

ÜMİT YILMAZ 200601039

BİLGİSAYAR

PROGRAMLAMA PROJE

PROBLEMİN BELİRLENMESİ

Yapacağımız proje STOK TAKİP OTOMASYONU.

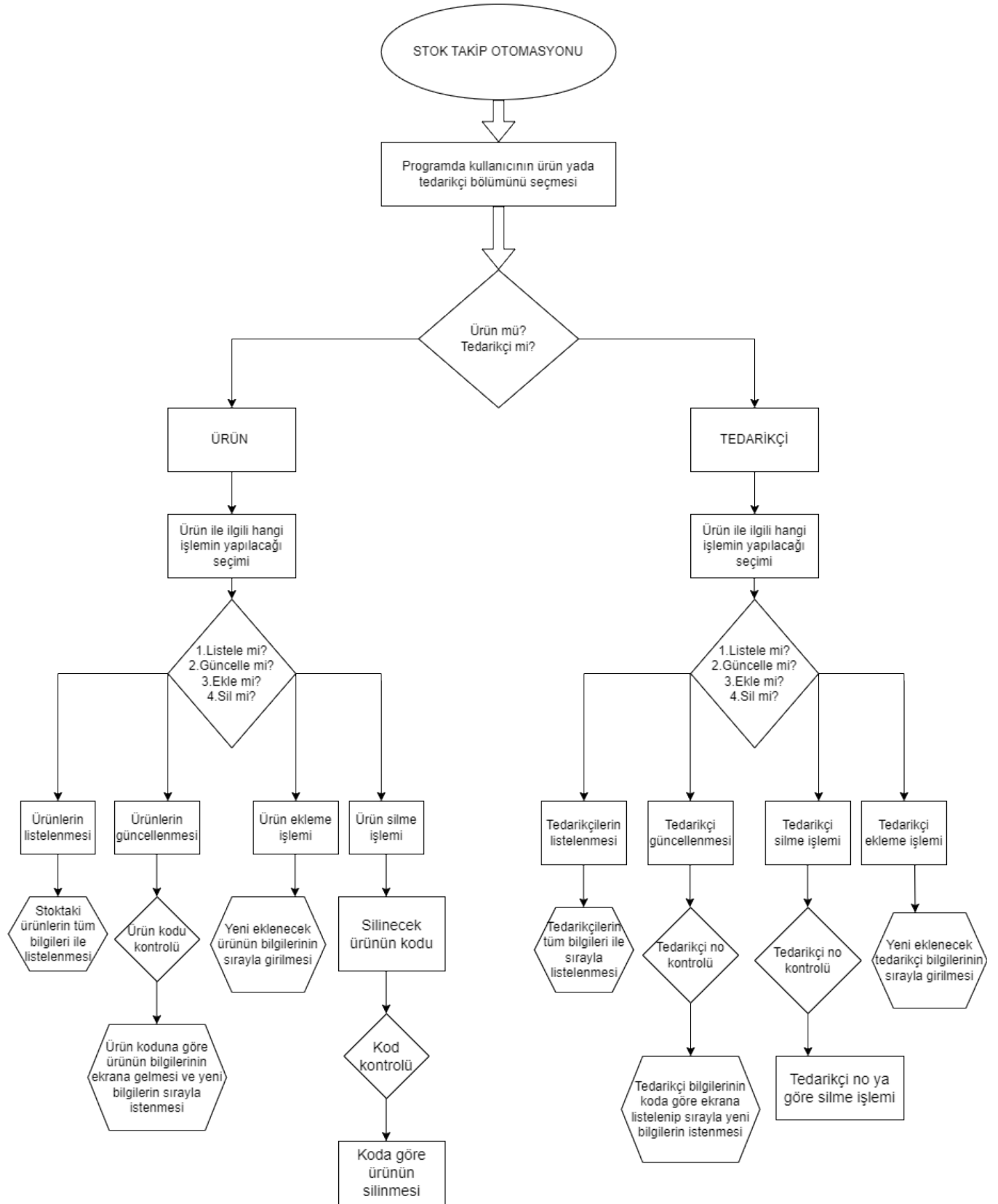
Bu programda başrolde iki ana yapı olacak. Bunlar ürünler ve tedarikçiler.

