



<http://kefad.ahievran.edu.tr>

Ahi Evran Üniversitesi Kırşehir Eğitim Fakültesi Dergisi

ISSN: 2147 - 1037

Bilişim Öğretmeni Adaylarının Programlama İlişkin Öz- Yeterliklerinin Yordayıcısı Olarak Bilişimsel Düşünme ve Problem Çözmeye İlişkin Yansıtıcı Düşünme Becerileri

Serdar Çiftci,
Meltem Çengel,
Muhammed Paf

DOI:.....

[Makale Bilgileri](#)

Yükleme:25/08/2017 Düzeltme:01/12/2017 Kabul: 14/03/2018

Özet

Öz-yeterlik, bireyin kendi yeteneklerine ve kapasitesine olan inancıdır. Kişi amaca ulaşabileceğine inanıyorsa bu durum performansına olumlu yansımaktadır. Programlama becerisi, birçok beceri alanıyla ilişkili olup, öz-yeterlik kavramı programlama becerisinin gelişiminde önemli bir yer tutmaktadır. Programlamaya ilişkin öz-yeterliği ise kişinin program yazarken amaca ulaşabileceğine olan inancı anlamı çıkarılabilir. Bu bağlamda araştırmanın amacı, bilgisayar ve öğretim teknolojileri eğitimi bölümü öğretmen adaylarının programlamaya ilişkin öz-yeterlik algılarının, bazı demografik değişkenler, Bilişimsel düşünme ve problem çözmeye ilişkin yansıtıcı düşünme becerileri ile ne ölçüde yordandığını ortaya koymaktır. Çalışmada öğretmen adaylarından 4 bölümden oluşan anket yardımıyla veriler toplanmıştır. Ankette, demografik verileri içeren kişisel bilgiler bölümü, Ramalingam ve Wiedenbeck (1998) tarafından geliştirilen ve Altun ve Mazman (2012) tarafından Türkçe'ye uyarlanan Programlama İlişkin Öz yeterlik Ölçeği, Kızılkaya ve Aşkar (2009) tarafından geliştirilen Problem Çözmeye İlişkin Yansıtıcı Düşünme Ölçeği ve Korkmaz, Çakır ve Özden (2017) tarafından geliştirilen Bilgisayarca Düşünme Ölçeği kullanılmıştır.

Çalışmada kullanılan veri toplama aracı ile Türkiye'de bulunan dört farklı üniversitenin Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümlerinde okuyan öğretmen adaylarından veriler elde edilmiştir. Verilerin analizinde Çoklu Regresyon analizi kullanılmıştır. Analiz sonuçlarına göre bağımsız değişkenler programlamaya ilişkin öz-yeterliği değeri ile anlamlı şekilde yordamaktadır ($F(3, 99) = 24.880, p < .001$). Modelin oluşmasına en önemli katkıyı bilişimsel düşünme, sonrasında problem çözmeye ilişkin yansıtıcı düşünme ve bilgisayar ile ilgili gelişmeleri takip etme katkı sunmaktadır. uyarlanmış R^2 değeri .413 düzeyindedir, ki bu durum model ile programlamaya ilişkin öz-yeterliğin %41'ini açıkladığını göstermektedir.

Anahtar Kelimeler: bilişim öğretmeni adayları, programlama ilişkin öz-yeterlik, bilişimsel düşünme, problem çözmeye ilişkin yansıtıcı düşünme

Sorumlu Yazar : Serdar Çiftci, Yrd.Doç.Dr., Adnan Menderes Üniversitesi, Türkiye, sciftci@gmail.com,

<https://orcid.org/0000-0001-5282-1861>

Meltem Çengel, Yrd.Doç.Dr., Adnan Menderes Üniversitesi, Türkiye, meltemcengel@gmail.com,

<https://orcid.org/0000-0002-0255-4600>

Muhammed Paf, Eğitim Programları ve Öğretim Yüksek Lisans Öğrencisi, muhammedpaf@gmail.com

Bu çalışma XV. European Conference on Social and Behavioral Sciences isimli konferansta sözlü olarak sunulmuştur.

Atf için: Bu bilgi KEFAD tarafından sağlanacaktır.

Giriş

Son dönemde, bireylerin katılımlarını ve performanslarını etkileyen unsurlar arasında öz-yeterlik kavramı gibi bireysel ve içsel kaynaklar önem kazanmaktadır (Chen, 2017). Öz-yeterlik genel olarak bireyin görevi ne ölçüde yapacağına ilişkin inancıdır (Huffman, Whetten, & Huffman, 2013). Bandura'ya (1994) göre öz-yeterlik sadece performans düzeyi ile ilgili "kehanet"lerde bulunan bir kavram değildir. Öz-yeterlik güdülenmeye ilişkin bileşenleri uyarır ve görevin gerçekleştirilmesi için gerekli bilişsel beceriler ve çabayı tetikler (Kher, Downey, & Monk, 2013). Davidsson, Lazron ve Ljunggren (2013) öz-yeterliğe ilişkin inancın; belirlenen görevi başarmak için gerekli etkinlikleri seçme becerisi, çaba harcama, zorluklarla başa çıkma ve performans açısından etkili olduğunu belirtmektedirler.

Özellikle son yıllarda kazanan bilgisayar ve programlamaya ilişkin becerilerin öğrenilmesi önem kazanmaktadır (Bargury et al., 2012; Bers, Flannery, Kazakoff, & Sullivan, 2014; Grout & Houlden, 2014; Jones, Mitchell, & Humphreys, 2013). Bilgisayar ile ilgili becerilerin öğrenilmesinde öz-yeterlik kavramının son derece önemli olduğu bilinmektedir (Karsten, Mitra, & Schmidt, 2012). Ancak bu becerilerin öğrenilmesi sırasında, öz-yeterlik algılarının düşük olması nedeniyle, yani programlamayı baştan zor kabul etmeleri nedeniyle, programlama dersinde başarısız olmaları olası olduğuna değinilmektedir (Askar & Davenport, 2009). Igbaria ve Iivari (1995) sınıfta teknoloji kullanma ile ilgili olumsuz algının nedenlerinden birinin düşük teknoloji öz-yeterliği olabileceğine dikkati çekmektedir.

