BÖLÜM 2: Algoritmalar Programlamaya Giriş

Dr. Öğretim Üyesi Erhan ERGÜN

UZEM | ONDOKUZ MAYIS ÜNİVERSİTESİ UZAKTAN EĞİTİM MERKEZİ

İÇERİK

- Programlama Nedir?
- Algoritma
- Akış Diyagramı Şekilleri
- Akış Diyagramı Örneği

Programlama Nedir?

Programlama, sınırlı sayıda kelime bilgisine sahip ancak istenileni harfiyen yerine getiren bir varlığa nasıl bir iş yaptırabiliriz sorusunun cevabı olarak da düşünülebilir. Programlanan makinenin yapacağı iş hakkında bir düşüncesi yada tercihi olamaz; bu konudaki bilgi ve düşünce programcıdadır.

Bilgisayar programlama söz konusu olduğunda, çok farklı diller ve yöntemler söz konusudur. Diller, insana yada makineye yakınlık (alt-orta-üst düzey diller), alana yönelik oluş (veri tabanı dilleri, yapay zeka dilleri, matematiksel diller), ya da programlama tekniği açısından (nesne-yönelimli diller, yapısal diller) sınıflandırılabilir.

Programlama, bir programlama dili ile ve genellikle bir programlama arayüzü (IDE: Integrated Development Environment - Tümleşik Geliştirme Ortamı) üzerinde yapılıyor olsa da aslında temelde dilden bağımsız bir faaliyet olarak ele alınmalıdır. Bir probleme özel geliştirilen programlama yaklaşımını pek çok dil ile kodlayıp çalıştırmak mümkündür.

Bir programlama problemini çözmek için tasarlanan yada izlenilen yola **algoritma** adı verilir.

Algoritma

Algoritma, bir yazılım problemini çözmek veya belirli bir amaca ulaşmak için tasarlanan yoldur. Bilgisayar biliminde bir işi gerçekleştirmek için tanımlanan, bir başlangıç durumundan başladığında, açıkça belirlenmiş bir son durumunda sonlanan, sonlu işlemler kümesidir.
Bir algoritmayı tasarlarken yada ifade ederken sözlü yada düz metin ifadelerinin genellikle yetersiz kaldığı görülür. Bu amaçla akış diyagramı (flow chart) adı verilen ifade yöntemi kullanılır. Akış diyagramlarında sadece işlemler gösterilmez, programın akışını da izlemek mümkündür.
Akış diyagramlarında belirli şekiller ve her şeklin temsil ettiği bir işlem kümesi vardır.

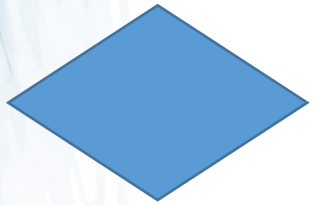
Akış Diyagramı Şekilleri

Başla/Dur: Başlangıç ve bitiş işlemleri için kullanılır

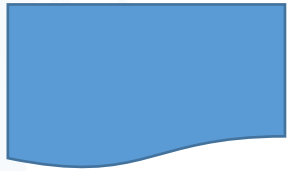
İşlem: Programın yürütülürken yapılan işlemler için kullanılır

Veri Girişi: Kullanıcıdan alınan veri için kullanılır

Akış Diyagramı Şekilleri



Sorgulama/Dallanma: Kutu içinde verilen şartın gerçekleşmesi ya da gerçekleşmemesi durumunda iki yoldan birine dallanmak için kullanılır



Çıktı: Programın ürettiği bir sonucu göstermek için kullanılır

Bu temel şekillerin dışında da sıralama, depolama, belge okuma gibi daha özelleştirilmiş şekiller de kullanılır ancak temel düzeyde bu beş şekil başlangıçtan bu güne kullanılagelmiştir ve temel ihtiyaçlarımızı karşılar.

Akış Diyagramı Örneği:

Akış diyagramı kullanımına örnek olarak 1'de 100'e kadar olan tamsayıları ekrana yazdıran bir algoritma verelim. Henüz bir programlama dilimiz olmamakla birlikte kullanacağımız işlemler ve terimler tüm kullanıcıların anlayacağı düzeyde seçilecektir.
Bir X değişkeni atayıp başlangıç değerini 1 olarak belirleyelim, X değişkenini önce yazdırıp sonra 100 olup olmadığını kontrol edelim. Henüz 100 değerine ulaşmamış ise X'i 1 artırıp yazdırma adımına geri dönelim.
Bu işlemleri 100 defa tekrarladığımızda X değişkeninin 1, 2, 3, ... ,100 değeri aldıktan sonra sınama aşamasında 100 değerinden sonra geri dönemediğini görürüz. Bu şekilde algoritma sonlanmış olacaktır.
Bu işlemlere ilişkin akış diyagramı sıradaki şekilde verilmiştir.
Aynı problemi farklı akış diyagramları ile çözmek de mümkündür. Örneğin X=100 yerine X≥100 sorgulaması da yapılabilir ve aynı sonuç elde edilir.

Akış Diyagramı Örneği: