



Bolu Abant İzzet Baysal Üniversitesi Eğitim Fakültesi Dergisi (BAİBÜefd)

Bolu Abant İzzet Baysal University
Journal of Faculty of Education



2022, 22(2), 529 – 551 . <https://dx.doi.org/10.17240/aibuefd.2022...-654547>

Bilgisayar Okuryazarlığı Eğitimi İçin Bilgisayar Programlama Öz Yeterlik Ölçeği: Türkçe Geçerlik ve Güvenirlilik Çalışması *

Computer Programming Self-Efficacy Scale for Computer Literacy Education: Turkish Validity and Reliability Study

Seyfullah Gökoğlu¹ 

Geliş Tarihi (Received): 3.12.2019

Kabul Tarihi (Accepted): 22.04.2022

Yayın Tarihi (Published): 30.06.2022

Öz: Bu çalışmada Tsai vd. (2019) tarafından geliştirilen Bilgisayar Programlama Öz Yeterlik (BPÖY) Ölçeği Türkçeye uyarlanmıştır. 174 üniversite öğrencisinden elde edilen veriler üzerinde geçerlik ve güvenilirlik analizleri yapılmıştır. Ölçeğin geçerliğini sınamak amacıyla açılımlı faktör analizi (AFA) ve doğrulayıcı faktör analizi (DFA) uygulanmıştır. AFA sonucunda ölçeğin orijinalinden farklı olarak üç faktörlü bir yapıda olduğu belirlenmiştir. Programlama yeterliği, programlama becerisi ve öz düzenleme olarak isimlendirilen faktörler toplam varyansın %68.853'ünü açıklamaktadır. DFA sonucunda ortaya çıkan uyum iyiliği indekslerinin kabul edilebilir düzeyde olduğu görülmüştür. Ölçeğin güvenilirliğini belirlemek amacıyla güvenilirlik katsayısı ve madde toplam korelasyonları hesaplanmıştır. 16 maddelik ölçeğin Cronbach alfa güvenilirlik katsayısı .936 olarak bulunmuştur. Madde toplam korelasyonları ise .430 ile .771 arasında değişmektedir. Öğrencilerin ölçekten aldığı puanlar doğrultusunda programlamaya yönelik öz yeterlik algılarının yüksek düzeyde olduğu belirlenmiştir. Cinsiyet bakımından öz düzenleme, programlama becerileri ve programlama yeterlikleri arasında anlamlı farklılık bulunamamıştır. Öğrencilerin programlama deneyimi arttıkça öz yeterlik algıları da artış göstermektedir. Araştırma kapsamında uyarlanan BPÖY ölçeğinin Türk öğrencilerin bilgisayar programlamaya yönelik öz yeterlik algılarının bilgi-işlemsel düşünme becerileri temele alınarak incelenmesi bakımından fayda sağlayacağı düşünülmektedir.

Anahtar Kelimeler: Bilgisayar programlama, bilgi-işlemsel düşünme, öz yeterlik, bilgisayar programlama öz yeterlik ölçeği, ölçek uyarlama

&

Abstract: In this study, the computer programming self-efficacy scale (CPSES) developed by Tsai et al. (2019) was adapted into Turkish. Validity and reliability analyses of its Turkish version were conducted on data obtained from 174 university students. To test the scale's validity, exploratory factor analysis (EFA) and confirmatory factor analysis (CFA) were both done. According to the results of the EFA, the scale had three factors – programming competence, programming skills, and self-regulation. Each factor differed from those of the original scale and accounted for 68.853% of the total variance. The goodness of fit indices of the CFA was found to be acceptable. In order to determine the reliability of the scale, reliability coefficient and item-total correlations were calculated. The Cronbach's alpha reliability coefficient of 16 items of the scale was found to be .936. The total item correlations ranged between .430 and .771. The student's scores revealed that their self-efficacy perceptions about programming were high. No significant difference was found among the three factors in terms of gender. The more programming experience students have, the better their self-efficacy perceptions improve. It is thought that Turkish version of CPSES will be beneficial for examining the self-efficacy perceptions of Turkish students in regards to computer programming, as based on computational thinking skills.

Keywords: Computer programming, computational thinking, self-efficacy, computer programming self-efficacy scale, scale adaptation

Atıf/Cite as: Gökoğlu, S. (2022). Bilgisayar okuryazarlığı eğitimi için bilgisayar programlama öz yeterlik ölçeği: Türkçe geçerlik ve güvenilirlik çalışması. *Bolu Abant İzzet Baysal Üniversitesi Eğitim Fakültesi Dergisi*, 22(2), 529-551. <https://dx.doi.org/10.17240/aibuefd.2022...-654547>

İntihal-Plagiarizm/Etik-Ethik: Bu makale, en az iki hakem tarafından incelenmiş ve intihal içermediği, araştırma ve yayın etiğine uyulduğu teyit edilmiştir. / This article has been reviewed by at least two referees and it has been confirmed that it is plagiarism-free and complies with research and publication ethics. <https://dergipark.org.tr/tr/pub/aibuefd>

Copyright © Published by Bolu Abant İzzet Baysal University, Since 2015 – Bolu

¹ Sorumlu Yazar: Dr. Seyfullah Gökoğlu, Bartın Üniversitesi, Fen Fakültesi, Bilgisayar Teknolojisi ve Bilişim Sistemleri Bölümü, gokoguluseyfullah@gmail.com, ORCID: <https://orcid.org/0000-0003-0074-7692>

1. GİRİŞ

21. yüzyıl öğrenen yeterliliklerinden biri olarak kabul edilen teknolojik okuryazarlık becerilerinin kazandırılması amacıyla bilgisayar programlama eğitimi giderek önem kazanmaktadır. Bunun bir sonucu olarak birçok ülkedeki çeşitli eğitim kademelerinde programlama ile ilgili eğitimler verilmekte ve çeşitli dersler yürütülmektedir. Bunun yanı sıra her yıl 180’den fazla ülkede milyonlarca öğrenciye ulaşan “Hour of Code” gibi etkinlikler düzenlenerek dünya genelinde programlama eğitiminin önemi vurgulanmaktadır (Çankaya vd., 2017).

