Phát triển ứng dụng CSDL 2

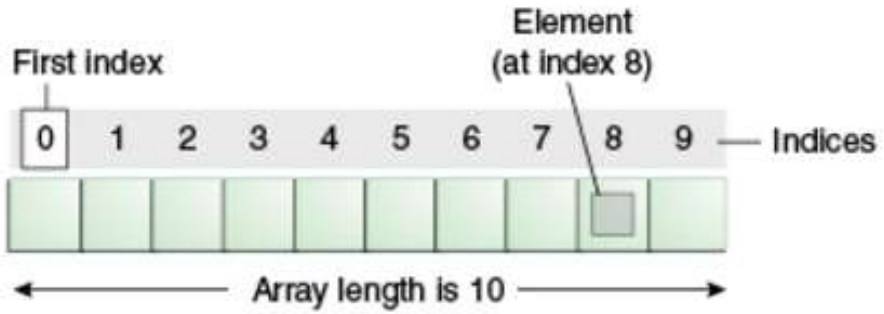
Người soạn: ThS. Phạm Minh Tú

Chương 04

- Nội dung
  - **Array**
  - Collection Framework

Array – Mảng

- Là một đối tượng lưu các giá trị có kích thước cố định.



An array of 10 elements.

Array

```
class ArrayDemo {
   public static void main(String[] args) {
      // declares an array of integers
      int[] anArray;

      // allocates memory for 5 integers
      anArray = new int[5];

      // initialize first element
      anArray[0] = 100;  anArray[1] = 200; anArray[2] = 300;   anArray[3] = 400;      anArray[4] = 500;

      System.out.println("Element at index 0: "
                         + anArray[0]);
      System.out.println("Element at index 1: "
                         + anArray[1]);
      System.out.println("Element at index 2: "
                         + anArray[2]);
      System.out.println("Element at index 3: "
                         + anArray[3]);
      System.out.println("Element at index 4: "
                         + anArray[4]);
   }
}
```

Array

- Khai báo mảng
  - int[] anArray;
- Tạo mảng
  - anArray = new int[10];
- Khởi tạo mảng
  - anArray[0] = 100;
  - anArray[1] = 200;
  - anArray[2] = 300;

Array

- Truy xuất các phần tử mảng
  - System.out.println("Element 1 at index 0: " + anArray[0]);
  - System.out.println("Element 2 at index 1: " + anArray[1]);
  - System.out.println("Element 3 at index 2: " + anArray[2]);
- Tạo và khởi tạo mảng

```
int[] anArray = {
    100, 200, 300,
    400, 500, 600,
    700, 800, 900, 1000
};
```

Multidimensional array

- Mảng nhiều chiều là mảng có phần tử là một mảng khác.
- Khai báo và khởi tạo mảng 2 chiều

```
String[][] names = {
  {"Mr. ", "Mrs. ", "Ms. "},
  {"Smith", "Jones"}
};
```

Multidimensional array

- Truy xuất mảng 2 chiều

```
System.out.println(names[0][0] + names[1][0]);
System.out.println(names[0][2] + names[1][1]);
```

Array

- Copy array
  - Lớp System hỗ trợ phương thức arraycopy để copy mảng a đến mảng b

```
public static void arraycopy(Object
src, int srcPos,Object dest, int
destPos, int length)
```

Array

- Copy array
  - Lớp System hỗ trợ phương thức arraycopy để copy mảng a đến mảng b

```
public static void arraycopy(Object
src, int srcPos,Object dest, int
destPos, int length)
```

Array

- Ví dụ

```
char[] copyFrom = { 'd', 'e', 'c', 'a', 'f', 'f', 'e',
                    'i', 'n', 'a', 't', 'e', 'd' };
char[] copyTo = new char[7];
System.arraycopy(copyFrom, 2, copyTo, 0, 7);
System.out.println(new String(copyTo));
```

Array

- Một số phương thức mảng
  - copyOfRange (): Copy mảng không cần tạo mảng đích
  - Ví dụ

```
char[] copyFrom = {'d', 'e', 'c', 'a', 'f', 'f', 'e',
                   'i', 'n', 'a', 't', 'e', 'd'};
char[] copyTo = java.util.Arrays.copyOfRange(copyFrom, 2, 9);
System.out.println(new String(copyTo));
```

Array

- Một số phương thức mảng
  - binarySearch(Object[] a, Object key): Tìm kiếm phần tử trong mảng bằng thuật toán tìm kiếm nhị phân

```
int []a =new int[]{4,5,7,2,3,6,7,1};
int index=Arrays.binarySearch(a, 3);
System.out.println(index);
```

Array

- Một số phương thức mảng
  - Sort(): Sắp xếp các phần tử trong mảng theo thứ tự tăng dần, dùng thuật toán quicksort

```
Short sArr[] = new Short[]{3, 13, 1, 9, 21};
Arrays.sort(sArr);
for (short number : sArr) {
     System.out.println("Number = " + number);
}
```

