**BLG 221**

**Veri Yapıları**

2010 Güz

**Proje 1**

Son Teslim Tarihi : 08.11.2010

Ad Soyad : Ozan Arkan Can

Numara : 040090573

Sınıf Eğitmeni : Gülşen ERYİĞİT

Asistan: Ahmet Aycan ATAK

CRN: 10950

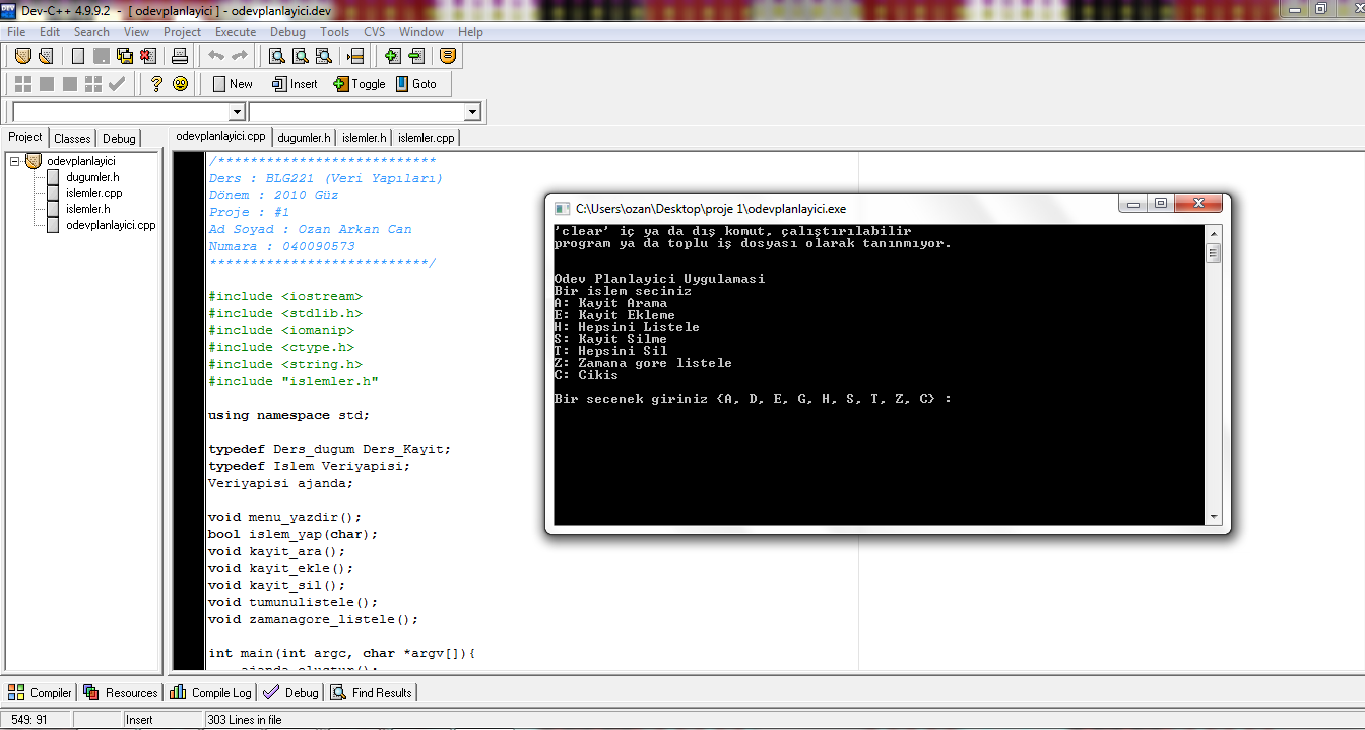
**1-Giriş**

Bu projede bir öğrencinin aldığı derslerde yapması gereken sınıf dışı çalışmaları planlaması için yardımcı olacak bir program yazılması hedeflenmiştir. Öğrenci her ders için çeşitli aktivitelerde bulunmaktadır. (ödev, proje, soru çözümü vb.) Her yapılacak iş öğrenci tarafından hangi dersle ilgili olduğu, işi tanımlayan anahtar bir sözcük ve bitiş tarihi belirtilerek girilmektedir. Program liste yapısında çalışmaktadır. Başlarken varsa dosyadan önceden girilmiş kayıtları okuyup gerekli işlemler yapıldıktan sonra kapatılırken listenin son hali tekrar dosyaya kaydedilmektedir.