Alanyazında bilgisayar programlamanın çeşitli bilişsel becerilerin geliştirilmesine katkıda bulunduğunu ortaya koyan çok sayıda araştırma bulunmakta ve özellikle öğrencilerin üst düzey düşünme ve problem çözme becerilerinin geliştirilmesinde yardımcı bir araç olarak kabul edildiği belirtilmektedir (Özmen & Altun, 2014). Son yıllarda bu beceriler arasında önemi vurgulanan bir diğer beceri de bilgi-işlemsel düşünme (BİD) becerisidir. BİD, programlamaya temel olan kavramlardan yararlanarak bireylerin düşüncelerini ve tasarımlarını sağlayan temel bir beceri olarak tanımlanmaktadır (Wing, 2006). BİD’e yönelik yapılan tanımlamaların temelinde BİD’in çeşitli bilişsel becerilere temel oluşturan önemli bir beceri olduğunun altı çizilmektedir. Örneğin Wing (2008) problem çözme becerisinden farklı olarak BİD’in bir çeşit analitik düşünme olduğunu vurgulamaktadır. International Society for Technology in Education (ISTE) (2015) BİD’i yaratıcılık, algoritmik düşünme, eleştirel düşünme, problem çözme ve iş birliğinin bir birleşimi olarak tanımlamaktadır. Tsai vd. (2019) ise yapılan tanımlamalardan yola çıkarak BİD’in bilgisayar bilimleri alanının yanı sıra çeşitli öğrenme bağlamlarında veya günlük yaşamda karşılaşılan sorunları çözmeye kullanılacak mantıksal düşünme, algoritma seçimi ve sistematik düşünme gibi çeşitli becerileri içerebileceğini söylemektedir. BİD ile ilgili yürütülen araştırmalarda BİD’in hangi alt becerilerden oluştuğuna dair kesin bir sınır çizilememiştir. Bu noktada Demir ve Seferoğlu (2017) BİD’in araştırmacılar tarafından çeşitli alt bileşenlere ayrıldığını belirtmektedir. Bunlardan bazıları; problem çözme, eleştirel düşünme, soyutlama, analitik ve algoritmik düşünme (Wing, 2008), eleştirel düşünme ve problem çözme (Ater-Kranov vd., 2010), koşullu mantık, algoritma oluşturma, hata ayıklama, simülasyon ve dağıtılmış işlem (Berland & Lee, 2011), problem çözme, algoritma inşa etme, hata yakalama, benzetim ve sosyalleşme (Kazimoğlu vd., 2012), yaratıcılık, algoritmik düşünme, eleştirel düşünme, problem çözme ve iş birliği (ISTE, 2015), soyutlama, algoritmik düşünme ve problem çözme (Kalelioğlu vd., 2016), algoritmik düşünme, iş birliği, eleştirel düşünme, problem çözme ve yaratıcılık (Korkmaz vd., 2017) olarak sıralanmaktadır.

Yukarıda yapılan tanımlamalar ve sıralanan alt bileşenler çerçevesinde BİD’in geleceğin dijital dünyasında bireylerin bilgiyi inşa etme ve problem çözme sürecinde yeni bir okuryazarlık becerisi olarak değerlendirileceği söylenebilir. BİD becerilerinin sadece bilgisayar bilimiyle uğraşan günümüz programcıları için değil, gelecekte dünyaya gelecek tüm bireyler için de bir gereklilik olacağı ifade edilmektedir (Tsai vd., 2019). Bu doğrultuda özellikle son yıllarda BİD becerilerinin öğretim programlarına entegre edilmesi amacıyla girişimler yoğunluk kazanmıştır. Eğitim kademelerinin tamamına yönelik olarak gerek bilgisayar biliminin temel kavramlarının gerekse bilgisayar programlamanın öğretilmesiyle ilgili çeşitli derslerin yürütülmesi bu girişimlerin birer göstergesi olarak değerlendirilebilir. Nitekim ISTE (2016) tarafından yayımlanan 21. yüzyıl öğrenen standartları listesinde bilgi-işlemsel düşünür olma özelliğinin yer alması yürütülen girişimlerin önemini ve değerini ortaya koymaktadır.

Alanyazında öğrencilerin BİD becerilerinin geliştirilmesi amacıyla çeşitli yöntemlere odaklanıldığı görülmektedir. Robotik uygulamalar, dijital oyunlar ve elektronik legolar BİD gelişimi için başvuru olan yaklaşımlar arasındadır (Çınar & Tüzün, 2017). Programlama eğitimi de BİD becerilerinin geliştirilmesi amacıyla kullanılan yöntemlerin önde gelenlerindedir. Bilgisayar programlama öğretimi öğrencilerin BİD becerilerini geliştirebilecek bir eğitim süreci olarak değerlendirilmektedir (Atmatzidou & Demetriadis,

2016; Kazimoglu vd., 2012). Bu bağlamda yapılan araştırmalarla programlama eğitiminin öğrencilerin BİD becerilerinin gelişiminde etkili olduğu ortaya konulmuştur (Alsancak Sırakaya, 2019, Brennan & Resnick, 2012, Oluk & Korkmaz, 2016; Oluk vd., 2018).

