

Programlama Dilleri Öğretimine Bir Model Önerisi:

Robot Programlama

Halil Ersoy, Rafet Orçun Madran, Yasemin Gülbahar

Başkent Üniversitesi, Bilgisayar ve Öğretim Teknolojiler Eğitimi Bölümü, Ankara
hersoy@baskent.edu.tr, omadran@baskent.edu.tr, gulbahar@baskent.edu.tr

Özet: Bilgisayar programlama, farklı becerilerin aynı anda kullanılmasını gerektiren bir problem çözüme ve üretim sürecidir. Bu becerileri kazandırmak için gerekli programlama eğitimi ve öğretimi farklı kurum ve kademelerde kimi zaman zorunlu dersler olarak, kimi zaman da isteğe bağlı kurslar şeklinde verilmektedir. Eğitimlerde başarıyı etkileyen olası faktörler arasında öğrencilerin programlamaya karşı tutumu, bilgisayar (yada bilişim teknolojileri) okuryazarlığı konusunda ön yeterlikleri ve seçilen dil ile amaçlanan becerilerin uyumu sayılabilir. Bu faktörlerin biri ya da bir kaç öğrencilerin programlama konusundaki başarılarını etkilediği gibi, bu becerilerin farklı alanlarda kullanılmasını da engellemektedir. Örneğin kullanılan dilden bağımsız olarak programlama becerisi, mantıksal düşünme ve algoritma oluşturma gibi bir çok alanda problem çözmeye yönelik becerileri ve hatta analitik düşünme becerisini de kazandırabilir.

Bu çalışmada, programlama becerisi kazandırmak, bir programlama dilinin öğrenimini kolaylaştırmak, motivasyonu yükseltmek ve başarıyı arttırmak için öğretim sürecinde robot programlama tekniklerini kullanan bir model önerilmektedir. Robot programlama dünyada ve son yıllarda Türkiye’de de farklı amaçlarla yapılan etkinliklerdir. Robot programlama etkinlikleri, programlama sürecini daha ilgi çekici hale getirmekte, yapılan öğretim etkinliklerinin öğrenciler tarafından daha anlamlı algılanmasını sağlamakta, rekabet ve takım çalışması gibi yeni öğrenme kuramlarına dayanan yöntemlerin kullanılmasına olanak sağlamaktadır. Özellikle son yıllarda bir çok öğrenci için erişimi kolaylaşan ve ucuzlayan robotların programlama alanında gerekli becerilerin kazanılmasına önemli katkılar sağlayacağı düşünülmektedir.

Anahtar Kelimeler: Robot programlama, programlama becerisi, programlama dilleri öğretimi

A Model Proposed for Teaching Programming Languages: Robotic Programming

Abstract: Computer programming is a problem solving and production process where different skills are executed simultaneously. Gaining the skills necessary for computer programming is possible in various schools and grades via either compulsory courses or voluntarily courses. Among the possible factors affecting the success in those trainings, students attitudes and perceptions toward the programming, adequacy in computer (or information technologies) literacy and consistency of the selected programming language and goals of the training. One or more of these factors may both affect the success of students in programming and impede the application of those skills in other field. For example, independent from the programming language selected, programming can provide problem solving skills like logical thinking and algorithm construction, and analitic thinking abilities.

In this study, in order to provide programming skills, to make learning programming language easy, to raise motivation and increase the success of students, a model using robotic programming

techniques is proposed. Robotic programming activities makes process of programming interesting, render teaching activities more meaningful and concrete for students, allow instructional methods based on new educational theories to be applied like competition and team working. It is thought that, robots which have been becoming cheaper and more accessible by the students in recent years will contribute to gain the skills necessary for programming.

Keywords: Robotic Programming, programming skills, teaching programming languages

1. Programlama Dili Nedir?

Elektronik cihazların farklı amaçlara hizmet edebilmek için programlanmaları gerekir. Programlama dili, bu amaç için hazırlanmış özel kelime ve sembollerden oluşan komutlar bütünüdür. Yazılan komutlar sonucunda bilgisayarın belli bir iş yapmasını sağlayan programlar, yani yazılımlar, ortaya çıkar. Farklı bilgisayarlar ve teknolojiler için oluşturulmuş neredeyse sayılamayacak kadar çok programlama dili mevcuttur. Her bir dilin kendine özgü deyim ve yazım kuralları ile amaca uygun komutlar oluşturulur ve çalıştırılır. Dile özgü komutların yazılması süreci de kodlama ya da programlama, ortaya çıkan son ürün ise program ya da uygulama olarak adlandırılır.