**2-Geliştirme ve Çalıştırma Ortamları**

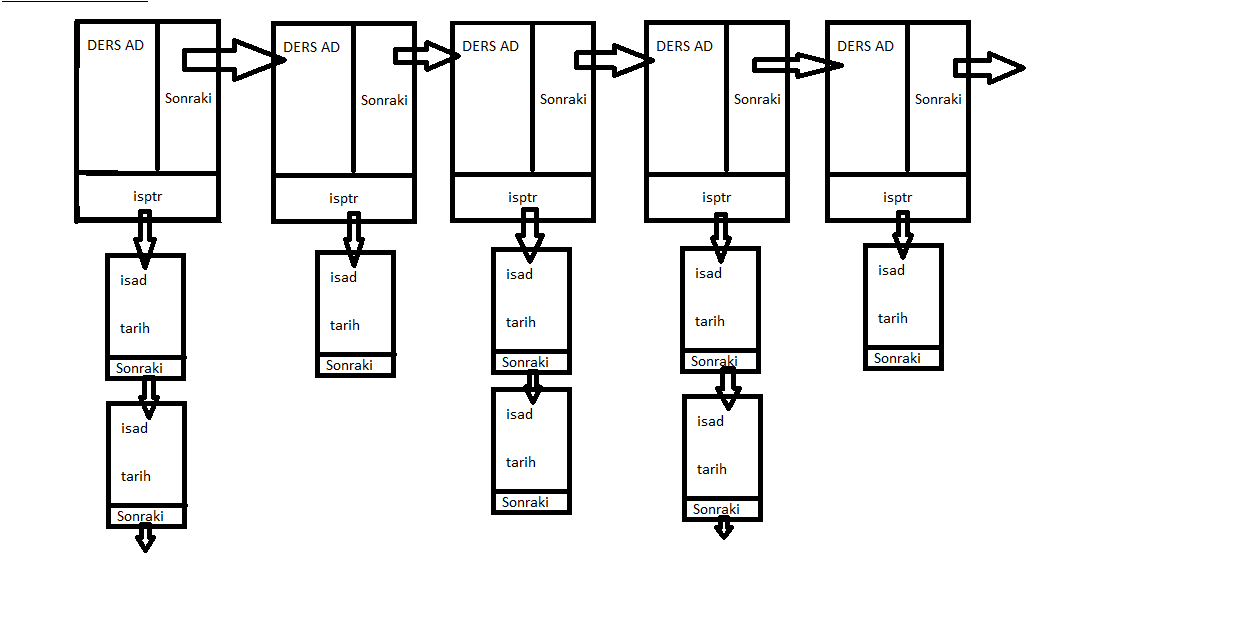
MS Windows

Kaynak kodu düzeltmede, derleme aşamasında ve çalıştırmada Dev-C++ kullanılmıştır.



**3-Veri Yapısı**

Veri yapısı olarak programda aşağıdaki gibi şematize edilmiş bir çoklu liste kullanılmıştır. Dersleri tutan bir liste ve her dersin işlerini tutan başka bir liste oluşturulmuştur. Ayrıca liste üzerinde daha rahat işlem yapabilmek için çeşitli durumlarda çağırılan ve aranan işlere işaret eden işaretçi listesi oluşturulmuştur.

****

Ders düğümü:

struct Ders\_dugum{

char ders[DERS\_AD\_UZUNLUK]; //Ders adını tutan karakter dizisi

Ders\_dugum \*sonraki; //Bir sonraki düğüme işaret eden işaretçi

Is\_dugum \*isptr; //Dersin iş listesinin başına işaret eden işaretçi

};

İş düğümü:

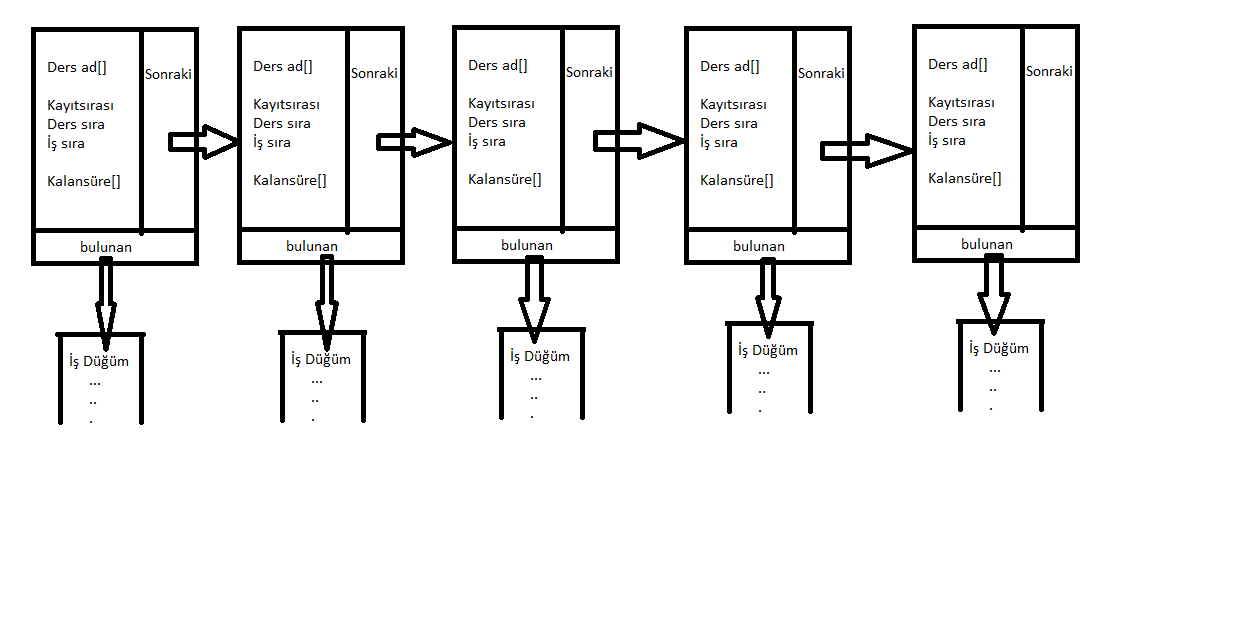
struct Is\_dugum{

char is[IS\_UZUNLUK]; //İşin adını tutan karakter dizisi

int tarih[5]; //Bitiş tarihini tutan dizi 0-4: gün, ay, yıl, saat, dakika

Is\_dugum \*sonraki; //Bir sonraki iş düğümünü işaret eden işaretçi

};

Aranan işler üzerinde işlem yapabilmek için kurulan liste yapısı aşağıda şematize edildiği gibidir. 

İşaretçi düğümü:

struct isaretci\_dugum{

int kayitsirasi;//isaretci listesi işleri tuttuğu için işlerin sayacı

int derssira;//Derslerin sayacı

int issayac;//Bir dersin işinin sayacı

char ders[DERS\_AD\_UZUNLUK]; //Ders adını tutan karakter dizisi

int kalansure[5];//Bitiş tarihine kalan süreyi tutan dizi

Is\_dugum \*bulunan;//Bulunan işe işaret eden işaretçi

isaretci\_dugum \*sonraki;

};

Dosya Kayıt Düğümü:

struct Dosya\_Kayit{

char ders[DERS\_AD\_UZUNLUK];// Ders adını tutan karakter dizisi

char is[IS\_UZUNLUK];// İşin adını tutan karakter dizisi

int tarih[5];// Bitiş tarihini tutan dizi 0-4: gün, ay, yıl, saat, dakika

};

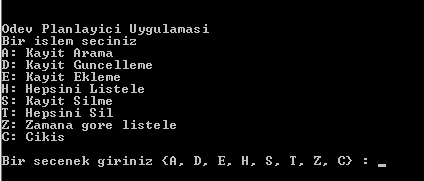
**4-Sonuç**

Proje aşamasında çoklu liste nasıl oluşturulur, çoklu liste üzerinde ilerlemeler nasıl yapılır öğrenilmiştir. İşaretçilerle ilgili yeni bilgiler edinilmiştir. Bolca segmantation fault edinilerek gerekli uyarılar alınmıştır .