Programlama eğitimin önemi ve öğrencilere kazandırdığı yeterlilikler bilinmesine karşın öğrencilerin birçoğu, özellikle de programlamaya yeni başlayanlar, bilgisayar programlamayı ileri düzey eğitim gerektiren ve uzman kişilerin yapabileceği zor bir iş olarak görmektedir (Genç & Karakuş, 2011). Gomes ve Mendes (2007) öğrencilerdeki bu algının programlamada genelleme, soyutlama, eleştirel düşünme gibi birçok becerinin bir arada kullanılmasının gerekliliğinden kaynaklandığını belirtmektedir. Pillay ve Jugoo (2005) ise bu duruma öğrencilerin problem çözme yaklaşımları ve çözüm üretebilme becerilerini etkili biçimde öğrenememelerinin sebep olduğunu vurgulamaktadır. Öğrencilerin bilgisayar programlamada zorluk çekmelerinin bir diğer nedeni olarak da düşük motivasyon ve öz yeterlik inancına sahip olmaları gösterilmektedir (Fang, 2012; Garner, 2009; Nilsen & Larsen, 2011).

Programlama eğitimi ile ilişkili olan önemli faktörlerden birisi olarak görülen öz yeterlik (Tsai vd., 2019), bireylerin farklı koşullar altındaki bireysel inanışlarını ve psikolojik durumlarını ve dolayısıyla da sergileyecekleri performansları değerlendirmek amacıyla kullanılacak psikolojik bir kavram olarak nitelendirilmektedir (Bandura, 1997). Bandura, bireyin sahip olduğu öz yeterlik inancının, bir görevi başarmak için harcayacağı çabanın düzeyini, karşılaşılabilecek zorluklarla başa çıkabilmek için göstereceği direnci, kullanacağı stratejileri ve performansını etkileyeceğini ileri sürmektedir. Öz yeterlik algısı yüksek olan bireyler düşük olanlara kıyasla zorluklarla başa çıkmada daha iyi olmakta, hedeflerine ulaşmak için daha fazla gayret göstermekte, daha az karmaşa yaşamakta ve bu nedenle de psikolojik olarak kendilerini daha iyi hissetmektedirler (Davidsson vd., 2010). Alanyazında öz yeterliğin genel akademik başarı ve beceri öğrenimi performansı ile ilişkili olduğu genel olarak kabul görmektedir (Bergey vd., 2015; Girasoli & Hannafin, 2008; Hoffman & Spataru, 2008). Bilgisayar okuryazarlığı bağlamında değerlendirildiğinde ise öğrencilerin bilgisayara ilişkin öz yeterliklerinin bilgisayar temelli öğrenme ortamlarındaki öğrenme performanslarıyla ilişkili olduğu belirtilmektedir (Moos & Azevedo, 2009). Bilgisayar okuryazarlığı günlük yaşamda bilgisayarı kullanma bilgi ve becerisi olarak tanımlanmaktadır (Korkut & Akkoyunlu, 2008). Bilgisayar okuryazarlığı becerilerinin öğrenilmesinde öz yeterlik kavramının önemli olduğu bilinmektedir (Karsten vd., 2012). Araştırmalar bilgisayar okuryazarlığı ile öz yeterlik arasında pozitif bir ilişki olduğunu ortaya koymaktadır (Korkut & Akkoyunlu, 2008; Kurbanoglu & Akkoyunlu, 2002). Bu çerçevede bilgisayar okuryazarlık düzeyi yüksek olan bireylerin, özellikle programlama gibi bilgisayar bilimleri ile ilgili alanlarda daha başarılı olacağı ve yüksek performans gösterecekleri söylenebilir. Bilgisayar okuryazarlığına benzer şekilde programlama öğreniminin önemli bileşenlerinden birisi olarak da değerlendirilen öz yeterliğin (Ramalingam & Wiedenbeck, 1998) öğrencilerin bilgisayar programlamaya dair performanslarını olumlu yönde yordadığı vurgulanmaktadır (Ramalingam vd., 2004).

Bilgisayar okuryazarlığı eğitimi içerisinde bilgisayar programlamanın önemi vurgulanarak yürütülen araştırmalar neticesinde öğrencilerin programlamaya yönelik öz yeterliklerini değerlendirmek amacıyla geliştirilmiş çeşitli ölçeklere (Altun & Kasalak, 2018; Kukul vd., 2017; Ramalingam & Wiedenbeck, 1998) ve bu ölçeklerin farklı programlama dillerine (Askar & Davenport, 2009; Govender & Basak, 2015) ve Türkçeye uyarlama çalışmalarına (Altun & Mazman, 2002; Korkmaz & Altun, 2014) rastlanmaktadır.

Ramalingam ve Wiedenbeck (1998) tarafından geliştirilen ölçek, programlamayı yeni öğrenen öğrencilerin C++ programlama dili hakkındaki öz yeterliklerini değerlendirmek amacıyla geliştirilen ilk ölçeklerdendir. 32 maddeden oluşan ve 7'li Likert tipinde hazırlanan ölçek 421 üniversite öğrencisine uygulanmıştır. Ölçeğin yapı geçerliğini sınamak amacıyla AFA kullanılmıştır. AFA sonucunda ölçeğin 4 faktörlü bir yapı altında açıkladığı toplam varyans %65.5 olarak bulunmuştur. Ortaya çıkan faktörler "bağımsızlık ve kararlılık", "basit programlama görevlerini gerçekleştirme", "karmaşık programlama görevlerini gerçekleştirme", ve "öz düzenleme" olarak isimlendirilmiştir. Ölçeğin güvenilirliğini ortaya koymak amacıyla iç tutarlık katsayısı hesaplanmış ve ölçeğin tamamı için Cronbach alfa değeri .98 olarak bulunmuştur.

