**Table of contents**

Mini Project:

**Object Oriented Programming**

Visualization of operations on tree data structures

**Professor: PhD Nguyen Thi Thu Trang**

Teaching Assistance: Nguyen Minh Chau

Group 2:

Ngo Quang Huy - 20184272

Nguyen Viet Thang – 20184306

Nguyen Duc Dat - 2018421

1. **Introduction…………………………………………………..……**3
2. **Duty Roster……………………………………………….….…….**4
3. **Requiments……………………………………………………..….**5
4. **Use Case Diagram………………………….…………….……..**6
5. **General Class Diagram……..……………………………..….**7
6. **Class Diagram For Each Package ……….…………….….**8
   1. **Class diagram for “tree.binary” package….……….….**8
   2. **Class diagram for “tree.node” package…………………**9
   3. **Class diagram for “tree.balance” package…………….**9
   4. **Class diagram for “tree.controller” package……….**10
   5. **Class diagram for “tree.exception” package……….**10
7. **Layout for GUI…………………………………………………..**11
8. **Preferrents…………………………………………………….….**11

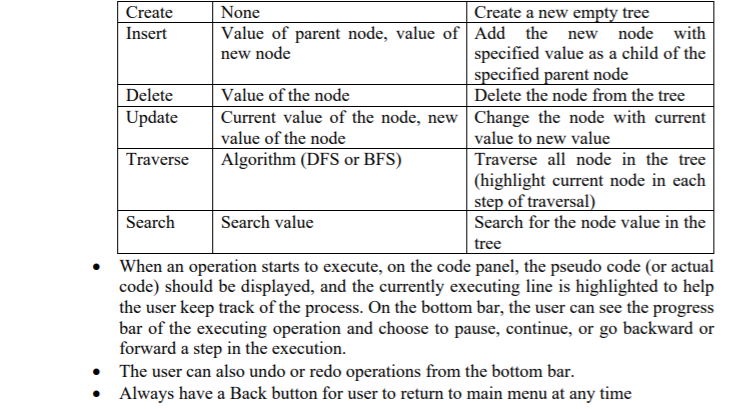
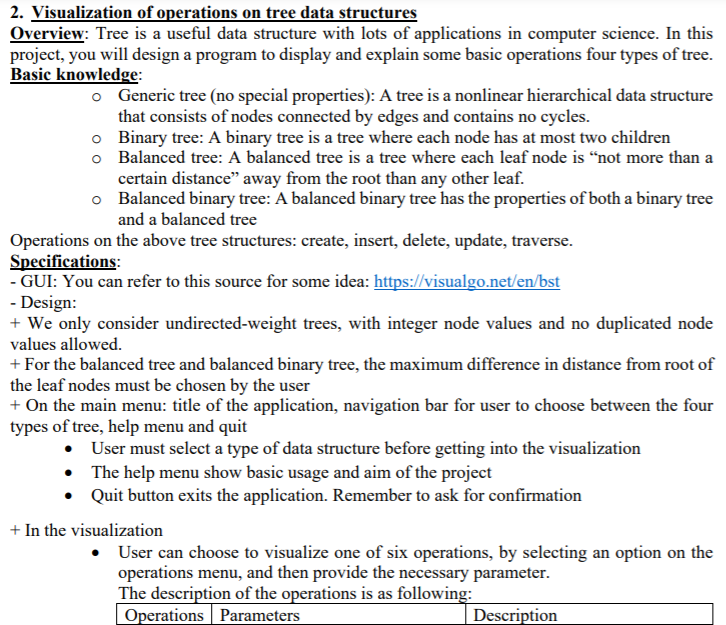
**1. Introduction**

* In [computer science](https://en.wikipedia.org/wiki/Computer_science), a tree is a widely used [abstract data type](https://en.wikipedia.org/wiki/Abstract_data_type) that simulates a hierarchical [tree structure](https://en.wikipedia.org/wiki/Tree_structure), with a root value and subtrees of children with a [parent node](https://en.wikipedia.org/wiki/Tree_(data_structure)#Terminology), represented as a set of linked [nodes](https://en.wikipedia.org/wiki/Node_(computer_science)).
* A tree data structure can be defined [recursively](https://en.wikipedia.org/wiki/Recursion) as a collection of nodes (starting at a root node), where each node is a data structure consisting of a value, together with a list of references to nodes (the "children"), with the constraints that no reference is duplicated, and none points to the root.
* Today our team will use Java FX to visualize how a tree work.