**Kullanım Klavuzu**

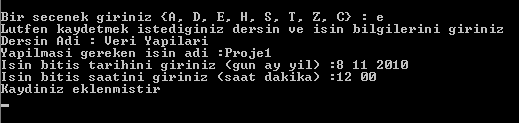
**A)Argümansız Çalıştırma**

Program argümansız olarak çalıştırıldığında ekrana bir menü gelecektir. Bu menüden yapılacak işlem seçilir.



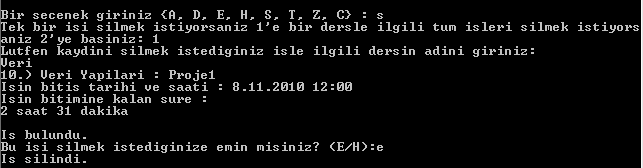
**Kayıt Ekleme:**

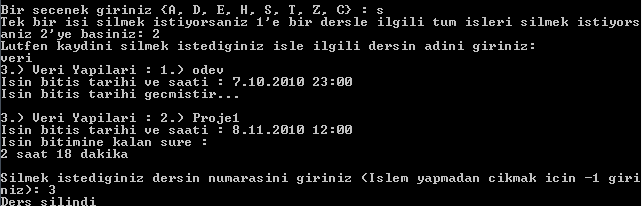
Menüye “e” veya “E” girilerek ekleme işlemi başlatılır. Önce dersin adı girilir. Sonra yapılacak işle ilgili anahtar sözcük girilir. İşin bitiş tarihi gün ay yıl olmak üzere aralarında birer boşluk bırakılarak sayısal olarak girilir. Son olarak işin bitiş saati saat dakika şeklinde aralarında birer boşluk bırakılarak girilir. Böylece kayıt eklenmiş olur.



**Kayıt Silme:**

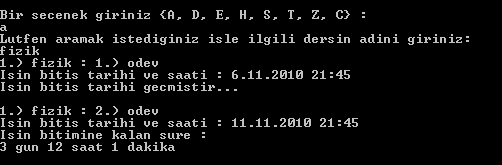
Menüye “s” veya “S” girilerek silme işlemi başlatılır. İki türlü silme işlemi vardır. Tek bir işi silme ve bir dersle ilgili tüm işleri silme. Hangi tür silme işlemi kullanılmak isteniyorsa seçim (1 veya 2) girilir. Silmek istenilen iş veya işlerle ilgili dersin ismi girilir. Eğer tek bir kayıt bulunduysa evet veya hayır girilerek bulunan kayıt silinir. 1 den çok kayıt bulunduysa seçiminize bağlı olarak iş veya ders numarası girilerek kayıt silinir.





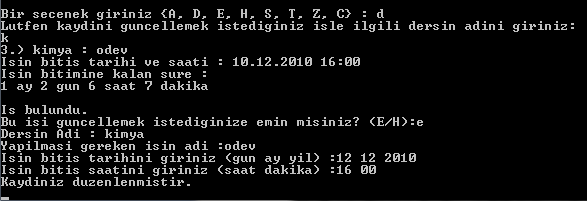
**Kayıt Arama:**

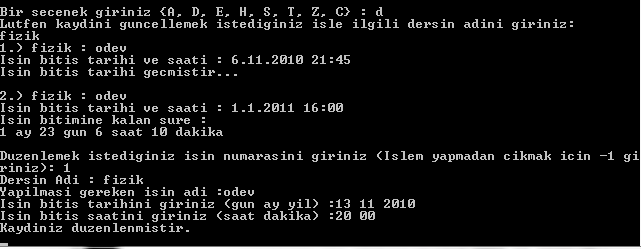
Menüye “a” veya “A” girilerek arama işlemi başlatılır. Arama yapılacak ders ismi girilir. Bulunan kayıtlar ekrana yazılır. Böylece arama işlemi tamamlanmış olur.

