

Bilgisayar Programlama Öğretiminde Yeni Bir Yaklaşım

Taner Arabacıoğlu¹, Halil İbrahim Bülbül², Ali Filiz³

¹ Adnan Menderes Üniversitesi, Aydın MYO, Bilgisayar Teknolojisi. ve Programlama Programı, 09010, Aydın

² Gazi Üniversitesi, Endüstriyel Sanatlar Eğitim. Fakültesi, Bilgisayar Eğitimi Bölümü, 06500, Ankara

³ Adnan Menderes Üniversitesi, Fen Edebiyat Fakültesi, Matematik Bölümü, 09010, Aydın

tarabacioglu@adu.edu.tr, bhalil@gazi.edu.tr, afiliz@adu.edu.tr

Özet: Programlama, herhangi bir problemin bir programlama dili kullanılarak çözülmesi için yazılan kod satırlarına verilen isimdir. Programlama mantığı öğretimi ise, programlama öğretiminin ilk ve en önemli basamağıdır. Her bilim dalında olduğu gibi başlangıcın olumlu olması diğer bir deyişle temelin sağlam atılması konunun anlaşılmasında oldukça önem taşımaktadır. Programlama dili eğitimi alan öğrencilerin yabancı dile ve programlamaya karşı olan olumsuz önyargıları, programlama dilinin öğretimini zorlaştırmaktadır. Eğitim kurumlarında programlama mantığı genelde teorik bir yöntemle verilmektedir. Teorik yöntemin öğretimde hem sıkıcı hem de çok etkili bir yöntem olmadığı bilinmektedir. Öğretimin uygulamalı bir yöntemle yapılması teorik yöntemle göre başarıyı arttıracığı açıktır. Bu çalışma ile programlama mantığı öğretiminde kullanılmak üzere, bir uygulama dili tasarlanmıştır. Uygulama dili kullanılarak programlama mantığı öğretiminde, programlama öğretimi etkililiğini arttırmak için teorik bilgilerin uygulamaya dönüştürülebilmesi amaçlanmıştır.

Anahtar Sözcükler: Programlama mantığı, programlama dili, uygulama dili

A New Approach to Computer Programming Teaching

Abstract: Programming is the name given for the code lines written for solving any problem by using a programming language. Teaching of programming logic is the first and most important phase of programming teaching. Like in each science branch, having the starting point as positive, that is to say having a sound ground is very important for understanding of the subject. The negative prejudgements of the students having programming training against the foreign language and programming make teaching of a programming language difficult. Programming logic in schools are generally given by means of a theoretical method. It is known that theoretical method is boring and not very effective one. It is obvious that giving the teaching through an applied method results in higher success than theoretical method. In this study, an application language is designed to use in the teaching of algorithm. Using the theoretical knowledge in application is aimed to increase the activities of programming teaching, using the application language in the teaching of algorithm.

Key Words: Programming Logic, programming language, application language

1. Giriş

Programlamada, problemin iyi anlaşılması ve iyi analiz edilmesinin önemi herkes tarafından bilinmektedir. Programlama dillerine ait komutlar birbirleri arasında farklılık göstermesine rağmen, çözüm için kullanılacak programlama mantığı tüm dillerde benzerdir.

Programlama öğretiminden önce gerçekleştirilen ve programlama için temel olarak adlandırılan algoritma diğer deyişle programlama mantığı öğretiminde ise konuşma dili kullanılmaktadır. Bunun sebebi ise herhangi bir programlama diline bağlı olmaksızın geliştirilmesinden ve kolay anlaşılabilmesinden kaynaklanmaktadır. Algoritma tasarımı sıra-

sında kullanılan komutlar kısa, öz ve anlaşılır olmalıdır.

Ülkemizde ise eğitim-öğretim süresi boyunca öğrencilere bir türlü kazandırılmayan yabancı dil seviyesi ve bir türlü kırılamayan önyargılar, programlama öğretiminin önündeki en büyük engellerden bir kaçıdır.

