

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

\*\*\*



# BÀI TẬP LỚN

*Đề tài:* **Visualization of operations on tree data structures**

*Học phần: Lập trình hướng đối tượng*

*Mã học phần: IT3070*

*Giảng viên hướng dẫn: Nguyễn Thị Thu Trang*

## Mục lục

<b>Thành viên và công việc .....</b>	<b>3</b>
<b>Mô tả dự án.....</b>	<b>3</b>
Yêu cầu .....	3
UseCase Diagram.....	4
<b>Thiết kế.....</b>	<b>5</b>

## I. Thành viên nhóm và công việc

### 1) Trần Thị Phương 20215123(Leader):

- Phân công công việc cho các thành viên
- Xây dựng cấu trúc project
- Xây dựng Class diagram
- Xây dựng các lớp
- Xây dựng giao diện
- Xây dựng controller

### 2) Lương Phúc Quang 20215125 :

- Xây dựng biểu đồ UseCase
- Xây dựng giao diện chính

### 3) Đỗ Hồng Quân 20194650

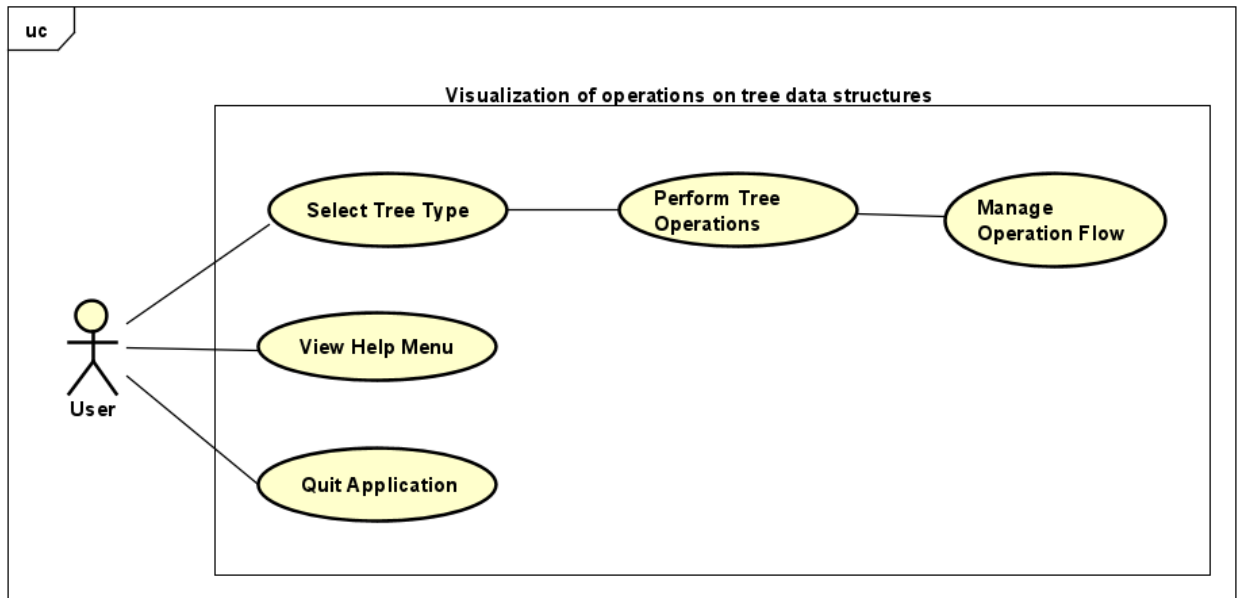
### 4) Dương Đăng Quang 20210716

## II. Mô tả dự án

### 1. Yêu cầu:

- Xây dựng mô hình trực quan hóa cấu trúc dữ liệu cây với các thao tác create, insert, update, traverse.
- Các loại cây :
  - Cây tổng quát: một nút có thể có nhiều nút con.
  - Cây nhị phân tìm kiếm: một nút có tối đa 2 nút con và các nút con nhỏ hơn nút gốc sẽ nằm ở cây con bên trái, còn lại nằm ở cây con bên phải.
  - Cây AVL: cây nhị phân tìm kiếm nhưng có thể tự cấu trúc để chiều cao cây con trái và cây con phải không lệch nhau quá 1.
  - Cây cân bằng: có cả hai tính chất của cây nhị phân tìm kiếm và cây cân bằng.
- Giao diện:
  - Các chức năng create, insert, delete, update, traverse, search.
  - Phần hiển thị code khi thực hiện
  - Các nút chức năng như dừng, tiếp tục, trở về