Bilgisayar programlamanın son yıllarda dünya genelinde K-12 eğitim müfredatlarında yer almaya başladığını ve programlama öz yeterliğinin ölçülmesi için bir araç geliştirilmesine ihtiyaç duyulduğunu gerekçe gösteren Kukul vd. (2017), ortaokul öğrencilerinin programlama öz yeterlik düzeylerini ölçmek amacıyla bir ölçek geliştirmiştir. 233 ortaokul öğrencisinden elde edilen veriler üzerinde yapılan analizler sonucunda 31 maddeden oluşan tek faktörlü bir ölçek geliştirilmiştir. Ölçeğin yapı geçerliğini belirlemek amacıyla AFA ve DFA uygulanmıştır. AFA sonucunda tek faktörlü yapının açıkladığı toplam varyans %41.15 olarak hesaplanmıştır. AFA sonucunda elde edilen faktör yapısının uygunluğu DFA ile test edilmiştir. Modelin uygunluğuna ilişkin analiz sonuçları üzerinde gerçekleştirilen modifikasyonlar neticesinde χ^2/df değeri 1.47, yaklaşık hataların ortalama karekökü (Root Mean Square Error of Approximation, RMSEA) .045, normlaştırılmış uyum indeksi (Normed Fit Index, NFI) .96, normlaştırılmamış uyum indeksi (Non-normed Fit Index, NNFI) .99, ortalama hataların karekökü (Root Mean Square Residuals, RMR) .061, karşılaştırmalı uyum indeksi (Comparative Fit Index, CFI) .99, fazlalık uyum indeksi (Incremental Fit Index, IFI) .99, iyilik uyum indeksi (Goodness of Fit Index, GFI) .85 ve düzeltilmiş iyilik uyum indeksi (Adjustment Goodness of Fit Index, AGFI) .83 olarak bulunmuştur. Ölçeğin güvenirlilik analizler sonucunda ise Cronbach alfa iç tutarlık katsayısı .95 ve iki yarı metodu sonucu .96 olarak belirlenmiştir.

Altun ve Kasalak (2018), literatürde blok temelli programlamaya ilişkin bir öz yeterlik algısı ölçeği bulunmaması nedeniyle bir ölçek geliştirmiştir. Blok temelli programlama eğitimi almış 329 ortaokul öğrencisinden elde edilen veriler kullanılarak gerçekleştirilen geçerlik ve güvenirlilik analizleri sonucunda ölçeğin toplamda 12 madde ve 2 alt faktörden oluştuğu belirlenmiştir. Basit blok temelli programlama görevleri olarak isimlendirilen birinci faktör toplam varyansın %11.462'sini, karmaşık blok temelli programlama görevleri olarak adlandırılan ikinci faktör ise toplam varyansın %46.763'ünü açıklamaktadır. Toplam varyansın %58.225'ini açıklayan 12 maddenin madde toplam korelasyonları .491 ile .702 arasında değişmektedir. AFA sonucunda ortaya çıkan 2 faktörlü yapı DFA ile sınanmış ve modelin oldukça iyi uyum değerleri gösterdiği bulunmuştur. Ölçeğin iç tutarlık katsayısı .893 olarak hesaplanmıştır.

Askar ve Davenport (2009), Ramalingam ve Wiedenbeck (1998) tarafından C++ programlama diline yönelik olarak geliştirilen öz yeterlik ölçeğini Java programlama diline uyarlamıştır. Ölçekteki madde sayısı değiştirilmeden C++ ile ilgili ifadeler Java programlama dili ile ilgili ifadelerle dönüştürülmüştür. 326 üniversite öğrencisine uygulanan ölçeğin güvenirlilik katsayısı .99 olarak hesaplanmıştır.

Govender ve Basak (2015), Askar ve Davenport (2009) tarafından Java programlama diline uyarlanan ölçeği 20 üniversite öğrencisine uyarlayarak ölçeğin geçerlik ve güvenirliliğini sınamıştır. 32 maddelik ölçeğin yapı geçerliği temel bileşenler analizi kullanılarak incelenmiş ve analiz sonucunda maddelerin döndürülmüş faktör yük değerlerinin .461 ile .912 arasında değişen 6 faktör altında toplandığı belirlenmiştir. Ortaya çıkan faktörler programlama becerisi, Java yapıları, karmaşıksızlık, zaman bilinci, anlaşılması için yeniden kodlama yeteneği ve öz motivasyon olarak isimlendirilmiştir. Ölçeğin güvenirliliği için Cronbach alfa katsayısı .973 olarak hesaplanmıştır.

Altun ve Mazman (2002), Ramalingam ve Wiedenbeck (1998) tarafından geliştirilen ölçeğin Türkçeye uyarlama çalışmasını gerçekleştirmiştir. Programlama dersi almış 152 üniversite öğrencisinden oluşan örneklemden elde edilen veriler üzerinde yapılan geçerlilik ve güvenirlilik çalışmaları sonucunda ölçeğin Türkçe formunun 2 faktör altındaki 9 maddeden oluştuğu bulunmuştur. Ölçekteki maddelerin ayırt edicilik düzeyinin belirlenmesi amacıyla madde-toplam test korelasyonları hesaplanmış ve 9 maddenin madde toplam korelasyonlarının .561 ile .855 arasında değiştiği belirlenmiştir. AFA sonucunda 2 faktörlü modelin toplam varyansın %80.814'ünü açıkladığı hesaplanmıştır. Ortaya çıkan faktörler basit programlama görevlerini gerçekleştirme ve karmaşık programlama görevlerini gerçekleştirme olarak isimlendirilmiştir. 2 faktörlü yapının doğrulanması amacıyla birinci düzey DFA yapılmış ve analiz sonucunda önerilen modifikasyonlar gerçekleştirilerek hesaplanan uyum indeksleri $\chi^2/df=1.55$, RMSEA=.062, S-RMR=.034, GFI=.95, AGFI=.90, CFI=.99, IFI=.99, NNFI=.99 ve NFI=.98 olarak bulunmuştur.

Ölçeğin güvenilirliğine dair Cronbach alfa katsayısı .928 ve McDonald ω (omega) katsayısı .956 olarak belirlenmiştir.

