  
            else {  
                return -1;  
            }  
        }  
    });  
}
```

Design by Minh An

## LinkedList – Ví dụ



- Sử dụng lại ví dụ với danh sách học viên.

Design by Minh An

## LinkedList vs ArrayList



ArrayList	LinkedList
1) ArrayList internally uses a <b>dynamic array</b> to store the elements.	LinkedList internally uses a <b>doubly linked list</b> to store the elements.
2) Manipulation with ArrayList is <b>slow</b> because it internally uses an array. If any element is removed from the array, all the bits are shifted in memory.	Manipulation with LinkedList is <b>faster</b> than ArrayList because it uses a doubly linked list, so no bit shifting is required in memory.
3) An ArrayList class can <b>act as a list</b> only because it implements List only.	LinkedList class can <b>act as a list and queue</b> both because it implements List and Deque interfaces.
4) ArrayList is <b>better for storing and accessing</b> data.	LinkedList is <b>better for manipulating</b> data.

Design by Minh An



# HashMap



- Là một triển khai của Map Interface
- Lưu trữ dữ liệu dưới dạng cặp key – value
- HashMap chỉ chứa các key duy nhất
- Có thể có 1 key là null, và nhiều value là null
- Duy trì các phần tử không theo thứ tự chèn
- Khởi tạo HashMap: `HashMap(<K, V>)`
- K là kiểu của key, V là kiểu của value
- Ví dụ:

```
HashMap<Integer, String> hashMap =  
    new HashMap<Integer, String>();
```

Design by Minh An

## HashMap – Một số phương thức



- `boolean equals(Object obj)`: Trả về true nếu obj có key và value bằng với key và value của đối tượng đang gọi
- `Object clone()`: Trả về một đối tượng là nhân bản của HashMap
- `boolean containsKey(Object key)`: Trả về true nếu HashMap chứa element có key bằng với key đã cho
- `Object put(Object key, Object value)`: chèn một element vào map hiện tại.
- `Object get(Object key)`: trả về value ứng với key được chỉ định
- `Object remove(Object key)`: xóa element có key được chỉ định
- `void clear()`: Xóa bỏ tất cả elements
- `boolean containsValue(Object value)`: Trả về true nếu HashMap chứa element có value bằng với value đã cho.

Design by Minh An

## HashMap – Một số phương thức



- `Set entrySet()`: Trả về một tập hợp các element
- `Set keySet()`: Trả về đối tượng Set có chứa tất cả các keys
- `boolean isEmpty()`: Trả về true nếu HashMap không chứa element nào
- `Set keySet()`: Trả về tập các key được chứa trong HashMap
- `putAll(Map m)`: Sao chép tất cả elements của map đã xác định vào HashMap.
- `int size()`: Trả về số element chứa trong HashMap
- `Collection values()`: Trả về tập các value của các elements chứa trong HashMap.

Design by Minh An

## Duyệt HashMap



- **Sử dụng phương thức `keySet()`**  

```
for (Integer key : hashMap.keySet()) {  
    Student value = hashMap.get(key);  
    System.out.println(key + " = " + value);  
}
```
- **Sử dụng phương thức `entrySet()`.**  

```
for (Entry<Integer, Student> entry : hashMap.entrySet())  
{  
    Integer key = entry.getKey();  
    Student value = entry.getValue();  
    System.out.println(key + " = " + value);  
}
```

Design by Minh An

## HashMap – Ví dụ



- Xây dựng từ điển với các yêu cầu
  - Tạo mới từ điển
  - Tra cứu từ trong từ điển
  - Thêm từ
  - Sửa từ trong từ điển
  - Xóa từ
  - ...

Design by Minh An

## Bài tập: Sử dụng ArrayList hoặc LinkedList



- Chương trình quản lý bán hàng gồm các bảng dữ liệu:

LoaiHang
ID
TenLoai

KhachHang
ID
TenKhachHang
LoaiHangID

Hang
ID
LoaiHangID
TenHang
SoLuong
DonGia

GioHang
ID
TenHang
LoaiHangID

Design by Minh An

## Bài tập: Sử dụng ArrayList hoặc LinkedList



- Cài đặt chương trình thực hiện thao tác:
  - Tạo danh sách loại hàng.
  - Tạo danh sách hàng trong đó loại hàng chỉ được chọn trong danh sách loại hàng.
  - Tạo danh sách khách hàng
  - Tạo giỏ hàng cho một khách mua hàng
  - Hiển thị danh sách loại hàng
  - Hiển thị danh sách hàng
  - Hiển thị danh sách khách mua hàng
  - Hiển thị giỏ hàng của khách

Design by Minh An

## Bài tập: HashMap



- Cài đặt chương trình giới thiệu các địa điểm du lịch:
  - Tạo danh sách ban đầu
  - Cho phép thêm / sửa / xóa địa điểm
  - Cho phép tìm kiếm địa điểm

Design by Minh An