### 2. Biểu đồ UseCase:

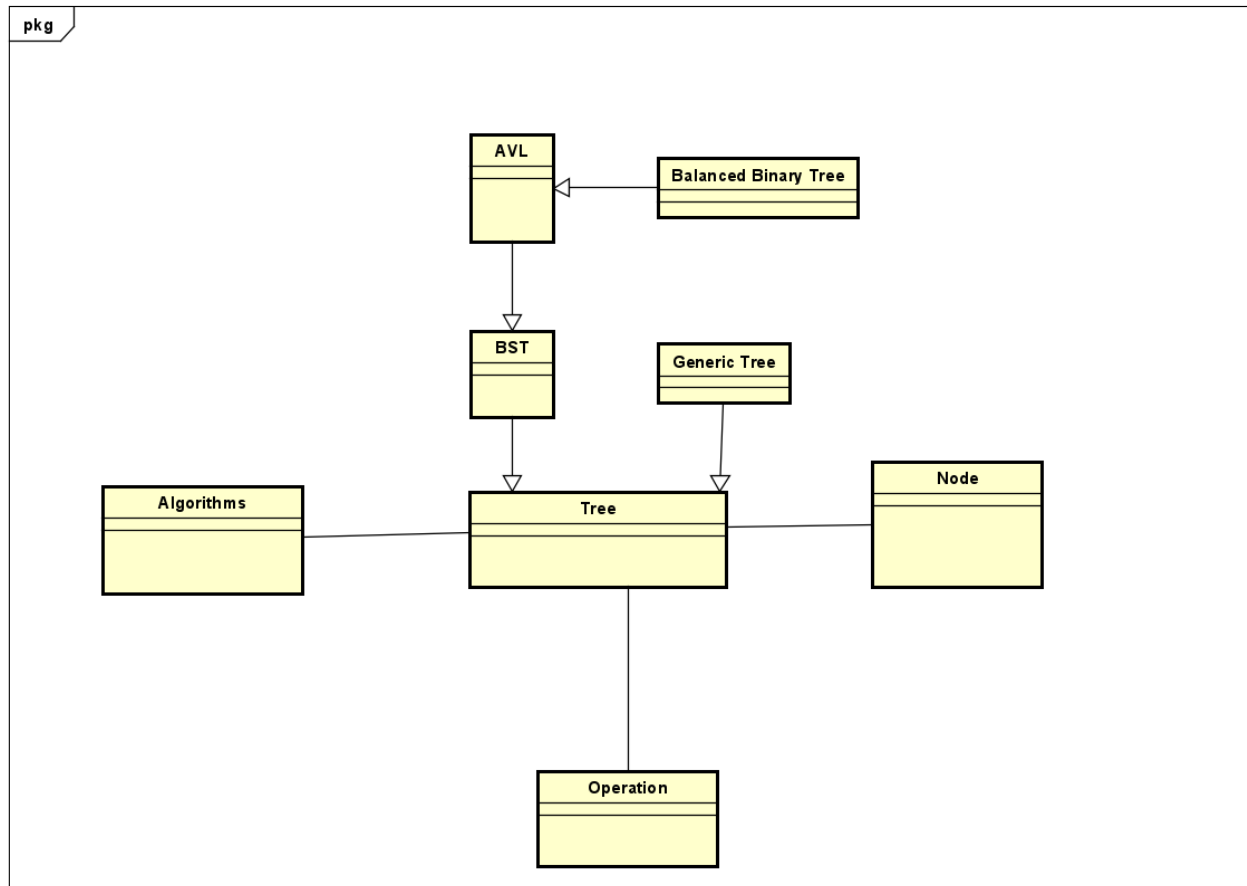


Người dùng có thể thực hiện các thao tác cơ bản sau:

- Chọn cấu trúc cây, chọn các chức năng mà mình muốn thông qua các nút bấm.
- Người dùng có thể tương tác với dữ liệu như thêm nút vào cây, tìm kiếm trên cây, xóa nút trên cây thông qua dữ liệu mà người dùng cung cấp.
- Người dùng có thể đổi qua giao diện của loại dữ liệu cây khác.

