

# Yazılım Ödevleri Kılavuzu

Bölümde hazırlanan yazılım ödevleriyle birlikte teslim edilecek raporların aşağıdaki yapıda olmaları gerekir:

## KAPAK

Dersin adı, kodu, projeyi yapan öğrenci ya da öğrencilerin adları, numaraları, ödevin teslim edildiği öğretim elemanının adı ve teslim tarihi belirtilecektir.

## BAŞLIK

### 1. Giriş

Projenin ayrıntılı tanımı ve amacı verilir.

### 2. Geliştirme ve Çalıştırma Ortamları

Projenin geliştirildiği ortam (işletim sistemi, derleyici, v.b.) ve çalıştırılması için gereken ortam (işletim sistemi, bellek gereksinimleri, v.b.) verilir. Program kısıtlamalarını belirleyen sabitler tanıtılır. Programın hangi kaynak dosyalarından oluştuğu, kaynak dosyalarının nasıl düzenlendikleri, programın nasıl derleneceği ve çalıştırılacağı açıklanır. Gerekliyse kullanıma ilişkin bilgiler ekran görüntüleriyle desteklenerek verilir.

### 3. Veri Yapıları ve Değişkenler

Projede kullanılan veri yapılarının şekiller yardımıyla açıklamaları yapılır. Kullanılan temel değişkenler ve işlevleri belirtilir.

### 4. Program Akışı

Programın akışının hangi altprogramdan başladığı ve nasıl sürdüğü şekiller yardımıyla (altprogram çağrı ağacı, akış diyagramı, v.b.) açıklanır. Her altprogram için kaynak kodunun içine yazılmış bulunan açıklamalar rapora eklenir (bkz. [Kod açıklamaları](#)). Nesneye dayalı dillerle geliştirilen ödevlerde, yukarıdakilere ek olarak, sınıf hiyerarşisi çizimle gösterilir.

### 5. Sonuç

Proje üzerine yorumlar, öneriler, geliştirme sırasında karşılaşılan zorluklar bu bölümde belirtilir.

## Kod Açıklamaları

Kaynak kodunun yazımında aşağıdaki kurallara uyulmalıdır:

### Kaynak Dosyaları

Her kaynak dosyasının başında açıklama olarak, dosyanın adı ve içerdiği veri yapıları, genel değişkenler ve altprogramların işlevleri yazılmalıdır.

### Altprogramlar

Her altprogramın başında açıklama olarak altprogramın adı, amacı, giriş parametreleri, çıkış parametreleri ve gerekiyorsa kullandığı veri yapıları / algoritmaları belirtilmelidir. Aşağıda örnek olarak bir C fonksiyonu için açıklama verilmiştir:

```

/*****
* Fonksiyon adı:      factorial
* Amacı:              giriş parametresinin faktöryelini hesaplar
* Giriş parametresi: faktöryeli hesaplanacak sayı (n)
* Çıkış parametresi: n!
* Veri yapıları:     ...
* Algoritması:       rekürsif olarak gerçekleştirilmiştir
*****/
float factorial(int n)
{
    ...
}

```

### Kritik Bölgeler

Kodun veri yapıları ya da algoritma açısından anlaşılması kritik bölgelerinde kod içine açıklama yazılacaktır.

### İsimler

Değişkenler, sabitler, altprogramlar, sınıflar gibi her türlü varlığa anlamlı isimler verilecektir.

### Girintileme

Tutarlı bir girintileme (indentation) yöntemi uygulanacaktır. Örneğin:

```

float factorial(int n)
{
    float result = 1;
    int i;

    for (i = 2; i <= n; i++)
        result *= i;
    return result;
}

```