



FENERBAHÇE ÜNİVERSİTESİ

TELEFON KAYIT SİSTEMİ

ALP EREN GÜRLE

TAHA YASİN ÖZTÜRK

Projenin amacı

► Bir telefon rehberinde bulunması gereken kayıt ekleme, kayıtları gösterme, kayıtları modifiye etme, kayıt arama ve silme kabiliyetlerine sahip olan bir sistem geliřtirmek. Geliřtirilen sistem, kullanıcıya komut satırı arayüzü sunarak rehberin kullanımını sağlayacaktır.

► Kullandığımız Araç:

► Proje kapsamında, ödev ve LAB'larda kullanılan Microsoft'un derleyicisi olan Visual Studio Community kullanılacaktır.

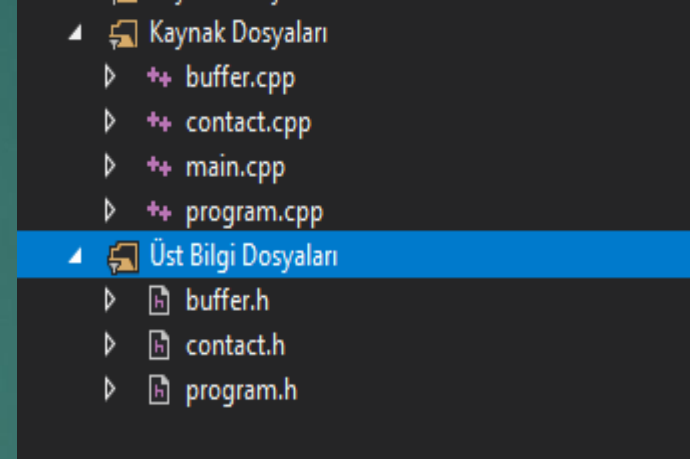


Yapım aşamaları

Öncelikle programımız 3 adet sınıftan (class) oluşmaktadır.

Bunlar program classı, contact classı ve buffer classıdır.

Her bir classın verilerini ve metotlarını kendi isimlerini taşıyan .h dosyalarında tanıttık (declaration) ve .cpp dosyalarında tanımladık (definition).



.h dosyaları şu şekilde oluşturduk

```
#ifndef _buffer_h
#define _buffer_h
#include "contact.h"

class Buffer {
protected:
    Contact* buffer;
    int size;
    int max_size;
public:
    Buffer();
    ~Buffer();
    void add(Contact);
    void remove(int);
    int getsize();
    Contact& operator[](int);
};

class SortedBuffer : public Buffer {
public:
    SortedBuffer();
    ~SortedBuffer();
    void sort();
};

#endif
```

Buffer.h

sortedbuffer objesi kullandık.

Bu obje dinamik bir array gibi davranarak readFile metodu çalıştırılırken dosyanın bütün içeriği bu objeye aktarılır

```
#ifndef _contact_h
#define _contact_h
#include <string>

class Contact {
private:
    std::string id;
    std::string isim;
    std::string soyisim;
    std::string cepno;
    std::string isno;
public:
    Contact(std::string, std::string, std::string, std::string, std::string);
    Contact();
    ~Contact();
    std::string getId();
    std::string getisim();
    std::string getsoyisim();
    std::string getcepno();
    std::string getisno();
    std::string toString();
    void setisim(std::string);
    void print_contact();
};

#endif
```

Contact.h

Programın ihtiyacı olan bilgileri oluşturduk

```
#ifndef _program_h
#define _program_h
#include "contact.h"
#include "buffer.h"

class Program {
private:
    SortedBuffer buffer;
    int status;
public:
    Program();
    ~Program();
    void readfile(std::string);
    void writefile(std::string);
    void add_contact(Contact);
    void remove_contact(std::string);
    void list_contacts();
    void modify_contact(std::string);
    void search_contact(std::string);
};

#endif
```

Program.h

Program objesi bizim kayıt ekleme, silme gibi işlemleri yaptığımız objedir. Bu fonksiyonlarda kişiyi modifiye etme ve arama metodlarını kullandık

