Test Design

Scenarios Setup

Name	Class	Scenario
setup1	BinaryTreeTest	A Binary tree with the following keys = {1,2,3,4,5,6,7,8,9,10}. The values of each node are the same as the keys (ex. key 1, value 1).
setup2	BinaryTreeTest	A Binary tree with the following keys = {50,100,0,25,-25,125,75,30}. The values of each node are the same as the keys (ex. key 50, value 50).
setup3	BinaryTreeTest	An empty binary tree.
setup4	AVLTreeTest	An AVL tree with the following keys = {1,2,3,4,5,6,7,8,9,10}. The values of each node are the same as the keys (ex. key 1, value 1).
setup5	AVLTreeTest	An AVL tree with the following keys = {50,100,0,25,-25,125,75,30}. The values of each node are the same as the keys (ex. key 50, value 50).
setup6	AVLTreeTest	An empty AVL tree ;

Test objective:
To check if add works properly, adding known numbers and then asserting the numbers’ existence in the Binary Tree by taking the postorder traversal of the tree and matching it with the expected String.

Class	Method	Scenario	Input values	Result
BinaryTree	Insert	setup2 setup1	postOrder() postOrder()	true true

Test objective:
To check if search works properly, searching 3 existent numbers and a non existent number in the Binary Tree.

Class	Method	Scenario	Input values	Result

BinaryTree	search	setup1	search(1)
			search(5)
			search(10)
			search(13)
				True
				True
				True
				Null

Test objective:
To check if remove works properly, removing and then searching the removed numbers, and removing a non existent number in the Binary Tree.

Class	Method	Scenario	Input Values	Result

BinaryTree	remove	setup2	remove(50)
			search(50)
			remove(25)
			search(25)
			remove(-25)
			search(-25)
			remove(13)
				true
				null
				true
				null
				true
				null
				false

Test objective:
To check if height works properly, checking the height of specific nodes and comparing them to the expected results.

Class	Method	Scenario	Input Values	Result

BinaryTree	height	setup2	height(50)
			height(50)
			height(100)
			height(-25)
				true
				false
				true
				true

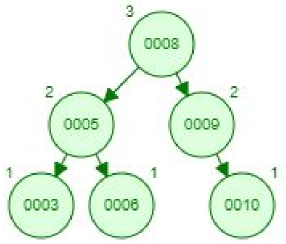
Test objective:
To check if insert works properly, checking the height of specific nodes and comparing them to the expected results.

Class	Method	Scenario	Input Values	Result

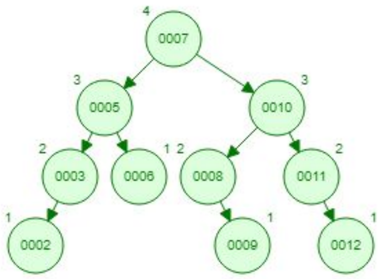
AVLTree	insert	setup4	searchAVLTree(1)
			searchAVLTree(5)
			searchAVLTree(10)
			searchAVLTree(13)
				true
				true
				true
				false

Test objective:
To check if the balance and insert methods work properly, inserting three values and checking if the tree has been balanced. In this case we want to test all the balance cases.

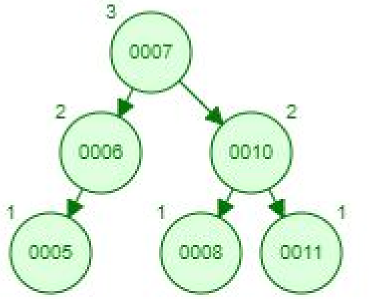
Class	Method	Scenario	Input values	Result

AVLTree	balance	setup6	case A: leftRotate(p)
			insert(5)
			insert(3)
			insert(8)
			insert(6)
			insert(9)
			insert(10)
				 3
				0008
				2
				0005
				2
				0009
				1
				0003
				1
				0006
				1
				0010
				Check if the key of the root is 8, his left children is 5, his right children is 9 and the right children of his left its 6

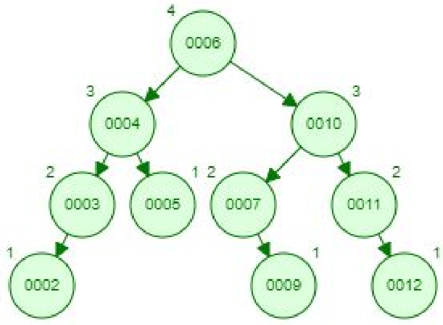
Class	**Method**	**Scenario**	**Input values**	**Result**

AVLTree	balance	setup6	case C: rightRotate(q)
			leftRotate(p)
			insert(5)
			insert(10)
			insert(3)
			insert(2)
			insert(7)
			insert(11)
			insert(12)
			insert(6)
			insert(8)
			insert(9)
				 4
				0007
				3
				0005
				3
				0010
				2
				0003
				1
				0006
				2
				0008
				2
				0011
				1
				0002
				1
				0009
				1
				0012
				Check if the key of the root is 7, his left children is 5, his right children is 10, the right children of his left its 6 and the left children of its right its 8.

Class	**Method**	**Scenario**	**Input values**	**Result**

AVLTree	balance	setup6	case D: rightRotate(p)
			insert(10)
			insert(11)
			insert(7)
			insert(6)
			insert(8)
			insert(5)
				 3
				0007
				2
				0006
				2
				0010
				1
				0005
				1
				0008
				0011
				1
				Check if the key of the root is 7, his left children is 6, his right children is 10 and the left children of his right its 8

Class	Method	Scenario	Input values	Result

AVLTree	balance	setup6	case F: leftRotate(q)
			rightRotate(p)
			insert(10)
			insert(11)
			insert(4)
			insert(3)
			insert(6)
			insert(12)
			insert(5)
			insert(7)
			insert(2)
			insert(9)
				 4
				0006
				3
				0004
				3
				0010
				2
				0003
				0005
				1
				2
				0007
				0011
				2
				1
				0002
				0009
				1
				0012
				1
				Check if the key of the root is 6, his left children is 4, his right children is 10 , the left children of his right its 7 and the right children of its left its 5

Test objective:
To check if remove works properly, removing and then searching the removed numbers, and removing a non existent number in the Binary Tree.

Class	**Method**	**Scenario**	**Input Values**	**Result**

AVLTree	remove	setup2	remove(50)
			search(50)
			remove(25)
			search(25)
			remove(-25)
			search(-25)
			remove(13)
				true
				null
				true
				null
				true
				null
				false