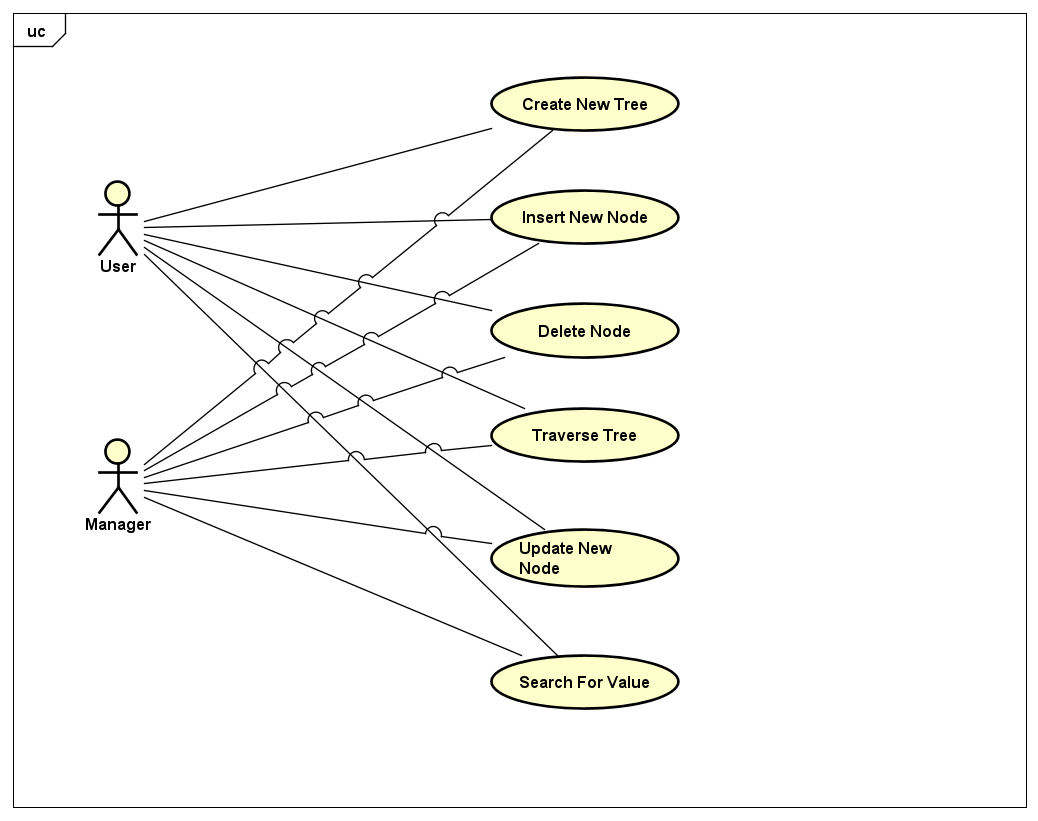
**2.** **Duty Roster**

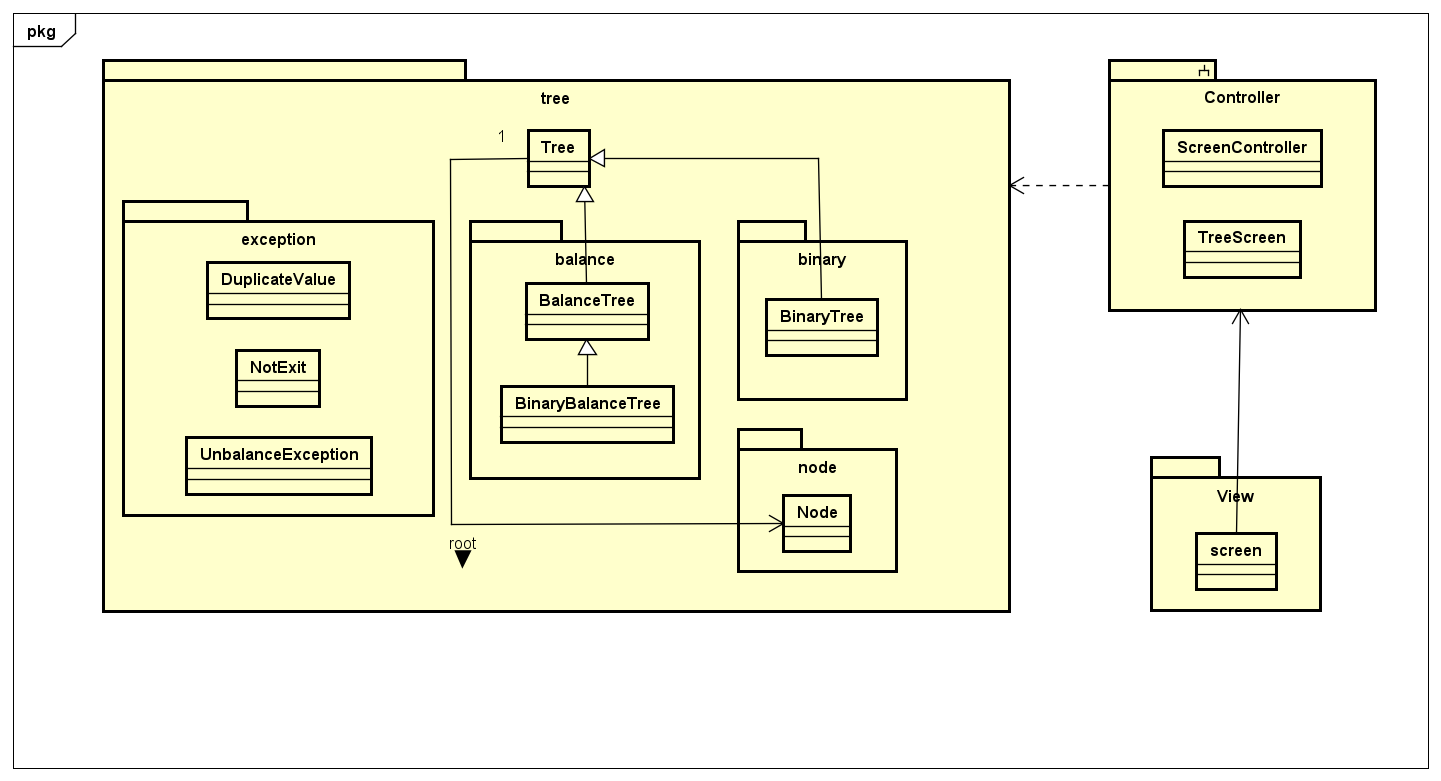
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Full Name | Email | Telephone Number | Responsibility of each member | Evaluate |
| Ngo Quang Huy | huy.nq184272@sis.hust.edu.vn | 0836869754 | Use JavaFX to design GUI for the project | Good |
| Nguyen Viet Thang | thang.nv184306@sis.hust.edu.vn | 0367120619 | Work with balance tree.  Support Huy to make GUI | Good |
| Nguyen Duc Dat | dat.nd184241@sis.hust.edu.vn | 0989481493 | Work with binary tree. Make usecase diagram and class diagram | Good |

