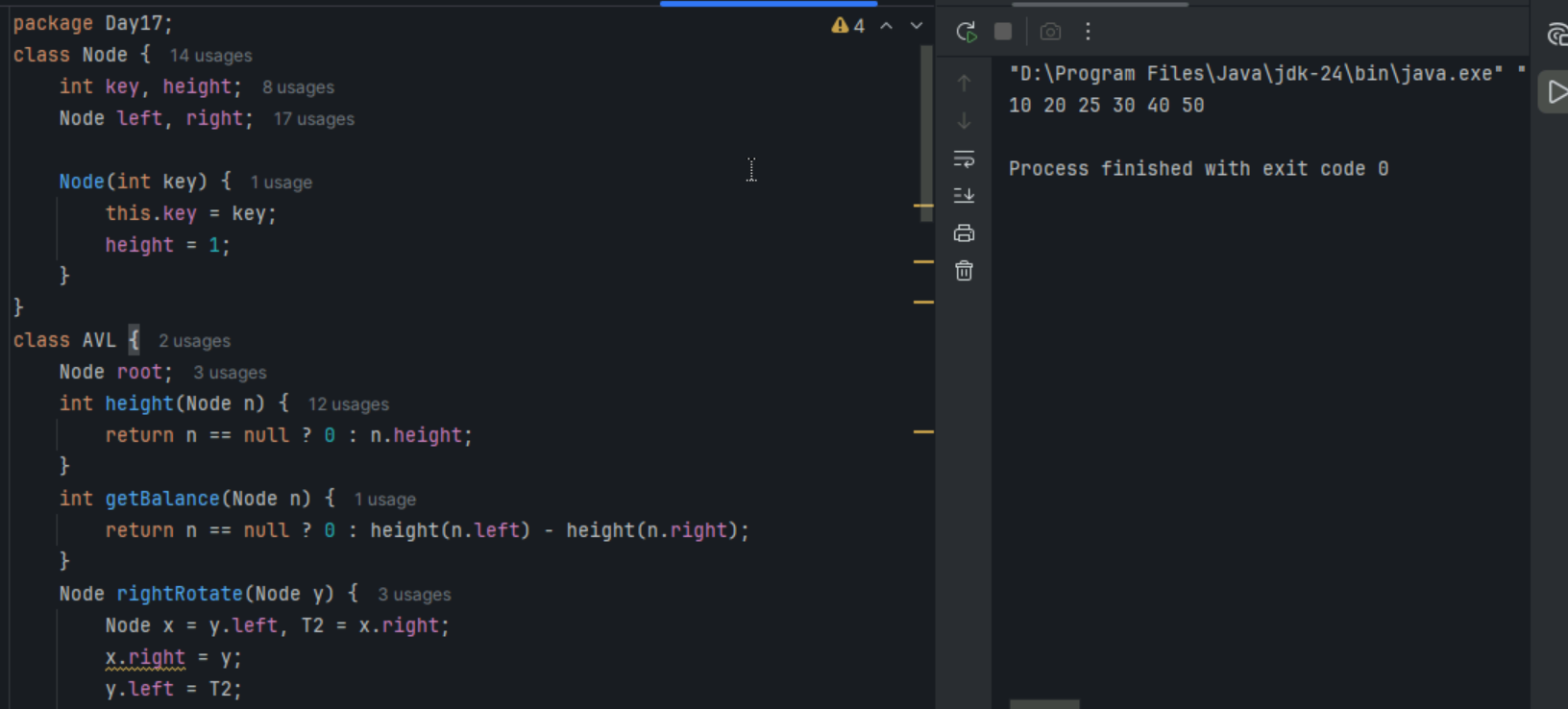
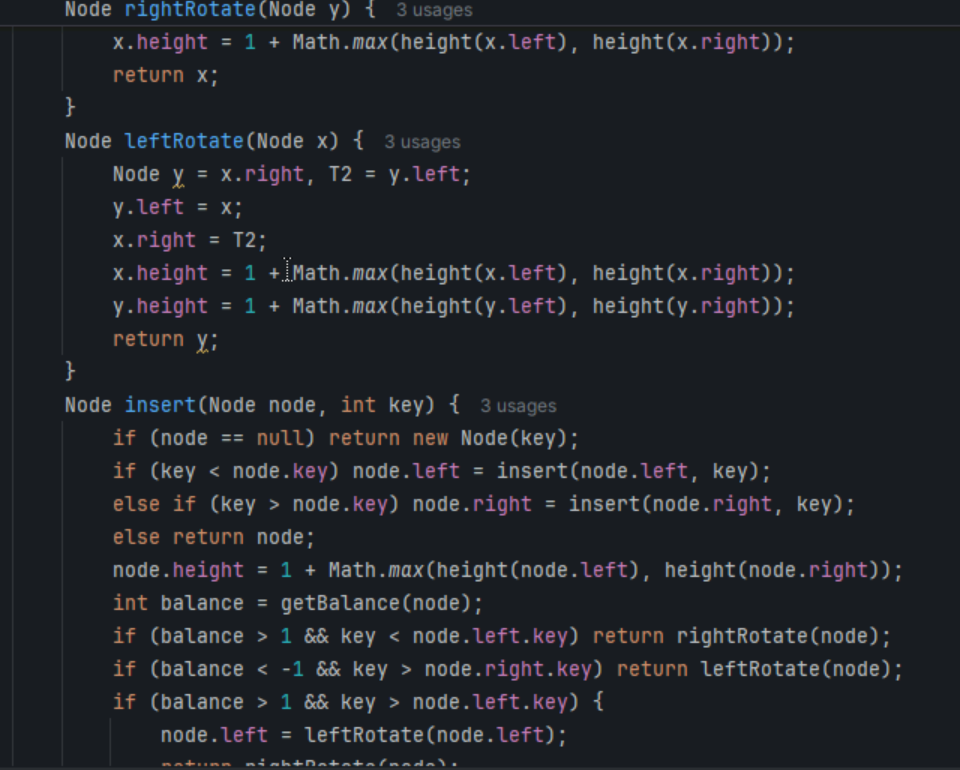
**Task1 - AVL Tree algorithm**

