

# PROGRAMLAMA DİLİ ÖĞRENMEDEKİ ZORLUKLAR VE ÇÖZÜM YAKLAŞIMLARI

Nazım İMAL<sup>1</sup>, Mehmet ESER<sup>2</sup>

<sup>1</sup> Bilecik Üniversitesi, Mühendislik Fakültesi, Bilecik

nazim\_imal@hotmail.com

<sup>2</sup> Bilecik Üniversitesi, Meslek Yüksekokulu, Bilecik

mehmet.eser@bilecik.edu.tr

## Özet

*En gelişmiş şekilde tasarlanan altyapıya sahip bile olsa, programlanmamış yada işletim sistemine sahip olmayan bir bilgisayardan faydalanabilmek mümkün değildir. Bir bilgisayar için donanım ne kadar zorunlu ise, en az onun kadar yazılım da zorunludur. Donanım yapıları farklı olsa bile, kullandıkları yazılımlar sayesinde, bilgisayarlar bilgi iletişim sisteminin bir servis sağlayıcısı, bir terminali ya da bağımsız bir elemanı haline gelirler. Bu yazılımları oluşturmak için, bilgisayarların makine ve assembler dillerinde direkt programlama yapmak imkansız denecek derecede çok çok zor olduğundan, makine ve assembler dillerine adapte olabilecek programlama dilleri geliştirilmiştir. Geliştirilen bu programlama dillerini kullanarak, kurumsal ve bireysel kullanıcıların, kendilerine ait özgün paket programlar yapabilmesi mümkün olmaktadır. Bu sebeple programlama dillerinden birini yada birkaçını öğrenmek, yazılım ve programlama ile uğraşan kişiler için zorunlu hale gelmiştir. Bu süreçte programlama dili öğrenmede farklı zorluklar ve hatalar ile karşılaşabilmektedir. Programlama dili öğrenmede ki hatalar, zorluklar ve çözümler çoğunlukla iç içe olduğu için burada birlikte irdelenecektir. Özellikle hataların sayısını artırılabilirliği etki yapan aynı anda farklı dillerin öğrenilmesi ve öğrenmede yanlış sıralama konuları üzerinde durulacaktır.*

## 1. Giriş

Programlama dili öğrenmeye ilk başlayanların sordukları “Hangi dili öğrenmeliyim?” sorusu en önemli konuyu içermektedir. Fakat bu soruyu cevaplamadan önce bilinmesi gereken konular ve cevaplanması gereken sorular vardır.

Bilinmesi gerekenlerin başında, problem çözme mantığı ve algoritma gelmektedir. Cevaplanması gerekenler sorular ise “İhtiyaç nedir, bu ihtiyaç hangi dil ile giderebilir ve bu dil gelecekte

kullanılabilir mi? “ tarzındaki sorularla ortaya çıkmaktadır. Tercih edilen dil ülkeden ülkeye, sektörden sektöre değişebilmekle beraber programlama dili öğrenmede metot ve öğrenme sırası çok daha fazla öne çıkmaktadır (Şekil 1).



Şekil 1. Programlama Dilleri

Diğer önemli husus ezberci bir yaklaşımın mı, yoksa sistematik bir yaklaşımın mı temel alınması gerektiğidir. Farklı kalıpları gerektirmeyecek, benzer programlar için ezberci yaklaşım çözüm olabilmekle beraber, farklı konularda farklı çözümler gerektiğinde sistematik yaklaşımın temel alınması zorunlu olmaktadır.

## 2. Algoritma ve Problem Çözme Tekniği

Programlama eğitimine başlayan bir öğrenci için öncelikle öğrenilmesi gereken konu algoritmadır. Algoritma; iyi tanımlanmış kuralların ve işlemlerin adım adım uygulanmasıyla bir sorunun giderilmesi veya sonuca en hızlı biçimde ulaşılması işlemi şeklinde tanımlanmaktadır.

Bir sorun yada problemi çözme işleminde modelleme yaparak, o modellemenin gerektirdiği basamakları oluşturabilmek algoritma sisteminin temel yapısını oluşturur. Algoritma sistemi ne kadar gerçekçi olarak oluşturulabilirse, problemlerin net bir şekilde çözümlenmesinde büyük kolaylık sağlayacaktır. Akış diyagramları

ise, algoritmaların kısa süre içinde algılanmasında kolaylık sağlamaktadır.