2. Programlama Dilleri Öğretimi

Programlama dilleri, bilgisayarın ilk ortaya çıkışından itibaren var olmuşlardır. Özellikle bilgisayarların kişisel olarak satın alınabilir hale gelmesinden sonra (1980 ve sonrası) artan bireysel olarak program yazma isteğiyle birlikte, programlama dili eğitimi farklı eğitim kurumlarında ve kademelerinde verilmeye başlanmıştır.

Günümüzde programlama dilini öğrenmek isteyen veya uzmanlık alanları gereği bilmek zorunda olan kişiler, farklı kurum ve kademelerde bu eğitimi almaktadırlar. Dünyada ve Türkiye’de eğitimin ilk kademesinden (ilköğretim) son kademesine kadar farklı öğretim programlarında programlama dili dersleri verilmektedir. Öte taraftan özel kurslar ile kişilere ihtiyaçları doğrultusunda programlama dersleri de verilmektedir. Verilen eğitimlerin

amaçları farklı olabilmektedir. Farklı düzeylerde programlama ile oluşturulan üründen çok, problem çözme becerilerinin gelişmesi için de dersler verilmektedir. Örneğin üniversitelerde ve meslek liselerinde programlamacı olmak için gerekli beceriler kazandırılmaktadır.

Bu derslerin büyük bir çoğunluğu bilgisayar laboratuvarlarında dersin sorumlusu tarafından, önce düz anlatım ardından da uygulama etkinlikleri ile verilmektedir. Kişilerin tek başlarına kendi olanakları ile programlama dili öğrenme şansları da vardır, ancak genelde donanım ve yazılım gereksinimi olmasından veya ön yeterlilik olan diğer bazı becerilere sahip olunmasından dolayı programlama dilleri, sınıf ve laboratuvarında verilen dersler ile daha etkili bir şekilde öğrenilmektedir.

2.1. Farklı Öğretim Kademeleri

Okudukları programlar gereği bilişim alanında meslek sahibi olacak bilgisayar mühendisleri, bilgisayar programcıları, bilgisayar öğretmenleri ve diğer alan öğrencileri üniversitelerin akademik programları gereği programlama derslerini almaktadırlar. Türkiye’de ilk ve orta öğretimde verilen bilgisayar okuryazarlığı derslerinin bazı aşamalarında da programlama yer almaktadır. Meslek ve teknik liselerde ise uzun yıllardır bilgisayar bölümlerinde programcılık eğitimi verilmektedir. Dünyada ise ilköğretim eğitim programında 10. ve 11. sınıflardaki bilgisayar derslerinde programlama öğrenimi yer almaktadır [1].

2.2. Klasik Öğretim Teknikleri

Çok farklı programlama dilleri olduğu halde, bu dillerin eğitimi üniversitelerde veya liseler-

de çok benzer öğretim teknikleri ile verilmektedir. Bu teknik bir uzman eşliğinde bilgisayar başında bire bir kod yazarak dili öğrenmeye çalışmaktır. Bu “uygulama” ağırlıklı etkinliklere genellikle teorik bilginin düz anlatım veya sunum yöntemi ile verildiği ön derslerle başlanır. Her hafta önce öğrenilecek konu ile ilgili sunum ya da anlatım yapılır, ardından program satır satır yazılıp açıklanarak öğrencilere gösterim yapılır, daha sonra da laboratuvarlarda öğrencilerin bireysel olarak kod yazacağı etkinlikler uygulanır. Dersler 2 ya da 3 saat teorik dersten sonra yine 2 ya da 3 saatlik uygulama saatleri şeklinde 14 - 15 haftalık dönemlerde “gösterip-yaptırma” tekniği kullanılarak verilir. Derslerin değerlendirmeleri yazılı sınavlar, uygulama sınavlar ve proje geliştirme süreçleri ile yapılmaktadır.

