

ISIM: NATHANAELLE BOPTI NGAH BONG
ÖGRENCİ NUMERASI: 24080410150

QUIZ

```
class program
{
    public class Node
    {
        public int Data;
        public Node Left;
        public Node Right;

        public Node(int data)
        {
            Data = data;
            Left = null;
            Right = null;
        }
    }

    public class BinarySearchTree
    {
        private Node root;

        public BinarySearchTree()
        {
            root = null;
        }

        public void Ekle(int data)
        {
            root = EkleRecursive(root, data);
        }

        private Node EkleRecursive(Node current, int data)
        {
            if (current == null)
            {
                return new Node(data);
            }

            if (data < current.Data)
            {
                current.Left = EkleRecursive(current.Left, data);
            }
            else if (data > current.Data)
            {
                current.Right = EkleRecursive(current.Right, data);
            }

            return current;
        }
    }
}
```

```

}
public bool Ara(int data)
{
    return AraRecursive(root, data);
}

private bool AraRecursive(Node current, int data)
{
    if (current == null)
    {
        return false;
    }

    if (data == current.Data)
    {
        return true;
    }
    else if (data < current.Data)
    {
        return AraRecursive(current.Left, data);
    }
    else
    {
        return AraRecursive(current.Right, data);
    }
}

public void Sil(int data)
{
    root = SilRecursive(root, data);
}

private Node SilRecursive(Node current, int data)
{
    if (current == null)
    {
        return current;
    }

    if (data < current.Data)
    {
        current.Left = SilRecursive(current.Left, data);
    }
    else if (data > current.Data)
    {
        current.Right = SilRecursive(current.Right, data);
    }
    else
    {
        if (current.Left == null)
        {
            return current.Right;
        }
        if (current.Right == null)
        {
            return current.Left;
        }
        current.Data = MinDeger(current.Right);
        current.Right = SilRecursive(current.Right, current.Data);
    }
}

```

```

    }
    return current;
}

private int MinDeger(Node node)
{
    int min = node.Data;
    while (node.Left != null)
    {
        min = node.Left.Data;
        node = node.Left;
    }
    return min;
}

public void PreOrderDolasim(Node node)
{
    if (node != null)
    {
        Console.Write(node.Data + " ");
        PreOrderDolasim(node.Left);
        PreOrderDolasim(node.Right);
    }
}

public void InOrderDolasim(Node node)
{
    if (node != null)
    {
        InOrderDolasim(node.Left);
        Console.Write(node.Data + " ");
        InOrderDolasim(node.Right);
    }
}

public void PostOrderDolasim(Node node)
{
    if (node != null)
    {
        PostOrderDolasim(node.Left);
        PostOrderDolasim(node.Right);
        Console.Write(node.Data + " ");
    }
}

public void LevelOrderDolasim()
{
    if (root == null) return;
    Queue<Node> queue = new Queue<Node>();
    queue.Enqueue(root);

    while (queue.Count > 0)
    {
        Node current = queue.Dequeue();
        Console.Write(current.Data + " ");

        if (current.Left != null)
        {
            queue.Enqueue(current.Left);
        }

        if (current.Right != null)

```

```

        {
            queue.Enqueue(current.Right);
        }
    }
}

public Node Root => root;
public void PreOrder() => PreOrderDolasim(root);
public void InOrder() => InOrderDolasim(root);
public void PostOrder() => PostOrderDolasim(root);
}

static void Main(string[] args)
{
    BinarySearchTree bst = new BinarySearchTree();
    bool running = true;

    while (running)
    {
        Console.WriteLine("1. Değer Ekle (Ekleme)");
        Console.WriteLine("2. Değer Sil (Silme)");
        Console.WriteLine("3. Değer Ara (Arama)");
        Console.WriteLine("4. Dolaşım Yöntemlerini Göster");
        Console.WriteLine("5. Çıkış");
        Console.Write("Seçiminizi yapın (1-5): ");

        if (int.TryParse(Console.ReadLine(), out int choice))
        {
            int data;

            switch (choice)
            {
                case 1:
                    Console.Write("Eklenecek tamsayı değeri girin: ");
                    if (int.TryParse(Console.ReadLine(), out data))
                    {
                        bst.Ekle(data);
                    }
                    else
                    {
                        Console.WriteLine("Hata: Geçersiz tamsayı formatı.");
                    }
                    break;

                case 2:
                    Console.Write("Silinecek tamsayı değeri girin: ");
                    if (int.TryParse(Console.ReadLine(), out data))
                    {
                        bst.Sil(data);
                    }
                    else
                    {
                        Console.WriteLine("Hata: Geçersiz tamsayı formatı.");
                    }
                    break;

                case 3: // Ara
                    Console.Write("Aranacak tamsayı değeri girin: ");
                    if (int.TryParse(Console.ReadLine(), out data))
                    {

