

PROGRAMLAMAYA GİRİŞ DERSİ

PROJE ÖDEVİ

Amaç: Düğün organizasyonu için program gerçekleştirilecektir. Bu programda her firmanın düğünleri kaydettiği bir Dugunler.Txt dosyası oluşturulsun.

Bu dosyada Dugun sınıfı bilgileri kaydedilsin.

Dugun sınıfının üyedeğişkenleri

DugunNo,
Davetliler dosyası
Patron(parayı ödeyecek):
Tarih
Saat
Damat adı,
Damadın baba adı
Gelin adı,
Gelinin annesi adı
Patron tel
Damat tel
Salon no
Kişi sayısı
Toplam ücret
...

Davetliler için Dugun sınıfındaki davetliler dosyası için girilen dosya oluşturulsun. Davetliler kayıt edilecektir.

Davetliler sınıfı

Üye değişkenler

No:
Adı:
Soyadı:
Akrabalık(gelin/damat?)
Adres
Telefon
...

Dosyalarda

kayıt ekleme, kayıt silme, güncelleme, arama işlemleri yürütülecek

Raporlar

Düğüne geleceklerin listesi

Düğüne gelemeyeceklerin listesi

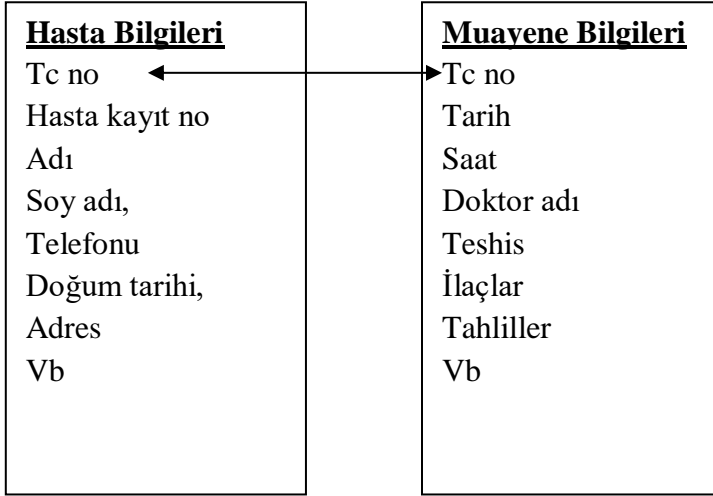
Düğün masaları yerleşim, Damat akrabaları masalarda birlikte, gelin akrabaları masalarda birlikte oturmalıdırlar.

Düğünde yer alan yemekler, masalardaki aksesuarlar, sahnede yer alacak sanatçı vs girilerek bir dosyaya kayıt edilmelidir.

Seçilen yemek ve diğer masraflar ile maliyet çıkarılacak.

Çoklu dosya yönetimi

Uygulamada tek bir dosya kullanımı genellikle yetersiz kalmaktadır. Gerek veri tekrarını önlemek, veri organizasyonunu sağlamak üzere çok sayıda ve bir biri ile ilişkili dosyalarda verilerimizi tutarız. Örneğin hastaların kayıtlarının tutulmasında hastanın değişmeyen ad, soy ad, tel, doğum tarihi vb. bilgileri yanında, hastanın her muayene olduğundaki bilgileri vardır. Muayene bilgileri hasta için sık değişen bir veridir. Ayrı bir dosyada tutulması zorunluluk gerektirir. Bu durumda en azından iki farklı dosya ile veri manipülasyonu gerçekleştirilmelidir. Şekil de dosyalar için muhtemel alanlar verilmiştir.