Buffer sınıfı

- ▶ Buffer sınıfı kayıtların tutulduğu bir contact pointer ile bu pointerla oluşturulan arrayin boyutunu ve maksimum boyutunu tutar. Bu array kendi kendine büyüme özelliğine sahiptir. Eğer kapasitesi dolduysa ve yeni bir obje eklenecekse gerekli memory işlemlerini yaparak boyutunu iki katına çıkarır. Bunları ise add metodu ile yapar. add metodu her eklemeye önce objenin boyutunun max. boyuta eşit olmadığına bakar, eğer eşitse eski bloğun iki katı olan yeni bir hafıza bloğu açar ve bütün içeriği oraya kopyalar. Ayrıca işlem yapılmadan önce eski hafıza bloğu geçici bir pointerla tutulur ve büyütme işlemi bittiğinde eski hafıza delete[] kullanılarak temizlenir ve olası bir memory leak engellenir.

```
void Buffer::add(Contact new_contact) {  
    if(size == max_size) { // Eğer buffer dolu ise yeni bir hafıza bloğu ayır.  
        Contact* temp = buffer; // Bufferin eski adresini tutar.  
        max_size *= 2; // Bufferin maksimum kapasitesini 2 katına çıkar.  
        Contact* new_buffer = new Contact[max_size];  
        for(int i = 0; i < size; i++) { // Eski bufferin içeriğini yeni buffere kopyalar.  
            new_buffer[i] = buffer[i];  
        }  
        buffer = new_buffer;  
        delete[] temp; // Eski bufferi siler.  
    }  
    buffer[size] = new_contact; // Yeni kişiyi ekle.  
    size++;  
}
```

Buffer metodları

- ▶ Remove fonksiyonu bir kayıtlı siler.
- ▶ Sort fonksiyonu alfabetik sıraya koyar.
- ▶ Kısa yoldan buffer arrayine ulaşmak için [] operatörünü aşırı yükleme yaptık.

```
void Buffer::remove(int index) {
    for(int i = index; i < size-1; i++) {
        buffer[i] = buffer[i+1];
    }
    size--;
}

// Buffer sınıfının contactları tutan buffer arrayine direkt olarak ulaşabilmek için [] operatörünü aşırı yükledi
Contact& Buffer::operator[](int index) {
    if(index >= size || index < 0) {
        cout << "Array index out of bound." << endl;
        exit(0);
    }
    return buffer[index];
}

SortedBuffer::SortedBuffer() : Buffer() { // Ana sınıfın constructorunu çağırır.
}

void SortedBuffer::sort() { // Arrayi alfabetik olarak sıralar. Bubble sort algoritmasını kullanır.
    for(int i = 0; i < getsize() - 1; i++) {
        for(int k = i+1; k < getsize(); k++) {
            if(buffer[i].getisim() > buffer[k].getisim()) {
                Contact temp = buffer[i];
                buffer[i] = buffer[k];
                buffer[k] = temp;
            }
        }
    }
}

SortedBuffer::~SortedBuffer() {
}
```

Program sınıfı

- ▶ readFile fonksiyonu dosyadan bilgileri okuyarak buffer objesinin içine atar.
- ▶ Buffer'ın içindeki bilgileri writeFile fonksiyonu ile dosyaların içine yazar.
- ▶ Bunları yaparken ise direkt olarak her işlemde dosyayı açıp okumak yerine öncelikle Buffer sınıfından inheritance ile oluşturulmuş bir sortedbuffer objesi kullanır. Bu obje dinamik bir array gibi davranarak readFile metodu çalıştırılırken dosyanın bütün içeriği bu objeye aktarılır ve bütün işlemler bu obje üzerinden yapılır. Aynı şekilde kullanıcının uygulama ile işi bittiğinde bu arrayin bütün içeriği writeFile metodu kullanılarak dosyaya geri yazdırılır

```
void Program::readFile(string filename) {
    ifstream file(filename); // Dosyayı aç
    if(!file.is_open()) { // Dosya adinin dogru olup olmadigini kontrol eder. Eger dosya acilamadiysa programi kapatir.
        cout << "Dosya acilamadi." << endl;
        exit(0);
    }
    string id, isim, soyisim, cepno, isno;
    while(file >> id >> isim >> soyisim >> cepno >> isno) { // Dosyanin sonuna gelesiyeye kadar satir satir oku.
        Contact newcontact(id, isim, soyisim, cepno, isno); // Yeni contact olustur ve buffera ekle.
        buffer.add(newcontact); // Dosyayi kapat
    }
    file.close();
}

void Program::writeFile(string filename) {
    ofstream file;
    file.open(filename); // Dosyayi ac
    for(int i = 0; i < buffer.getsize(); i++) { // Her satira bir kayıt yazdır.
        file << buffer[i].toString() << endl;
    }
    file.close();
}
```

