

ISPARTA UYGULAMALI BİLİMLER ÜNİVERSİTESİ -TEKNOLOJİ FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ  
2022-2023 GÜZ DÖNEMİ BLG-101 ALGORİTMA VE PROGRAMLAMAYA GİRİŞ DERSİ ÖDEV RAPORU

Ad Soyad:Muhittin Keser  
Numara:2112721056

Dil Kursu Otomasyonu – V0.1

Proje dil kursunda bulunan öğrencilerin ve öğretmenlerin takibini kolaylaştırmak amacıyla geliştirilmiştir. Bu kapsamda ilk olarak dil kursları hakkında geniş çaplı bir araştırma yapılmış olup isterler belirlenmiştir.

**PROGRAM GEREKSİNİM ANALİZİ :**

kursta verilecek diller ve seviyeleri eklenmeli. ( her dilden 3 seviye olacak şekilde A-B-C ) örn: ingilizce/A , almanca/C gibi diller eklendikten sonra her dile öğretmen atanmalı. eğer kayıtlı olmayan bir dilde öğretmen eklenirse hata mesajı çıkmalı dillere göre öğrenciler eklenmeli ve her dile eklenen öğrenci sonucu o dilin doluluk oranı 1 artmalı. eğer doluluk max seviyedeysen öğrenci eklenememeli yıl sonu öğrencilerin toplam ödeyeceği ücret ve öğretmenlerin toplam alacağı maaş sözleşmede anlaşılan ders saatine ve o dilin saat başı ücretine göre hesaplanmalı öğrenci ve öğretmen silinebilmeli/düzeltilibilmeli sınıf bilgileri görüldükten sonra sınıfta hangi dilin verildiği , öğretmen adı , toplam doluluk oranı , ve öğrenciler gösterilmeli

**VERSİYON 0.1 ÖZELLİKLERİ:**

Ders ekleme, öğretmen ekleme ( eğer öğretmene atanacak dil kayıtlı değilse hata mesajı ) , öğrenci ekleme ( öğrencinin eklendiği sınıfın doluluğunu artırma ) , öğretmen yıl sonu toplam Maaş hesaplama

```
Seç C:\Users\thevi\Desktop\Yeni klas+r (3)\dilkursu.exe
|-----HOŞGELDİNİZ-----|
1)ders ekle (ön koşul)
2)öğretmen ekle (2.ön koşul)
3)öğrenci ekle
4)öğretmen ücret hesapla
----Dil kursu Otomasyonu v0.1----
```

Resim 1: Proje Ana Menüsü

```
C:\Users\thevi\Desktop\Yeni klas+r (3)\dilkursu.exe
|-----HOŞGELDİNİZ-----|
1)ders ekle (ön koşul)
2)öğretmen ekle (2.ön koşul)
3)öğrenci ekle
4)öğretmen ücret hesapla
----Dil kursu Otomasyonu v0.1----
```

1  
dilini adını ve seviyesini girin : ingilizce/A  
saat başı ücretini girin : 71

Resim 2 : ders ekleme

```
öğretmenin vereceği eğitimi belirtin : 81
|-----HOŞGELDİNİZ-----|

1)ders ekle (ön koşul)
2)öğretmen ekle (2.ön koşul)
3)öğrenci ekle
4)öğretmen ücret hesapla

----Dil kursu Otomasyonu v0.1----
2
öğretmen adı :muhittin
öğretmen soyadı : keser
öğretmenin vereceği dil :ingilizce/A
girdiğiniz dil kayıtlı. şimdi cinsiyet ve öğretmenin vereceği eğitim saatini belirtin
öğretmenin cinsiyeti:e
öğretmenin toplam vereceği eğitim saati : 81
```

Resim 3 : öğretmen ekleme

```
|-----HOŞDELDİNİZ-----|

1)ders ekle (ön koşul)
2)öğretmen ekle (2.ön koşul)
3)öğrenci ekle
4)öğretmen ücret hesapla

----Dil kursu Otomasyonu v0.1----
4
hangi öğretmenin yıl sonu ücretini hesaplayacaksınız ? muhittin
öğretmenin eğitim verdiği dili girin: ingilizce/A
öğretmen bulundu. ücreti hesaplanıyor...
öğretmenin alacağı toplam ücret : 5751
```

Resim 4 : öğretmen yıl sonu ücret hesaplama

```
|-----HOŞDELDİNİZ-----|

1)ders ekle (ön koşul)
2)öğretmen ekle (2.ön koşul)
3)öğrenci ekle
4)öğretmen ücret hesapla

----Dil kursu Otomasyonu v0.1----
3
öğrenci adı : huseyin
veliöğrenci soyadı :
öğrencinin eğitim göreceği dil : ingilizce/A
böyle bir dil var , belirttiğiniz dil sınıfına 1 kişi ekleniyor...
sınıfa bir kişi eklendi
doluluk oranı : 1
```