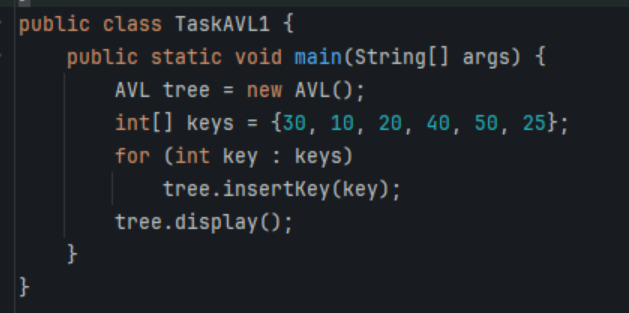
1. Insert the node like in a regular BST.
2. Update the height of the ancestor nodes.
3. Check the balance factor of each node.
4. If unbalanced, perform the required rotation.

**Task2**

****

****

****

****

**Task3 - Algorithm for RedBlack tree**

Insert an Element - Red Black Tree −

1. Check 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 the 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.

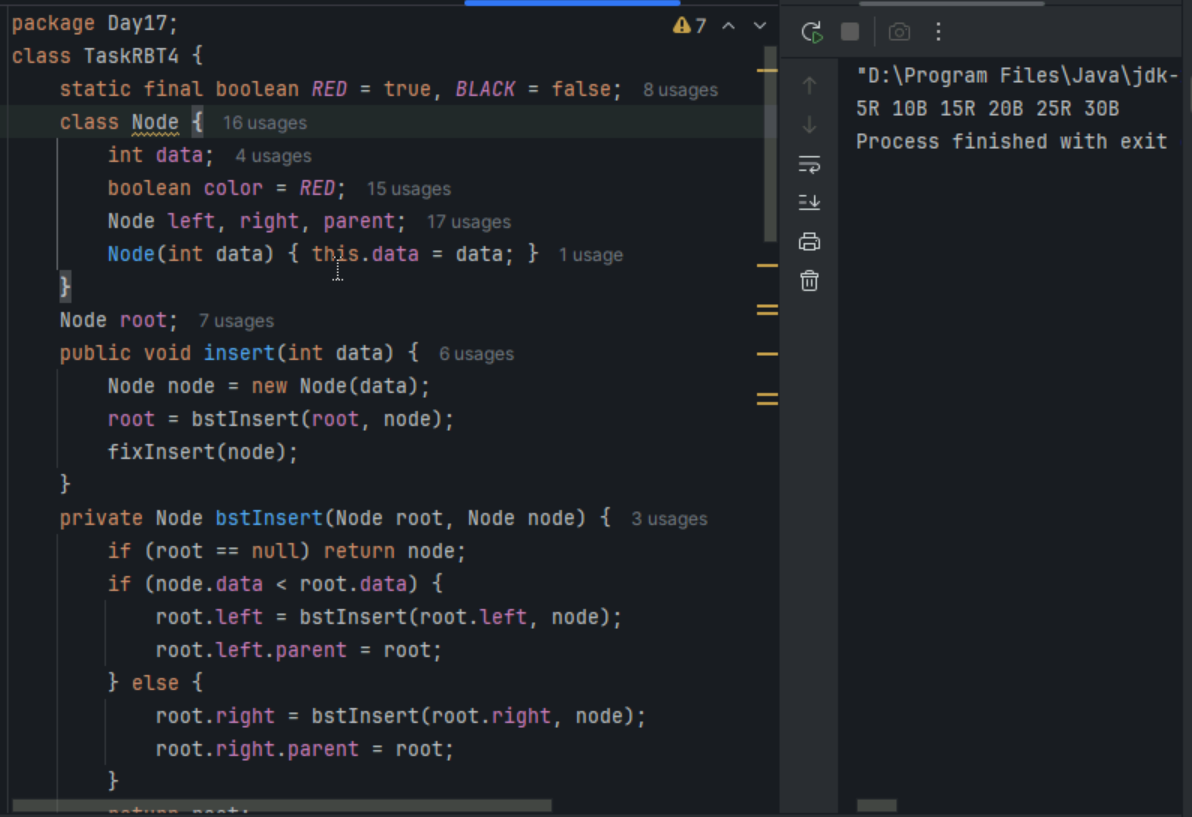
The conditions are −

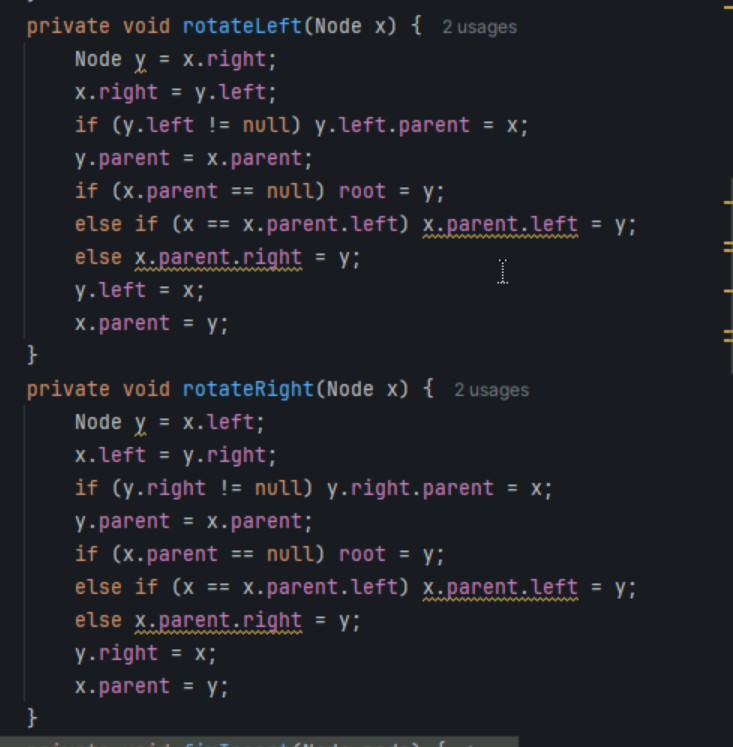
- If parent of new node is Red and neighbour node is empty or NULL, then rotate left or right rotation.

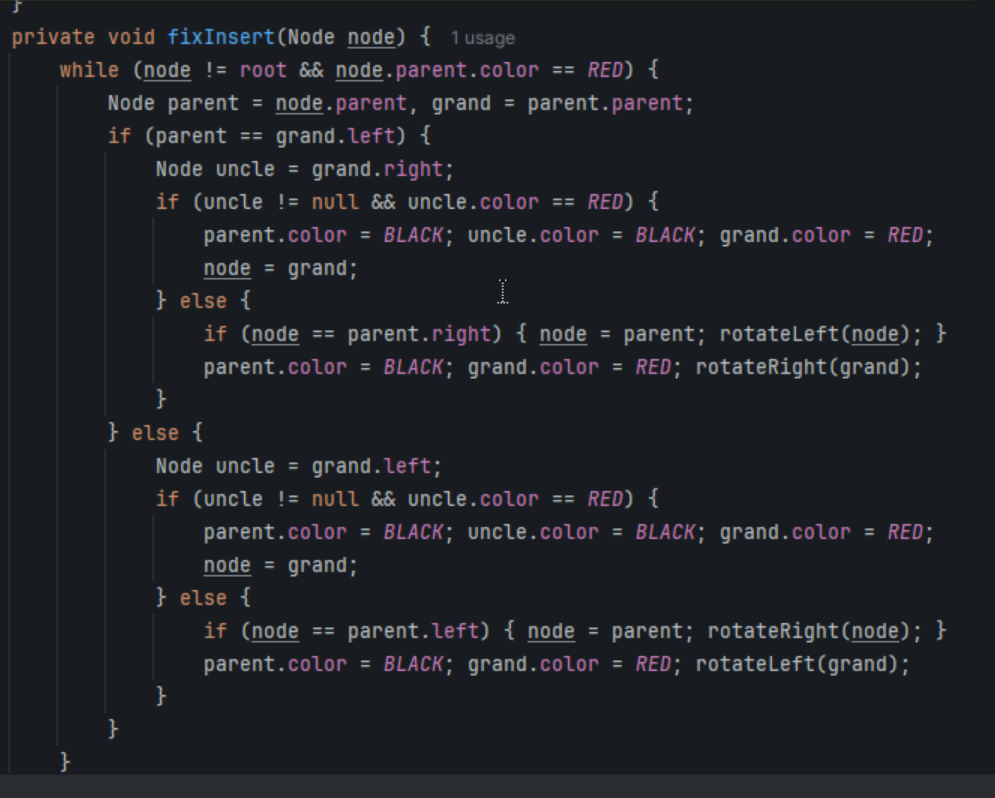
- In Left-Left Rotation flip the color of the parent and grandparent.

Make the parent as Grandparent and grandparent as child

**Task 4**

****

****

****

****