Ülkemizde programlamaya ilişkin öz-yeterlik ile ilgili çalışmalar incelendiğinde, programlama dersi alan öğrencilerin öz-düzenleyici stratejileri ile başarıları arasındaki ilişkilere odaklanan (Haşlaman & Aşkar, 2007), Bilgisayar ve Öğretim Teknolojileri Eğitimi (BÖTE) ve farklı bölümlere devam eden ve programlama dersi alan öğrencilerin programlamaya ilişkin öz-yeterlik algıları (Gezgin & Adnan, 2016; Mazman & Altun, 2013), programlamaya ilişkin öz-yeterlik ve tutum arasındaki ilişkiler (Özyurt & Özyurt, 2015; Yağcı, 2016) gibi konular ele alınmıştır. Bununla birlikte, Psycharis ve Kallia (2017) bilgisayar programlama becerisinin, lise öğrencilerinin sorgulama becerileri, problem çözme ve matematik öz-yeterlikleri üzerindeki etkisini incelemiştir. Başka bir çalışmada ise; Bilgisayar Bilimleri öğrencilerine verilen problem çözme öğretiminin programlama öz-yeterliği ve başarıları üzerindeki etkileri incelenmiştir (Maddrey, 2011).

Var olan çalışmalara gözden geçirilerek; bu çalışmanın temel amacı, programlamaya ilişkin öz-yeterliğin yordayıcılarını ortaya koymak olarak belirlenmiştir. Bu amaçla, programlamaya ilişkin öz-yeterlik ile ilişkili olduğu düşünülebilen, problem çözmeye ilişkin yansıtıcı düşünme, bilişimsel düşünme ve bazı demografik değişkenler ele alınmıştır.

Kuramsal Çerçeve

Problem Çözmeye İlişkin Yansıtıcı Düşünme

Program yazma; problem çözme, soyutlama, matematiksel mantık, sınama, hata ayıklama ve sorunları çözme gibi karmaşık ve zorlu görevleri içermektedir (Saeli ve diğerleri, 2011). Program yazma süreci, problem çözme sürecine benzer aşamalar içerir. Soloway (1993)'e göre; programlama süresince, bireyler problem çözme/tasarlama/düşünme stratejilerini etkin olarak kullanırlar. Bunun temel nedeni, programlama esnasında, öğrenenlerin öncelikle problemin nasıl çözüleceğini matematiksel olarak bulmaları, sonrasında bunu bilgisayara nasıl aktaracakları ve doğru olarak yazılım diline aktarmayı düşünmelerinin gerekliliğidir (Psycharis & Kallia, 2017). Program yazma süreci, problemin nasıl etkili çözülebileceğine ilişkin derinlemesine okuma ve bilişsel farkındalık becerilerini de gerektirir. Bu nedenle, program yazma, programın niteliğini geliştirmek üzere kararların ve eylemlerin sürekli gözden geçirilmesini ve alternatiflerin üretilmesini içerir (Kalelioğlu, 2015). Govender ve diğerlerine (2014) göre ise program yazma sürecinde gerekli olan bazı beceriler açık olarak öğretilmemektedir. Bu noktada gerekli olan becerilerden biri olarak problem çözmenin öğretiminin, yalnızca sorunlar üzerinde uygulama yapmak değil, aynı zamanda problem çözme sürecinin öğretilmesini içermesi gerektiği belirtilmektedir.

Özellikle problemin çok iyi yapılandırılmadığı örneklerde (ill-structured problems); süreç belirsizleşmekte, pek çok çözüm ve değerlendirme olasılığı bulunmaktadır. Bu tür problemlerin çözümünde sürece ilişkin olarak yansıtıcı düşünme; sürecin kontrolü, izlenmesi ve deneyim kazanılması aşamalarını içermesi nedeniyle önemlidir (Lee, Teo, & Zealand, 2011). Yansıtıcı düşünme, eleştirel düşünme sürecinin önemli bir parçasıdır. Ayrıca analiz etme süreçleri ve neler olduğuna ilişkin karar süreçlerini içermektedir. Öğrencilere bir adım geri atıp, bir sorunu nasıl çözdükleri ile ilgili fikir sağladığı ve hedefledikleri çözüme ulaşabilmek için kullandıkları stratejilerin ne ölçüde işe yaradığını anlamalarını sağladığı için de önemlidir (Kalelioğlu, 2015). Yansıtıcı düşünme, durumdan kuşkulama, tereddüt etme, şaşırma, zihinsel güçlük içirme, sorgulama, şüpheyi giderecek materyal bulma etkinliklerini kapsamaması bakımından düşünme adına uygulanılan işlemlerden ayrılmaktadır. Yansıtma kavramının farklı bağlamlarda farklı tanımlarla kullanılmaktadır. Bu çalışmada yansıtıcı düşünme problem çözme bağlamı ile sınırlandırılmıştır.