Korkmaz ve Altun (2014) ise Ramalingam ve Wiedenbeck (1998) tarafından geliştirilen ölçeği mühendislik öğrencilerinin C++ programlama diline yönelik öz yeterlik düzeylerini belirlemek amacıyla Türkçeye uyarlamıştır. Farklı üniversitelerde öğrenim gören 378 elektrik-elektronik mühendisliği ve bilgisayar mühendisliği öğrencisine uyarlanan ölçeğin geçerliği AFA ve madde ayırt edicilik gücü ile sınanmıştır. AFA sonucunda Türkçeye uyarlanan ölçeğin, madde toplam korelasyonları .588 ile .779 arasında değişen 28 maddeden oluştuğu ve tek faktör altında açıklanan toplam varyansın %52.591 olduğu belirlenmiştir. Madde ayırt edicilik gücü hesaplaması için ölçekteki her bir madde için elde edilen ham puanlar üzerinden %27'lik üst ve alt gruplar belirlenerek bağımsız t-testi uygulanmıştır. Ölçekteki maddelerin t-testi sonuçlarının -21.370 ile -13.351 arasında değiştiği gözlemlenmiş ve ölçeğin geneli için t-testi sonucu -37.133 olarak hesaplanmıştır ($p < .001$). Ölçeğin iç tutarlık katsayısı .966 olarak bulunmuştur. Analizler sonucunda ölçeğin Türk kültüründe eğitim gören mühendislik öğrencilerinin öz yeterliklerini ölçmek amacıyla kullanılabilir geçerli ve güvenilir bir ölçek olduğu ifade edilmiştir.

Alanyazındaki, bilgisayar programlamaya yönelik öz yeterlik algısını ölçmek amacıyla geliştirilen ölçeklerin tamamının geçerlik ve güvenilirlik bakımından yeterli düzeyde olduğu raporlaştırılmıştır. Ancak ölçekler içerik bakımından ele alındığında bireylerin bilgisayar programlama ile ilgili problem çözmeye dair genel BİD becerileri yerine belirli bir programlama dilinin sözdizimi yapısına odaklandıkları görülmektedir. Bu tür ölçekler geleceğin programlama uzmanlarının bilgisayar programlamaya yönelik öz yeterlik algılarını değerlendirmek için uygun olarak görülebilirken tüm öğrenciler için bir gereklilik olarak görülen ve 21. yüzyıl öğrenen standartları arasında yer alan BİD becerilerinin değerlendirilmesi bakımından yetersiz kalacaklardır. BİD, sadece bilgisayar bilimleri ve programlama alanlarında öğrenim gören öğrenciler için değil, farklı disiplin ve alanlarda öğrenim gören öğrenciler için de sahip olunması ve geliştirilmesi gereken bir beceridir. Bu çerçevede çeşitli eğitim kademelerinde bilgisayar ve programlama ile ilgili bölümlerin dışındaki alanlarda da öğrenim gören öğrencilere temel bilgisayar ve programlama dersleri verilmektedir. BİD becerileri, programlamaya yönelik öz yeterlik ve bilgisayar okuryazarlığı arasındaki ilişki göz önünde bulundurulduğunda alanyazındakilerden farklı olarak diğer disiplinlerdeki öğrencilerin de bilgisayar programlama ile ilgili BİD becerilerinin doğru bir şekilde değerlendirilebilmesi için daha uygun ölçme araçlarına ihtiyaç olduğu düşünülmektedir. Bu çerçevede öğrencilerin genelinde bilgisayar programlama becerilerine ilişkin algılarının okuryazarlık düzeyinde değerlendirilmesi uygun bir çözüm yolu olarak görülebilir. Bu durumu gerekçe gösteren Tsai vd. (2019), öğrencilerin bilgisayar programlamaya yönelik öz yeterliklerini incelemek amacıyla BİD becerilerini esas alan genel bir öz yeterlik ölçeği geliştirmiştir.

1.1. Araştırmanın amacı

Bu araştırmanın amacı Tsai vd. (2019) tarafından geliştirilen bilgisayar okuryazarlığı eğitimi için BPÖY ölçeğinin Türkçeye uyarlanmasıdır. Bu amaç doğrultusunda aşağıdaki sorulara yanıt aranmıştır:

- 1- Öğrencilerin bilgisayar programlamaya yönelik öz yeterlik düzeyleri nasıldır?
- 2- Bilgisayar programlamaya yönelik öz yeterlik algısı cinsiyete göre farklılık göstermekte midir?
- 3- Bilgisayar programlama deneyimi ile programlamaya yönelik öz yeterlik algısı arasında ilişki var mıdır?

Türkçeye ve Türk kültürüne uyarlanan ölçeğin son yıllarda artan bir şekilde önem kazanan programlama öğretimi, robotik uygulamalar gibi alanlarda öğrencilerin programlamaya ilişkin öz yeterliklerinin değerlendirilmesi amacıyla öğretmenler, müfredat tasarımcıları, bilgisayar bilimleri eğitimcileri ve araştırmacılar tarafından kullanılabilir nitelikte ve faydalı olması beklenmektedir.

2. YÖNTEM

2.1. Çalışma grubu

Araştırmanın çalışma grubu 69 kız (%39.7) ve 105 erkek (%60.3) olmak üzere bilgisayar ve programlama ile ilgili bölümlerde öğrenim gören toplam 174 üniversite öğrencisinden oluşmaktadır. Öğrencilerin %47.1'i meslek yüksekokulunda (N=82), %52.9'u ise fakültede (N=92) öğrenim görmektedir. Öğrencilerin %46.6'sının (N=81) 1 yıl ve daha az, %20.7'sinin (N=36) 1-3 yıl arası, %23'ünün (N=40) 3-5 yıl arası, %7.5'inin (N=13) 5-7 yıl arası ve %2.3'ünün (N=4) 7 yıl ve üzeri programlama deneyimi bulunmaktadır. Katılımcılara ilişkin demografik bilgiler Tablo 1'de sunulmuştur.

Tablo 1.