- Bu modify fonksiyon ID'si girilen kişinin bilgilerini ID,isim,Soyadi,Cep Numarası,Iş numarası'nı düzenler.

```
void Program::modify_contact(string id) {  
    for(int i = 0; i < buffer.getsize(); i++) {  
        if(buffer[i].getid() == id) {  
            string id, isim, soyisim, cepno, isno;  
            cout << "ID'yi giriniz: ";  
            cin >> id;  
            cout << "Isim giriniz: ";  
            cin >> isim;  
            cout << "Soyadi giriniz: ";  
            cin >> soyisim;  
            cout << "Cep numarasi giriniz: ";  
            cin >> cepno;  
            cout << "Is numarasini giriniz: ";  
            cin >> isno;  
            Contact newcontact = Contact(id, isim, soyisim, cepno, isno);  
            buffer[i] = newcontact;  
            return;  
        }  
    }  
    cout << "Bu ID'ye kayitli bir kisi bulunmamaktadır." << endl;  
}
```


Main Fonksiyonu

- ▶ Main fonksiyonu başlangıçta bir program objesi oluşturur ve kullanıcının program ile işi bitesiye kadar bir while döngüsü oluşturur.

```
{
    Program program = Program(); // Program objesi
    string filename;
    cout << "Kayıtların tutulduğu dosya adını giriniz. (Örnek: telefon.txt) : ";
    cin >> filename;
    program.readFile(filename);
    int input;
    cout << "Telefon rehberi programına hoş geldiniz." << endl;
    while(true) {
        printMenu();
        cin >> input;
        if(input == 0) {
            break;
        }
        else if(input == 1) { // Kisi ekleme
            Contact newcontact = create_contact();
            program.add_contact(newcontact);
        }
        else if(input == 2) { // Kisi silme
            string id;
            cout << "Silmek istediğiniz kişinin ID'sini giriniz: ";
            cin >> id;
            program.remove_contact(id);
        }
        else if(input == 3) { // Kisi degistirme
            string id;
            cout << "Degistirmek istediğiniz kişinin ID'sini giriniz: ";
            cin >> id;
            program.modify_contact(id);
        }
        else if(input == 4) { // Kisi arama
            string type;
            cout << "Kisiyi hangi parametre ile aramak istersiniz ? (id, isim, soyisim, cepno, isno)" << endl;
            cin >> type;
            program.search_contact(type);
        }
        else if(input == 5) { // Kisileri listele
            program.list_contacts();
        }
    }
}
```

Contact sınıfı

- ▶ Bu sınıfta bir kişinin bilgilerini tutar.
- ▶ Bir kişinin ID,Adı,Soyadı,Cep numarası, İş numarası'nı tutar.

```
void Contact::print_contact() { // Kişinin bilgilerini yazdırır.
    cout << "ID: " << id << endl;
    cout << "Adi: " << isim << endl;
    cout << "Soyadi: " << soyisim << endl;
    cout << "Cep Numarasi: " << cepno << endl;
    cout << "Is numarası: " << isno << endl;
    cout << endl;
}

string Contact::toString() {
    return id + " " + isim + " " + soyisim + " " + cepno + " " + isno;
}

void Contact::setisim(string isim) {
    this->isim = isim;
}

#endif
```

```
#ifndef _contact_h
#define _contact_h
#include <string>

class Contact {
private:
    std::string id;
    std::string isim;
    std::string soyisim;
    std::string cepno;
    std::string isno;
public:
    Contact(std::string, std::string, std::string, std::string, std::string);
    Contact();
    ~Contact();
    std::string getid();
    std::string getisim();
    std::string getsoyisim();
    std::string getcepno();
    std::string getisno();
    std::string toString();
    void setisim(std::string);
    void print_contact();
};

#endif
```