Programlama süreçleri, problem çözme ve yansıtıcı düşünme arasındaki ilişkilere odaklanan pek çok çalışma bulunmaktadır (Fessakis, Gouli, & Mavroudi, 2013; Liao & Bright, 1991). Örneğin; Bergin, Reilly ve Traynor (2005), bilişüstü ve yönetsel süreçlere ilişkin olarak daha çok strateji kullanan öğrencilerin programlama süreçlerinde daha başarılı olduğunu belirtmektedirler. Akcaoglu ve Koehler (2014) ise oyun-tasarımı ve öğrenme grubuna katılan öğrencilerin problem çözme

becerilerinin katılmayanlara göre anlamlı şekilde arttığını belirtmektedirler. Ke (2014) ise ele aldığı örnek olayda, bilgisayarda oyun programlamanın; öğrencilerin matematik ile ilişkili kavram ve problem çözmeye ilişkin becerilerine katkı sunduğunu belirtmektedir. Psycharis ve Kallia (2017)'nin çalışması ise programlamanın öğrencilerin sorgulama, problem çözme ve matematiğe ilişkin öz-yeterlikleri üzerinde anlamlı bir etkisi olup olmadığına ilişkindir. Çalışmaya göre, bilgisayarda programlama, öz-yeterlik ile ilgili belirleyiciler üzerinde anlamlı bir etki yaparken, problem çözme üzerinde anlamlı bir etkide bulunmamaktadır.

Bilişimsel Düşünme

Bilişimsel düşünme (Computational Thinking) ile ilgili alan yazında isimlendirme karmaşası söz konusudur. Türkçe alanyazında yapılan çalışmalarda farklı isimlendirmeler bulunmaktadır. Komputasyonel düşünme (Şahiner & Kert, 2016), bilgi-işlemsel düşünme (Demir & Seferoğlu, 2017), bilgisayarca düşünme (Çatlak, Tekdal & Baz, 2015), bilişimsel düşünme (Sayın & Seferoğlu, 2016), hesaplamalı düşünme (MEB, 2017), kompütasyonel düşünme (Aldağ & Tekdal, 2015; Şahiner & Kert, 2016) ve bilgi-işlemsel düşünme (Barut, Tuğtekin & Kuzu, 2016; Kalelioğlu, Gülbahar, Akçay ve Dogan, 2015) farklı kullanılan isimlendirmelerden bazılarıdır. Buna ek olarak; programlama yerine yazılım geliştirme, kodlama, program yazma, bilgisayar programı yazma gibi ifadeler de kullanılmaktadır. Bu çalışmada “bilişimsel düşünme” ve “programlama” ifadeleri temel alınmıştır.

Bilgisayarda program yazma becerileri ile ilişkilendirilen önemli değişkenlerden biri de bilişimsel düşünmedir (Román-González, 2015). İlk olarak Wing (2006) tarafından ortaya atılan bilişimsel düşünme kavramı, problem çözme yollarını, sistemler tasarlamayı ve insan davranışlarını anlamayı içerir. Bilgisayara ilişkin kavramları bilmek, “değişken” gibi programlama sürecinde kullanılan kavramları bilmeye işaret ederken; uygulamaları bilmek programlama sürecinde ortaya çıkan, döngü, tekrar, koşul gibi problem çözme uygulamalarını anlamayı; bakış açılarını anlamak ise bireyin kendisinin ve diğerlerinin teknolojik dünya ile ilişkisini kavramı içerir (Lye & Koh, 2014). Bilişimsel düşünme yalnızca bilgisayar bilimcilerin kullanması gereken bir kavram değildir (Kim, Kim, & Kim, 2013). Hatta Kafai ve Burke (2013), bilişimsel düşünme kavramından bahsedebilmek için teknolojik araçlara gereksinim duyulmadığına değinmektedirler.

Wing (2006) bilişimsel düşünmeyi, eleştirel düşünme ve var olan bilgilerin matematik, fen ve genel olarak FETEMM (Fen bilimleri, teknoloji, mühendislik ve matematik-STEM) alanındaki karmaşık problemlerin çözümüne uygulanması olarak açıklamaktadır. Aho (2012)'ya göre bilişimsel düşünme problemlerin parçalara ayrılması, soyutlama, mantıksal düşünme, algoritma yaratma ve hata ayıklama olarak formüle edilebilir. Son dönemde ise bilişimsel düşünme; farklı hiyerarşik seviyelerde soyutlama, algoritmik düşünme, otomatizasyon, parçalara ayırma, modelleme, kalıplar

çıkarma, tekrarlama, ölçekleme ve sembolik temsil gibi problem çözme stratejilerini ve bilgisayar bilimi yapılarını kapsayan bir kavram olarak tanımlanabilir (Grover & Pea, 2013; Korkmaz, Çakir, & Özden, 2017; Ni & Guzdial, 2012; Psycharis & Kallia, 2017; Werner & Denning, 2009). Bununla birlikte bu kavramı, analitik ve problem çözme becerilerini içermenin yanında, alışkanlıklar, hata ve sorunlar ile karşılaşıldığında takınılan tutum ve bilgisayar bilimlerinde kullanılan yaklaşımları benimseme ile ilişkilendiren çalışmalar da bulunmaktadır (Bers et al., 2014).

Doğrudan bilgisayarda programlamaya ilişkin öz-yeterlik ve bilişimsel düşünme arasındaki ilişkilere odaklanan çalışmalar olmamakla birlikte, Pellas (2014) çalışmasında öğrencilerin derse katılımının, bilgisayara ilişkin öz-yeterlik, üst bilişsel özdüzenleyici stratejiler ve benlik saygısı tarafından anlamlı şekilde yordandığını ve bu bağımsız değişkenler arasında anlamlı düzeyde ilişki olduğunu belirtmektedir. Robotik ile programlamaya ilişkin yürütülen deneysel çalışmada Jaipal-Jamani ve Angeli (2017), öğretmen adaylarının robotik ile öğretime ilişkin öz-yeterlik inançlarının, alan bilgisi düzeylerinin ve bilişimsel düşünme düzeylerinin arttığına işaret etmektedir. Başka bir çalışmada ise (Psycharis & Kallia, 2017), öğrencilere matematik ile ilişkilendirilerek gerçekleştirilen programlama öğretiminin matematiğe ilişkin öz-yeterlikleri üzerinde etkisi olduğuna değinmektedir. Pellas ve Peroutseas (2016) nitel yöntem uyguladıkları çalışmalarında, programla dersin yürütülen çalışmalarında, öğrencilerin programlamaya ilişkin yeterlikleri ve bilişimsel düşünceleri arasında ilişkilere işaret ettiğine değinmektedir.