Programlama eğitimi verilirken bazen algoritma dersleri herhangi bir dil olmaksızın verilirken, bazılarında ise bir dil ile birlikte verilmektedir. Tablo 1'de de görüleceği üzere, alçak seviyeli programlama dilleri assembler ve makine diline yaklaşımı, orta ve yüksek seviyeli diller ise makine ve assembler dilinden uzaklığı ifade etmektedir. Bu sebeple, programlama eğitimi verilirken kullanılacak olan dilin seviyesi büyük önem arz etmektedir. Programlama eğitimine başlangıç açısından dil kullanılmaksızın yalnızca algoritma kurulumu öğretimi de uygun olmasına rağmen, dil ile birlikte algoritma anlatılacaksa, mutlaka, anlaşılması kolay, kuralları basit olan yüksek seviyeli bir dil tercih edilmelidir.

Tablo 1. Örnek programlama dilleri ve seviyeleri

Seviye	Örnek Programlama Dilleri
Çok yüksek seviyeli diller	VisualBasic, VB.NET, Acces, Foxpro ...
Yüksek seviyeli diller	Pascal ,Basic ,Fortran...
Orta seviyeli diller	C ,C++, C# , Java ,ADA...
Düşük seviyeli diller	Assembly...
Makina dili	1 ve 0

Diğer durumda, yani, anlaşılması biraz daha zor olan orta yada alçak seviyeli dil tercih edilirse, algoritma öğrenilebilmekle birlikte, dilin gerektirdiği ekstra kuralların da öğrenilmesi zorluğu ortaya çıkacaktır. Oysa ki bu adımda amaç, henüz dil öğrenmek değil, dil öğrenimine temel teşkil eden algoritmayı öğrenmektir.

## 2.1. Aynı Anda Farklı Dil Öğrenme Hatası

Aynı anda farklı dil öğreniminde en büyük sıkıntı yazım kurallarında olmaktadır. Her dilin kendine has yazım kuralları ve kalıbı bulunmaktadır. Aynı anda birden fazla dil öğrenmeye kalkıldığında, diller arası karışıklık olması muhtemeldir. Aslında bir dilin kuralları öğrenilip daha sonra diğer dile geçildiğinde önce ki bilgilerle daha kolay bir öğrenim olmakta, üstelik diller arasında karşılaştırma fırsatı da doğmaktadır.

## 2.2. İhtiyaç

Programlama dillerini en genel haliyle Web ve Masaüstü Programlama dilleri olarak ikiye ayırabiliriz. Dil öğreniminde ihtiyaca bağlı ve aynı zamanda güncel bir dil öğrenmek, daha sonra ortaya çıkabilecek zaman, emek ve para kayıplarını önleyecektir. İhtiyaç veya ihtiyaçlar belirlenip ona göre seçim yapıldıktan sonra öğrenme yada öğretme aşamasına geçilmelidir. Eğer ihtiyaçlar göz önüne alınmadan bir seçim yapılacak olursa, öğrenilmiş olan dil işe yaramayabilir yada öğrenci gereksiz bilgilerle zaman ve emek kaybına uğramış olur. Örneğin matematiksel alt yapısı zayıf yada zor tanımlanabilen bir dili mühendislik için tercih etmek ileri aşamalarda sorun olmaktadır.

## 2.3. Ezberci Yaklaşım

Ezberlenen bilgi, unutulma ihtimali yüksek olan bilgidir. Bu yüzden sadece ezberlenen, fakat yorumu yapılamayan bilgiler bilişimde değerli değildir. Bir şey bilmek değil, bilgiye ulaşmanın yollarını bilmek günümüzde daha büyük önem taşımaktadır. Problemler hiç değişmiyor olsaydı onları ezberler, bir tablo haline getirir, bilgisayara yükler ve her sorun rahatlıkla çözülebilirdi. İnsana faktörüne de gerek kalmazdı. Bu sebeple çözüm yaklaşımları öğrenilerek, bunlar karşılaşılan her probleme uygulanarak kullanılmaya çalışılmalıdır. Basit bir örnekle ifade etmeye çalışalım:

Ezberci yaklaşım: Apartmanın girişinden birinci kattaki evimin kapısına kadar 6 basamak çıkarım. Kapıya ulaşırım.

Sistematik yaklaşım: Apartmanın girişinden birinci kattaki evimin kapısı için ard arda gelen basamaklar bitinceye kadar çıkarım. Kapıya ulaşırım.

Sadece basamak durumlarını göz önüne alarak bu yaklaşımları bir robota yüklediğimizi varsayalım. Her iki robot da kapıya ulaşır.