Ürün kısmında struct olarak bir ürün yapısı oluşturulacak. Bu yapıda ürünlere ait **kategori (sabit), ürün kodu, ürün adı, ürün satış fiyatı, stok miktarı** verileri tutulacak. Programda ürün bölümünde ürünleri listele, sil, güncelle ve ekle bölümleri olacak. **Ürünleri listele** kısmına girdiğimizde o an stokta ne kadar varsa tüm bilgileri ile birlikte tüm ürünler listelenecek. **Ürün güncelle** kısmına girdiğimizde bizden güncellemek istediğimiz ürünün kodu ile güncellenecek ürünün bilgileri güncelleyebileceğiz. **Ürün ekle** kısmında stokta bitmiş bir ürün varsa o ürün için ekleme yapabileceğimiz gibi yeni bir ürünü de tüm bilgileri ile(tedarikçi no vs.) ekleyebileceğiz. **Ürün sil** kısmı çıkarmak istediğimiz bir ürünü programdan silmemiz için var.

Tedarikçi kısmında tedarikçi için bir struct yapısı kurulacak.

Bu struct yapısında **tedarikçi no, tedarikçi adı, adres, şehir (sabit)** bilgileri tutulacak. Tedarikçi bölümüne girdiğimizde tedarikçi bilgilerini güncelleme, listeleme, ekleme ve silme gibi işlemler yapılabilecek. Tedarikçi güncelle kısmında tedarikçi no ya göre bir güncelleme işlemi yapabileceğiz. Tedarikçi listeleme kısmı ile tedarikçileri listeleyebileceğiz. Tedarikçi sil kısmında sistemden silinmesi gerek bir tedarikçiyi silebileceğiz. Tedarikçi ekleme işlemi ile yeni bir tedarikçi ekleme işlemi yapabileceğiz.

AKIŞ VE ALGORİTMA ŞEMASI



STRUCT YAPILARI

```
struct urun{
```

```
int urunkod;  
char urunisim[20];  
int urunkategori;  
int urun_stok_miktari;  
int urun_satis_fiyat;
```

```
};
```

```
struct
```

```
    adres{ char  
    sehir[20]; char  
    ilce[20]; char  
    ulke[20];
```

```
};
```

```
struct tedarikci{
```

```
    int tdrk_no;  
    char tdrk_isim[20];  
    char tdrk_adres[100];
```

```
};
```

```
typedef struct{
```

```
    int gun;
```

```

        int ay;
        int yıl;
    } tarih;
    -----
    struct stokGiris{
        int stkgrs_id;
        int tdrk_no;
        int urunkod;
        int stkgrs_alis_miktarı;
        int stkgrs_alis_fiyat;
        tarih stkgrs_tarih;
    };
    Int main ()
    {
        Struct urun urunler[30];
        Struct tedarikci tedarikçiler[10];

    }

```

DOSYA İŞLEMLERİ

Urun ekle fonksiyonunda Urun.txt dosyasında önce okuma yapılır bu okuma yapılarak eklenecek öğrenci numarasında başka öğrenci var mı kontrol yapılır. Daha sonra append(a) modunda öğrenci eklenir.

```

void urun_ekle(urun urunler[],urun yedekUrun[ ],urun *ptrUrun[],int
*i)
{
    int k=*i;

```

```

int x=0;
FILE *fp=fopen("Urun.txt","r");
while(!feof(fp))
{
    //AYNI NUMARADA URUN VARMI KONTOL YAPILIYOR
    fscanf(fp,"%d %s %s %f
%d",&yedekUrun[x].kodu,&yedekUrun[x].adi,&yedekUrun[x].ktg.ad
,&yedekUrun[x].satis_fiyati,&yedekUrun[x].stok_miktari);

    if(yedekUrun[x].kodu==ptrUrun[k]->kodu)
    {
        printf("\n\nAyni Urun Koduna Sahip Bir Urun Var!!\n");
        printf("!!Lutfen urun kodunuzu kontrol edin.\n");
        printf("!!Yeni bir urun kodu deneyin yada bu urun koduyla
islem icin ANAMENU den stok kismina gidiniz\n\n");
        return ;
    }
    x++;
}
fclose(fp);
fp=fopen("Urun.txt","a");
if(fp==NULL)
{
    printf("!!Urun Dosyasi Acilmadi!");
}

fprintf(fp,"\n%d %s %s %2.f %d",ptrUrun[k]->kodu,ptrUrun[k]-
>adi,ptrUrun[k]->ktg.ad,ptrUrun[k]->satis_fiyati,ptrUrun[k]-
>stok_miktari,ptrUrun[k]->stok_miktari);
fclose(fp);

