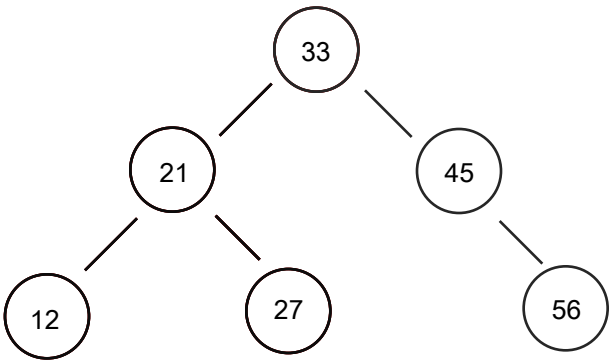
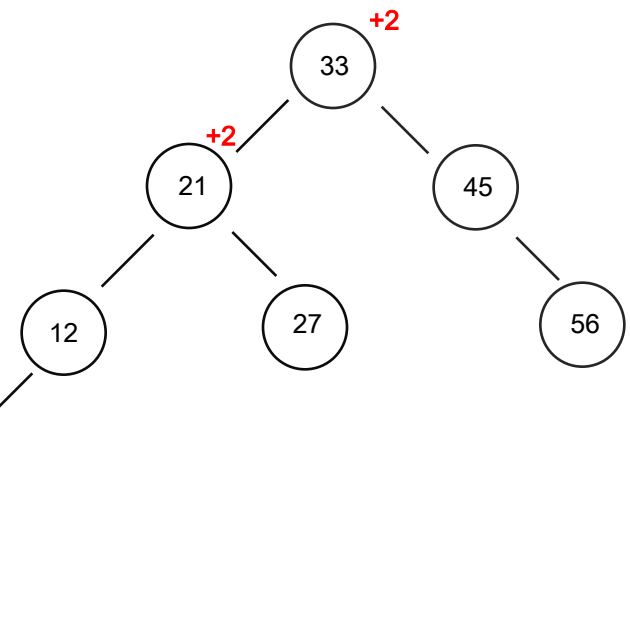
PROGRAM CALISTIRILMASI

```
Run: Main
"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" ...
1.kez RR Rotate Uygulandi.
2.kez RR Rotate Uygulandi.
1.kez RL Rotate Uygulandi.
PreOrder dolasma ile AVL Agac yapisi :
33 21 12 27 45 56

Agac Yapisina `5` ve `4` elemanlari eklendi

1.kez LL Rotate Uygulandi.

PreOrder dolasma ile AVL Agac yapisi :
33 21 5 4 12 27 45 56


Process finished with exit code 0
```