****

**Kayıt Güncelleme:**

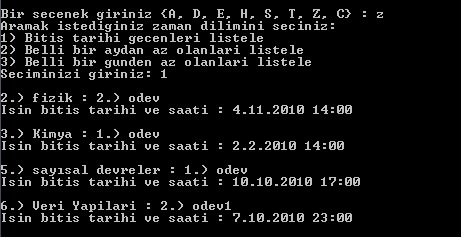
Menüye “d” veya “D” girilerek güncelleme işlemi başlatılır. Güncelleme yapılacak işle ilgili dersin ismi girilir. Bulunan kayıtlar ekrana yazılır. Eğer tek bir kayıt bulunduysa evet veya hayır denilerek işleme devam edilir.1 den çok kayıt bulunduysa güncelleme yapılacak işin numarası girilir. Sonra yeni bilgiler girilir. Böylece güncelleme işlemi tamamlanmış olur.

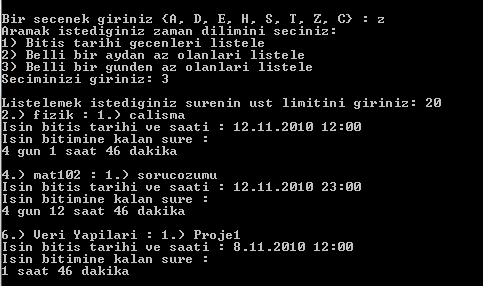


****

**Zamana Göre Listeleme:**

Menüye “z” veya “Z” girilir. Bitiş tarihi geçenleri, belli bir aydan az olanları ve belli bir günden az olanları listeleme şeklinde 3 tür listeleme şekli vardır. Hangi tür listeleme yapılmak isteniyorsa önce o seçilir. Eğer belli bir günden az veya belli bir aydan az olanlar listelenmek isteniyorsa bu değerlerde sayısal olarak girilir.





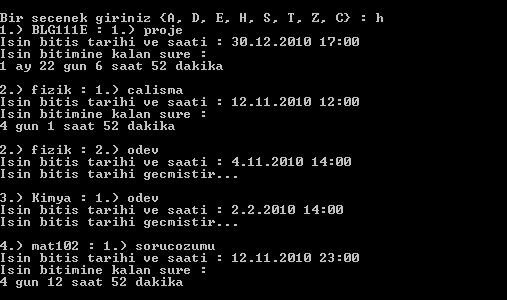
**Hepsini Sil:**

Menüye “t” veya “T” girilerek tüm ajanda boşaltılır.



**Hepsini Listele:**

Menüye “h” veya “H” girilerek tüm liste ekrana yazdirilir.



**Çıkış:**

Programdan çıkmak için menüde “c” veya “C” girilir. Gelen soruya “e” yanıtı verilerek program sonlandırılır.

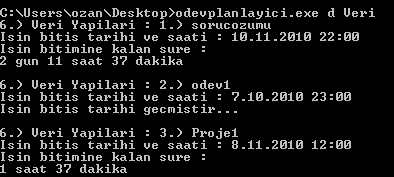


**B)Argümanlı Çalıştırma**

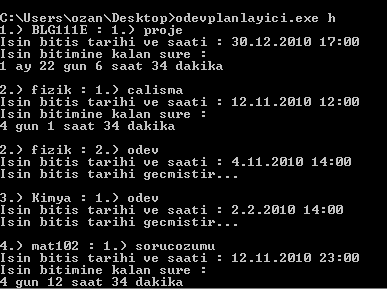
1) >odevplanlayici.exe t dosyaismi girilerek tüm ajanda ismi verilen dosyaya yazdırılır. Dosya isminin uzantılı olması gerekmektedir.



2)>odevplanlayici.exe d dersimsi girilerek arama yapılan dersle ilgili işlelistelenir.



3) >odevplanlayici.exe h girilerek tüm liste yazdırılır.



4) >odevplanlayici.exe b girilerek bitiş tarihi geçen işler yazdırılır.