Peki başka bir apartmanda basamak sayısı değiştiğinde ilk robotun hali ne olur? Elbette değişen durumlara ayak uyduramaz. Yeniden program yüklenmesi gerekir. Bu iki apartmanda da doğru çalışması istense ne olur? Elbette her iki apartman için gerekli bilgilerin olduğu yeni bir program yüklenmelidir.

Peki apartman sayısı 1000 adet ise? İşte ezberci yaklaşımın hayata uygulanmasındaki zorluklar burada ortaya çıkmaktadır. Algılama ve elektromekanik sınır değerlerini aşmamak şartı ile ikinci robot her türlü basamak konumu ve sayısı için çalışır, değişiklik ve müdahale gerektirmez.

Esnek yaklaşımları olduğundan değişen durumlara uyum sağlayabilir.

Kısacası, ezberci yaklaşımda her tür özel durum için öğrenilen bilgi sadece o özel durumda kullanılabilir. Diğer durumlarda o bilginin geçerliliği yoktur. Zihinde sürekli olarak taşınan, ama kullanılmayıp sadece hamallığı yapılan bir bilgi yükü olur. Sistemik yaklaşımda ise bilginin birden çok ilişkisi bulunur. Öğrenilen bir bilgi birden fazla yere uygulanabildiğinden, verilen eğitimden çok daha fazla verim sağlanabilir.

### 3. Sonuç

Programlama dili eğitimi, eğitimin sonunda nasıl bir programcı yetiştirilmek isteniyorsa, o doğrultuda olmalıdır. Bir işletmede sahip olunan yazılıma yönelik olarak, sadece o yazılımın gerektirdiği, belirli kalıplarla sınırlandırılmış ve nispeten kısa süreli bir eğitim makul olmakla beraber, farklı konularda farklı problemlerle karşılaşacak ve çözüm yolları üretecek bir programcının eğitimi çok daha geniş perspektifte olmalıdır. Bu sebeple, farklı sorunlara çözüm üretmesi istenen bir programcı eğitiminde geniş yelpazede sorunlara çözüm olabilecek bir yada birkaç programlama dili öğretimi hedef alınmalıdır.

Programlama dili öğrenmek uzun bir tecrübeyi gerektirir ve dolayısıyla bir süreçtir. Yukarıda bahsedilen konular, bu sürecin temelinde neler yapılması ve yapılmaması gerektiği konularını kapsamaktadır. Programlama eğitimi, başlangıç aşamasında ya hiç dil olmaksızın algoritma temelli, yada algoritma ile birlikte yüksek temelli bir dil ile birlikte başlatılmalıdır. İlerleyen aşamalarda orta ve alçak seviyeli bir dil ile eğitim devam ettirilebilir.

Programlama eğitimi sırasında bir dil öğrenilirken, ikinci bir dilin aynı periyotta öğrenilmeye çalışılması, sorun olabilmektedir. İlk dil için ayrılan periyot sona erdikten sonra, yeni dil için farklı bir periyot tercih edilmesi daha verimli olmaktadır. Tercih edilen dilin ihtiyaçlara ne derece cevap verebildiği de önem taşır. Doğru amaca hizmet edebilmek için uygun alt yapıya sahip bir dil tercih edilmelidir.

Öğrenilen dil ayrıca, farklı sorunlar ve programlar karşısında programcı tarafından esnek bir tarzda kullanılabilmelidir. Her türlü sorun yada problem karşısında farklı bir uygulamanın bilinmesini gerektiren dilleri öğrenmek çok zor olmaktadır, öğrenilmiş olsa bile, bu dillerle çözüm üretmek çok zahmetli ve uzun bir süreç almaktadır.

Bu sebeplerden ötürü, programcılık eğitiminde kolaydan zora doğru, ezbercilikten ziyade

sistemik öğrenmeyi hedefleyen, ihtiyaçları göz önünde bulundurarak, faydalı olmayı hedefleyen bir eğitim yaklaşımı temel alınmalıdır.

### 4. Kaynaklar

[1] Alaybeyoğlu A, Morkaya Ö. " Ülkemizdeki <http://www.enginkuzu.org/bilgi-ogrenme-algoritma.php>

[2] Türk Dil Kurumu, Web sözlüğü

[3] <http://enginyoyen.com/blog/tr/programcilik-ve-egitimi>

[4] Programcılık Mantığı, Kerem KÖSEOĞLU, Pusula Yayıncılık, 2004