Katılımcılara İlişkin Demografik Bilgiler

		N	%
Cinsiyet	Kız	69	39.7
	Erkek	105	60.3
Öğrenim Düzeyi	Bilgisayar Programcılığı	82	47.1
	Bilgisayar ve Öğretim Teknolojileri Eğitimi	40	23.0
	Bilgisayar Mühendisliği	52	29.9
Programlama Deneyimi	1 yıl ve daha az	81	46.6
	1-3 yıl arası	36	20.7
	3-5 yıl arası	40	23
	5-7 yıl arası	13	7.5
	7 yıl ve üzeri	4	2.3

2.2. Bilgisayar programlama öz yeterlik (BPÖY) ölçeği

BPÖY ölçeği, Tsai vd. (2019) tarafından lise ve üniversite düzeyinde öğrenim gören öğrencilerin programlama öğrenimine yönelik algılarını ortaya koymak amacıyla geliştirilmiştir. Ölçek maddeleri Berland ve Lee (2011) tarafından ortaya konulan BİD yaklaşımı çerçevesinde bireylerin programlama becerilerinin okuryazarlık düzeyinde değerlendirilmesi amacıyla oluşturulmuştur. Berland ve Lee (2011), BİD'i bilgisayar programlama sürecinde gerekli olan anlama, tasarlama, yazma, düzenleme ve yeniden kullanma becerilerine odaklanan koşullu mantık, algoritma oluşturma, hata ayıklama, simülasyon ve dağınık hesaplama olmak üzere beş alt bileşen çerçevesinde ele almıştır. Bu bileşenlerle ilgili olarak Tsai vd. (2019) tarafından hazırlanan 24 maddelik taslak form 106 öğrenciye uygulanmıştır. Ölçeğin yapı geçerliği AFA ile sınanmıştır. AFA öncesinde verilerin analiz için uygunluğunu belirlemek amacıyla yapılan Kaiser-Meyer-Olkin (KMO) testi sonucu .91 ve Bartlett Sphere testi anlamlı olarak bulunmuştur ($p=.000$). AFA sonrasında ölçeğin mantıksal düşünme, algoritma, hata ayıklama, kontrol ve iş birliği olarak isimlendirilen 5 alt faktör ve 16 maddeden oluştuğu belirlenmiştir. Ölçeğin güvenirliliğini belirlemek amacıyla Cronbach alfa (α) katsayısı hesaplanmıştır. Orijinal ölçeğe ilişkin analizler sonucunda elde edilen bulgular Tablo 2'de sunulmuştur.

Tablo 2.

Orijinal Ölçeğe İlişkin Geçerlik ve Güvenirlik Analizi Sonuçları

Faktörler	Açıklanan Varyans (%)	α	Madde Sayısı
Mantıklı Düşünme	23.71	.96	4
İş Birliği	19.01	.95	3
Algoritma	16.32	.92	3
Kontrol	14.66	.84	3
Hata Ayıklama	10.16	.94	3
Toplam	83.87	.96	16

Ölçeğin mantıklı düşünme faktörü altında yer alan maddeler ile öğrencilerin mantıksal koşullar kullanarak bir program yazabilme becerileri ile ilgili algıları belirlenmeye çalışılmıştır. İş birliği faktörü altında öğrencilerin programlama işinin iş birlikli yapısı hakkındaki algıları değerlendirilmiştir. Algoritma faktörü altında öğrencilerin programlama yaparken bağımsız olarak bir problem çözmek için bir algoritma oluşturabilme becerileri ölçülmüştür. Kontrol faktörü altında öğrencilerin bir program editörünü kontrol edebilme becerileri incelenmiştir. Hata ayıklama faktörü altında ise öğrencilerin program hatalarını düzeltebilme becerilerini değerlendiren maddeler yer almaktadır.

2.3. Ölçeğin Türkçe formunun oluşturulması

Ölçeğin Türkçeye uyarlama çalışması için ilgili araştırmacılardan gerekli izinler alındıktan sonra orijinal ölçekte yer alan 16 madde, iki İngilizce alan uzmanı tarafından Türkçeye çevrilmiştir. Yapılan çeviriler iki bilgisayar ve öğretim teknolojileri eğitimi (BÖTE) alan uzmanı tarafından incelenerek kavramların ve ifadelerin anlam ve dil yapısı bakımından uygunluğu kontrol edilmiş ve gerekli düzeltmeler yapılmıştır. Oluşturulan Türkçe form, dilsel eşdeğerlik ve anlam uyumluluğunun sağlanabilmesi amacıyla iki farklı İngilizce alan uzmanı tarafından tekrar İngilizceye çevrilmiştir. Yapılan çeviriler öğretim teknolojileri ve programlama alanında uzman olan farklı 2 öğretim üyesinin görüşüne sunulularak Türkçe ve İngilizce çeviriler arasındaki tutarlık incelenmiş, cümle yapılarının ve anlamlarının bozulup bozulmadığı kontrol edilmiştir. Yapılan değerlendirmeler sonucunda oluşturulan Türkçe form uygulama öncesinde üniversitelerin bilgisayar programcılığı bölümünde görev yapan 3 farklı öğretim görevlisine inceletilerek ifadelerin öğrenciler tarafından anlaşılabilirliği açısından onay alınmıştır. İlgili kurumlardan gerekli izinler alındıktan sonra ölçek, üniversitelerin bilgisayar mühendisliği, bilgisayar ve öğretim teknolojileri eğitimi ve bilgisayar programcılığı bölümlerinde öğrenim gören öğrencilere çevrimiçi olarak uygulanmıştır.