Resim 5 : öğrenci ekleme ve sınıfa bir kişi ekleme

## PROJE KODLARI

```
#include <iostream>
using namespace std;
#include <fstream>
#include <string.h>
#include <windows.h>

struct diller{

    char dilAdivSeviye[20] ;
    int dilSaatBasiUcret ;
    int sinifDoluluk = 0;

};

struct ogrenciler {

    char ad[40] ;
    char soyad[20] ;
    char dil[20] ;
    char cinsiyet ;
    int egitimSaati ; // sözleşmeye göre kaç saat eğitim göreceği .
    int toplamUcret ; // sözleşmeye göre yıl sonu toplam ücret

};

struct ogretmenler {

    char ad[40] ;
    char soyad[20] ;
    char dil[20] ;
    char cinsiyet ;
    int egitimSaati ; // sözleşmeye göre kaç saat eğitim vereceği .
    int toplamUcret ; // sözleşmeye göre yıl sonu alacağı ücret .

};

diller dil ;
ogretmenler ogretmen ;
ogrenciler ogrenci ;

void dilEkle(){

    cout << "dilini adını ve seviyesini girin : " ; cin >> dil.dilAdivSeviye ;
    cout << "saat başı ücretini girin : " ; cin >> dil.dilSaatBasiUcret ;

    dil.sinifDoluluk = 0 ;

    ofstream dilBilgileri("dilbilgileri.txt" , ios::app);
    dilBilgileri.write((char*)&dil , sizeof(dil));

}

void ogretmenEkle(){

    cout <<"öğretmen adı : " ;
    cin >> ogretmen.ad ;
    cout << "öğretmen soyadı : ";
    cin >> ogretmen.soyad ;
    cout << "öğretmenin vereceği dil : " ;
    cin >> ogretmen.dil ;

    bool durum = true ;
    // öğretmene atanan dil önceden eklenmişmi kontrol edelim :
```

## PROJE KODLARI

```
// ilk başta dilbilgileri.txt dosyasının içindeki tüm nesneleri okumalıyım klasik algoritmayla .
ifstream dilBilgileriOku("dilbilgileri.txt");
dilBilgileriOku.seekg(0,ios::end);
int kacDil = dilBilgileriOku.tellg() / sizeof(dil) ;

while(true){
for(int i = 0 ; i<kacDil ; i++){

    dilBilgileriOku.seekg(i*sizeof(dil) , ios::beg) ;
    dilBilgileriOku.read((char*)&dil , sizeof(dil));
    if(strcmp(ogretmen.dil , dil.dilAdivSeviye) == 0 ){
        durum = false ;
        break ;
    }

}

if(durum){
    cout << "girdiğiniz dil kayıtlı değil tekrar deneyin" << endl ;
    cin >> ogretmen.dil ;

}
else{
    cout << "girdiğiniz dil kayıtlı. şimdi cinsiyet ve öğretmenin vereceği eğitim saatini belirtin" << endl ;
    cout << "öğretmenin cinsiyeti:" ;
    cin >> ogretmen.cinsiyet ;
    cout << "öğretmenin toplam vereceği eğitim saati : " ;
    cin >> ogretmen.egitimSaati ;
    break ;
}

}

ofstream ogretmenBilgileri("ogretmenbilgileri.txt" , ios::app);
ogretmenBilgileri.write((char*)&ogretmen , sizeof(ogretmen));

}

void ogrenciEkle(){
    cout << "öğrenci adı : " ; cin >> ogrenci.ad ;
    cout << "öğrenci soyadı : " ; cin >> ogrenci.soyad ;
    cout << "öğrencinin eğitim göreceği dil : " ; cin >> ogrenci.dil ;

    /////
    bool durum = false ;

    ifstream dilBilgileriOku("dilbilgileri.txt");
    dilBilgileriOku.seekg(0,ios::end);
    int kacDil = dilBilgileriOku.tellg() / sizeof(dil) ;

    diller yenediller ;
    for(int i = 0 ; i<kacDil ; i++){

        dilBilgileriOku.seekg(i*sizeof(dil) , ios::beg) ;
        dilBilgileriOku.read((char*)&dil , sizeof(dil));
        if(strcmp(ogrenci.dil , dil.dilAdivSeviye) == 0 ){
            cout << "böyle bir dil var , belirttiğiniz dil sınıfına 1 kişi ekleniyor..." << endl ;
            dil.sinifDoluluk++ ; // dilin sınıf doluluğunu arttırdıktan sonra dilbilgileri.txt te o veriyi
                                // güncellememiz ve buradan yazdırmamız gerekiyor .

            durum = true ;
        }

    }
```