Yöntem

Katılımcılar

Araştırmanın 166 katılımcısı bulunmaktadır. Bunların 93'ü erkek (%56), 73'ü kadındır (%44). Katılımcılar Adnan Menderes Üniversitesi (n=48, %28.9), Muğla Sıtkı Koçman Üniversitesi (n=14, %8.4), Süleyman Demirel Üniversitesi (n=53, %31.9) ve Konya Selçuk Üniversitesi, Ereğli Eğitim Fakültesi'nden (n=50, %30.1) olmak üzere dört farklı üniversiteye devam etmektedir. Araştırma katılımcılarının %48.2 (n=80) ikinci sınıfa devam ederken, %22.9 (n=38) üçüncü sınıfa, %28.9 (n=48) dördüncü sınıfa devam etmektedir. %24.1 (n=40) Anadolu Lisesi, %65.7 (n=109) Mesleki veya teknik lise, %10.2'si (n=17) ise diğer lise türlerinden mezun olmuştur. 160 katılımcının (%96.4) bilgisayarı var iken, 6 katılımcının (%3.6) bilgisayarı yoktur. 17 öğrenci (%14.8) bilgisayar ile ilgili gelişmeleri takip ederken, 98 öğrenci (n=59.0) bu gelişmeleri takip etmemektedir.

Araçlar

Uygulanan araçta öğrencilerin bazı demografik ve bilgisayar kullanma alışkanlıklarını belirlemeye yönelik 10 maddelik kişisel bilgiler formu, programlamaya ilişkin öz-yeterlik ölçeği, problem çözmeye ilişkin yansıtıcı düşünme ölçeği ve bilişimsel düşünme ölçeği kullanılmıştır.

Üniversite öğrencilerinin C++ programını kullanabilme öz-yeterliklerini belirlemek amacıyla Ramalingam ve Wiedenbeck (1998) tarafından geliştirilen Programlamaya İlişkin Öz-yeterlik Ölçeği'nin Türkçeye uyarlamasını Altun ve Mazman (2012) tarafından gerçekleştirilmiştir. Uyarlama çalışması sonucunda dokuz maddeden oluşan iki faktörlü bir yapıya ulaşılmıştır. "Karmaşık programlama görevleri" ve "basit programlama görevleri" olmak üzere iki boyuttan oluşan ölçeğin iç tutarlılık katsayısı .928 bulunmuş ve bu yapı toplam varyansın %80,814'ünü açıklamıştır. Bu yapıya ilişkin model doğrulayıcı faktör analizi çalışması da gerçekleştirilmiştir.

Kızılkaya ve Aşkar (2009) tarafından geliştirilen Problem Çözmeye İlişkin Yansıtıcı Düşünme ölçeği 14 madde ve üç alt boyuttan oluşmaktadır. Temel alınan üç boyut; sorgulama, değerlendirme ve nedenleme boyutlarıdır. Sorgulama boyutuna örnek olarak "Problemi okuduğumda verilen ve istenenleri belirlemek için kendime sorular sorarım" maddesi verilebilir. Değerlendirme boyutuna örnek olarak "Çözüm yollarımı tekrar tekrar değerlendirip bir sonraki problemi daha iyi çözmeye çalışırım" maddesi, nedenleme boyutuna örnek olarak "Problem çözerken, yaptığım işlemlerin nedenini düşünerek, bulduğum sonuçla ilişkisini kurmaya çalışırım" gösterilebilir. Ölçek maddeleri 5'li likert tipine göre puanlanmıştır. Maddelerin içerdiği eylem sıklıkları "Her zaman", "Çoğu zaman", "Bazen", "Nadiren", "Hiçbir zaman" seviyelerinde düzenlenmiştir. Bu seviyeler; Her zaman=5, Çoğu zaman=4, Bazen=3, Nadiren=2, Hiçbir zaman=1 olarak puanlanmıştır. Ölçeğe ilişkin doğrulayıcı faktör analizi sonuçları GFI= 0,92, AGFI= 0,89, NNFI= 0,93, CFI= 0,95, RMSR= 0,08, RMSEA= 0.071 olarak hesaplanmıştır.

Korkmaz, Çakır ve Özden (2017) tarafından geliştirilen ve üniversite öğrencileri üzerinde uygulanan ölçek toplam 22 maddeden ve beş faktörden oluşmaktadır. Ölçek beşli Likert tipli bir ölçek olup, beş faktör altında toplanabilen 22 maddeden oluşmaktadır. Ölçeğin faktör yapılarının doğrulanmasına yönelik doğrulayıcı faktör analizi yapılmıştır. Ölçek yaratıcılık (4 madde), algoritmik düşünme (4 madde), işbirliklilik (4 madde), eleştirel düşünme (4 madde), problem çözme (6 madde) boyutlarından oluşmaktadır. Ölçeğin toplam cronbach alpha güvenirlik puanı, .81 düzeyindedir. Ölçek toplam varyansın %56.1'ini açıklamaktadır.