Yapılan bu çalışma ile algoritma mantığının, bilgisayar programcısı olacak kişilere kolayca kavratılabilmesi açısından alışıldan farklı bir yöntemle sunulmaktadır. Programlama mantığı öğretiminin kolaylaştırılması ve basit bir arayüz kullanılarak öğrenciye bir editör alışkanlığı kazandırılması hedeflenmektedir.

2. Programlama Mantığı Öğreticisi Tasarımı

Programlama mantığı öğretiminin kolaylaştırılması ve programlama mantığının teorik temelden pratiğe dönüştürülmesinin ne derece önemli olduğu yukarıda belirtilmişti. Bu amaçla programlama mantığını yeni bir yaklaşımla öğretmeyi amaçlayan bir uygulama dilinin geliştirildiği öğretici isimli sistem aşağıdaki özellikleri içermektedir;

- Windows ortamında çalışmaktadır.
- Dosya boyutunun oldukça küçük olması sebebiyle mail yoluyla ya da bir disketle rahatlıkla taşınabilir.
- Görselliği ön planda tutmaktadır
- Türkçe olduğu için de yabancı dile karşı olan ön yargılardan sıyrılmaktadır.
- Tahtada veya kağıt üzerinde yapılan işleri bilgisayar ortamına taşımakta ve çalışmayı zevkli hale getirmektedir.
- Bir bilgisayar destekli eğitim materyalidir.
- Bunların sonucu olarak da öğrencinin tamamen soyut kavramlarla çalışmasını önlediği ve öğrenci üzerindeki zihinsel yükü azalttığı için başarıyı olumlu yönde etkilemektedir.

2.1. Sisteme Genel Bir Bakış

Şekil 1'de öğretici sisteminin bir görüntüsü verilmiştir.



Şekil 1. Öğretici sistemi ara yüzü görünümü

Yukarıdaki ara yüz, kavramların ve sonuçların anlaşılmasını kolaylaştırmak amacıyla tasarlanmıştır. Ara yüz, 5 adet pencere ve 12 adet düğmeden oluşmaktadır.

Sol üst köşede editör penceresi, sağda yardım penceresi ve her bir komuta ait yardım metinlerini gösteren düğmeler, altta ise yazılan programları çözümleyen pencereler bulunmaktadır. Çözümleme pencereleri, kullanılan komutlar, kullanılan değişkenler, **Eğer** ifadesi pencereleridir. Yukarıda bahsedilen pencereler kullanılan komutlara bağlı olarak karşımıza çıkmaktadır. Örneğin Eğer ifadesi kullanıldıysa **Eğer** çözümleme penceresi, görüntülenecektir.

Sağ tarafta bulunan yardım düğmelerinin her biri bir komutu temsil etmektedir. Hangi komut ile ilgili bilgi alınmak isteniyorsa o düğme tıklanır. Alttaki pencerede de ilgili yardım konuları görüntülenir.

Sol üst köşedeki editör penceresinde, program çalıştığında imleç varsayılan olarak konumlanır ve programlar buraya yazılır. Editör penceresinin hemen altında bulunan **Kaydet** düğmesine tıklanıldığında yazılan program .RTF formatında kaydedilir. **Aç** düğmesi ile de daha önceden yazılan ve kaydedilen programlar editör penceresine getirilir. Hemen yanındaki **Çalıştır** düğmesi yazılan programı derler ve sonuçlarını görmemizi sağlar. **DURDUR** düğmesi ise

programın çalışmasını sona erdirir. **DURDUR** düğmesine basılmadan önce yazılan program mutlaka kaydedilmelidir.

