

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:

```

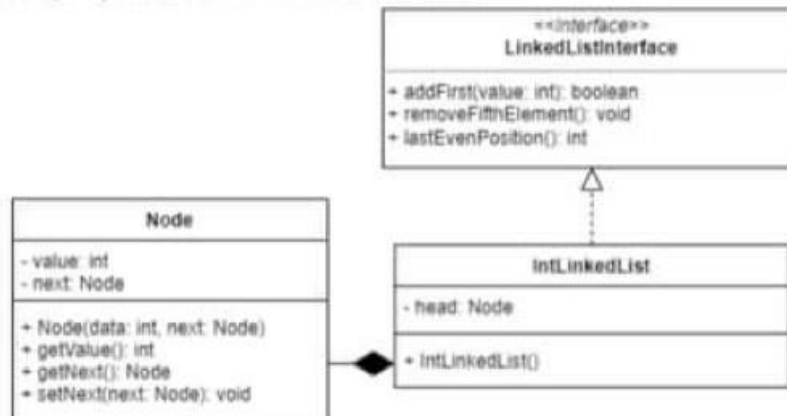
51203083_LeVanHung_01
├── Cau1
│   ├── IntLinkedList.java
│   ├── LinkedListInterface.java
│   ├── Test.java
│   └── Node.java
├── Cau2
│   └── Cau2.java
└── Cau3
    └── Cau3.java
    
```

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 void removeFifthElement()** để xóa phần tử thứ 5 trong danh sách. Nếu danh sách có ít hơn 5 phần tử thì trả về ngoại lệ *NoSuchElementException* và không xóa. Ngược lại nếu danh sách có số lượng phần tử ≥ 5 , thì xóa phần tử thứ 5 của danh sách.
- (1.0 điểm)** Sinh viên hiện thực phương thức **public int lastEvenPosition()** để trả về vị trí của phần tử chẵn cuối cùng trong danh sách. Vị trí phần tử trong danh sách bắt đầu từ 0, nghĩa là phần tử *head* có vị trí là 0. Trường hợp danh sách rỗng hoặc danh sách không chứa số chẵn thì trả về -1.

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 reverseSentence(String str)** đảo ngược thứ tự từ của một câu. 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: Cho câu "I am a student" thì kết quả của hàm này là "student a am I".

Gợi ý: Sinh viên dùng **Stack<String>** và hàm **split()** của **String**.

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 countSquareNumber(int []a, int k)** để trả về số lượng số chính phương trong mảng **a** bằng phương pháp đệ quy.

Số chính phương là số lấy căn bậc 2 ra một số tự nhiên.

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 ---