Programlama derslerinin hiç biri tek başına öğrencinin iyi bir programcı olabilmesi amacını taşımamaktadır. Bunun sebebi, programlamanın, tek bir derste edinilmesi zor olan bir çok beceriye ihtiyaç duymasıdır. Bir programcının iyi bir program geliştirebilmesi için dilden bağımsız olarak çözüm için bir algoritma geliştirmesi, daha sonra o algoritmaya yönelik kodları yazması, olası hataları düzeltilmesi ve en sonunda ürünü ortaya çıkarması gerekmektedir. Kimi büyük yazılım geliştirme projelerinde bu adımların farklı takımlar tarafından yürütüldüğünü düşünürsek, bu becerilerin ne kadar önemli olduğu anlaşılabilir. Bu nedenle, eğitim programları incelendiğinde programcılık becerilerinin birkaç derste kazandırıldığı görülmektedir. Genellikle ilk derslerde programlama dilinin kendisinden çok algoritma geliştirmeye yönelik içerik sunulmaktadır. Daha sonraki derslerde hedeflenen dil ile problem çözmeye yoğunlaşmaktadır.

2.3. Programlama Sürecinde Başarı Faktörleri

Programlama dili derslerinde başarıyı etkileyen bir çok faktör vardır. Bunlardan bazıları motivasyon , programlamaya karşı tutum, programlama dilinin karmaşıklığı [2] ve öğretim yöntemi ve tasarımı olarak sıralanabilir [3].

Programlama eğitimi sırasında düşük olan motivasyon, öğrencinin zamanla dersten soğumasına ve çaba sarf etmemesine yol açabilir. Programlama sürecinin uzun saatler boyunca bilgisayar önünde çalışmayı gerektiren bir iş olduğunu düşünen bir çok öğrenci, bu süreçleri sıkıcı bulabilir.

Programlamaya karşı olumsuz tutum ve algılar, diğer derslerde de olabileceği gibi, başarıyı engelleyen önemli bir faktördür. Programlama dili eğitiminin kendi kariyerleri için önem taşımadığını düşünen bir çok öğrencide de yine olumsuz tutumlar yer almaktadır. Örneğin matematik öğretmeni olacak bir aday dört yıllık üniversite eğitimi sırasında 3 adet programlama ders almaktadır. Bu dersleri kendi alanlarıyla alakasız olarak gören bir çok öğrenci programlama dili derslerine karşı ilgisiz ve isteksiz olarak devam ediyor olabilir. Bunlara ek olarak bilgisayar ve teknoloji kullanımına yönelik genel tutumlar da programlama dili derslerinde etkilidir.

Eğitim için seçilen programla dili başarıyı etkileyebilir [4]. Programlama dilleri, çoğunlukla, “öğrenilmeleri kolay olsun” kaygısıyla geliştirilmemişlerdir [2]. Bu nedenle özellikle ilk programlama dili derslerinde, seçilen dilin karmaşıklığı öğrenmeyi güçleştirebilir. İlk derslerde asıl amaç dilden bağımsız olarak algoritma mantığının öğretilmesidir. Daha sonra istenilen dile özgü deneyim kazanılması beklenir. Ancak öğrenciler geliştirdikleri algoritmaları denemek için de bir programlama dili kullanmalıdırlar. Diğer bir ifade ile, seçilen bir programlama dilinin yazım kurallarına uygun komutlar yazılır ve çalıştırılır. Eğer dilin yazım kuralları çok katı ise, yazım ortamı öğrenciye anlık destek vermiyorsa, ya da hata ayıklama araçları gelişmemiş ise, öğrenci kodlama aşamasındaki zorluklardan dolayı algoritmaya odaklanamayabilir. Analiz etmekten çok ezberlemenin ve hatırlamaya çalışmanın arasında vaktini harcar, sonuç olarak motivasyonu ve başarısı düşebilir.

Klasik öğretim teknikleri de programlama başarısını etkileyebilir [2]. Programlama başından sonuna farklı bilgi ve becerilerin kullanılması gereken bir süreçtir. Öğrenciler daha önceden var olan akademik alışkanlıklarından derste yeterince yararlanamayabilir. Ezberi çok iyi olan bir öğrenci sadece komutları hatırlayarak problemi çözemez. Okuması kuvvetli bir öğrenci de yeni bir problemde algoritma geliştiremeyebilir. Kuramsal bilginin uygulama ile “problem çözme” becerisi haline gelmesi bir çok öğrencide ulaşılmayan bir hedef olarak kalabilir.