**3. Requirements**

****

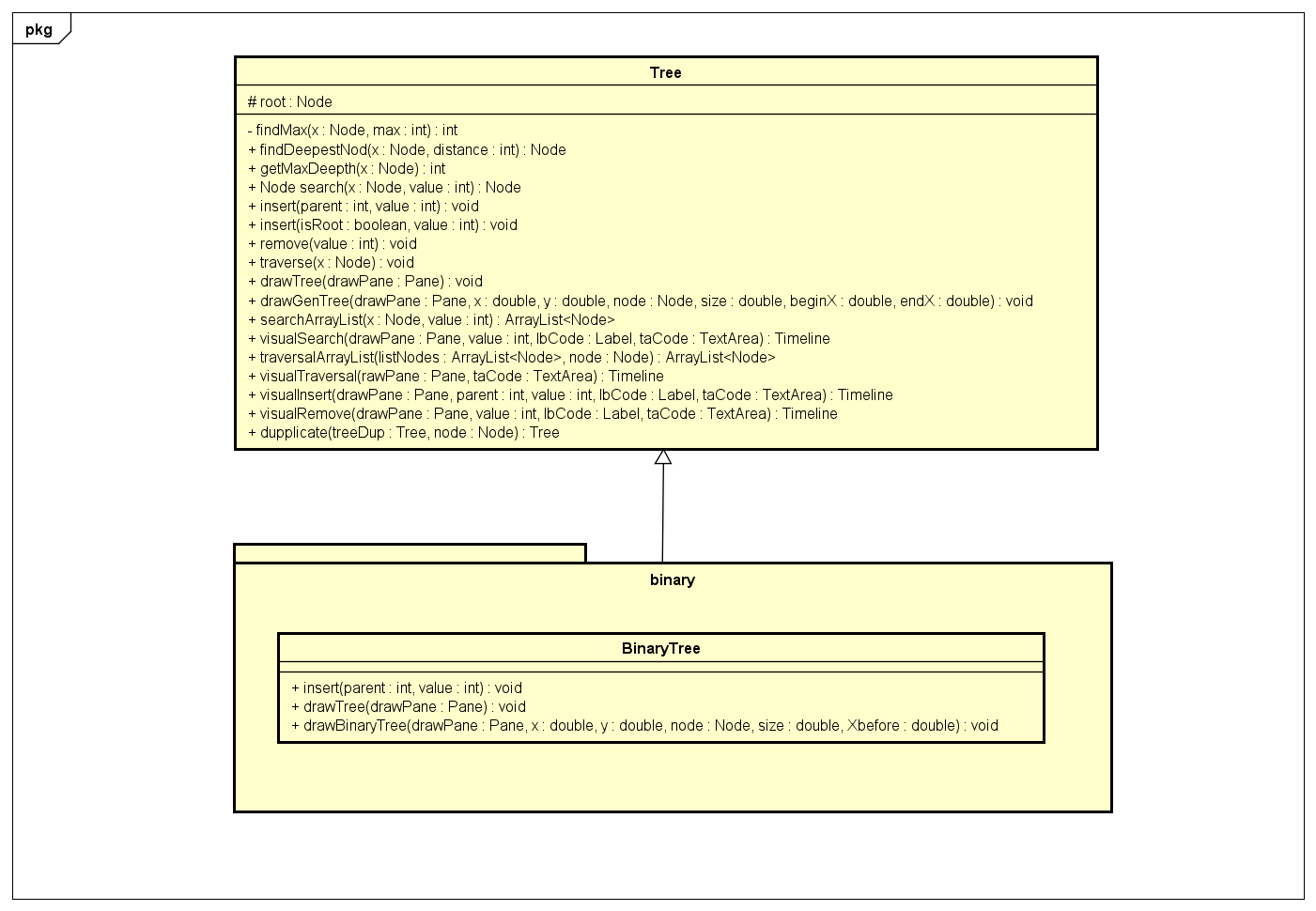
**4. UseCase Diagram**



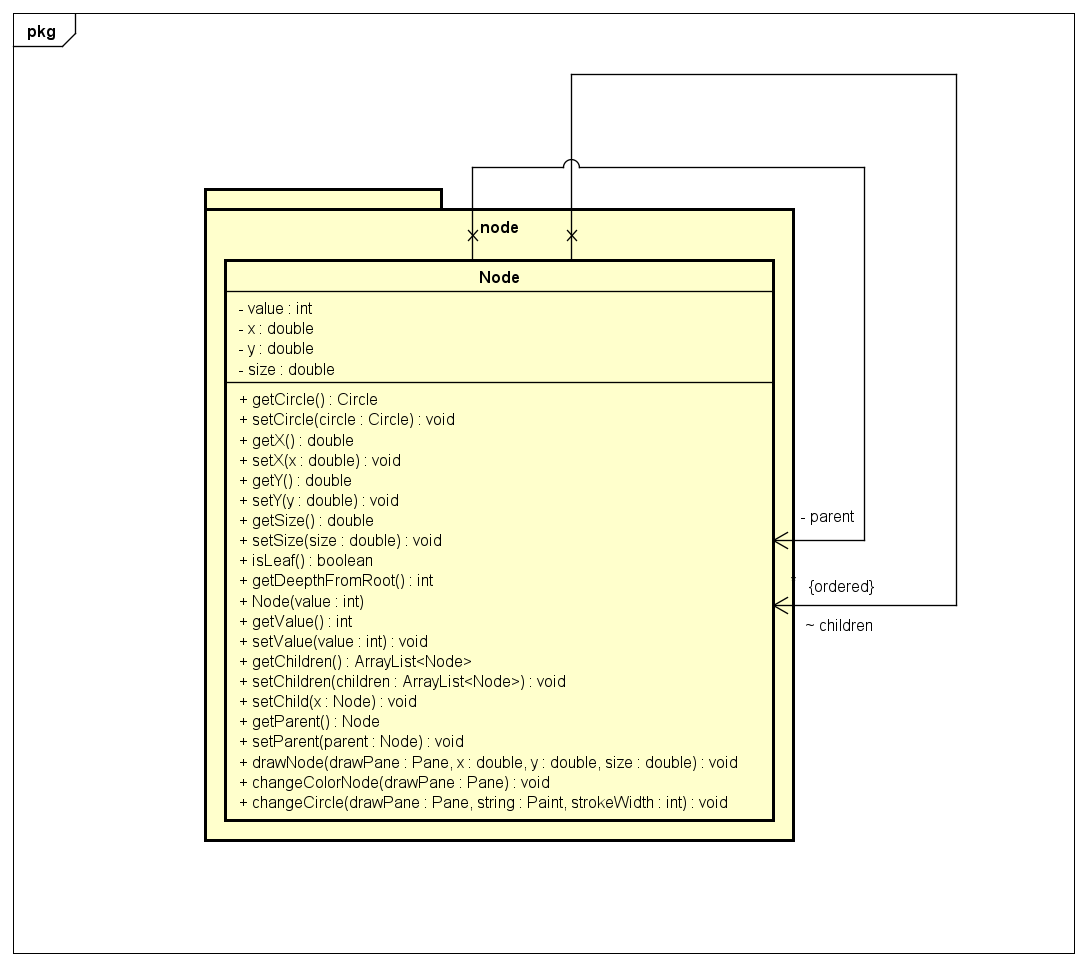