2.4. Verilerin analizi

Türkçeye uyarlanan ölçek formunun geçerliği faktör analizi kullanılarak incelenmiştir. Ölçeğin orijinal boyutları ile Türkçeye uyarlanan formunun boyutları arasındaki uyuma bakmak ve yapı geçerliğini sınamak amacıyla AFA uygulanmıştır. AFA sırasında veri setinden azami varyansı elde etmek ve fazla sayıdaki değişkeni azaltarak daha küçük sayıda bileşen altında toplamak amacıyla faktörleştirme tekniği olarak temel bileşenler analizi kullanılmıştır (Tabachnick & Fidell, 2007). AFA öncesinde verilerin ve örneklemin temel bileşenler analizine uygunluğu KMO katsayısı ve Barlett testi uygulanarak incelenmiştir. AFA sonucunda, ölçekteki her bir maddenin hangi faktör altında yer alacağı, maddelerin faktörler ile ilişkisini gösteren yük değerlerine bakılarak belirlenmiştir. Gerçekleştirilen analizde madde faktör yük değerlerinin .30 ve daha yüksek olması esas alınmıştır (Büyüköztürk, 2007; Field, 2000; Kline, 2011). AFA sonrasında ortaya çıkan yapının doğrulanması amacıyla DFA uygulanmıştır. DFA sırasında uyum indeksleri olarak χ^2/df , RMSEA, SRMR, CFI, GFI, AGFI ve NFI kullanılmıştır. Ölçeğin güvenilirliğini belirlemek amacıyla Cronbach alfa güvenilirlik katsayısı ve ölçekte yer alan her bir maddenin ölçtüğü özellik açısından ayırt edicilik düzeyinin belirlenmesi amacıyla madde toplam korelasyonları hesaplanmıştır.

2.5. Araştırmanın etik izni

Yapılan bu çalışmada “Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesi” kapsamında uyulması belirtilen tüm kurallara uyulmuştur. Yönergenin ikinci bölümü olan “Bilimsel Araştırma ve Yayın Etiğine Aykırı Eylemler” başlığı altında belirtilen eylemlerden hiçbiri gerçekleştirilmemiştir.

3. BULGULAR

3.1. Geçerlik ve güvenilirlik analizleri

Çalışma grubundan elde edilen verilerin AFA'ya uygunluğunun sınanması amacıyla KMO katsayısı hesaplanmış ve “barlett spheccricity” testi uygulanmıştır. KMO katsayısı .919 ve Barlett testi anlamlı olarak bulunmuştur ($\chi^2=1895.592$, $p=.000$). Elde edilen sonuçlar veri setinin faktör analizi için uygun olduğunu göstermektedir (Hair vd., 1998; Leech vd., 2005; Tavşancıl, 2010).

AFA için gerekli ön koşulların sağlandığı belirlendikten sonra Türkçeye uyarlanan ölçeğin faktör yapısının orijinal ölçek ile uyumluluğunu sınamak amacıyla 5 faktörlü bir model üzerinden “varimax” döndürme tekniği kullanılarak temel bileşenler analizi gerçekleştirilmiştir. Varimax, AFA sırasında bağımsız alt faktörlerin belirlenebilmesi amacıyla faktörler arasındaki en hassas ayrımı veren ve en sık kullanılan döndürme yöntemlerinden biri olarak önerilmektedir (Ho, 2006). Analiz sonucunda ortaya çıkan varyans değerleri Tablo 3’te gösterilmiştir.

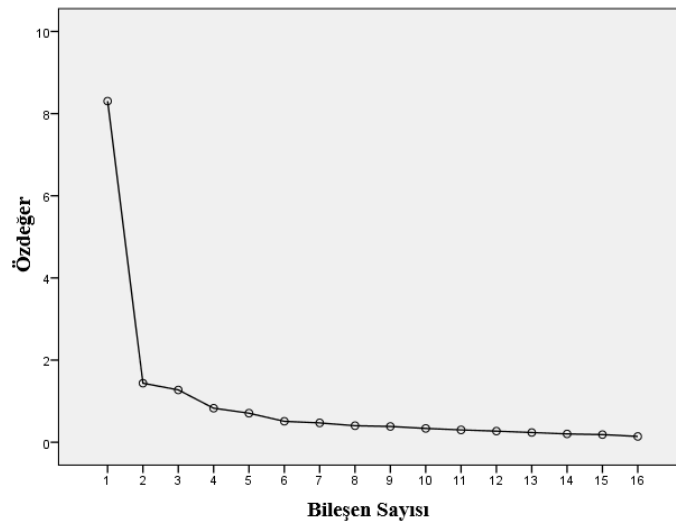
Tablo 3.

Açıklanan Toplam Varyans Tablosu

Faktörler	Başlangıç Öz Değerleri			Yüklenen Faktörlerin Karelerinin Dağılımı		
	Toplam	Varyansa Katkısı (%)	Kümülatif (%)	Toplam	Varyansa Katkısı (%)	Kümülatif (%)
1	8.304	51.900	51.900	8.304	51.900	51.900
2	1.438	8.988	60.888	1.438	8.988	60.888
3	1.274	7.965	68.853	1.274	7.965	68.853
4	.829	5.180	74.033	.829	5.180	74.033
5	.708	4.424	78.457	.708	4.424	78.457

Tablo 3 incelendiğinde 5 faktörlü yapının açıkladığı toplam varyans %78.457 olmasına rağmen öz değeri 1’den büyük olan 3 faktör olduğu görülmektedir. Faktör analizinde öz değeri 1 ve 1’in üzerinde olan faktörler kararlı olarak kabul edilmektedir (Pedhazur & Schmelkin, 1991). Benzer şekilde Çokluk vd. (2016), öz değeri 1’in üzerinde olan faktörlerin (Kaiser ölçütü) önemli birer faktör olarak değerlendirilmesi gerektiğini vurgulamaktadır.

Faktör sayısına karar vermek amacıyla kullanılması önerilen bir diğer ölçüt de yamaç-birikinti grafiğidir. Yamaç-birikinti grafiği baskın faktörleri ortaya koyarak faktör sayısını azaltmaya yardımcı olmaktadır (Çokluk vd., 2016). Şekil 1’de 5 faktörlü yapıya ilişkin analiz sonucunda elde edilen yamaç-birikinti grafiği gösterilmiştir. Grafik incelendiğinde çizgi eğiminin 3. faktörden itibaren belirgin bir şekilde plato yaptığı görülmektedir. Bu noktanın faktör sayısına karar vermede bir kesme noktası olarak değerlendirilebileceği ve modelin faktör sayısının ne olması gerektiğine işaret ettiği belirtilmektedir (Çokluk vd., 2016; Kalaycı, 2010).