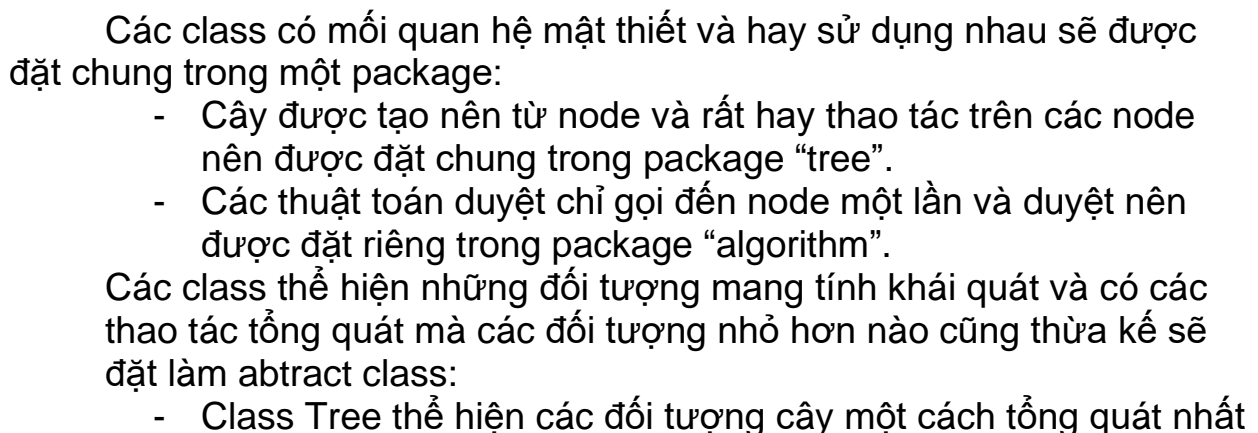
### 3. Thiết kế

#### a) Biểu đồ Class tổng quát



- Class Tree dùng để biểu thị các đối tượng là cấu trúc dữ liệu cây, đây là một abstract class.
- Các class BSTree, GenericTree, AVLTree, BalanceTree biểu thị các đối tượng hẹp hơn thuộc mỗi loại cấu trúc dữ liệu khác nhau.
- Class Node biểu thị các đối tượng Node là một phần tạo nên cây.
- Class Algorithm biểu diễn các thuật toán ta có thể thực hiện trên cấu trúc dữ liệu cây như duyệt theo chiều rộng(BFS), duyệt theo chiều sâu(DFS).

pkg



bao gồm các hành động như tạo cây rỗng, chèn, xóa, tìm kiếm. Vì thế Tree là một abstract class.

- Các lớp con thừa kế lớp Tree bao gồm các class BSTree và GenericTree.
- Cây AVL và cây cân bằng là trường hợp của cây nhị phân tìm kiếm vì thế 2 class này sẽ kế thừa lớp BSTree.
- Cấu trúc dữ liệu của node trên cây nhị phân tìm kiếm và cây tổng quát được kế thừa từ lớp cha là lớp Node, sau đó node của cây AVL là node đặc biệt với cấu trúc mở rộng từ node của cây nhị phân nên AVLNode sẽ là lớp con của Bnode.
- Phương thức duyệt của Algorithm chỉ có thể gọi mà không được tạo nên Algorithm được thiết kế là một interface với các đối tượng implement là các thuật toán như duyệt theo chiều rộng, chiều sâu...

Mối quan hệ giữa các class:

- Node và tree là mối quan hệ Aggregation vì khi tạo cây mà cần có node thì phải tạo nhưng cây có thể không có node nào.
- Tree và Algorithm là mối quan hệ Association vì phương thức traverse của tree gọi đến các thuật toán sử dụng mà không có ràng buộc gì.

Một số chi tiết về các class:

- Các cây sẽ có attribute là root node, constructor được tạo với giá trị là số nguyên và các node con hoặc node anh em.
- Các method :
  - Create: tạo một cây rỗng.
  - Insert: nhận vào một số nguyên, tạo ra một node và gắn node đó vào cây.
  - Delete: nhận vào một giá trị nguyên và xóa node trên cây có giá trị bằng giá trị nhận vào.
  - Search: nhận vào một số nguyên tìm ra node có giá trị bằng số nguyên đó
  - Traverse: nhận vào một đối tượng là một thuật toán và duyệt cây.