**5. General Class Diagram**

**6. Class Diagram For Each Package**

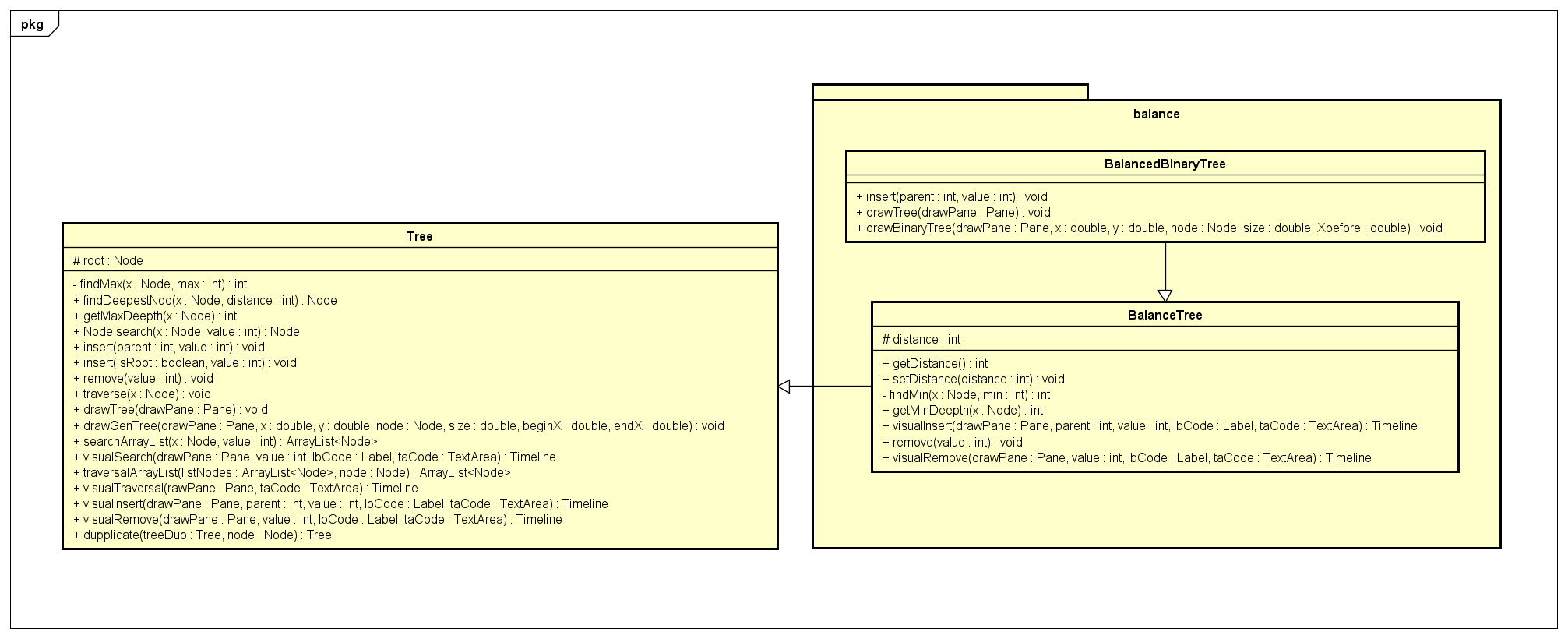
**6.1. Class diagram for “tree.binary” package.**