(TC NO) her iki dosyadada yer almaktadır. Bu bir veri tekrarı gibi görünsede bize iki dosya arasındaki ilişkiyi kurmamızı sağlamada yardımcı olacaktır. Örneğin, Hastanın genel bilgilerini bir kere kayıt ettiğimizde, hasta her seferinde muayeneye geldiğinde sadece muayene bilgileri dosyasına muayene ile ilgili bilgiler kayıt eklenecektir. Kime ait olduğu da TC NO ile bilinecektir. Yani iki dosya arasında bilgilere ulaşımı TC no bilgisi sağlayacaktır.

```

//çok sayıda ilişkili dosya kullanımı
#include <windows.h>
#include <fstream>
#include <iostream>
#include <stdio.h>
#include <string>

using namespace std;

int main(int argc, char *argv[])
{
    // hasta.txt dosyasında yer alacak alanlar
    int secim, i = 1;
    string Hasta_Adi;
    string Hasta_Soyadi;
    string Tc;
    string Tel;
    int Hasta_Yas;
    char Hasta_cinsiyet;
    //muayene.txt dosyasında yer alacak alanlar
    //string Tc;    iki dosya arasındaki ilişkiyi sağlamak için
    string tarih;
    string DK_Adi;
    string Teshis;
    double Ucret;

    char cevap = 'e';

    do
    {
        cout << "    Ne yapmak istiyorsunuz ? " << endl;
        cout << "    1) Hasta kayıt ekleme " << endl;
        cout << "    2) Randevu kayıt " << endl;
        cout << "    3) programdan çıkış " << endl;
        cin >> secim;
    } while (secim != 1 && secim != 2 && secim != 3);

    if (secim == 1)
    {
        ofstream DosyaYaz;
        DosyaYaz.open("Hasta.txt", ios::app);
        do
        {
            cout << "\n Hasta TC      :"; cin >> Tc;
            cout << "\n Hasta adi      :"; cin >> Hasta_Adi;
            cout << "\n Hasta soyadi    :"; cin >> Hasta_Soyadi;
            cout << "\n Hasta Yasi      :"; cin >> Hasta_Yas;
            cout << "\n telefonu      :"; cin >> Tel;
            cout << "\n cinsiyet       :"; cin >> Hasta_cinsiyet;

            DosyaYaz << Tc << " " << Hasta_Adi << " " << Hasta_Soyadi << " "

```

```

        << Hasta_Yas << ' ' << Tel << ' '
        << Hasta_cinsiyet << "\n";
        cout << "\n baska kayıt yapacak misin?(e/h) "; cin >> cevap;
    } while (!(cevap == 'h'));
    DosyaYaz.close();
    cout << "dosyaya yazım tamamlandı. ";
}
if (secim == 2)
{
    string TC;
    cout << " Hasta TC :"; cin >> TC;
    ifstream DosyaOku("Hasta.txt");
    while (!DosyaOku.eof())
    {
        DosyaOku >> Tc >> Hasta_Adi >> Hasta_Soyadi
        >> Hasta_Yas >> Tel >> Hasta_cinsiyet;
        if (TC == Tc)
        {
            cout << "\n Hasta Bilgileri \n";
            cout << " TC no      :" << Tc << endl;
            cout << "adi        :" << Hasta_Adi << endl;
            cout << "soy adi    :" << Hasta_Soyadi << endl;
            cout << "Yaşı       :" << Hasta_Yas << endl;
            cout << "telefonu   :" << Tel << endl;
            cout << "cinsiyeti:" << Hasta_cinsiyet << endl;
            cout << " randevu istiyor mu? (e/h)" << endl;
            cin >> cevap;
            if (cevap == 'e')
            {
                DosyaOku.close();
                ofstream DosyaYaz;
                DosyaYaz.open("Muayene.txt", ios::app);
                cout << "\n Randevu Bilgileri \n";
                cout << "\n Tarih          :"; cin >> tarih;
                cout << "\n Doktor Adi       :"; cin >> DK_Adi;
                cout << "\n Teshis           :"; cin >> Teshis;
                cout << "\n Muayene Ucreti    :"; cin >> Ucret;

                DosyaYaz << Tc << " " << tarih << ' ' << DK_Adi << ' '
                << Teshis << ' ' << Ucret << "\n";
                DosyaYaz.close();
                system("PAUSE");
                exit(1);
            }
        }
        else
        {
            exit(1);
        }
    }
}

system("PAUSE");
return EXIT_SUCCESS;
}

```

Hastanın hangi tarihlerde muayene olduğunu sorgulamak istediğimizde, önce TC no ile hasta “Hasta.txt” dosyasında sorgulanır. Hasta dosyada kayıtlı ise, ekrana getirilir. Hastanın muayene bilgileri “Muayene.txt” dosyasında yer aldığından dolayı, hastanın TC no ile muayene bilgileri “Muayene.txt” dosyasında araştırılır. Hasta farklı zamanlarda birden fazla muayene olmuş ise her birisi ekrana getirilecektir. Aşağıdaki kod bloğu böyle bir sorgulamayı gerçekleştirmek için kullanılabilir.

```

if (secim == 3)
{
    string TC;