Analiz

Yaklaşık 20 dakika süren uygulamalar sonrasında veriler SPSS ortamına aktarılmış ve verilerin analizi için çoklu doğrusal regresyon analizi yapılmıştır. Çoklu regresyon analizinin gerçekleştirilebilmesi için karşılanması gereken varsayımlar, yordayan değişkenlerin ve bağımlı değişkenin doğrusal olması, hataların ve artıkların normal olarak dağılması ve yordayan ile ilişkisiz olmasıdır. Bununla birlikte, çoğunlukla sorun oluşturan durumlardan biri, çoklu bağlaşımdır. Çoklu bağlaşımda, değişkenler arasında yüksek ilişki olduğu durumlarda gözlenmektedir. Çoklu

bağlaşımın olmamasına, tolerans ve VİF değerleri kontrol edilerek karar verilmiştir (Leech, Barrett, & Morgan, 2005).

Bulgular ve Yorum

Programlamaya ilişkin öz-yeterliğin, problem çözmeye ilişkin yansıtıcı düşünme, bilişimsel düşünme ve bilgisayar ile ilişkili yayınları takip etmeye göre yordanıp yordanamayacağını belirlemek üzere çoklu doğrusal regresyon analizi yapılmıştır.

Öncelikle Tablo 1’de ilgili değişkenlere ilişkin betimsel ve ilişkisel istatistikler sunulmuştur.

Tablo 1. Programlamaya İlişkin Öz-yeterlik ve Bağımsız Değişkenlerin Ortalama, Standart Sapma ve İlişki Düzeyleri

Değişkenler	X	SD	1	2	3
Programlama İlişkin Öz-yeterlik	41.796	12.371	.563*	.554*	-.365*
1. Bilişimsel Düşünme	84.796	11.310	-	.585*	-.233**
2. Problem çözmeye ilişkin yansıtıcı düşünme	54.815	9.295		-	-.293*
3. Bilgisayar ile ilgili gelişmeleri takip etme	.845	.364			-

* p< .001

** p< .009

Tablo 1’e göre, programlamaya ilişkin öz-yeterlik ile bilişimsel düşünme arasında .563 düzeyinde, problem çözmeye ilişkin yansıtıcı düşünme arasında, .554 düzeyinde ve bilgisayar ile ilgili gelişmeleri takip etme arasında -.365 düzeyinde anlamlı ilişkiler bulunmaktadır (p< .001). bilişimsel düşünme ile problem çözmeye ilişkin yansıtıcı düşünme arasında .585 düzeyinde, bilgisayar ile ilgili gelişmeleri takip etme arasında -.233 düzeyinde anlamlı ilişkiler, problem çözmeye ilişkin yansıtıcı düşünme ile bilgisayar ile ilgili gelişmeleri takip etme arasında -.293 düzeyinde anlamlı ilişkiler bulunmaktadır. Tablo 2’de ilgili değişkenleri içeren regresyon modeli özeti sunulmuştur.

Tablo 2. Programlama İlişkin Öz-yeterliğin bilişimsel düşünme, problem çözmeye ilişkin yansıtıcı düşünme ve bilgisayar ile ilgili gelişmeleri takip etme ile yordanmasına ilişkin Çoklu Regresyon Model Özeti

Değişken	B	Std. Hata	β
Bilişimsel Düşünme	.377	8.974	.345
Problem çözmeye ilişkin yansıtıcı düşünme	.391	.103	.294
Bilgisayar ile ilgili gelişmeleri takip etme	-6,756	.127	.199
Sabit	-3,856	2.706	

Analiz sonuçlarına göre bağımsız değişkenler programlamaya ilişkin öz-yeterliği değeri ile anlamlı şekilde yordamaktadır (F (3, 99)= 24.880, $p < .001$). Üç bağımsız değişken de modelin oluşmasına anlamlı katkı sunmaktadır. Tablo 2’de sunulan beta değerlerine göre, modelin oluşmasına en önemli katkısı bilişimsel düşünme, sonrasında problem çözmeye ilişkin yansıtıcı düşünme ve bilgisayar ile ilgili gelişmeleri takip etme katkı sunmaktadır. Analiz sonuçlarına göre uyarlanmış R^2 değeri .413 düzeyindedir, ki bu durum model ile programlamaya ilişkin öz-yeterliğin %41’ini açıkladığını göstermektedir.

Tartışma ve Öneriler

Öz-yeterlik genel olarak bireyin bir beceriyi ne ölçüde yapacağına ilişkin inancı (Huffman, Whetten, & Huffman, 2013), programlamaya ilişkin öz yeterlik ise bireyin program yazarken amaca ulaşabileceğine olan inancı olarak etmektedir. Askar ve Davenport (2009) programlama becerilerin öğrenilme sürecinde, öz-yeterlik algılarının düşük olmasının programlama dersinde başarısız olma durumlarını tetikleyebildiğini ifade etmektedirler. Araştırma kapsamında yapılan analiz sonuçlarına göre; programlamaya ilişkin öz-yeterliğin bilişimsel düşünme, problem çözmeye ilişkin yansıtıcı düşünme ve bilgisayar ile ilgili gelişmeleri takip etme tarafından anlamlı şekilde yordandığı göstermektedir. Jaipal-Jamani ve Angeli (2017) ilkökul öğretmen adaylarının robotik öz-yeterliği ve bilişimsel düşüncelerine ilişkin yürüttükleri deneysel çalışmada, öğrencilerin robotiğe ilişkin öz-yeterlikleri ve bilişimsel düşünme becerilerinin deneysel süreçten anlamlı düzeyde etkilendiğini belirlemişlerdir.

