Ton Duc Thang University
Faculty of Information Technology

ĐỀ KIỂM TRA 2
MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT
THỜI GIAN: 50 PHÚT

Quang D. C.
dungcamquang@tdtu.edu.vn

I. Lưu ý khi làm và nộp bài

- Sinh viên tạo thư mục **MSSV_HoTen**.
- Chép các file được cung cấp sẵn vào thư mục này.
- Sinh viên **không** chỉnh sửa tên file, tên lớp, tên phương thức cho trước. Sinh viên không được code thêm vào file **Student.java**.
- Sau khi làm xong, sinh viên kiểm tra lại bài làm. Sau khi kiểm tra bài làm hoàn tất, sinh viên tiến hành xóa các file **Student.java**, **Main.java**, **data.inp** và các file .class; chỉ nộp lại file **BST.java** và **Node.java**. Sinh viên nén thư mục **MSSV_HoTen** lại dưới dạng **.zip** (thư mục **MSSV_HoTen** chứa các file bài làm của sinh viên).
- Cấu trúc thư mục đúng khi nộp lại:

```
51203083_LeVanHung_01
├── BST.java
└── Node.java
```

II. Đề bài (7 điểm)

Một lớp tạo cây nhị phân tìm kiếm được định nghĩa để lưu trữ các đối tượng học sinh **Student**.

Mỗi học sinh **Student** sẽ chứa thông tin gồm *id* (mã học sinh), *name* (tên học sinh) và *score* (điểm số).

Các file cung cấp sẵn:

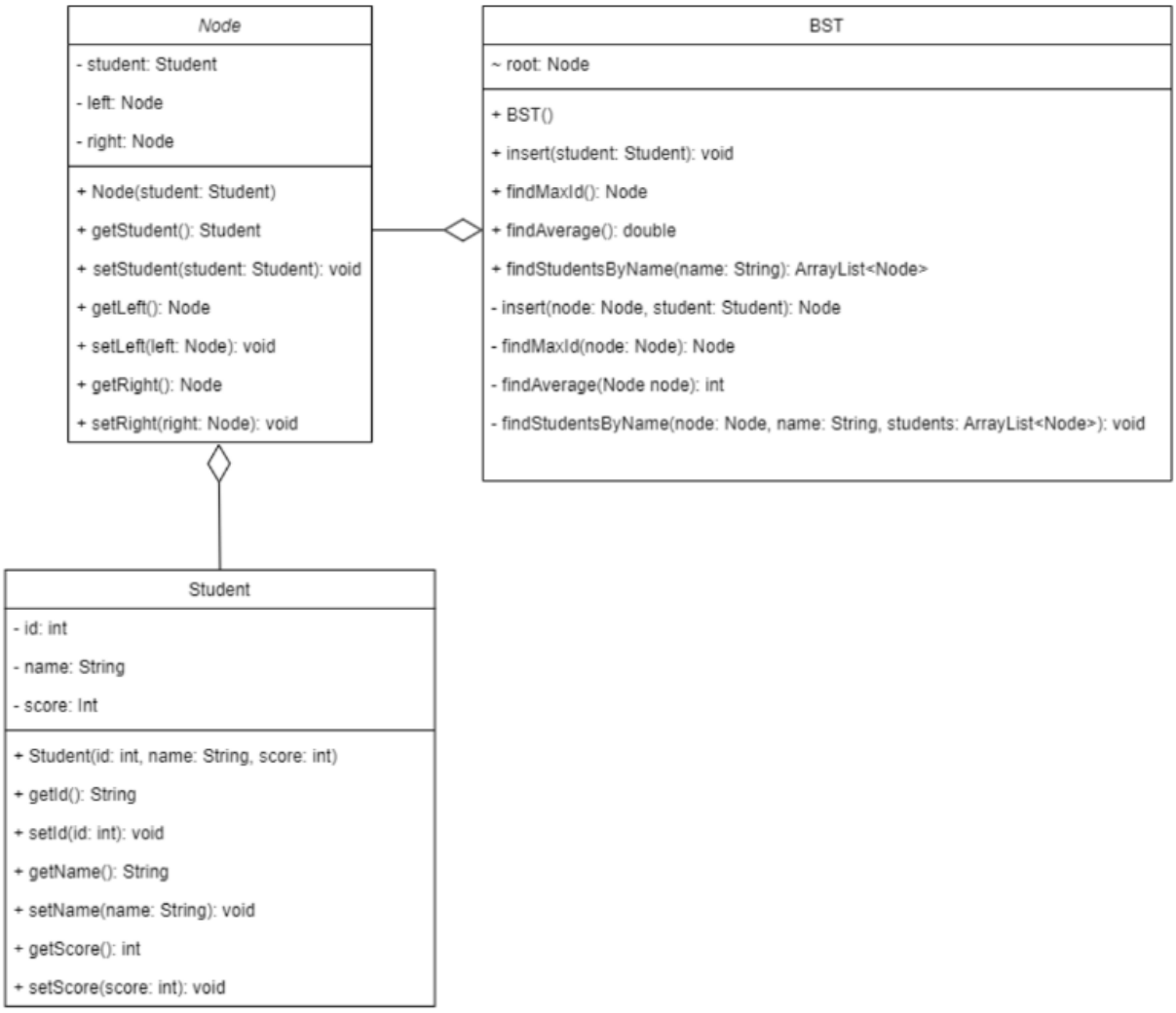
Student.java – Chứa lớp Student và các phương thức ***get*** để lấy dữ liệu.

