



***BLM 102- Algoritmalar ve Programlama I
2019-2020 Bahar Dönemi***

Telefon Kayıt Sistemi

***Proje Teslim Raporu
10 Haziran 2020***

Taha Yasin Öztürk, Alp Eren Gürle

1	GİRİŞ	1
1.1	Projenin Amacı.....	1
1.2	Proje Ekibi.....	1
2	GELİŞTİRİLEN UYGULAMA	2
2.1	Kullanılan Araçlar	2
2.2	Tasarım	2
3	SONUÇLAR.....	9

1 Giriş

1.1 Projenin Amacı

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

1.2 Proje Ekibi

Alp Eren Gürle:
Okul numarası:190301028
Doğum Tarihi:13.10.2000
Doğum Yeri: Denizli
Mezun Olduğu Lise: Uğur Anadolu lisesi

Taha Yasin Öztürk:
Okul numarası:190301027
Doğum Tarihi:20.11.2000
Doğum Yeri: Denizli
Mezun Olduğu Lise: Erbakır Fen Lisesi

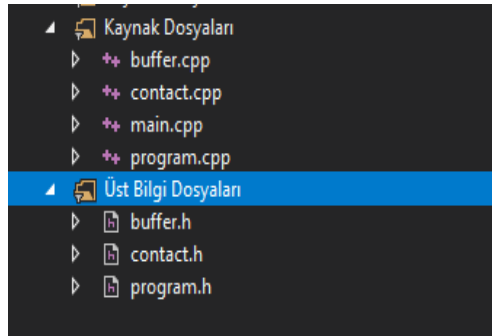
2 Geliştirilen Uygulama

2.1 Kullanılan Araçlar

Bu tasarım geliştirilirken Microsoft'un derleyicisi olan Visual Studio Community aracı kullanılacaktır..

2.2 Tasarım

- /1 Öncelikle programımız 3 adet sınıftan (class) oluşmaktadır.
- /2 Bunlar program classı, contact classı ve buffer classıdır.
- /3 Her bir classın verilerini ve metotlarını kendi isimlerini taşıyan .h dosyalarında tanıttık (declaration) ve .cpp dosyalarında tanımladık(definition).



```
#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

- /4 sortedbuffer objesi kullandık.
- /5 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

/6 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

/7 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 eklemekten 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 buffera 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 sınıfında

- ▶ readFile fonksiyonu dosyadan bilgileri okuyarak buffer objesinin içine atar.
- ▶ Buffer'in 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 adının doğru olup olmadığını kontrol eder. Eğer dosya açılmadıysa programı kapatır.
        cout << "Dosya açılmadı." << endl;
        exit(0);
    }
    string id, isim, soyisim, cepno, isno;
    while(file >> id >> isim >> soyisim >> cepno >> isno) { // Dosyanın sonuna gelene kadar satır satır oku.
        Contact newcontact(id, isim, soyisim, cepno, isno); // Yeni contact oluştur ve buffera ekle.
        buffer.add(newcontact); // Dosyayı kapat
    }
    file.close();
}

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

```

Main Fonksiyonu

- Main fonksiyonu başlangıçta bir program objesi oluşturur ve kullanıcının program ile işi bitmesiye 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 değiştirme
            string id;
            cout << "Değiştirmek 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 sınıfında ise

- 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 numarasi: " << 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.
```

EKSIKLERİ:

Program 1'den fazla isminiz varsa çalışmıyor.İkinci isminizi SoyIsime atıyor.

Programın dosyayı açabilmesi için dosyanın absolute yeri girilmesi gerekiyor.

3 Sonular

Latex'in kullanımı ile CV hazırlamayı ğrendik, C++ sınıf yapısını pekiştirdik. C++ readFile ve writeFile ğrendik. Buffer'in kullanımı araştırdık. add metodu'nu kullandık. Add metodu her eklemekten nce objenin boyutunun max. boyuta eşıt olmadığına bakar,

Ayrıca bu proje ile gnlerce hatta haftalarca bir kod zerinde dřnmenin ,uğraşıp geliřtirmenin deneyimini yaşıadık.

Hazırlanan sunum video'su adresi: [youtube.com/
/channel/UCbCm70s36DzEUOLlqCDLGw?viewas=subscriber](https://youtube.com/channel/UCbCm70s36DzEUOLlqCDLGw?viewas=subscriber)

Dosyaların github adresi: github.com/tahaoztrk, github.com/alpgurlee