

## ĐỀ KIỂM TRA 1

### MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

#### THỜI GIAN: 45 PHÚT

Quang D. C.

dungcamquang@tdtu.edu.vn

#### I. Lưu ý khi làm và nộp bài

- Sinh viên tạo thư mục **MSSV\_HoTen**.
- Trong thư mục **MSSV\_HoTen**, sinh viên chép thư mục **Cau1** đã nhận được vào thư mục trên và tạo thêm thư mục **Cau2** chứa file **Cau2.java** và thư mục **Cau3** chứa file **Cau3.java** cùng cấp với thư mục **Cau1**. Sinh viên làm bài trực tiếp trong các thư mục này.
- Sinh viên **không** chỉnh sửa tên file, tên lớp, tên phương thức cho trước.
- Sau khi làm xong, sinh viên nén thư mục **MSSV\_HoTen** lại dưới dạng **.zip** (thư mục **MSSV\_HoTen** chứa thư mục Cau1, Cau2 và Cau3).
- Cấu trúc thư mục đúng khi nộp lại:

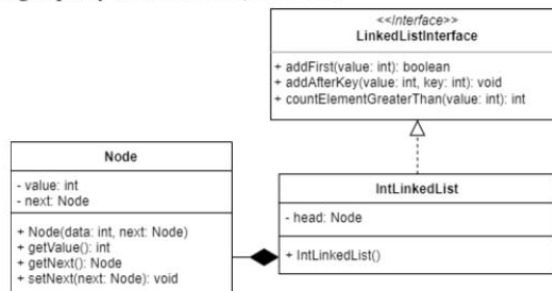
```
51203083_LeVanHung_01
├── Cau1
│   ├── IntLinkedList.java
│   ├── LinkedListInterface.java
│   ├── Test.java
│   └── Node.java
├── Cau2
│   └── Cau2.java
└── Cau3
    └── Cau3.java
```

#### II. Đề bài

##### Câu 1 (4 điểm):

Sinh viên được cung cấp sẵn các file:

- **Node.java**: là nơi định nghĩa Node để tạo Linked List.
- **LinkedListInterface.java**: là interface định nghĩa các phương thức phải có trong lớp **IntLinkedList**.
- **Test.java**: là nơi chứa sẵn khởi tạo và lời gọi đến các phương thức của lớp **IntLinkedList**. Sinh viên dùng lớp này để kiểm tra lại bài làm.



Sinh viên dựa vào sơ đồ lớp và các file cho sẵn trong thư mục **Cau1**, tạo thêm file **IntLinkedList.java**, hiện thực lớp **IntLinkedList** và định nghĩa các phương thức:

- (1.5 điểm)** Sinh viên hiện thực phương thức **public boolean addFirst(int value)** để thêm một phần tử có giá trị *value* vào đầu danh sách. Nếu trong danh sách đã có giá trị tồn tại bằng với giá trị *value* truyền vào thì không thêm và trả về *false*, ngược lại nếu không trùng thì thêm vào danh sách và trả về *true*. Sinh viên có thể tự hiện thực thêm phương thức *toString()* để tự kiểm tra lại bài làm, tuy nhiên phải đảm bảo phương thức này không phát sinh lỗi. (Sinh viên phải hiện thực interface để chạy với phương thức *main* được cung cấp sẵn và phải hiện thực đúng phương thức *addFirst* để được tính điểm các câu bên dưới)
- (1.5 điểm)** Sinh viên hiện thực phương thức **public boolean addAfterKey(int value, int key)** để thêm phần tử chứa giá trị *value* ngay sau phần tử có giá trị bằng tham số *key* đầu tiên trong danh sách, nếu thêm thành công thì trả về *true*, nếu danh sách không có phần tử nào có giá trị bằng *key* thì không thêm và trả về *false*.
- (1.0 điểm)** Sinh viên hiện thực phương thức **public int countElementGreaterThan(int value)** để trả về số lượng phần tử trong danh sách có giá trị lớn hơn tham số *value* truyền vào.

Sinh viên sử dụng file **Test.java** để kiểm tra lại bài làm.

**Câu 2 (3 điểm):** Trong file **Cau2.java** đã tạo bên trên, sinh viên hiện thực lớp **Cau2** và định nghĩa phương thức **public static String reverseIntList(int[] arr)** dùng Stack đảo ngược một mảng số nguyên và trả về chuỗi nối các kết quả đảo ngược.

Tại câu này, sinh viên được phép sử dụng thư viện Stack có sẵn của Java bằng cách thêm thư viện chứa Stack với câu lệnh **import java.util.Stack;**

*Ví dụ:* Ví dụ cho mảng `int arr[] = {1,2,3,4,5}` thì kết quả trả về sẽ là chuỗi "54321".

*Gợi ý:* Sinh viên dùng `Stack<Integer>` tạo chuỗi ngược để trả về.

Sinh viên tự viết phương thức **main** để kiểm tra lại bài làm. Lưu ý phương thức **main** gây lỗi cho bài làm thì sinh viên bị 0 điểm cả bài.

**Câu 3 (3 điểm):** Trong file **Cau3.java** đã tạo bên trên, sinh viên hiện thực lớp **Cau3** và định nghĩa phương thức **public static int productOfEvenDigits(int n)** để trả về tích các chữ số là chữ số chẵn trong số nguyên dương *n* bằng phương pháp **đệ quy**.

*Sinh viên dùng vòng lặp để giải bài này sẽ không được điểm.*

Sinh viên tự viết phương thức **main** để kiểm tra lại bài làm. Lưu ý phương thức **main** gây lỗi cho bài làm thì sinh viên bị 0 điểm cả bài.

--- HẾT ---