



SAKARYA
ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ YAZ OKULU VERİ YAPILARI FİNAL ÖDEVİ

Şule Akçay

G181210027

sule.akcay3@ogr.sakarya.edu.tr

Ödevde ilk önce Node classını oluşturarak başladım. Kişi sınıfının genel bilgilerini tutması için kişi sınıfına bağlı pointer oluşturdum. yüksekliği tutan heighta sıfır değerinin atadım.

```
class Node{
public:
    Kisi* data;
    Node* left;
    Node* right;
    int height;
    Stack stack;
    Node(Kisi* dt, Node* lt=NULL, Node* rt=NULL){
        data = dt;
        left = lt;
        right = rt;
        height = 0; // Eklenen her yeni node ilk önce yaprak olarak eklenir.
        stack.push('0');
    }
};
```

Daha sonra ise kişi sınıfını oluşturdum. Kişi için istenilen bilgileri yazdım.(isim,doğum yılı,yas,kilo)

Temel classları oluşturduktan sonra ise main dosyasında text dosyasını okutma işlemi yaptım.İlk önce satır satır okuma işlemi yaptıktan sonra # işaretine göre bilgileri ayırdım.

Kisi sınıfından oluşturduğum pointer ile listeye atadım. Burada öğrendiğim bilgilerden birisi ise “stoi” nin ne işe yaradığı string tipini integer tipine çeviriyor. Kullanma amacım ise doğum yılı, kilo ve yaşı integer veri tipinde almam.Text dosyasından veri okutmak dışında ödevimde vector kullanmadım.

```
int main(){
    vector<string> linebyline;
    ifstream dict_file("./bin/Kisiler.txt");
    string line;

    while(getline(dict_file, line))
    {
        string new_line;
        new_line = line + "\n";
        linebyline.push_back(new_line);
    }

    stringstream ss;
    AVLTree* agac = new AVLTree();
    for(int i=0; i<linebyline.size(); i++){
        vector<string> liste;
        ss<<i;

        string satir = linebyline.at(i);
        split(satir, '#', liste);
        Kisi* kisi = new Kisi(liste.at(0), stoi(liste.at(1)), stoi(liste.at(2)));
        agac->Ekle(kisi);
    }

    agac->Postorder();

    delete agac;
```

Text dosyasından verileri okuttuktan sonra asıl kısma geldim.avltree dosyasında ise asıl işlemleri yaptım.Kişileri yaşa göre ekledim.Kişinin yaşı aynı ve küçük eşit ise düğümün soluna,büyük ise düğümün sağına yerleştirme işlemlerini yaptım.

```

if(sub_Node == NULL){
    sub_Node = new Node(new_one);
}
else if(new_one->yas <= sub_Node->data->yas){
    sub_Node->left = AraveEkle(sub_Node->left, new_one);
    if(Yukseklık(sub_Node->left) - Yukseklık(sub_Node->right) == 2){
        if(new_one->yas < sub_Node->left->data->yas){
            sub_Node = SolCocukIleDegistir(sub_Node);
            sub_Node->stack.push('Y');
        }
        else{
            sub_Node->left = SagCocukIleDegistir(sub_Node->left);
            sub_Node = SolCocukIleDegistir(sub_Node);
            sub_Node->stack.push('Y');
        }
    }
}
else if(new_one->yas > sub_Node->data->yas){
    sub_Node->right = AraveEkle(sub_Node->right, new_one);
    if(Yukseklık(sub_Node->right) - Yukseklık(sub_Node->left) == 2){
        if(new_one->yas > sub_Node->right->data->yas){
            sub_Node = SagCocukIleDegistir(sub_Node);
            sub_Node->stack.push('Y');
        }
        else{
            // Sol Sag Durumu
            sub_Node->right = SolCocukIleDegistir(sub_Node->right);
            sub_Node = SagCocukIleDegistir(sub_Node);
            sub_Node->stack.push('Y');
        }
    }
}
}
}

```

Ayrıca düğümün hareketlerini tutan ve yerleştiren metotları burada yazdım.İstenildiği gibi Preorder metodu kullandım.Düğümde bulunan yığıtlar düğümün hareketlerinden sorumlu,düğüm aşağı inerse A harfi pushladım.

```

Node* AVLTree::SolCocukIleDegistir(Node* sub_Node){
    Node* tmp = sub_Node->left;
    sub_Node->left = tmp->right;
    tmp->right = sub_Node;

    // Yukseklıkları Guncelle
    sub_Node->height = Yukseklık(sub_Node);
    tmp->height = max(Yukseklık(tmp->left), sub_Node->height) + 1;
    sub_Node->stack.push('A');
    return tmp;
}

Node* AVLTree::SagCocukIleDegistir(Node* sub_Node){
    Node* tmp = sub_Node->right;
    sub_Node->right = tmp->left;
    tmp->left = sub_Node;

    sub_Node->height = Yukseklık(sub_Node);
    tmp->height = max(Yukseklık(tmp->right), sub_Node->height)+1;
    sub_Node->stack.push('A');
    return tmp;
}

void AVLTree::Postorder(Node* sub_Node){
    if(sub_Node != NULL){
        Postorder(sub_Node->left);
        Postorder(sub_Node->right);
        cout<<sub_Node->data->isim<<"", "<<sub_Node->data->dogum_yili<<"", "<<sub_Node->data->kilo<<"  Yigit: "<< sub_Node->stack.pop()<<endl;
    }
}

```

Stack sınıfında push ve pop işlemini gerçekleştirdim. Push metodun da kapasite kalmıyorsa kapasiteyi 2 katına çıkardım ve harfi atadım.

Pop metodun da ise harfi atıp okutma işlemini yaptım.

```
Stack::Stack(){
    elemanlar = NULL;
    stackBasi = -1;
    elemanSayisi = 0;
    kapasite = 0;
}

void Stack::push(const char& eleman){
    if(dolumu()){
        yerAc(max(1, 2*kapasite));
    }
    stackBasi++;
    elemanlar[stackBasi] = eleman;
    elemanSayisi++;
}

char Stack::pop(){
    char elemanim;
    while (true)
    {
        elemanim = top();
        stackBasi--;
        elemanSayisi--;
        if(elemanSayisi == 0) break;
        cout<<elemanim<<" ";
    }
    return elemanim;
}
```

Ödevimde anlayamadığım nokta O ve A harfini doğru yazdırmasına rağmen koşul yerini yanlış yazdığım için D ve Y harfini yanlış yazıyor.Bu ödevde öğrendiğim yapılar stack ve AVL ağaç yapısı oldu.

```
g++ -I ./include -o ./lib/stack.o -c ./src/stack.cpp
g++ -I ./include -o ./lib/avltree.o -c ./src/avltree.cpp
g++ -I ./include -o ./bin/main.exe ./lib/avltree.o ./lib/Kisi.o ./lib/stack.o ./src/main.cpp
./bin/main.exe
A A D Leyla, 1992, 60 Yigit: O
D Murtaza, 1982, 85 Yigit: O
D Y D Mehtap, 1987, 74 Yigit: O
D Mehmet, 1974, 82 Yigit: O
D D Y D Ahmet, 1980, 72 Yigit: O

C:\Users\Suleakcay\Desktop\G181210027_SuleAkca_1A\G181210027_SuleAkca_1A>
```

Hocam nedenini bilmiyorum ama çıktısında harfleri başa koyuyor. Düzeltmeye çalıştım ama olmadı.