Array

- Một số phương thức mảng
  - Nếu sắp xếp giảm dần, dùng:

public static <T> void sort(T[] a, **Comparator<? super T> c**)

```
Short sArr[] = new Short[]{3, 13, 1, 9, 21};
Comparator<Short> comp = Collections.reverseOrder();
Arrays.sort(sArr, comp );
for (short number : sArr) {
     System.out.println("Number = " + number);
}
```

Array

- Một số phương thức mảng
  - equals(int[] a, int[] a2): So sánh 2 mảng bằng nhau.

```
Object o1 = new Object();
Object o2 = new Object();
Object[] a1 = { o1, o2 };
Object[] a2 = { o1, o2 };
System.out.println(a1.equals(a2)); // prints false
System.out.println(Arrays.equals(a1, a2)); // prints
true
```

Array

- Một số phương thức mảng
  - Fill(): Gán giá trị đến các phần tử trong mảng

```
int arrnum[] ={5,6,9,2,10};
for(int i=0;i<arrnum.length;i++){
  System.out.println(arrnum[i]+" ");
}
Arrays.fill(arrnum,0);
for(int i=0;i<arrnum.length;i++){
  System.out.println(arrnum[i]+" ");
}
```

Array

- Một số phương thức mảng
  - Tham khảo
  - https://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html

Chương 04

- Nội dung
  - Array
  - **Collection Framework**

Collection Framework

- Còn được gọi là container, là đối tượng nhóm nhiều phần tử trong một đơn vị. Collection có thể được dùng để lưu trữ, truy xuất, thao tác.
  - Lưu trữ: add,…
  - Truy xuất: get,…
  - Thao tác: sort,…

Collection Framework

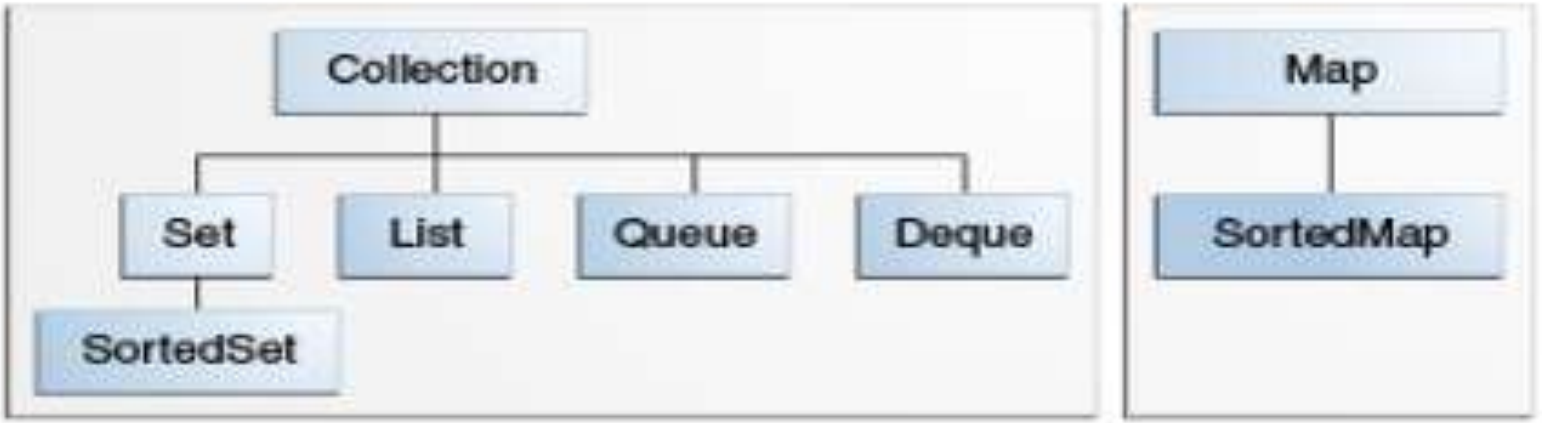
- Collection bao gồm các thành phần sau:
  - Interface: Trình bày kiểu dữ liệu trừu tượng
  - Implementation: Hiện thực kiểu dữ liệu trừu tượng
  - Algorithms: Các phương thức thực hiện tính toán như: search, sort,…

Collection Framework

- Ưu điểm
  - Giảm nổ lực lập trình: Collection cung cấp sẵn các phương thức, cấu trúc cũng như thuật toán thao tác với dữ liệu đó
  - Tăng hiệu năng chương trình: Collection có hiệu suất cao
  - https://docs.oracle.com/javase/tutorial/collections/