Öğretici sisteminin açılışında Şekil 2'deki mesaj görüntülenir. (Algoritma tasarımı veya komutlar hakkında ipuçları almak için yardım düğmelerini ve programlama ipuçları düğmesini kullanabilirsiniz)



Şekil 2. Öğretici sistemi açılış mesajı

Yukarıda görüntülenen mesaj, öğretici sistemine ait özelliklerinin tam olarak kullanılabilmesini amaçlamaktadır. Yardım düğmeleri, genel yazım kuralları ve her bir komutun kullanımı ile ilgili bilgiler vermektedir. Uygulama dilinde olduğu gibi yardım konularının da Türkçe olması anlaşılabilirliği arttıracaktır. Aktif olarak yardım kullanımı programlamada sıkça karşılaşılan hatalara, anında çözüm bulabilmek açısından oldukça önem taşımaktadır. Şekil 3'te ise yardım düğmelerinden bir tanesinin görüntüsü bulunmaktadır.



Şekil 3. Yardım düğmesi görüntüsü

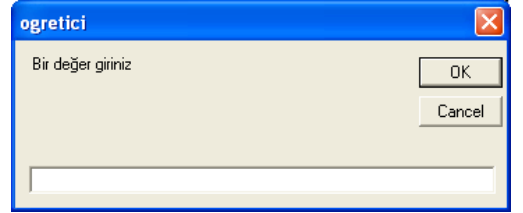
Ayrıca **Programlama İpuçları** düğmesiyle, verilen bir problemin anlaşılmasını dolayısıyla çözümünü kolaylaştıracak ipuçlarını sunmaktadır. Bu ipuçları, problem çözümü yöntemleridir. Çözümleme ise, problemde geçen

her bir kelimenin komut olarak karşılığının bulunabilmesidir. Eğer bir problem anlaşılmalı ve çözümlenmiş ise geriye kalan tek şey kodlamanın yazılmasıdır. Programlama ipuçları düğmesinin kullanımı ise Şekil 4'te gösterilmiştir.



Şekil 4. Programlama ipuçları düğmesinin kullanım görüntüsü

Öğretici sisteminde veri girişi ise Şekil 5'te gösterildiği gibi yapılmaktadır.



Şekil 5. Veri giriş görüntüsü

Öğretici sistemi, zihinsel olarak yapılan işlemleri görselleştiren bir sistemdir. Problem çözümüne ve yazılacak programlara yardımcı olmaktadır. Kullanıcılara kendi dillerinde, emir kipinden oluşan uygulama komutlarıyla program yazabilmeyi sağlar. Bunlar **Oku**, **Yaz**, **Eğer**, **Döngü**, **İken** ve **Tekrarla** komutlarından oluşmaktadır.

2.2 Çalışma Sonuçları

Bu bölümde öğretici sisteminin çalışmasını anlatmak için iki adet örnekten yararlanılmıştır. Birinci örnek, girilen iki sayıdan, birinci sayı, ikinci sayıdan küçük ise sayıların toplamını hesaplayan, büyük ise de sayıların çarpımını hesaplatan bir algoritma tasarımı istenirse eğer

şağıdaki kod satırları çözümü oluşturacaktır.

```
başla ;  
oku a ;  
oku b ;  
eğer a<b ise c:=a+b değilse  
c:=a*b ;  
yaz c ;  
dur .
```

Programa girdi olarak sırasıyla 4 ve 5 değerleri verildiğinde çözüm sonucu Şekil 6'daki gibi görüntülenecektir.



Şekil 6. Öğretici sistemi çalışma görüntüsü

Öğretici sistemini kullanılmadan gerçekleştirilecek çözümde ise doğruluk tablosu adı verilen yöntem ile gerçekleştirilecekti. Şekil 7'de üstteki problemin doğruluk tablosu kullanılarak tahtada ya da kâğıt üzerindeki çözümü gerçekleştirilmiştir.

a	b	c	eğer	ise	değilse	Ekran
4	5	9	a < b	a + b	a * b	9
			4 < 5	4 + 5	4 * 5	

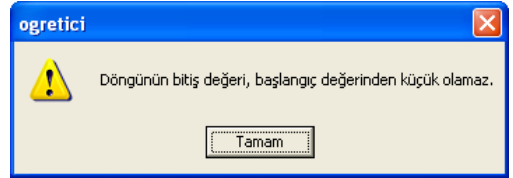
Şekil 7. Doğruluk tablosu kullanılarak gerçekleştirilen çözüm

Şekil 7'de de görüldüğü gibi çözüm tamamen teorik olarak gerçekleştirilmiştir. Çözüm sırasında oluşabilecek bir hata durumunu ders öğretim elemanı dışında belirlenme şansı yoktur. Dolayısıyla yapılan yanlış, öğretim elemanı tarafından düzeltilinceye kadar ortaya çıkmayacaktır.