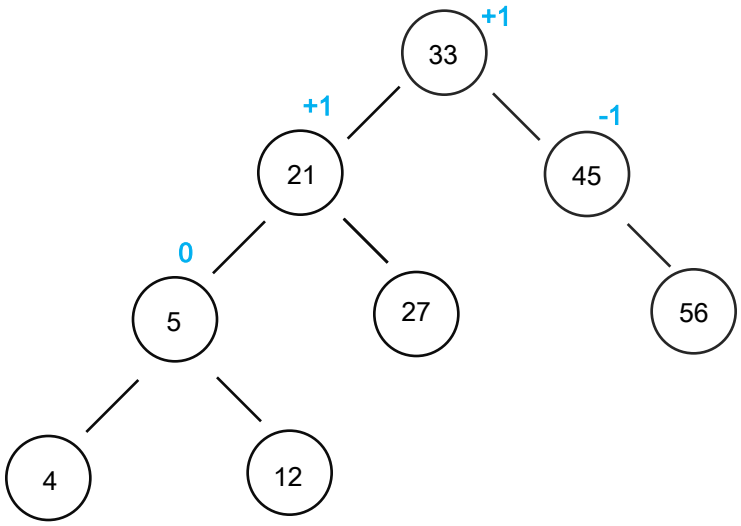
ILK DURUMDA AVL AGACIMIZ



ELEMANLAR EKLENINCE AGAC YAPISI



ROTATE ISLEMI SONUNDAKI AVL AGAC YAPISI



Main Sınıfı

```java
package com.company;

// @author Onur Akbal

public class Main {

    public static void main(String[] args) {
        AVLTree agac = new AVLTree();

        agac.kok = agac.ekle(agac.kok,  icerik: 12);
        agac.kok = agac.ekle(agac.kok,  icerik: 21);
        agac.kok = agac.ekle(agac.kok,  icerik: 33);
        agac.kok = agac.ekle(agac.kok,  icerik: 45);
        agac.kok = agac.ekle(agac.kok,  icerik: 56);
        agac.kok = agac.ekle(agac.kok,  icerik: 27);

        /* ilk bastaki AVL Agaci

             33
            /  \
          21   45
         /  \     \
        12  27    56

        */

        System.out.println("PreOrder dolasma ile AVL Agac yapisi :");
        agac.preOrder(agac.kok);
        System.out.println("\n");

        System.out.println("Agac Yapisina `5` ve `4` elemanlari eklendi\n");


        agac.kok = agac.ekle(agac.kok,  icerik: 5);

        agac.kok = agac.ekle(agac.kok,  icerik: 4);    // EN SOL TARAFA IKI ELEMAN EKLEYEREK DENGEYI BOZUYORUZ.


        System.out.println("\nPreOrder dolasma ile AVL Agac yapisi :");
        agac.preOrder(agac.kok);
        System.out.println("\n");
        /* Sonraki Rotate isleminden sonra bu hali alacaktir
        Dengeli haldedir.

               33
              /  \
            21   45
           /  \     \
          5   27    56
         /  \
        4   12

        */

    }
}
```

Dugum Sınıfı

Main.java × AVLTree.java × Dugum.java ×

```java
package com.company;

public class Dugum {
    int icerik, boy;
    Dugum sol, sag;

    Dugum(int d) {
        icerik = d;
        boy = 1;
    }
}
```

AVLTree Sınıfı

Main.java AVLTree.java Dugum.java

```java
package com.company;

public class AVLTree {
    Dugum kok;

    int boy(Dugum N) {
        if (N == null)
            return 0;

        return N.boy;
    }

    // Maksimum olani dondurme
    int max(int a, int b) {
        return (a > b) ? a : b;
    }
    int sayac=0;
    int sayac1=0;
    int sayac2=0;
    int sayac3=0;

    Dugum rightRotate(Dugum y) {
        Dugum x = y.sol;
        Dugum T2 = x.sag;

        // Dondurme islemi
        x.sag = y;
        y.sol = T2;

        // boyu guncellemek icin
        y.boy = max(boy(y.sol), boy(y.sag)) + 1;
        x.boy = max(boy(x.sol), boy(x.sag)) + 1;


        return x;
    }

    Dugum leftRotate(Dugum x) {
        Dugum y = x.sag;
        Dugum T2 = y.sol;

        // Dondurme Islemi
        y.sol = x;
        x.sag = T2;

        //  boy guncelleme
        x.boy = max(boy(x.sol),boy(x.sag)) + 1;
        y.boy = max(boy(y.sol), boy(y.sag)) + 1;

        // dugum tipinde y yi dondur
        return y;
    }

    // Denge faktoru hesaplanmasi
    int getBalance(Dugum N) {
        if (N == null)
            return 0;

        return boy(N.sol) - boy(N.sag);
    }

    Dugum ekle(Dugum dugum, int icerik) {

        if (dugum == null)
            return (new Dugum(icerik));

        if (icerik < dugum.icerik)
            dugum.sol = ekle(dugum.sol, icerik);
        else if (icerik > dugum.icerik)
            dugum.sag = ekle(dugum.sag, icerik);
        else
            return dugum;

        dugum.boy = 1 + max(boy(dugum.sol),
                boy(dugum.sag));

        // denge kontrolu icin balance hesaplama
        int balance = getBalance(dugum);

        // Sol Sol Rotasyon
        if (balance > 1 && icerik < dugum.sol.icerik) {
            sayac++;
            System.out.println(sayac+ ".kez LL Rotate Uygulandi.");
            return rightRotate(dugum);
        }

        // Sag Sag Rotasyon
        if (balance < -1 && icerik > dugum.sag.icerik) {

            sayac3++;
            System.out.println(sayac3+ ".kez RR Rotate Uygulandi.");
            return leftRotate(dugum);
```

```
        }
        // Sol Sag Rotasyon
        if (balance > 1 && icerik > dugum.sol.icerik) {
            sayac2++;
            System.out.println(sayac2+ ".kez LR Rotate Uygulandi.");
            dugum.sol = leftRotate(dugum.sol);
            return rightRotate(dugum);
        }

        // Sag Sol Rotasyon
        if (balance < -1 && icerik < dugum.sag.icerik) {
            sayac1++;
            System.out.println(sayac1+ ".kez RL Rotate Uygulandi.");
            dugum.sag = rightRotate(dugum.sag);
            return leftRotate(dugum);
        }
        return dugum;
    }

    // pre order dolasma
    void preOrder(Dugum dugum) {
        if (dugum != null) {
            System.out.print(dugum.icerik + " ");
            preOrder(dugum.sol);
            preOrder(dugum.sag);
        }
    }
}
```