## PROJE KODLARI

```
        ofstream yeniDilBilgileri("yenidilbilgileri.txt" , ios::app) ;

        yenidiller = dil ;
        yeniDilBilgileri.write((char*)&yenidiller , sizeof(yenidiller));
        yeniDilBilgileri.close();

    }

    if(durum){
        dilBilgileriOku.close(); // remove ve rename işlemlerinden önce kullanılan metin belgesini
                                //close() yapmalıyız.

        remove("dilbilgileri.txt");
        rename("yenidilbilgileri.txt" , "dilbilgileri.txt");
        cout << "sınıfa bir kişi eklendi" << endl ;

    }
    else{
        remove("yenidilbilgileri.txt");
    }

    ifstream yeniSorgula("dilbilgileri.txt");
    yeniSorgula.seekg(0,ios::end);
    int kac= yeniSorgula.tellg() / sizeof(dil) ;

    for(int i = 0 ; i<kac ; i++){
        yeniSorgula.seekg(i*sizeof(yenidiller) , ios::beg);
        yeniSorgula.read( (char*)&yenidiller, sizeof(yenidiller) );

        if(strcmp(yenidiller.dilAdivSeviye , ogrenci.dil ) == 0 ) {
            cout << "doluluk oranı : " << yenidiller.sinifDoluluk << endl ;
        }

    }

// en son yeni öğrenciyi metin belgesine ekleyelim .
    ofstream ogrenciBilgileri("ogrencibilgileri.txt" , ios::binary |ios::app);
    ogrenciBilgileri.write((char*)&ogrenci,sizeof(ogrenci));
    ogrenciBilgileri.close() ;

}

void ogretmenUcretHesapla(){
    char ogretmenAdi[40] ;
    char ogrdil[40] ;
    int ucret ;
    cout <<"hangi öğretmenin yıl sonu ücretini hesaplayacaksınız ? " ; cin >> ogretmenAdi ;
    cout << "öğretmenin eğitim verdiği dili girin: " ; cin >> ogrdil ;

    ifstream ogretmenSorgula("ogretmenbilgileri.txt");
    ogretmenSorgula.seekg(0,ios::end);
    int sayac1 = ogretmenSorgula.tellg() / sizeof(ogretmen) ;
    for(int i = 0 ; i<sayac1 ; i++){
        ogretmenSorgula.seekg(i*sizeof(ogretmen) , ios::beg);
        ogretmenSorgula.read((char*)&ogretmen ,sizeof(ogretmen) );

        if(strcmp(ogretmen.ad , ogretmenAdi ) == 0){
            cout << "öğretmen bulundu. ücreti hesaplanıyor.." << endl ;

            ifstream dilbilgileriOku("dilbilgileri.txt");
            dilbilgileriOku.seekg(0,ios::end);
            int sayac2 = dilbilgileriOku.tellg()/sizeof(dil);

            for(int i = 0 ; i<sayac2 ; i++){
```

## PROJE KODLARI

```
        dilbilgileriOku.seekg(i*sizeof(dil));
        dilbilgileriOku.read((char*)&dil , sizeof(dil));

        if(strcmp(dil.dilAdivSeviye , ogrdil) == 0){
            cout << "öğretmenin alacağı toplam ücret : " << dil.dilSaatBasiUcret * ogretmen.egitimSaati <<
endl
            << endl ;
        }

    }

}

}

}

}

int main() {

    system("color 4D") ;
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), BACKGROUND_RED | FOREGROUND_RED
| FOREGROUND_GREEN | FOREGROUND_BLUE | FOREGROUND_INTENSITY);
    setlocale(LC_ALL, "Turkish");

    cout << "|-----HOŞDELDİNİZ-----| " << endl << endl ;
    cout << "      1)ders ekle (ön koşul) " << endl ;
    cout << "      2)öğretmen ekle (2.ön koşul) " << endl ;
    cout << "      3)öğrenci ekle " << endl ;
    cout << "      4)öğretmen ücret hesapla " << endl << endl ;
    cout << " ----Dil kursu Otomasyonu v0.1----" << endl ;
    char secim ;
    cin >> secim ;

    switch(secim){

        case '1' :

            dilEkle();

            break ;

        case '2' :

            ogretmenEkle();

            break ;

        case '3' :

            ogrenciEkle();

            break ;

        case '4' :

            ogretmenUcretHesapla();

        }

    main();
    return 0;
}
```