```

```

        if (bst.Ara(data))
        {
            Console.WriteLine($"Sonuç: {data} ağaçta bulundu.");
        }
        else
        {
            Console.WriteLine($"Sonuç: {data} ağaçta
bulunamadı.");
        }
    }
    else
    {
        Console.WriteLine("Hata: Geçersiz tamsayı formatı.");
    }
    break;

case 4:
    Console.WriteLine("\n--- DOLAŞIM SONUÇLARI ---");
    bst.PreOrder();
    bst.InOrder();
    bst.PostOrder();
    bst.LevelOrderDolasim();
    break;

case 5:
    running = false;
    Console.WriteLine("Programdan çıkılıyor. Güle güle!");
    break;

default:
    Console.WriteLine("Geçersiz seçim. Lütfen 1 ile 5 arasında
bir sayı girin.");
    break;
    }
    }
    else
    {
        Console.WriteLine("Hata: Lütfen bir sayı girin.");
    }

    Console.WriteLine("\nDevam etmek için bir tuşa basın...");
    Console.ReadKey();
}
}
}

```

```
1. Deger Ekle (Ekleme)
2. Deger Sil (Silme)
3. Deger Ara (Arama)
4. Dolasim Yöntemlerini Göster
5. Çikis
Seçiminizi yapin (1-5): 1
Eklenecek tamsayi degeri girin: 5

Devam etmek için bir tusa basin...
.1. Deger Ekle (Ekleme)
2. Deger Sil (Silme)
3. Deger Ara (Arama)
4. Dolasim Yöntemlerini Göster
5. Çikis
Seçiminizi yapin (1-5): 1
Eklenecek tamsayi degeri girin: 8

Devam etmek için bir tusa basin...
.1. Deger Ekle (Ekleme)
2. Deger Sil (Silme)
3. Deger Ara (Arama)
4. Dolasim Yöntemlerini Göster
5. Çikis
Seçiminizi yapin (1-5): 1
Eklenecek tamsayi degeri girin: 8

Devam etmek için bir tusa basin...
.1. Deger Ekle (Ekleme)
2. Deger Sil (Silme)
3. Deger Ara (Arama)
```

```
Devam etmek için bir tusa basin...
.1. Deger Ekle (Ekleme)
2. Deger Sil (Silme)
3. Deger Ara (Arama)
4. Dolasim Yöntemlerini Göster
5. Çikis
Seçiminizi yapin (1-5): 2
Silinecek tamsayi degeri girin: 9
```

```
Devam etmek için bir tusa basin...
.1. Deger Ekle (Ekleme)
2. Deger Sil (Silme)
3. Deger Ara (Arama)
4. Dolasim Yöntemlerini Göster
5. Çikis
Seçiminizi yapin (1-5): 3
Aranacak tamsayi degeri girin: 9
Sonuç: 9 ağaçta bulunamadi.
```

```
Devam etmek için bir tusa basin...
.1. Deger Ekle (Ekleme)
2. Deger Sil (Silme)
3. Deger Ara (Arama)
4. Dolasim Yöntemlerini Göster
5. Çikis
Seçiminizi yapin (1-5): 4

--- DOLASIM SONUÇLARI ---
5 1 3 8 1 3 5 8 3 1 8 5 5 1 8 3
Devam etmek için bir tusa basin...
```