

**Bài 7: BÀI TẬP MINH HỌA THAO TÁC VỚI VECTOR**

Vector và đối tượng nguyên thủy- tình huống slide 12 vector

Tình huống giả thuyết	Các phương thức sử dụng
<ul style="list-style-type: none">➤ Tạo một Vector lưu trữ các số nguyên kèm theo các điều kiện sau:<ul style="list-style-type: none">• Các item không được xuất hiện quá 1 lần• Xóa một phần tử tại vị trí chỉ định• Tìm giá trị nhỏ nhất trong Vector• Xuất ra các giá trị trong Vector	<ul style="list-style-type: none">➤ add(Object item)➤ remove(int index)➤ contains(Object item)➤ Collections.min()➤ iterator()➤ size()

Thực hiện

Xây dựng lớp Vector_PrimitiveDemo.java thao chứa các thao tác với Vector

```
Vector_PrimitiveDemo.java

private Scanner scanner;
private Vector<Integer> nums;
public Vector_PrimitiveDemo() {
    //khai báo vector
    nums = new Vector<>();
    scanner = new Scanner(System.in);
}

//Thêm các phần tử không trùng nhau
public void scanUnique() {
    System.out.println("Enter the number of elements:");
    int n = scanner.nextInt();
    scanner.nextLine();
    for (int i = 0; i < n; i++) {
        System.out.printf("%dth element: ", (i + 1));
        int num = scanner.nextInt();
        scanner.nextLine();
        if (nums.contains(num)) {
            System.out.printf("Element(%d) has existed!\n", (num));
        } else {
            nums.add(num);
            i++;
        }
    }
}

//Thêm vào một số phần tử mà không kiểm soát trùng lặp
public void scan_find() {
```



```
System.out.println("Enter the number of elements:");
int n = scanner.nextInt();
scanner.nextLine();
System.out.printf("Enter a list of number(%d
elements):%n",n);
for (int i = 0; i < n; i++) {
    System.out.printf("%dth element:%n ",(i + 1));
    nums.addElement(scanner.nextInt());
    scanner.nextLine();
}
}
```

```
//In vector sử dụng vòng for
public void printList() {
    for (int num : nums) {
        System.out.print(num + " ");
    }
    System.out.println();
}
```

```
//Tìm phần tử bằng giá trị cho trước
public void findElements() {
    System.out.println("Enter the number you want to find:");
    int num = scanner.nextInt();
    scanner.nextLine();
    int count = 0;
    for (int i = 0; i < nums.size(); i++) {
        if (nums.elementAt(i) == num) {
            System.out.println(num + " was found at " + i);
            count++;
        }
    }
    System.out.printf("Total %d elements was found\n",count);
}
```

```
//Xóa phần tử với giá trị nhập vào. Xóa khi nào hết tất cả cá
phần tử tương tự trong tập hợp
public void removeElements() {
    System.out.println("Enter the element you want to remove
from list:");
    int e = scanner.nextInt();
    scanner.nextLine();
    int count = 0;
    boolean hasMore;
    do {
```



```
        hasMore = nums.removeElement(e);
        if(hasMore) count++;
    } while (hasMore);
    System.out.println("Total removed: " + count);
    System.out.println("Number list after removed:");
    printList();
}
```

//Sử dụng lớp tiện ích để sắp xếp tập hợp.

```
public void printSortedList() {
    Comparator<Integer> c=new Comparator<Integer>() {
        @Override
        public int compare(Integer o1, Integer o2) {
            return o1-o2;
        }
    };
    nums.sort(c);
    printList();
}

}
```



Lớp chứa hàm main vận hành bài toán

```
public class Main{
public static void main(String[] args) {
    Vector_PrimitiveDemo primitiveDemo=new
Vector_PrimitiveDemo();
    //enter unique element
    primitiveDemo.scanUnique();
    System.out.println("Entered list:");
    primitiveDemo.printList();
    primitiveDemo.findElements();
    primitiveDemo.removeElements();
    System.out.println("List after sorting:");
    primitiveDemo.printSortedList();
}
}
```