Programın Çıktısı

- ▶ Programın dosyayı açabilmesi için dosyanın absolute yeri girilmesi gerekmektedir.
- ▶ Örnek: C:/telefon.txt

```
C:\Users\alpgu\OneDrive\Masaüstü\Telefon kayıtları\Debug\Project8.exe
Kayıtların tutulduğu dosya adını giriniz. (Örnek: telefon.txt) : C:/telefon.txt
Telefon rehberi programına hoş geldiniz.
1-Kayıt Ekle
2-Kayıt Sil
3-Kayıt modifiye etme
4-Kayıt Arama
5-Kayıtları göster
Yapmak istediğiniz işlemin numarasını giriniz. Programdan çıkmak için 0 giriniz.
```

Kayıt ekleme çıktısı

```
Yapmak istediginiz islemin numarasini giriniz. Programdan cikmak icin 0 giriniz.5
Mevcut kayıt sayısı: 12
-----
ID: 0
Adi: 0
Soyadi: 0
Cep Numarasi: 0
Is numarasi: 0

ID: 1234567
Adi: Alp
Soyadi: Gürle
Cep Numarasi: 5546448611
Is numarasi: 1234567890
```

```
1-Kayıt Ekle
2-Kayıt Sil
3-Kayıt modifiye etme
4-Kayıt Arama
5-Kayıtları göster
Yapmak istediginiz islemin numarasini giriniz. Programdan cikmak icin 0 giriniz.1
ID'yi giriniz: 1234567
İsim giriniz: Alp
Soyadi giriniz: Gürle
Cep numarasi giriniz: 5546448611
Is numarasini giriniz: 1234567890
```

```
ID: 0
Adi: 0
Soyadi: 0
Cep Numarasi: 0
Is numarasi: 0

ID: 1234567
Adi: Alp
Soyadi: Gürle
Cep Numarasi: 5546448611
Is numarasi: 1234567890

ID: 324325
Adi: Anil
Soyadi: Guvenir
Cep Numarasi: 3845
Is numarasi: 348532

ID: 3245435
Adi: Ayse
Soyadi: Yilmaz
Cep Numarasi: 23784
Is numarasi: 786578

ID: 325236
Adi: Baris
Soyadi: Uzuner
Cep Numarasi: 3287
Is numarasi: 43578346

ID: 234673
Adi: Buse
Soyadi: Yavas
Cep Numarasi: 2134
Is numarasi: 435346

ID: 565433
Adi: Kedi
Soyadi: Miyav
Cep Numarasi: 35476
Is numarasi: 345346

ID: 2387432
Adi: Mehmet
Soyadi: Ucak
Cep Numarasi: 34534
Is numarasi: 346

ID: 1234
Adi: Melisa
Soyadi: Kisa
Cep Numarasi: 5465465456
Is numarasi: 56465465

ID: 1310
Adi: Oğuz
Soyadi: Gürle
Cep Numarasi: 5334725747
Is numarasi: 02582653605
```

```
CA\Users\alpgu\OneDrive\Masaüstü\Telefon kayıt(Debug)\Project8.exe
```

```
Kayıtların tutulduğu dosya adını giriniz. (Örnek: telefon.txt) : C:/telefon.txt
Telefon rehberi programına hoş geldiniz.
1-Kayıt Ekle
2-Kayıt Sil
3-Kayıt modifiye etme
4-Kayıt Arama
5-Kayıtları göster
Yapmak istediğiniz işlemin numarasını giriniz. Programdan çıkmak için 0 giriniz.2
Silme istediğiniz kişinin ID'sini giriniz: 123456
Bu ID'ye kayıtlı bir kişi bulunamadı.
1-Kayıt Ekle
2-Kayıt Sil
3-Kayıt modifiye etme
4-Kayıt Arama
5-Kayıtları göster
Yapmak istediğiniz işlemin numarasını giriniz. Programdan çıkmak için 0 giriniz.2
Silme istediğiniz kişinin ID'sini giriniz: 123
Bu ID'ye kayıtlı bir kişi bulunamadı.
1-Kayıt Ekle
2-Kayıt Sil
3-Kayıt modifiye etme
4-Kayıt Arama
5-Kayıtları göster
Yapmak istediğiniz işlemin numarasını giriniz. Programdan çıkmak için 0 giriniz.2
Silme istediğiniz kişinin ID'sini giriniz: 2342
1-Kayıt Ekle
2-Kayıt Sil
3-Kayıt modifiye etme
4-Kayıt Arama
5-Kayıtları göster
Yapmak istediğiniz işlemin numarasını giriniz. Programdan çıkmak için 0 giriniz.2
```