```

```
}
```

Urun Guncelle fonksiyonunda ilk olarak okuma modunda bilgiler alınırken güncellenmek istenen urunun numarası if içerisinde kontrol edilir. Numara eşit olunca yeni veriler girilip bu sefer yeniden dosya oluşturma modunda yeni verilerler urunler tekrar yazılır.

```
void urun_guncelle(urun yedekurun[],int urunkodu)
{
    FILE *fp=fopen("Urun.txt","r");
    int i=0;
    char satir;

    while(!feof(fp))
    {
        fscanf(fp,"%d %s %s %f
%d",&yedekurun[i].kodu,&yedekurun[i].adi,&yedekurun-
>ktg.ad,&yedekurun[i].satis_fiyati,&yedekurun[i].stok_miktari);
        if(yedekurun[i].kodu==urunkodu)
        {

            printf("Urun Kodu:\n");
            scanf("%d",&yedekurun[i].kodu);
            printf("Urun Adi:\n");
            scanf("%s",&yedekurun[i].adi);
            printf("Urun Kategori Adi:\n");
            scanf("%s",&yedekurun[i].ktg.ad);
            printf("Urun Fiyati:\n");
            scanf("%f",&yedekurun[i].satis_fiyati);
            printf("Urun Stok Miktarı:\n");
            scanf("%d",&yedekurun[i].stok_miktari);
        }
    }
}
```

```

        i++;
    }
    fclose(fp);
    fopen("Urun.txt","w");
    for(int a=0; a<(i-1); a++)
    {
        fprintf(fp,"%d %s %s %3f
%d\n",yedekurun[a].kodu,yedekurun[a].adi,yedekurun[a].ktg.ad,yede
kurun[a].satis_fiyati,yedekurun[a].stok_miktari);
    }
    fclose(fp);

```

Ogrenci sil fonksiyonunda ilk olarak yine dosya okuma modunda bilgiler yedek bir urun yapısı içersinde tutulur. Aynı zamanda verilen değişkenler sayesinde silinecek urunun nosu da tutulur .Ve dosya w modunda açılarak bu sefer silinecek urune kadar yazılır dosya. Silinecek veride bir sonraki değerdeki bilgiler ustune yazılır bu sayede silinmiş olur.

```

void urun_sil(urun yedekUrun[])
{
    FILE *fp=fopen("Urun.txt","r");
    int i=0,a=0,k=0,l=0;
    int urunNo;
    if(fp==NULL)
    {
        printf("Dosya Acilmadi");
    }
    printf("Silinecek Urun No: ");
    scanf("%d",&urunNo);

```



```

while(!feof(fp))
{
    fscanf(fp,"%d %s %s %f
%d",&yedekUrun[i].kodu,&yedekUrun[i].adi,&yedekUrun[i].ktg.ad,
&yedekUrun[i].satis_fiyati,&yedekUrun[i].stok_miktari);
    if(yedekUrun[i].kodu==urunNo)
    {
        l=i;
        k++;
    }
    i++;
}

fclose(fp);
fp = fopen("Urun.txt","w");
for(a=0; a<(i-1); a++)
{
    if(yedekUrun[a].kodu==yedekUrun[l].kodu)
    {
        strcpy(yedekUrun[a].adi,yedekUrun[a+1].adi);
        strcpy(yedekUrun[a].ktg.ad,yedekUrun[a+1].ktg.ad);
        yedekUrun[a].kodu = yedekUrun[a+1].kodu;
        yedekUrun[a].satis_fiyati=yedekUrun[a+1].satis_fiyati;
        yedekUrun[a].stok_miktari=yedekUrun[a].stok_miktari;
    }
}
if(k==0)
{
    printf("Urun Bulunamadi!\n");
    return;
}

```

```

    }
    for(a=0; a<(i-2); a++)
    {
        fprintf(fp,"%d %s %s %f
%d\n\n",yedekUrun[a].kodu,yedekUrun[a].adi,yedekUrun[a].ktg.ad,y
edekUrun[a].satis_fiyati,yedekUrun[a].stok_miktari);
    }
    fclose(fp);
}

```

Fputc içinde fgetc kullanarak satır satır okuma yapılır ve konsolda gösterilir.

```

void urun_listele()
{
    FILE *fp;
    fp=fopen("Urun.txt","r");
    printf("-----\n");
    while(!feof(fp))
    {
        fputc(fgetc(fp));
    }
    printf("-----\n");
    fclose(fp);
}

```