```

```

cout << " Hasta TC :"; cin >> TC;
ifstream DosyaOku("Hasta.txt");
while (!DosyaOku.eof())
{
    DosyaOku >> Tc >> Hasta_Adi >> Hasta_Soyadi
        >> Hasta_Yas >> Tel >> Hasta_cinsiyet;
    if (TC == Tc)
    {
        cout << "\n Hasta Bilgileri \n";
        cout << " TC no      :" << Tc << endl;
        cout << " adi        :" << Hasta_Adi << endl;
        cout << " soy adi    :" << Hasta_Soyadi << endl;
        cout << " Yaşı       :" << Hasta_Yas << endl;
        cout << " telefonu  :" << Tel << endl;
        cout << " cinsiyeti:" << Hasta_cinsiyet << endl;
        TC = Tc;
        DosyaOku.close();
        ofstream DosyaYaz;
        DosyaOku.open("Muayene.txt", ios::in);
        cout << " randevu bilgileri \n" << endl;
        cout << setw(11) << " TC " << setw(10) << " Tarih " << setw(12) << " Doktor
Adi " << setw(10)
            << " Teshis " << setw(6) << " Ucret" << endl;
        while (!DosyaOku.eof())
        {
            DosyaOku >> Tc >> tarih >> DK_Adi >> Teshis >> Ucret;

            if (TC == Tc)
            {
                cout << setw(11) << Tc << setw(10) << tarih << setw(12) << DK_Adi
                    << Teshis << setw(6) << Ucret << endl;
            }
        }
        DosyaYaz.close();
        system("PAUSE");
        exit(1);
    }
}
}

```

İstenenler:

1. Proje için gerekli işlemleri yapan C++ kaynak kod dosyası
2. Projede yapılan işlemlerin basit bir şekilde anlatıldığı rapor dosyası

Ödevin son teslim tarihi :

Sistemde belirtilen gün ve saattir

UYULMASI GEREKEN KOD DÜZENİ

Her C++ dosyasının başında aşağıdaki yorum bloğu bulunacaktır. Yorum bulunmayan her C++ dosyası için 10 puan kırılacaktır. (pdf üzerinden kopyalayıp yapıştırmanız problem çıkartabilir)

```
/******  
**                                     SAKARYA ÜNİVERSİTESİ  
**                                     BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ  
**                                     BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ  
**                                     PROGRAMLAMAYA GİRİŞİ DERSİ  
**  
**                                     ÖDEV NUMARASI.....:  
**                                     ÖĞRENCİ ADI.....:  
**                                     ÖĞRENCİ NUMARASI:  
**                                     DERS GRUBU.....:  
*****/
```

Değişken isimleri anlamlı olmalıdır. Örneğin tek sayı adedini tutacak bir değişken için

YANLIŞ		DOĞRU
<code>int a;</code>	yerine	<code>int tekAdedi;</code>

Her bir küme parantezinin altındaki kodları yazmadan önce **tab** tuşuna basılarak boşluk bırakılmalıdır.

YANLIŞ	DOĞRU
<pre>if(true) { int a =6; cout<<"Merhaba"; if(false) cout<<"Nasilsin"; }</pre>	<pre>if(true) { int a =6; cout<<"Merhaba"; if(false) cout<<"Nasilsin"; }</pre>

Her döngü ve koşul işlemlerinden önce yapılan işlem hakkında yorum yazılmalıdır

YANLIŞ	DOĞRU
<code>int t = 0;</code>	<code>int toplam = 0;</code>

```
for(int i=0;i<100;i++)  
    t+=i;
```

```
//1 den 100 e kadar olan sayılar toplanıyor  
for(int i=0;i<100;i++)  
    toplam+=i;
```

ÖDEV TESLİM KURALLARI

TESLİM EDİLECEK DOKÜMANLAR

- Ödev için bir rapor dosyası hazırlanacaktır.
- Dosya 'pdf' formatına çevrilip yollanması gerekmektedir.

DERLEYİCİ VE TESLİM TARİHİ

Derleme için **Visual Studio 2017** veya üzeri C++ derleyicisi kullanılacaktır.

Zamanında teslim edilmeyen ödevler değerlendirilmeyecektir.

Teslim saati sistemde belirtilen tarih ve saattir

KOD PAYLAŞMAK YASAKTIR

Ödevler bireyseldir ve verilen bütün ödevler karşılaştırılacaktır. Birbirine çok benzeyen ödevler **kopya** muamelesi görecektir. Öğretim üyesi kopya durumunda ödevi değerlendirmez veya gerekli soruşturma ve ceza işlemlerini başlatabilir.