3. Öneri: Robot Programlama

Pek çok programlama dili ve programlama mantığının öğretilmesindeki güçlükler göz önüne alınarak uygulanmış farklı yaklaşımlar vardır. Örneğin, kullanılan dilin kendisinin “zor” olmasına karşı daha basit diller (mini language) [4] önerilmiştir. Bu tür diller yazılım geliştirmekten çok programlama dili eğitimi için geliştirilmiş, basit komutlardan ve farklı etkileşim teknikleri ile donatılmış dillerdir.

Başka bir yaklaşım konunun farklı derslerde farklı dilleri ve öğretim tekniklerini kullanarak öğretilmesi olabilir.

Sonuçta, programlama dili öğretilirken çoğu işlem ve kavram öğrenciler açısından soyut kalmakta ve öğrenciler öğrendikleri bilgileri somutlaştırmakta zorlanmaktadır. Robot programlama bu sürecin somutlaştırılmasını sağlayacak bir yaklaşım olabilir.

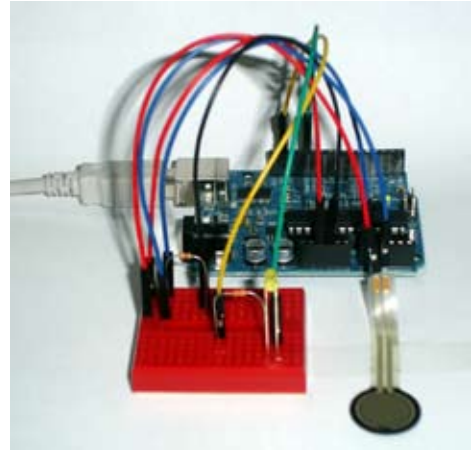
Robot programlama konusunda farklı boyutta robotlardan ve uyumlu dillerden bahsedilebilir. Bu çalışmada ülkemizde temin edilmesi kolay bir platform olan Arduino™ [5] robot ürünleri örnek olarak kullanılmıştır.

Arduino bir giriş/çıkış (input/output) kartı ve Java tabanlı bir dilin yer aldığı geliştirme ortamından oluşan bir fiziksel programlama platformudur. Arduino tek başına çalışan etkileşimli nesnelere geliştirmek için kullanılabilir.

ceği gibi bilgisayar üzerinde çalışan yazılımlara da (ör: Adobe Flash) bağlanabilir. Arduino IDE kod editörü ve derleyici olarak görev yapan, aynı zamanda derlenen programı karta yükleme işlemini de yapabilen, her platformda çalışabilen Java programlama dilinde yazılmış bir uygulamadır. Geliştirme ortamı, sanatçıları programlamayla tanıştırmak için geliştirilmiş Processing™ [6] yazılımından yola çıkılarak geliştirilmiştir [8]. Açık donanım ve açık kaynak kodlu bir mimariye sahip olan Arduino kullanımı için hazır üretilmiş kartlar satın alınabilir, ayrıca Arduino donanım referans tasarımları Creative Commons dağıtılmaktadır ve Arduino web sitesinden indirilebilir. Aşağıda tasarım ve uygulama aşamalarından kısaca bahsedilmiş olan örnek uygulama fiziksel platformların çalışma prensibini özetlemektedir.

3.1. Örnek Bir Uygulama

Arduino robot programlama paket ile oluşturulabilecek uygulamalardan bir tanesi, üzerine basılınca bunu algılayıp alarm veren küçük bir robot ve programlaması. Resim 1’de görüldüğü üzere basit bağlantılar ile “basınç algılayıcısı” hazır hale geliyor.



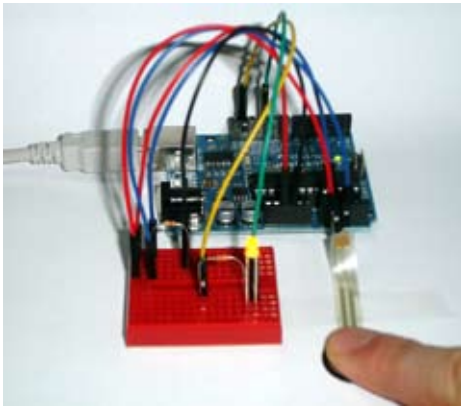
Resim 1: Hazır Robot

Bilgisayara bağlı ucu ile Resim 2’deki program yazılıp robota yükleniyor.