****

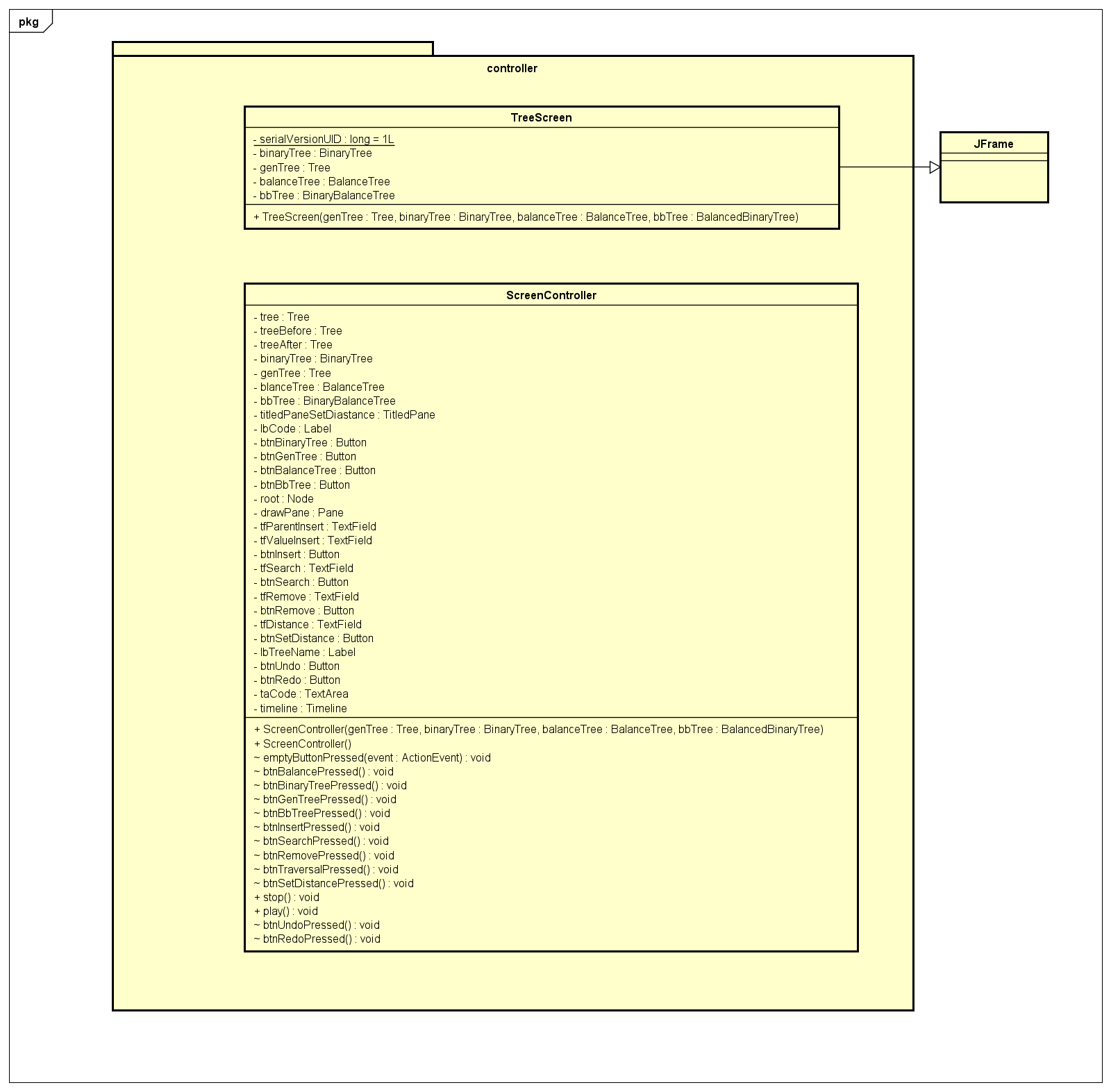
**6.2. Class diagram for “tree.node” package.**

