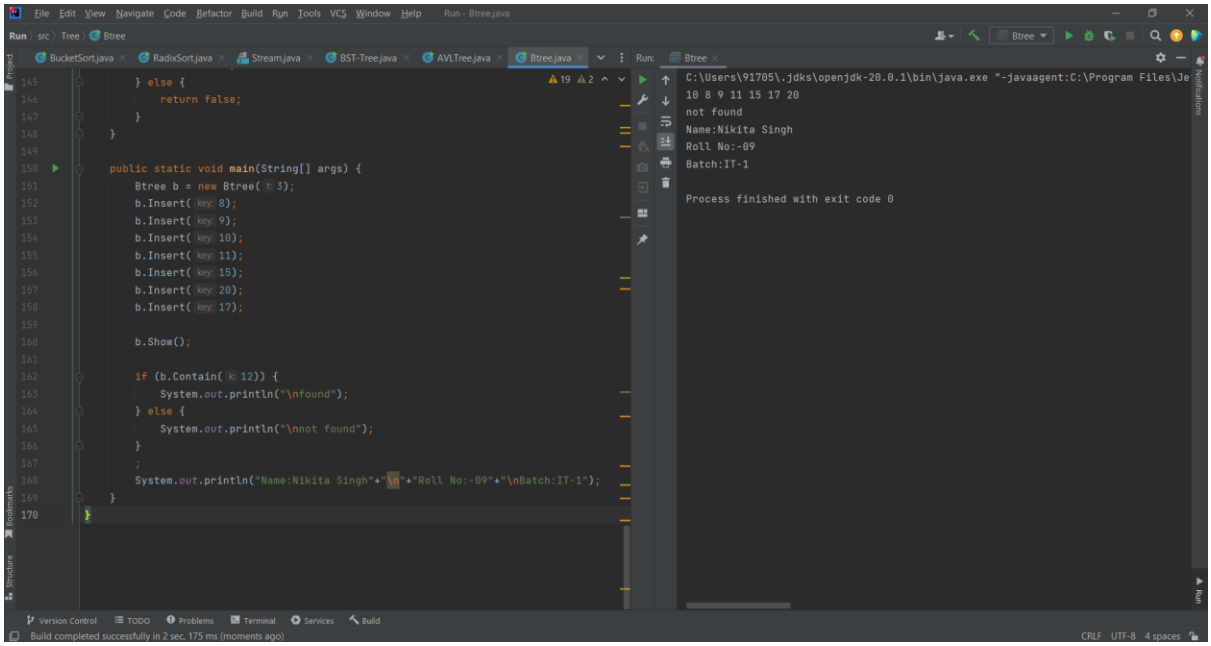
Code:

```
// Searching a key on a B-tree in Java
package Tree;
public class Btree {

    private int T;

    // Node creation
    public class Node {
        int n;
        int key[] = new int[2 * T - 1];
        Node child[] = new Node[2 * T];
        boolean leaf = true;

        public int Find(int k) {
            for (int i = 0; i < this.n; i++) {
                if (this.key[i] == k) {
                    return i;
                }
            }
            return -1;
        };
    }

    public Btree(int t) {
        T = t;
        root = new Node();
        root.n = 0;
        root.leaf = true;
    }

    private Node root;

    // Search key
    private Node Search(Node x, int key) {
        int i = 0;
        if (x == null)
            return x;
        for (i = 0; i < x.n; i++) {
            if (key < x.key[i]) {
                break;
            }
            if (key == x.key[i]) {
                return x;
            }
        }
        if (x.leaf) {
            return null;
        } else {
            return Search(x.child[i], key);
        }
    }

    // Splitting the node
    private void Split(Node x, int pos, Node y) {
        Node z = new Node();
        z.leaf = y.leaf;
        z.n = T - 1;
        for (int j = 0; j < T - 1; j++) {
```

```
            z.key[j] = y.key[j + T];
        }
        if (!y.leaf) {
            for (int j = 0; j < T; j++) {
                z.child[j] = y.child[j + T];
            }
        }
        y.n = T - 1;
        for (int j = x.n; j >= pos + 1; j--) {
            x.child[j + 1] = x.child[j];
        }
        x.child[pos + 1] = z;

        for (int j = x.n - 1; j >= pos; j--) {
            x.key[j + 1] = x.key[j];
        }
        x.key[pos] = y.key[T - 1];
        x.n = x.n + 1;
    }

    // Inserting a value
    public void Insert(final int key) {
        Node r = root;
        if (r.n == 2 * T - 1) {
            Node s = new Node();
            root = s;
            s.leaf = false;
            s.n = 0;
            s.child[0] = r;
            Split(s, 0, r);
            insertValue(s, key);
        } else {
            insertValue(r, key);
        }
    }

    // Insert the node
    final private void insertValue(Node x, int k) {

        if (x.leaf) {
            int i = 0;
            for (i = x.n - 1; i >= 0 && k < x.key[i]; i--) {
                x.key[i + 1] = x.key[i];
            }
            x.key[i + 1] = k;
            x.n = x.n + 1;
        } else {
            int i = 0;
            for (i = x.n - 1; i >= 0 && k < x.key[i]; i--) {
            }
            ;
            i++;
            Node tmp = x.child[i];
            if (tmp.n == 2 * T - 1) {
                Split(x, i, tmp);
                if (k > x.key[i]) {
                    i++;
                }
            }
            insertValue(x.child[i], k);
        }
```

```
    }

    public void Show() {
        Show(root);
    }

    // Display
    private void Show(Node x) {
        assert (x == null);
        for (int i = 0; i < x.n; i++) {
            System.out.print(x.key[i] + " ");
        }
        if (!x.leaf) {
            for (int i = 0; i < x.n + 1; i++) {
                Show(x.child[i]);
            }
        }
    }

    // Check if present
    public boolean Contain(int k) {
        if (this.Search(root, k) != null) {
            return true;
        } else {
            return false;
        }
    }

    public static void main(String[] args) {
        Btree b = new Btree(3);
        b.Insert(8);
        b.Insert(9);
        b.Insert(10);
        b.Insert(11);
        b.Insert(15);
        b.Insert(20);
        b.Insert(17);

        b.Show();

        if (b.Contain(12)) {
            System.out.println("\nfound");
        } else {
            System.out.println("\nnot found");
        }
        ;
        System.out.println("Name:Nikita Singh"+"\n"+"Roll No:-
09"+"Batch:IT-1");
    }
}
```

Output:-


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Run - Btree.java
Run src Tree Btree
BucketSort.java
RadixSort.java
Stream.java
BST-Tree.java
AVLTree.java
Btree.java
} else {
return false;
}
}
public static void main(String[] args) {
Btree b = new Btree(t: 3);
b.Insert(key: 8);
b.Insert(key: 9);
b.Insert(key: 10);
b.Insert(key: 11);
b.Insert(key: 15);
b.Insert(key: 20);
b.Insert(key: 17);
b.Show();
if (b.Contain(k: 12)) {
System.out.println("\nfound");
} else {
System.out.println("\nnot found");
}
;
System.out.println("Name:Nikita Singh"+"\n"+"Roll No:-09"+"\nBatch:IT-1");
}
}
Run: Btree
C:\Users\91705\.jdks\openjdk-20.0.1\bin\java.exe "-javaagent:C:\Program Files\Je
10 8 9 11 15 17 20
not found
Name:Nikita Singh
Roll No:-09
Batch:IT-1
Process finished with exit code 0
Version Control
TODO
Problems
Terminal
Services
Build
Build completed successfully in 2 sec, 175 ms (moments ago)
CRLF UTF-8 4 spaces