

TELEFON REHBERİ

Algoritma ve Programlama

Proje

Ahmet Batuhan Yılmaz, Hayat Zehra Demir, Rauf Fuzun, Mert Meriç Karadeniz



NASIL ÇALIŞIYOR?

Menü Sistemi

Menü sisteminde cout komutu ile istediğiniz işlem için hangi sayıyı tuşlamanızı söylüyor.

Case ile tuşladığımız başlığa çağrı yapılır çağrı yapıldıktan sonra ekrana o başlık başlatılır.0 tuşuna bastığımızda ise rehberden çıkılır.

tel.Yaz ile dosyalar içerisinde rehber.txt metin belgesine kayıt bilgileri gönderilir.

```
Telefon tel("Telefon");
tel.Oku("rehber.txt");
int secim;
do
{
    cout << "1-Telefon Kaydi Ekle\n2-Kayitlari Listele\n3-Kayitlari Guncelle\n4-Kayitlari Ara\n5-Kayit Sil\n>>";
    cin >> secim;
    switch (secim)
    {
        case 1:
            tel.yeniKayit();
            break;
        case 2:
            tel.ekranaYazdir();
            break;
        case 3:
            tel.Guncelle();
            break;
        case 4:
            tel.Ara();
            break;
        case 5:
            tel.Sil();
            break;
        case 0:
            cout << "Gule Gule!";
            break;
        default:
            cout << "Yanlis Secim\n";
            break;
    }
} while (secim!=0);
tel.Yaz("rehber.txt");
```

NASIL ÇALIŞIYOR?

Telefon Sistemi

Telefon sınıfı rehberden kalıtım ile üretiliyor ve bir isime sahip bunu program içinde bir daha kullanmadık.

```
public: Rehber(){
};
class Telefon : public Rehber {
public: string sahibiad;
int no = 0;
public: Telefon(string ad) {
    this->sahibiad = ad;
}
public: void Oku(string dosya) {
    fstream dosyaoku;
    dosyaoku.open(dosya, ios::in);
    if (dosyaoku.is_open()) {
        string tp;
        no = 0;
        while (getline(dosyaoku, tp)) {
            vector <string> tokens;
            stringstream check1(tp);
            string intermediate;
            while (getline(check1, intermediate, ' '))
            {
                tokens.push_back(intermediate);
            }
            int start = 0;
            this->kayitlar[no].kayitno = stoi(tokens[0]);
            this->kayitlar[no].ad = tokens[1];
            this->kayitlar[no].soyad = tokens[2];
            this->kayitlar[no].cep = tokens[3];
            this->kayitlar[no].is = tokens[4];
            no++;
        }
        dosyaoku.close();
    }
}
```

NASIL ÇALIŞIYOR?

Rehber ve Kayıt Sistemi

Rehber sınıfı, 100 adet Kayıt nesnesinde dizi içeriyor. Yani 100 adet kayıt Rehber sınıfına kaydedilebiliyor.

YeniKayıt, kullanıcıdan isim, soyisim, cep telefonu numarası ve iş numarası bilgilerini alıp ana programdaki 'no' değişkeninin değerindeki 100 dizilik kayıtlar dizisinin no. elemanına girilen bilgileri kopyalıyor. no değişkeni bir arttırılıyor, böylece 0'dan başlayan no değişkeni kaç kayıt varsa o kadar artıyor. Ekrana Yazdır methodu, telefondaki kayıtlardan ismi boş olmayan yani kaydı yapılmış kayıtları listeliyor.

```
}  
public: void yeniKayit() {  
  
    cout << "-----Yeni Kayit-----\n-Ad>>";  
    cin >> kayitlar[no].ad;  
    cout << "-Soyad>>";  
    cin >> kayitlar[no].soyad;  
    cout << "-Cep Tel>>";  
    cin >> kayitlar[no].cep;  
    cout << "-Is Tel>>";  
    cin >> kayitlar[no].is;  
    kayitlar[no].kayitno = no + 1;  
    cout << "Kayit Basarili!\nKayit Numarasi: "<<kayitlar[no].kayitno<<&endl;  
    no++;  
}
```

```
class Kayit {  
public:  
    int kayitno;  
    string ad;  
    string soyad;  
    string cep;  
    string is;  
    Kayit(int kayitno,string ad,string soyad,string cep,string is) {  
        this->kayitno = kayitno;  
        this->ad = ad;  
        this->soyad = soyad;  
        this->cep = cep;  
        this->is = is;  
    }  
    Kayit() {  
    }  
    void yazdir() {  
        cout << setfill('-') << setw(16) << this->kayitno << setw(16) << this->ad << setw(16) <<this->soyad << setw(16)  
        << this->cep << setw(16) << this->is << &endl;  
    }  
};  
bool kayitvarmi(Kayit kayit) {  
    if (kayit.ad != "") {  
        return true;  
    }  
    else return false;  
}  
class Rehber {  
public:Kayit kayitlar[100];
```

NASIL ÇALIŞIYOR

Güncelle, arama , tel.ekranayazdır ve sil sistemi

- Güncelle methodu önce kullanıcıdan id alıyor, sonra o id'ye göre o kaydın ad soyad gibi bilgilerini güncelliyor.
- Ara methodu, kullanıcıdan 1 ile 4 arasında hangi filtreye göre aranmasını gönderdikten sonra bilgi verilir ve eğer öyle bir kayıt varsa ekrana gösterilir.Kullanıcının gönderdiği bilgilerle birisi bulunamazsa ekrana uyarı hatası verilir
- Sil methodu kullanıldığında o ID numarasına sahip kayıtları siler ve rehber.txt dosyasını günceller.
- Tel.ekranayazdır methoduna bu parametreleri göndererek hangi alanda(Ad,soyad vs) arama yapılacağını switch case yapısıyla kontrol ediyor ve metin ile uyuşan kayıtları seçime göre listeliyor. mauris.

NASIL YAPILDI

Projeye başlamadan önce 4 arkadaş konuşarak kendimize bir plan çizdik. Bizden istenen yeni kayıt, kayıt düzenleme vb. istekleri nasıl yapabileceğimiz hakkında konuştuk ve işlemlere başladık. İlk denememizde çıkmaza girdik ve kodu tekrardan yazdık. Kodu en iyi şekilde yazabilmek için internetteki bilgi kaynaklarına baktık. Yeni bilgileri de kendimize kattık. Olabildiğince obje odaklı kodlamaya çalıştık.

TEŞEKKÜRLER

- AHMET BATUHAN YILMAZ
- MERT MERİÇ KARADENİZ
- HAYAT ZEHRA DEMİR
- MEHMET RAUF FUZUN