İkinci örnek incelendiğinde aradaki fark da ortaya çıkacaktır. Girilen iki adet sayının toplamını, ekrana 3 kez yazdıran bir algoritma tasarımı Şekil 8'de görüldüğü gibi gerçekleştirilirse, döngü komutunun başlangıç ve bitiş değerlerinin yanlış verilmesi sonucunda, Şekil 9'da görülen uyarı penceresi ekrana gelecektir.



Şekil 8. Öğretici sistemi çalışma görüntüsü



Şekil 9. Uyarı penceresi

Üstteki örnekler değerlendirildiğinde öğretici sisteminin kullanımında oluşan farklar ortaya konmuştur.

3. Sonuç

Bu çalışmada, programlama mantığı öğretimi için kullanılabilir bir eğitim materyali geliştirilmiştir. Geliştirilen bu materyal, programlama mantığı öğretimi sürecinde bir takım zorluklar yaşanan bir problemin çözüm basamaklarından olan çözümü deneme ve çözümü geliştirme basamaklarını kolaylaştırmayı amaçlamaktadır.

Söz konusu çözüm basamakları, teorik yöntemle yapılan algoritma tasarımlarında, kullanılamamaktadır. Diğer bir deyişle öğrenci

yaptığı yanlış görememekte, göremediği için de yaptığı çözümün doğruluğundan hiçbir zaman emin olamamaktadır. Bununla birlikte öğretici sisteminin kullanılmasıyla bu eksiklik giderilmektedir.

Teorik yöntem olan doğruluk tablosu kullanılarak gerçekleştirilen çözümün bazı sakıncaları bulunmaktadır. Bu çözüm yöntemi tahata veya kağıt üzerinde yapılmakta ve soyut düşünebilme yeteneği üzerine kurulmuştur. Dolayısıyla öğrencinin tek başına çalışmasını destekleyen nitelikte bir yöntem değildir. Programlamaya yeni başlayanlar için, sürekli bir başkasının desteğine ihtiyaç duymasına neden olmaktadır. Bununla birlikte, doğru sonuca yönlendirmediği için moral bozukluğuna neden olabilmektedir.

Türkçe olan, bir uygulama dili tasarlanmıştır. Dil olarak Türkçe kullanılması, yabancı dile olan önyargıları ve yabancı dil eksikliğini ortadan kaldıracak, öğretim materyali olarak bir uygulama dilinin kullanılması da çok sayıda komut ve özellik içermesinden dolayı öğrencilere oldukça karmaşık gelen ve öğretim sırasında aksaklıklara yol açan programlama dillerinin kullanımını da azaltacaktır.

4. Kaynaklar

- [1] Eker, M., "Algoritmayı Anlamak", *Nirvana*, Ankara, 2004.
- [2] Ramadhan, H.A., "Programming by Discovery", *Journal of Computer Assisted Learning*, 16, 2000, 83-93.
- [3] Shackelford, R.L. and LeBlanc, R.L., "Introducing Computer Science Fundamentals Before Programming", *Frontiers in Education Conference*, 285-289, 1997, Pittsburg.
- [4] Mansoor, A. And Mohammed, H., "Design of an Arabic Programming Language", *Computer Languages*, 21, 1995, 191-201.
- [5] Guibert, N. and Patric, G., "Teaching and Learning Programming with a Programming by Example System", *International Symposium on End User Development*, 2003, Bonn.
- [6] Adamchik, V. and Gunawardena, A., "A Learning Objects Approach to Teaching Programming", *International Conference on Information Technology: Computers and Communication*, 96-99, 2003, Las Vegas.