Main.java – Chứa các dòng lệnh để kiểm thử với testcase cho sẵn.

Node.java – Node chứa đối tượng Student. Sinh viên phải định nghĩa file này.

BST.java – Chứa lớp định nghĩa cây nhị phân tìm kiếm Student. Sinh viên code thêm vào các phương thức trống của file này.

data.inp – Chứa dữ liệu để đọc vào cây nhị phân. (format dòng theo thứ tự là id,name,score)



Sinh viên định nghĩa lớp **Node** theo sơ đồ lớp bên trên với phương thức khởi tạo có tham số là *this.student = student* và *left = right = null*.

Lớp BST đã được định nghĩa sẵn các phương thức public để gọi đến root (trừ phương thức của câu 3), sinh viên đọc hiểu mã nguồn và định nghĩa vào các phương thức còn trống:

1) (2 điểm) **private Node insert(Node node, Student student)** để thêm một node chứa đối tượng Student vào cây. Thuộc tính để xét khi insert là mã số (*id*) của học sinh. Cây nhị phân tìm kiếm mang tính chất giá trị của các nút bên trái sẽ bé hơn giá trị nút đang xét và giá trị các nút bên phải sẽ lớn hơn hoặc bằng các nút đang xét. Trong file **Main.java** cung cấp sẵn phương thức đọc file gọi đến phương thức **insert** của cây. Sinh viên làm đúng câu này thì các câu bên dưới mới được tính điểm.
2) (1 điểm) **private Node findMaxId(Node node)** để tìm ra Node chứa học sinh có mã học sinh lớn nhất.
3) (2 điểm) Hiện thực phương thức **private int findAverage(Node node)** để tính tổng điểm số của tất cả các học sinh có trong cây. Và hiện thực **public double findAverage()** để tính trung bình điểm số của tất cả học sinh. (Gợi ý: Trung bình điểm số là lấy tổng điểm số của tất cả học sinh chia cho số nút của cây)

4) (2 điểm) **private void findStudentsByName(Node node, String name, ArrayList<Node> students)** để tìm ra các nút chứa học sinh có tên trùng khớp với giá trị ***name*** truyền vào và thêm vào danh sách ***students***. (Không quan trọng thứ tự của các phần tử trong danh sách)

Lớp **Main.java** đã cung cấp sẵn một số testcase mẫu trong phương thức main, kết quả in ra màn hình khi sinh viên làm đúng với testcase mẫu:

```
Cau 2:
15
Cau 3:
13.75
Cau 4:
4 1 6
```

Sinh viên có thể chỉnh sửa phương thức main để kiểm tra thêm các trường hợp khác.

Sau khi đã kiểm thử xong bài làm, sinh viên chỉ giữ lại 02 file **BST.java, Node.java** và xóa các file khác (kể cả file .class), nén bài làm và nộp theo hướng dẫn ở Phần I.

--- HẾT ---