Bilişimsel düşünme ve problem çözmeye becerisi ilişkili olmakla birlikte, bilişimsel düşünme ile birlikte gelişen problem çözmeye becerisi matematik gibi diğer alanlara aktarılabilir. Wing (2006) tanımında BD’yi problem çözmeye bir yolu olarak ifade etmektedir. Problem çözmeye ilişkin beceriler ile programlamaya ilişkin öz-yeterlik arasında sıkı bağlar olduğu Maddrey (2011)’in çalışmasında da ortaya konulmuştur. Bu çalışmada matematik içermeyen problem çözümü

öğretiminin öğrencilerin programlamaya ilişkin öz-yeterliğini arttığını gözlenmiştir. Programlama öğretiminin sorgulama becerileri, problem çözme ve matematik öz-yeterliği arasındaki ilişkileri araştıran başka bir çalışmada ise programlama öğretiminin sorgulama becerilerini ve matematik öz-yeterliğini anlamlı düzeyde etkilediği, ancak problem çözme becerilerine anlamlı düzeyde katkı sunduğu belirtilmektedir (Psycharis & Kallia, 2017). Bununla birlikte, Kalelioğlu (2015) ilköğretim öğrencilerine programlama öğretimini içeren deneysel çalışmasında, programlama becerisi öğretiminin problem çözmeye ilişkin yansıtıcı düşünme üzerinden anlamlı bir etkisi olmadığını ifade etmiştir.

Araştırma sonuçlarında dikkat çeken bir diğer bulgu ise programlamaya ilişkin öz-yeterlik ile bilgisayar ile ilgili gelişmeleri takip etme arasında ters yönlü bir ilişki vardır. Bu sonuca göre programlamaya ilişkin öz-yeterlik becerileri yüksek olan bireylerin alandaki gelişmeleri daha az takip ettikleri söylenilebilir. Bu durum kendini-değerlendirme türündeki ölçeklerde orta çıkabilecek yanlılık kavramı ile ilişkili olabilir. Örneğin, bilgisayar öz yeterliliği kavramı ile ilgili bazı çalışmalar, özyeterliliğin önemli olduğu kadar, kendini-değerlendirmeye dayalı bu kavramın yansızlıktan uzak olmasının da önemli olduğunu ifade etmektedirler. Yansızlık kendi performansını olduğundan fazla ya da az algılamama ile ilişkilidir (Pajares & Graham, 1999). Genel olarak, kendi performansını ve özyeterliliğini olduğundan biraz daha iyi algılamak sonrasındaki çabayı ve bu çabanın sürekliliğini arttırabilir (Bandura, 1986; Pajares, 1997). Ancak kendi-performansını olanın çok üzerinde algılayan bireyler çalışma yöntemini değiştirmek veya kendini daha çok geliştirme ile ilgili çaba harcamayabilir (Zimmerman, Bonner, and Kovach, 1996). Konu ile daha çok ilgili öğrenciler, kendi performanslarını daha doğru değerlendirebilmektedirler (Zell ve Krizan, 2014). Benzer şekilde, Ehrlinger ve diğerleri (2008) de belirli bir alanda, herhangi bir konu ile ilgili daha deneyimli kişilerin kendi performanslarına ve bilişsel kapasitelerine ilişkin daha çok geri bildirim aldıklarına işaret etmektedirler.

Öğrencilerin bilgisayara ilişkin özyeterlik bildirimlerindeki doğruluk ve yanlılığı inceleyen çalışmada, daha deneyimli öğrencilerin özyeterliklerini daha düşük değerlendirdikleri gözlenmiştir. Bununla birlikte, bilişsel yeteneği daha yüksek öğrenciler kendini-değerlendirme ölçeklerinde performansları ile daha tutarlı ve doğru bildirimlerde bulunmuşlardır (Aesaert, Voogt, Kuiper, van Braak, 2017). Buradan yola çıkarak gelişmeleri daha çok takip eden öğrencilerin kendi yeterliliklerini daha düşük buldukları düşünülebilir.

Araştırma bulgularından yola çıkarak, farklı kademelerde uygulamaya konan programlama derslerinin bir parçası olarak problem çözmeye ilişkin yansıtıcı değerlendirme ve bilişimsel düşünme becerilerinin eğitim programlarına dahil edilmesi önerilebilir. Bununla birlikte, programlamaya ilişkin özyeterliliği yordayan diğer değişkenlerin belirlenmesine yönelik çalışmalar yürütülebilir. Diğer bir

araştırma konusu olarak, programlamaya ilişkin gerçek performans, programlamaya ilişkin öz-yeterlik ve bu iki kavram arasındaki ilişkileri etkileyen diğer değişkenleri inceleyen çalışmalara gereksinim duyulduğu söylenebilir.

Kaynakça

- Aho, A. V. (2012). Computation and computational thinking. *Computer Journal*, 55(7), 833–835. <http://doi.org/10.1093/comjnl/bxs074>
- Akcaoglu, M., & Koehler, M. J. (2014). Cognitive outcomes from the Game-Design and Learning (GDL) after-school program. *Computers and Education*, 75, 72–81. <http://doi.org/10.1016/j.compedu.2014.02.003>
- Aldağ, H., & Tekdal, M. (2015). Bilgisayar kullanımı ve programlama öğretiminde cinsiyet farklılıkları. *Proceeding of 1. Uluslararası Çukurova Kadın Çalışmaları Kongresi*, 236–243.
- Altun, A., & Mazman, S. G. (2012). Programlamaya ilişkin öz yeterlilik algısı ölçeğinin Türkçe formunun geçerlilik ve güvenirlik çalışması. *Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi*, 3(2), 297–308.
- Askar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for Java programming among engineering students. *The Turkish Online Journal of Educational Technology*, 8(1), 1303–6521. Retrieved from <http://files.eric.ed.gov/fulltext/ED503900.pdf>
- Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Englewood Cliffs, NJ: Prentice Hall.
- Bandura, A. (1994). Self-efficacy. In V. S. Ramachaudran (Ed.), *Encyclopedia of human behavior* (pp. 71–81). New York: Academic.
- Bargury, I. Zur, Haberman, B., Cohen, A., Muller, O., Zohar, D., Levy, D., & Hotoveli, R. (2012). Implementing a new Computer Science Curriculum for middle school in Israel. *2012 Frontiers in Education Conference Proceedings*, 1–6. <http://doi.org/10.1109/FIE.2012.6462365>
- Barut, E., Tuğtekin, U., & Kuzu, A. (2016). Robot uygulamalar ile bilgi işlemsel düşünme becerilerine bakış. İçinde *3rd International Conference on New Trend in Education (ICNTE 2016)*.
- Bergin, S., Reilly, R., & Traynor, D. (2005). Examining the role of self-regulated learning on introductory programming performance. *First International Workshop on Computing Education Research*, 81–86. <http://doi.org/10.1145/1089786.1089794>
- Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers and Education*, 72, 145–157. <http://doi.org/10.1016/j.compedu.2013.10.020>
- Chen, I. S. (2017). Computer self-efficacy, learning performance, and the mediating role of learning engagement. *Computers in Human Behavior*, 72, 362–370. <http://doi.org/10.1016/j.chb.2017.02.059>