Collection Framework

- Interface



Collection Framework

- Interface
  - Set: Tập các phần tử không trùng nhau
  - List: Tập các phần tử có thứ tự, các phần tử có thể trùng
  - Queue: Một tập các phần tử có thứ tự ưu tiên để xử lý và hoạt động theo nguyên tắc FIFO
  - Deque: Một tập các phần tử có thứ tự ưu tiên để xử lý và hoạt động theo nguyên tắc FIFO và LIFO
  - Map: Một đối tượng ánh xạ các khóa đối với các giá trị

Collection Framework

- Interface
  - Sortedset: giống như set nhưng các phần tử theo thứ tự tăng
  - SortedMap: Giống như Map nhưng các khóa sắp xếp theo thứ tự tăng

Collection Framework

- Collection
  - Các phương thức:
    - Int size(),
    - boolean isEmpty(),
    - boolean contains(Object element),
    - boolean add(E element),
    - boolean remove(Object element),
    - Iterator<E> iterator().

Collection Framework

- Các duyệt qua từng phần tử trong collection
  - (1) aggregate operations (JDK 8)
  - (2) for-each
  - (3) Iterators.

Collection Framework

- Các duyệt qua từng phần tử trong collection
  - (1) aggregate operations (JDK 8)

```
myShapesCollection.parallelStream()
.filter(e -> e.getColor() == Color.RED)
.forEach(e -> System.out.println(e.getName()));
```

Collection Framework

- Các duyệt qua từng phần tử trong collection
  - (2) for-each

```
for (Object o : collection)
    System.out.println(o);
```

Collection Framework

- Các duyệt qua từng phần tử trong collection
  - (3) Iterators.

```
public interface Iterator<E> {
    boolean hasNext();
    E next();
    void remove(); //optional
}
```

  - Ví dụ

```
static void filter(Collection<?> c) {
    for (Iterator<?> it = c.iterator(); it.hasNext(); )
          it.remove();
  }
```

Collection Framework

- Bulk
  - addAll
  - removeAll
  - retainAll
  - containsAll

Collection Framework

- Chuyển các phần tử trong collection tới mảng 1 chiều

```
Object[] a = c.toArray();
```

Collection Framework

- Set
  - Các class thực thi:
    - AbstractSet
    - ConcurrentHashMap.KeySetView
    - ConcurrentSkipListSet
    - CopyOnWriteArraySet
    - EnumSet
    - HashSet
    - JobStateReasons
    - LinkedHashSet
    - TreeSet

Collection Framework

- HashSet
  - Không đảm bảo thứ tự phần tử
  - Có thể chứa phần tử null
  - not synchronized

Collection Framework

- HashSet
  - Ví dụ: Lưu tên của 5 học sinh vào một tập hợp Hashset

```
Set<String> hs=new HashSet<>();
int i=1;
do
  {
       System.out.println("Nhap ten: ");
       hs.add(new Scanner(System.in).nextLine());
  }while(++i<=5);
```

Collection Framework

- HashSet
  - Ví dụ: Nhập tên một học sinh và tìm kiếm tên học sinh có tồn tại trong Hashset hay không?

Nhập tên học sinh:

```
System.out.println("Nhap ten can tim kiem: ");
String name=new Scanner(System.in).nextLine();
```

Collection Framework

- HashSet
  - Ví dụ: Nhập tên một học sinh và tìm kiếm tên học sinh có tồn tại trong Hashset haykhoong?

Tìm kiếm: Cách 1

```
Iterator<String> it=hs.iterator();
while(it.hasNext())
{
    if(name.equals(it.next()))
    {
            System.out.println("Ten da tim thay");
            return;
    }
}
```

Collection Framework

- HashSet
  - Ví dụ: Nhập tên một học sinh và tìm kiếm tên học sinh có tồn tại trong Hashset haykhoong?

Tìm kiếm: Cách 2

```
if(hs.contains(name))
    System.out.println("Ten tim thay");
else
    System.out.println("Ten khong tim thay");
```

Collection Framework

- List
  - Các class thực thi:
    - AbstractList
    - AbstractSequentialList
    - ArrayList
    - AttributeList
    - CopyOnWriteArrayList
    - LinkedList
    - RoleList
    - RoleUnresolvedList
    - Stack
    - Vector

Collection Framework

- ArrayList
  - Các phần tử có thứ tự thêm vào
  - Có thể chứa phần tử null
  - not synchronized

Collection Framework

- ArrayList
  - Ví dụ

```
List<String> hs=new ArrayList();
int i=1;
do
{
        System.out.println("Nhap ten: ");
        hs.add(new Scanner(System.in).nextLine());
}while(++i<=5);

System.out.println("Nhap ten can tim kiem: ");
String name=new Scanner(System.in).nextLine();

if(hs.contains(name))
   System.out.println("Ten tim thay");
else
   System.out.println("Ten khong tim thay");
```

Collection Framework

- Tham khảo Collection FrameWork
- https://docs.oracle.com/javase/tutorial/collections/interfaces/index.html

Hỏi đáp