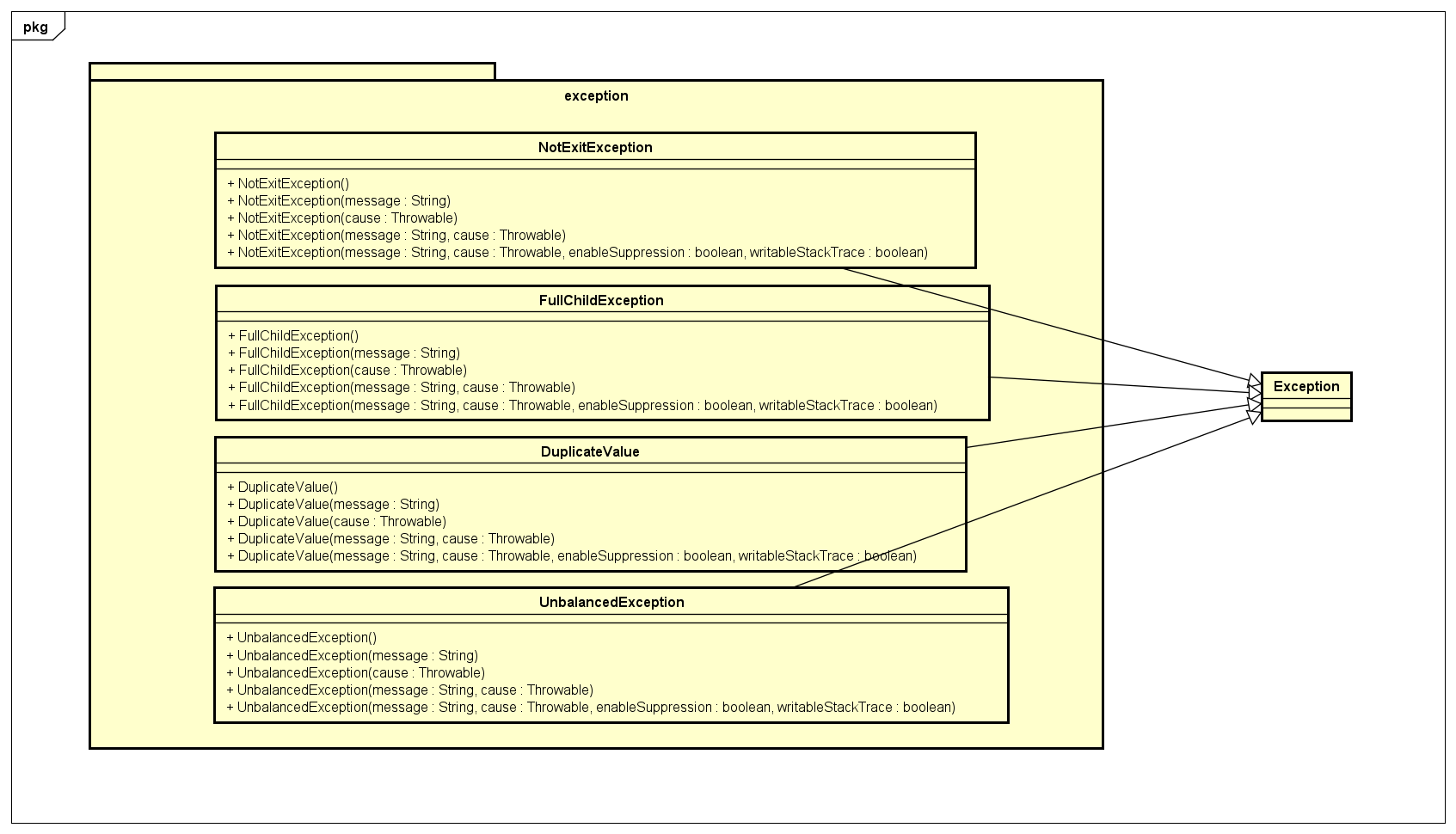
****

**6.3. Class diagram for “tree.balance” package.**

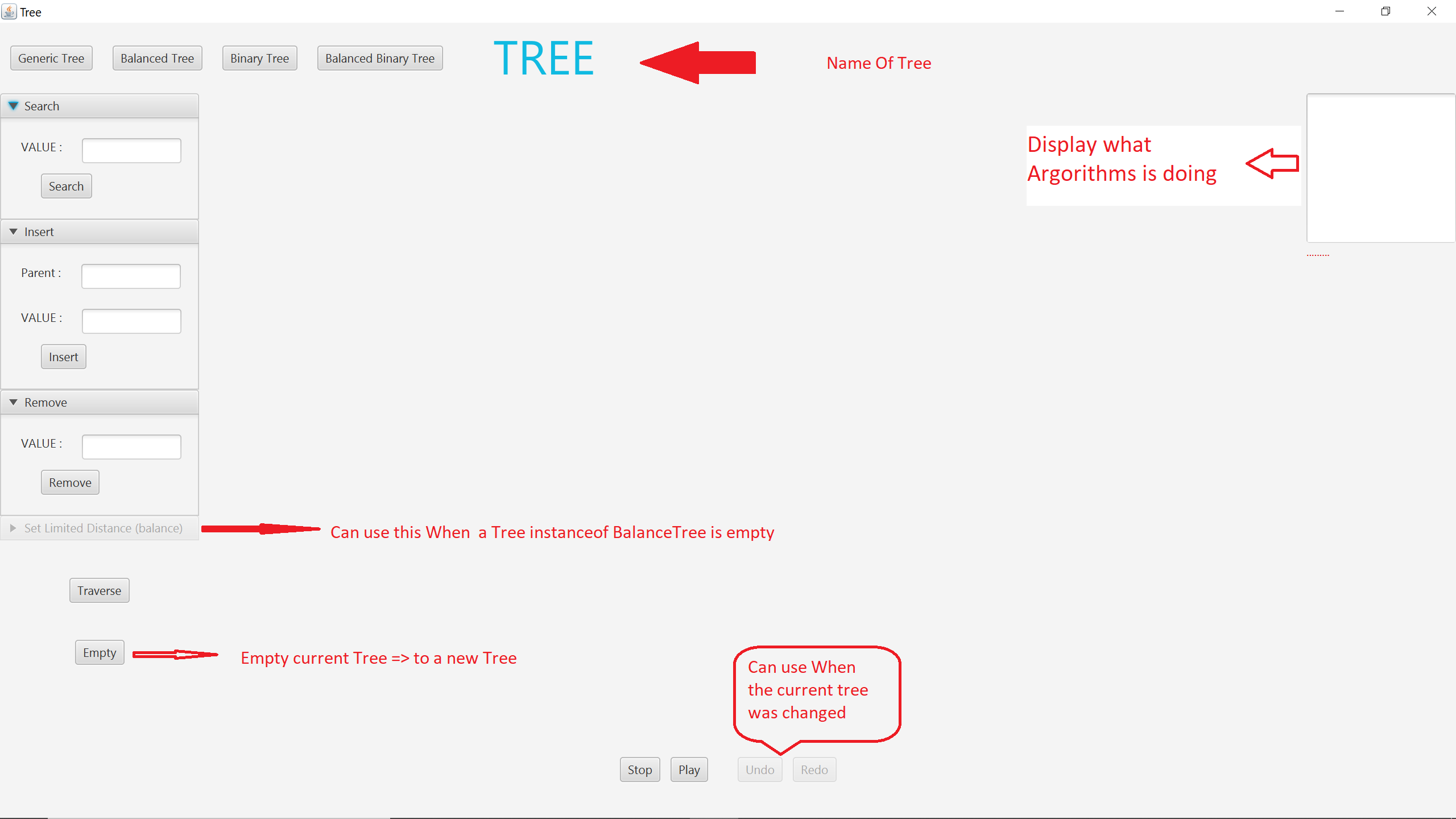


**6.4. Class diagram for “tree.controller” package**

****

**6.5. Class diagram for “tree.exception” package **

**7. Layout For GUI**



**8. Preferrences**

* [**https://stackoverflow.com/questions/19330731/tree-implementation-in-java-root-parents-and-children?fbclid=IwAR1Z4bNnALXsv8LrAn-RxMYVjdpsp1DdzhdS6E423Ge2YhZYPUtCX8NMZ0k**](https://stackoverflow.com/questions/19330731/tree-implementation-in-java-root-parents-and-children?fbclid=IwAR1Z4bNnALXsv8LrAn-RxMYVjdpsp1DdzhdS6E423Ge2YhZYPUtCX8NMZ0k)
* [**https://en.wikipedia.org/wiki/Tree\_(data\_structure)**](https://en.wikipedia.org/wiki/Tree_(data_structure))
* **https://www.youtube.com/watch?v=814FJoUKFLw**