Kayıt silme Çıktısı

```
ID: 666666  
Adi: asdfgh  
Soyadi: ghjkl  
Cep Numarasi: 123456  
Is numarasi: 654321  
-----
```

```
Yapmak istediginiz islemin numarasini giriniz. Programdan cikmak icin 0 giriniz.3  
Degistirmek istediginiz kisinin ID'sini giriniz: 324325  
ID'yi giriniz: 666666  
Isim giriniz: asdfgh  
Soyadi giriniz: ghjkl  
Cep numarasini giriniz: 123456  
Is numarasini giriniz: 654321  
1-Kayit Ekle  
2-Kayit Sil  
3-Kayit modifiye etme  
4-Kayit Arama  
5-Kayitlari goster  
Yapmak istediginiz islemin numarasini giriniz. Programdan cikmak icin 0 giriniz.5  
Mevcut kayıt sayısı: 12
```

Kayıt modifiye etme Çıktısı

```
5-Kayıtları göster
Yapmak istediğiniz işlemin numarasını giriniz. Programdan çıkmak için 0 giriniz.5
Mevcut kayıt sayısı: 12
-----
ID: 0
Adi: 0
Soyadi: 0
Cep Numarası: 0
Is numarası: 0

ID: 1234567
Adi: Alp
Soyadi: Gürle
Cep Numarası: 5546448611
Is numarası: 1234567890

ID: 3245435
Adi: Ayşe
Soyadi: Yılmaz
Cep Numarası: 23784
Is numarası: 786578

ID: 325236
Adi: Baris
Soyadi: Uzuner
Cep Numarası: 3287
Is numarası: 43578346

ID: 234673
Adi: Buse
Soyadi: Yavas
Cep Numarası: 2134
Is numarası: 435346

ID: 565433
Adi: Kedi
Soyadi: Miyav
Cep Numarası: 35476
Is numarası: 345346

ID: 2387432
Adi: Mehmet
Soyadi: Ucak
Cep Numarası: 34534
Is numarası: 346

ID: 1234
Adi: Melisa
Soyadi: Kısa
Cep Numarası: 5465465456
Is numarası: 56465465

ID: 2342
Adi: Onur
Soyadi: Kısa
Cep Numarası: 435
Is numarası: 14

ID: 1310
Adi: Oğuz
Soyadi: Gürle
Cep Numarası: 5334725747
```

```
Yapmak istediğiniz işlemin numarasını giriniz. Programdan çıkmak için 0 giriniz.4
Kisiyi hangi parametre ile aramak istersiniz ? (id, isim, soyisim, cepno, isno)
isim
Kisinin ismini giriniz: Alp
ID: 1234567
Adi: Alp
Soyadi: Gürle
Cep Numarası: 5546448611
Is numarası: 1234567890
```

Kayıt arama ve Kayıtları göster Çıktısı