

**Öğrenci Adı :**

**Öğrenci No**

**E-Mail**

Sinif sınıfı

MyClass sınıfı içerisinde adress olarak char türenden bir label değişkeni tutulur.Bu label Sınıf ismini temsil eder.İçerisinde isim ata() ,isim getir() , ogrenci\_ekle() , ogrenci isimleri getir() ,isme\_gore\_ara() ,ogrenci\_degistir() ve bir ~Sinif () yıkıcı fonksiyonu bulunur.

Ogrenci\_degistir() gelen Student nesnesini gelen indeks numarasına taşır

void ogrenci\_degistir(Ogrenci \*ogrenci, int degistirilecekIndeks)

    {

        ogrenci\_isimleri->adresleDegistir(ogrenci, degistirilecekIndeks);

    }

Isme\_gore\_ara() gelen sınıf adına göre dizi içerisinde arama işlemi yapar ve bulunan öğrencinin adresini döndürür

Ogrenci \*isme\_gore\_ara(const char karakter) const throw(ElemanYok)

    {

        for (int i = 0; i < ogrenci\_isimleri->length(); i++)

        {

            if (\*ogrenci\_isimleri->elementAt(i)->isim\_getir() == karakter)

                return ogrenci\_isimleri->elementAt(i);

        }

        throw ElemanYok("Eleman bulunamadi");

    }

Okul sınıfı

Sinif\_ekle() dizi içerisinde sınıf nesnesinin adresini ekler

    void sinif\_ekle(Sinif \*sinif)

    {

        sinif\_isimleri->add(sinif);

    }

Sinif\_degistir() gelen iki sinif indeksleri Isme\_gore\_ara() fonksitonu sayesinde bulur ve ArrayList içerisinde bulunan swap fonksiyonu sayesinde yerdeğiştirme yapar

    void sinif\_degistir(char birinci\_sinif, char ikinci\_sinif)

    {

        Sinif \*birinciSinif = isme\_gore\_ara(birinci\_sinif);

        Sinif \*ikinciSinif = isme\_gore\_ara(ikinci\_sinif);

        int indeks\_bir = sinif\_isimleri->indexOf(birinciSinif);

        int indeks\_iki = sinif\_isimleri->indexOf(ikinciSinif);

        sinif\_isimleri->degistir(indeks\_bir, indeks\_iki);

    }

Yonetim sınıfı

Program menüsündeki işlemleri, gerekli fonksiyonlar ve ilişkiler yöneten bir sınıdfır.İçerisinde private olarak imlec\_pozisyon\_ayarla() , dosya\_oku() ,okul\_cizdir() ,ogrenci\_degistir() , ogrenci\_ara() ve public olarak menuCizdir() fonksiyonları bulunur.Ekstra olarak ArrayListşhpp sınıfı içerisine yerdeğiştirme için private olarak degistir() fonksiyonu bulunur

dosya\_oku() () fonksiyonu fstrem kütüphanesini kullanalar ödevde bulunan Okul.txt isimli dosyayı satır satır okumamızı sağlar.Okunan her satır içerisindeki boşluk hariç ilk karekteri sınıf olarak alir ve ger kalan karakterleri öğrenci olarak alarak bir sınıf oluşturur ve School.hpp isimlı sınıfımzda bulunan okul

okul\_icerigini\_cizdir()

    void okul\_icerigini\_cizdir()

    {

        x\_konumu = 1;

        y\_konumu = 1;

        system("cls");

        for ( int i = 0; i < okul->sinif\_isimleri\_getir()->length(); i++)

        {

            Sinif \*gecici\_okul\_nesnesi = okul->sinif\_isimleri\_getir()->elementAt(i);

            char \*okul\_ismi = gecici\_okul\_nesnesi->isim\_getir();

            imlec\_konumunu\_ayarla(x\_konumu, y\_konumu);

            cout << "Okul :" << \*okul\_ismi;

            for (unsigned int j = 0; j < gecici\_okul\_nesnesi->ogrenci\_isimleri\_getir()->length(); j++)

            {

                Ogrenci \*ogrenci = gecici\_okul\_nesnesi->ogrenci\_isimleri\_getir()->elementAt(j);

                char \*ogrenci\_ismi = ogrenci->isim\_getir();

                y\_konumu++;

                imlec\_konumunu\_ayarla(x\_konumu, y\_konumu);

                cout << \*ogrenci\_ismi << " (" << (void \*)ogrenci\_ismi << ")" << endl;

            }

            if (y\_konumu > max\_konumuY)

            {

                max\_konumuY = y\_konumu + 2;

            }

            y\_konumu = 1;

            x\_konumu += 15;

        }

    }

# Student sınıfı

Studen sınıfı içerisinde adress olarak char türenden bir label değişkeni tutulur.Bu label öğrenci ismini temsil eder.İçerisinde setLabel() ,getLabel() , changeAdress() ve bir ~Student() yıkıcı fonksiyonu bulunur.

setLabel() gelen karaktere göre öğrenci adı ataması yapar

getLabel() çağırıldığında öğrenci adının adressini döndürür

changeAdress() gelen karakter adresini öğrenci adının bulunduğu adrese atar

~Student() sınıf sonlandığında çöpleri toplar.