Homework 3: Abstract Data Type (ADT) and List ADT

**Bài 1.** Tạo danh sách (List) bằng kiểu dữ liệu mảng (array) như sau:

//Tạo giao diện ListInterface kế thừa giao diện Iterable

**public** **interface** ListInterface<T> **extends** Iterable<T>{

**public** **void** add(T data); //Thêm phần tử vào danh sách

**public** T get(**int** i); //Lấy giá trị phần tử thứ i

**public** **void** set(**int** i, T data); //Đặt data vào vị trí i của danh sách

**public** **void** remove(T data); //Loại phần tử data khỏi danh sách

**public** **void** isContain(T data); //Kiểm tra phần tử data có trong danh sách

**public** **int** size(); //Kích thước danh sách

**public** **boolean** isEmpty(); //Danh sách có rỗng hay không

}

//Tạo lớp SimpleArrayList cài đặt giao diện ListInterface

**public** **class** SimpleArrayList<T> **implements** ListInterface<T> {

**private** T[] array;

**private** **int** n = 0;

**private** **int** defaultSize = 100;

**public** SimpleArrayList() {

array = (T[]) **new** Object[defaultSize];

}

**public** SimpleArrayList(**int** capacity) {

// Hàm dựng với kích thước mảng là capacity

}

**public** **void** add(T data) {

// Hoàn thành thân hàm}

**public** T get(**int** i) {

// Hoàn thành thân hàm

}

**public** **void** set(**int** i, T data) {

// Hoàn thành thân hàm

}

**public** **void** remove(T data) {

// Hoàn thành thân hàm

}

**public** **boolean** isContain(T data) {

// Hoàn thành thân hàm

}

**public** **int** size() {

// Hoàn thành thân hàm

}

**public** **boolean** isEmpty() { // Hoàn thành thân hàm

}

**public** Iterator<T> iterator() {

//Trả về toàn bộ danh sách

// Hoàn thành thân hàm

}

}

**Bài 2.** Tạo kiểu danh sách móc nối (SimpleLinkedList) theo mẫu sau.

**public** **class** SimpleLinkedList<T> {

**class** Node {

T data;

Node next;

}

**private** Node top = **null**;

**private** Node bot = **null**;

**private** **int** n = 0;

**public** **void** add(T data) {

// Thêm phần tử vào đầu danh sách

}

**public** **void** addBot(T data) {

// Thêm phần tử vào cuối danh sách

}

**public** T get(**int** i) {

// Lấy phần tử ở vị trí thứ i

**return** **null**;

}

**public** **void** set(**int** i, T data){

// Gán giá trị ở vị trí i bằng data

}

**public** **boolean** isContain(T data) {

// Kiểm tra trong danh sách có chứa phần tử data hay không?

**return** **false**;

}

**public** **int** size() {

// Trả lại thông tin số phần tử có trong danh sách

**return** 0;

}

**public** **boolean** isEmpty() {

// Kiểm tra danh sách có rỗng hay không?

**return** **true**;

}

**public** T removeTop() {

// Xóa phần tử ở đầu danh sách, trả lại giá trị data của phần tử đó **return** **null**;

}

**public** T removeBot() {

// Xóa phần tử ở cuối danh sách, trả lại giá trị data của phần tử đó **return** **null**;

}

**public** **void** remove(T data) {

// Xóa tất cả các phần tử có giá trị bằng data

}

}