- Çatlak, Ş., Tekdal, M., & Baz, F. Ç. (2015). Scratch Yazılımı İle Programlama Öğretiminin Durumu: Bir Doküman İnceleme Çalışması. *Journal of Instructional Technologies & Teacher Education*, 4(3). Tarihinde adresinden erişildi <http://www.jitte.org/article/view/5000163313>
- Davidsson, K., Larzon, L., & Ljunggren, K. (2013). Self-Efficacy in Programming among STS Students. Retrieved from <http://www.it.uu.se/edu/course/homepage/datadidaktik/ht10/reports/Self-Efficacy.pdf>
- Demir, Ö., & Seferoglu, S. S. (2017). Yeni Kavramlar, Farklı Kullanımlar: Bilgi-İşlemsel Düşünmeyle İlgili Bir Değerlendirme (ss. 801–830).
- Ehrlinger, J., Johnson, K., Banner, M., Dunning, D., & Kruger, J. (2008). Why the unskilled are unaware: Further explorations of (absent) self-insight among the incompetent. *Organizational Behavior and Human Decision Processes*, 105,98-121.
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. *Computers and Education*, 63, 87–97. <http://doi.org/10.1016/j.compedu.2012.11.016>
- Gezgin, D. M., & Adnan, M. (2016). Makine Mühendisliği ve Ekonometri Öğrencilerinin Programlamaya İlişkin Öz Yeterlik Algılarının İncelenmesi. *Ahi Evran Üniversitesi Kırşehir Eğitim Fakültesi Dergisi*, 17(2), 509–525.
- Govender, I., Govender, D., Havenga, M., Mentz, E., Breed, B., Dignum, F., & Dignum, V. (2014). Increasing self-efficacy in learning to program: exploring the benefits of explicit instruction for problem solving. *TD The Journal for Transdisciplinary Research in Southern Africa*, 10(1), 187–200.
- Grout, V., & Houlden, N. (2014). Taking Computer Science and Programming into Schools: The Glyndŵr/BCS Turing Project. *Procedia - Social and Behavioral Sciences*, 141, 680–685. <http://doi.org/10.1016/j.sbspro.2014.05.119>
- Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43. <http://doi.org/10.3102/0013189X12463051>
- Haşlamam, T., & Aşkar, P. (2007). Programlama Dersi ile İlgili Özdüzenleyici Öğrenme Stratejileri ve Başarı Arasındaki İlişkinin İncelenmesi. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 32, 110–122.
- Huffman, A. H., Whetten, J., & Huffman, W. H. (2013). Using technology in higher education: The influence of gender roles on technology self-efficacy. *Computers in Human Behavior*, 29(4), 1779–1786. <http://doi.org/10.1016/j.chb.2013.02.012>
- Igbaria, M., & Iivari, J. (1995). The effects of self-efficacy on computer usage. *Omega*, 23(6), 587–605.
- Jaipal-Jamani, K., & Angeli, C. (2017). Effect of Robotics on Elementary Preservice Teachers' Self-Efficacy, Science Learning, and Computational Thinking. *Journal of Science Education and Technology*, 26(2), 175–192. <http://doi.org/10.1007/s10956-016-9663-z>

- Jones, S. P., Mitchell, B., & Humphreys, S. (2013). Computing at school in the UK : from guerrilla to gorilla. *Communications of the ACM*, (April), 1–13. <http://doi.org/10.1016/j.compedu.2013.10.020>
- Kafai, Y. B., & Burke, Q. (2013). Computer Programming Goes Back to School. *Phi Delta Kappan*, 95(1), 61–65. <http://doi.org/10.1177/003172171309500111>
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, 200–210. <http://doi.org/10.1016/j.chb.2015.05.047>
- Kalelioğlu F, Gülbahar Y, Akçay S, Dogan D. (2014). Curriculum Integration Ideas for Improving the Computational Thinking Skills of Learners through Programming via Scratch.. *7 th International Conference on Informatics in Schools: Situation, Evolution and Perspectives: İstanbul; 22/09/2014 - 25/09/2014*
- Karsten, R., Mitra, A., & Schmidt, D. (2012). Computer self-efficacy : A meta-analysis. *Journal of Organizational Ans End User Computing*, 24(4), 54–80.
- Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers and Education*, 73, 26–39. <http://doi.org/10.1016/j.compedu.2013.12.010>
- Kher, H. V., Downey, J. P., & Monk, E. (2013). A longitudinal examination of computer self-efficacy change trajectories during training. *Computers in Human Behavior*, 29(4), 1816–1824. <http://doi.org/10.1016/j.chb.2013.02.022>
- Kim, B., Kim, T., & Kim, J. (2013). Paper-and-Pencil Programming Strategy toward Computational Thinking for Non-Majors: Design Your Solution. *Journal of Educational Computing Research*, 49(4), 437–459. <http://doi.org/10.2190/EC.49.4.b>
- Kizilkaya, G., & Aşkar, P. (2009). Problem çözmeye yönelik yansitici düşünme becerisi ölçeğinin Geliştirilmesi. *Eğitim ve Bilim*, 34(154), 82–92.
- Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558–569. <http://doi.org/10.1016/j.chb.2017.01.005>
- Lee, C., Teo, T., & Zealand, N. (2011). Shifting Pre-service Teachers' Metacognition Through Problem Solving, 3, 570–578.
- Leech, N. L., Barrett, K. C., & Morgan, G. A. (2005). *SPSS for Intermediate Statistics: Use and Interpretation* (2nd ed.). New Jersey: Lawrence Erlbaum Associates.
- Liao, Y.-K. C., & Bright, G. W. (1991). Effects of Computer Programming on Cognitive Outcomes: A Meta-Analysis. *Journal of Educational Computing Research*, 7(3), 251–268. <http://doi.org/10.2190/E53G-HH8K-AJRR-K69M>
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through

- programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61.
<http://doi.org/10.1016/j.chb.2014.09.012>
- Maddrey, E. (2011). *The Effect of Problem-Solving Instruction on the Programming Self- efficacy and Achievement of Introductory Computer Science Students*. Information Sciences. Nova Southeastern University.
- Mazman, S. G., & Altun, A. (2013). The effect of introductory to programming course on programming self efficacy of CEIT students. *Journal of Instructional Technologies & Teacher Education*, 2(3), 24–29.
- Ni, L., & Guzdial, M. (2012). Who AM I?: Understanding high school computer science teachers' professional identity. *SIGCSE '12: Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 499–504. <http://doi.org/10.1145/2157136.2157283>
- Özyurt, Ö., & Özyurt, H. (2015). A study for determinin computer programming students' attitudes towards progmming and their programming self-efficacy. *Eğitimde Kuram ve Uygulama Articles /Makaleler Journal of Theory and Practice in Education*, 11(1), 51–67.
- Pajares, F., & Graham, L. (1999). Self-efficacy, motivation constructs, and mathematics performance of entering middle school students. *Contemporary Educational Psychology*, 24, 124-139.
- Pellas, N. (2014). The influence of computer self-efficacy, metacognitive self-regulation and self-esteem on student engagement in online learning programs: Evidence from the virtual world of Second Life. *Computers in Human Behavior*, 35(2), 157–170.
<http://doi.org/https://doi.org/10.1016/j.chb.2014.02.048>
- Pellas, N., & Peroutseas, E. (2016). Gaming in Second Life via Scratch4SL: Engaging High School Students in Programming Courses. *Journal of Educational Computing Research*, 54(1), 108–143.
<http://doi.org/10.1177/0735633115612785>
- Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science*, 45(5), 583–602. <http://doi.org/10.1007/s11251-017-9421-5>
- Ramalingham, V., & Wiedenbeck, S. (1998). Development and Validation of Scores on a Computer Programming Self-Efficacy Scale and Group Analyses of Novice Programmer Self-Efficacy. *Journal of Educational Computing Research*, 19(4), 367–381.
- Román-González, M. (2015). Computational Thinking Test: Design Guidelines and Content Validation Computational Thinking Test : Design Guidelines and Content Validation. *Proceedings of EDULEARN15 Conference*, (July), 2436–2444. <http://doi.org/10.13140/RG.2.1.4203.4329>
- Saeli, M., Perrenet, J., Jochems, W. M. G., Zwaneveld, B., Nederland, O. U., & Centrum, R. D. M. (2011). Teaching Programming in Secondary School: A Pedagogical Content Knowledge Perspective. *Informatics in Education*, 10(1), 73–88. <http://doi.org/10.1145/2016911.2016943>

- Sayın, Z., & Seferođlu, S. S. (2016). Yeni bir 21. yūzyıl becerisi olarak kodlama eđitimi ve kodlamanın eđitim politikalarına etkisi. *Akademik Biliřim Konferansı*, 3–5.
- Soloway, E. (1993). Should we teach students to program? *Communications of the ACM*, 36(10), 21–24. <http://doi.org/10.1145/163430.164061>
- řahiner, A., & Kert, S. (2016). Komputasyonel Dūřünme Kavramı ile İlgili 2006-2015 Yılları Arasındaki Çalıřmaların İncelenmesi. *European Journal of Science and Technology, Avrupa Bilim ve Teknoloji Dergisi*, 5, 38–43.
- Werner, L., & Denning, J. (2009). Pair Programming in Middle School. *Journal of Research on Technology in Education*, 42(1), 29–49. <http://doi.org/10.1080/15391523.2009.10782540>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33. <http://doi.org/10.1145/1118178.1118215>
- Yađcı, M. (2016). Effect of attitudes of information technologies (IT) preservice teachers and computer programming (CP) students toward programming on their perception regarding their self-sufficiency for programming Biliřim teknolojileri (BT) öđretmen adaylarının. *International Journal of Human Sciences*, 13(1). <http://doi.org/10.14687/ijhs.v13i1.3502>
- Zell, E., & Krizan, Z. (2014). Do people have insight into their Abilities? A metasynthesis. *Perspectives on Psychological Science*, 9, 111-125.
- Zimmerman, B. J., Bonner, S., & Kovach, R. (1996). *Developing Self-regulated learners: Beyond achievement to self-efficacy*. Washington, DC: American Psychological Association.