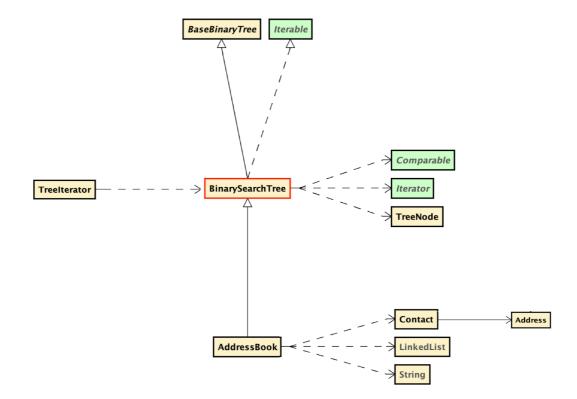
Author StuID: 2020215138 Author Name: Huang Kaisheng

## **UML Class Relationship Diagram:**



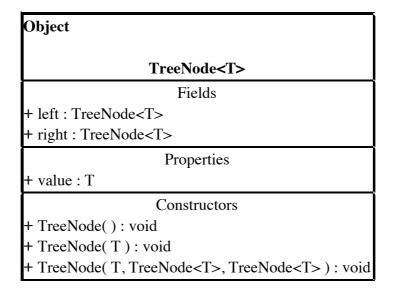
## **UML Diagram of BaseBinaryTree**

# Fields # root: TreeNode<T> Constructors + BaseBinaryTree(): void + BaseBinaryTree(T): void Properties + isEmpty(): boolean + makeEmpty(): void + getRoot(): T

## **UML Diagram of BinarySearchTree**

# BinarySearchTree<T extends Comparable<T>> Constructors + BinarySearchTree(): void Methods - get(T, TreeNode<T>): T + get(T): T - insert(T, TreeNode<T>): TreeNode<T> + insert(T): void + iterator(): Iterator<T> - remove(T, TreeNode<T>): TreeNode<T> + remove(T): void

## **UML Diagram of TreeNode**



## **UML Diagram of Treelterator**

## Fields restack: Stack<TreeNode<T>> raversalNode: TreeNode<T> Constructors TreeIterator( BaseBinaryTree<T>): void Methods hasNext(): boolean next(): T

## **UML Diagram of Address**

# Address Properties + city: String + state: String + street: String + zipCode: int Constructors + Address( String, String, String, int ): void + Address( String, String, String, String): void Methods + compareTo( Address): int «volatile» «synthetic» «bridge» + compareTo( Object): int

## **UML Diagram of Contact**

+ toString(): String

# Fields ~ firstName : String ~ lastName : String Properties + address : Address + name : String + phone : long + phoneAreaCode : long «readOnly» + phoneFormatted : String Constructors + Contact( String, String, Address, String ) : void

## **UML Diagram of AddressBook**

## BinarySearchTree

### AddressBook

### Constructors

+ AddressBook(): void

### Methods

- + getContactByAreaCode( int ) : LinkedList<Contact>
- + getContactByCity( String ) : LinkedList<Contact>
- + getContactByZipCode( int ) : LinkedList<Contact>