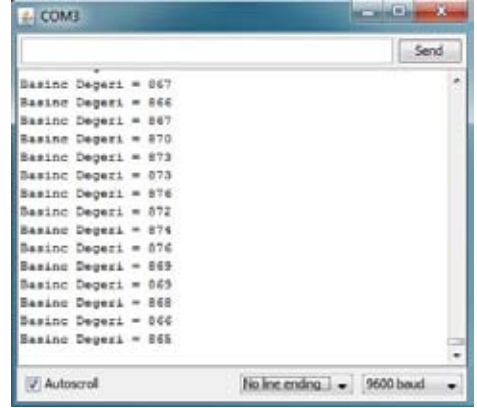
Resim 2: Arduino Kod Editörü

Daha sonra tek bir düğme ile robota yükleniyor. Yükleme tamamlandı robot çalışmaya başlıyor. Eğer basınç algılayıcısının üzerine bir cisim ile dokunulursa (Resim 3’de parmak basılıyor), robotun üzerindeki bir devreye akım verip (örnekte sarı led yanıyor) olası başka işlemleri yapabilir.



Resim 3: Robot Basıncı Algılayıcı Halde

Aynı zamanda, bilgisayar ekranında basınç algılayıcısının algıladığı basınçta rakamsal olarak sürekli izlenebilir (Resim 4).



Resim 4: Anlık Basınç Değerleri

Bu değerler, kod editöründe değişkenler olarak edinilebilir. Bu sayede programın akışına göre farklı çıktılar ek edilebilir. Örneğin bir “if” karar yapısı ile, robotun basıncın 100’ün üzerinde olduğunda tepki vermesi sağlanabilir. Yine farklı değerler için farklı ledler yakılabilir.

4. Sonuç

Sonuç olarak, programlamayı öğrenmek farklı bir düşünce yapısı geliştirmeyi gerektirir ve bu sürecin kolaylaştırılması ancak öğretilen kavram ve süreçlerin somutlaştırılması ile olanaklıdır. Programlama dilleri öğretiminde sürekli olarak soyut kavramlar ile çalışılmasının neden olduğu olumsuzlukların giderilmesinde Arduino ve benzeri platformlar önemli bir çözüm sunmaktadır. Önerilen yaklaşımın, öğrencilerin yazdıkları kodun çalışır halini fiziksel olarak gözlemleyebilmeleri programlama ile ilgili kavramların somutlaşmasına yardımcı olacağı düşünülmektedir.

5. Kaynaklar

[1] Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. ACM K-12 Task Force. **A Model Curriculum for K-12 Computer Science. Final Report of the ACM K-12 Task Force Curriculum Committee.** Erişim Tarihi 29 Aralık 2010. <http://www.csta.acm.org/Curriculum/sub/CurrFiles/K-12ModelCurr2ndEd.pdf>, (2003)

[2] Jenkins, T. “On the difficulty of learning to program”. **In 3rd annual Conference of LTSN-ICS**, Loughbrorrough University, Leicestershire, UK (2002)

[3] Mayer, R. E. “Some conditions of meaningful learning for computer programming: Advance organizers and subject control of frame order”. **Journal of Educational Psychology**, 68 (2), pp. 143-150. (1976)

[4] Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., & Miller, P. “Mini-languages: A Way to Learn Programming Principles”. **Education and Information Technologies**, 2 (1), pp. 65-83. (1997)

[5] Arduino™, Açık kaynak kodlu robotik programlama platformu. <http://www.arduino.cc/> (Son Erişim tarihi:17 Ocak 2011)

[6] Processing™, Açık kaynak kodlu programlama dili ve platformu. [http:// processing.org/](http://processing.org/) (Son erişim tarihi 17 Ocak 2011)

[7] Wikipedia, “Arduino”, <http://tr.wiki pedia.org/wiki/Arduino> (Son erişim tarihi 17 Ocak 2011)