Day 17 - 104608492 - Shirisha Perapagu

AVL Tree and Red-Black Tree

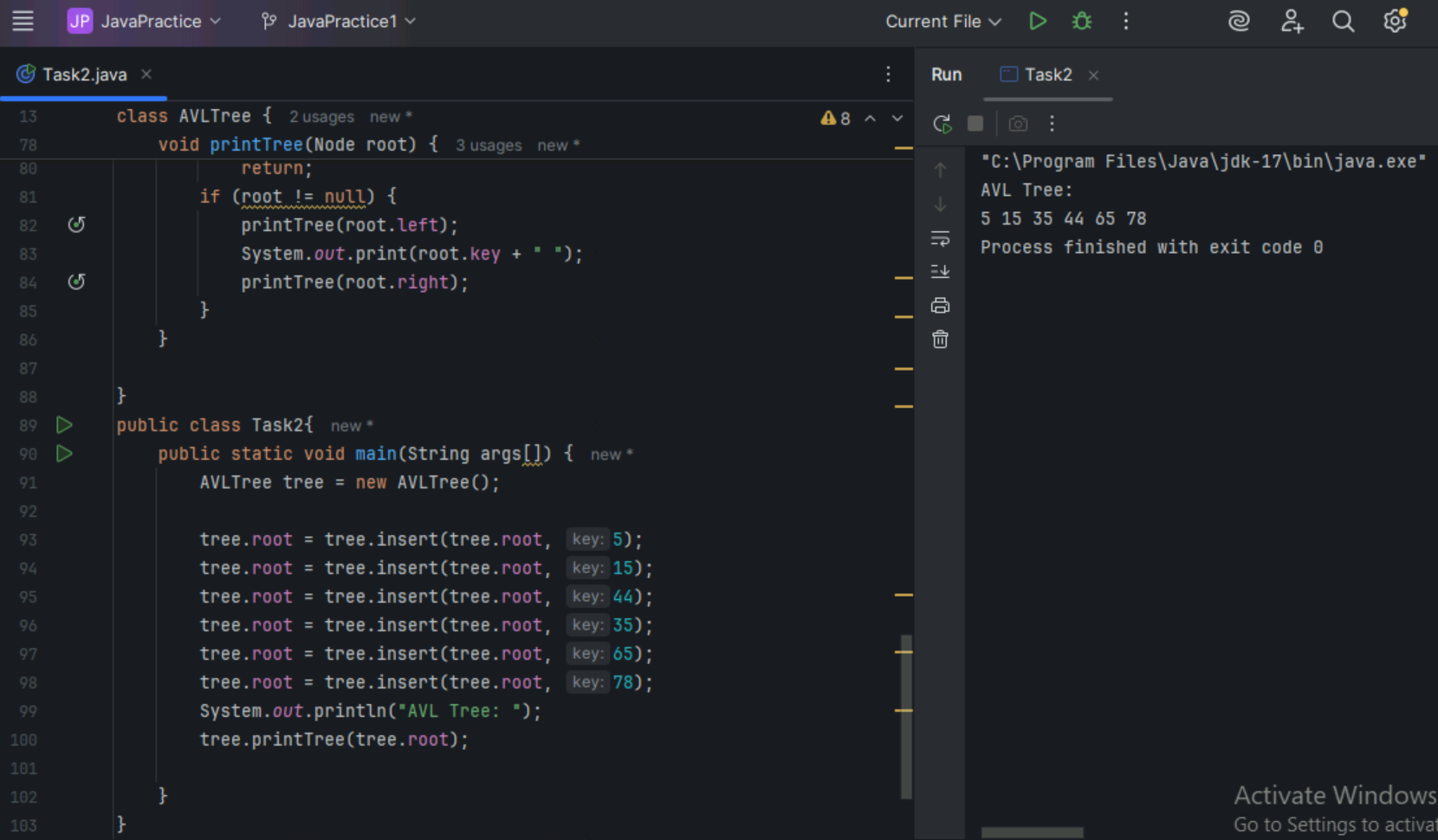
Task 1

AVL Tree Insertion algorithm

1. Start from root node
2. If tree is empty, root becomes the new node.
3. If tree is not empty, if new node is less than current node go to left and if new node is greater than current node go to right.
4. Calculate the height of each node.
5. Calculate the balance factor of each node.
6. If balance factor is not between -1 and 1, do rotations.
7. Repeat steps 4-6 until root is updated.
8. After rotation, tree is balanced.

Task 2

AVL Tree Traversal



Task 3:

Red-Black Tree Insertion Algorithm

Insert an Element - Red Black Tree −

1. Check if tree is empty. If empty, then insert new node - color Black. (Because Root Node - Black in color)

2. else if Tree - not empty then insert new node as leaf node to the end and color - Red.

3. If parent of new node is Red and its neighbours(parent’s) node is also Red,

then Flip the color of both neighbour and Parent and Grandparents (If it is not Root Node Otherwise Flip the color of the Parent and neighbour only) i.e., Black.

4. If parent of new node is Red and its neighbours(parent’s) node is empty or NULL,

then Rotate (either Left-Left or Left-Right rotation) the new node and parent.

5. we have two types of rotation

- Left Left Rotation and

- Left Right Rotation.

6. we apply Rotation in some conditions only.

The conditions are −

- If parent of new node is Red and neighbour node is empty or NULL, then rotate left or right rotation.

- In Left-Left Rotation flip the color of the parent and grandparent.

Make the parent as Grandparent and grandparent as child

Task 4

Red-Black Tree Traversal