Şekil 1. Beş faktörlü modele ilişkin yamaç-birikinti grafiği

Beş faktörlü modele ilişkin elde edilen veriler doğrultusunda Türkçeye uyarlanan ölçeğin 3 faktörlü bir yapıdan oluşabileceği düşünülmüştür. Bu doğrultuda AFA’nın ölçekteki maddeleri önceden belirlenen

faktör sayısı yerine öz değeri 1'den büyük faktörler altında toplayacak şekilde tekrarlanmasına karar verilmiştir. Tekrarlanan analize ilişkin sonuçlar Tablo 4'te sunulmuştur.

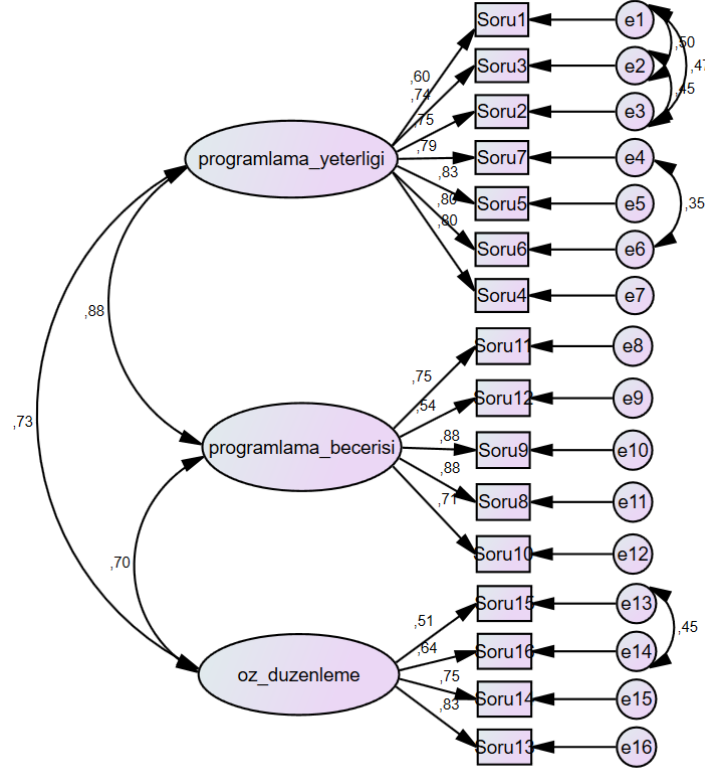
Tablo 4.*AFA Sonuçları*

Maddeler	Ortak Faktör Varyansı	1. Faktör	2. Faktör	3. Faktör
Soru 1	.760	.842		
Soru 3	.735	.808		
Soru 2	.747	.802		
Soru 7	.668	.700		
Soru 5	.684	.697		
Soru 6	.660	.627		
Soru 4	.659	.607		
Soru 11	.759		.797	
Soru 12	.610		.774	
Soru 9	.729		.619	
Soru 8	.706		.606	
Soru 10	.529		.553	
Soru 15	.735			.837
Soru 16	.734			.790
Soru 14	.649			.653
Soru 13	.651			.542
Özdeğer		8.304	1.438	1.274
Açıklanan Varyans (%)		29.943	23.146	15.763

Varimax döndürme tekniği kullanılarak gerçekleştirilen temel bileşenler analizi sonucunda 3 faktörlü yapının toplam varyansın %68.853'ünü açıkladığı belirlenmiştir. Bu oran çok faktörlü desenler için yeterli kabul edilmektedir (Büyüköztürk, 2007; Tavşancıl, 2010). Ortaya çıkan faktörler, ilgili literatür (Altun & Mazman, 2002; Askar & Davenport, 2009; Govender & Basak, 2015; Korkmaz & Altun, 2014; Kukul vd., 2017; Ramalingam & Wiedenbeck, 1998) ve içerdikleri maddeler göz önünde bulundurularak programlama yeterliği (1. faktör), programlama becerisi (2. faktör) ve öz düzenleme (3. faktör) olarak isimlendirilmiştir. Programlama yeterliği faktörü, öğrencilerin bir program editörünü kullanarak programın yapısını ve sonuçlarını yordayabilmelerine yönelik algılarını değerlendiren 7 maddeden oluşmaktadır. Programlama becerisi faktörü altında öğrencilerin programlama prosedürleri ve karşılaşılan sorunların üstesinden gelebilme becerileri ile ilgili algılarını ölçen 5 madde bulunmaktadır. Öz düzenleme faktörü ise öğrencilerin programlama hakkındaki duygu, düşünce ve davranışlarını yönlendirmeleri ile ilgili algılarını ortaya koyan 4 madde içermektedir.

AFA sonrasında ortaya çıkan 3 faktörlü modelin yapı geçerliğini değerlendirmek amacıyla DFA uygulanmıştır. Analiz sonucunda model üzerinde herhangi bir modifikasyon yapılmadan önce ulaşılan uyum iyiliği indeksleri hesaplanmıştır ($\chi^2/df=3.304$, RMSEA=.115, SRMR=.0663, CFI=.874, GFI=.793, AGFI=.721 ve NFI=.830). DFA sırasında ortaya çıkan modifikasyon önerileri dikkate alınarak modele katkıda bulunacağı öngörülen maddeler arasında modifikasyon işlemleri gerçekleştirilmiştir. Ortaya çıkan modifikasyon önerileri incelendiğinde sırasıyla e1-e2, e13-e14, e1-e3, e2-e3, e4-e6 maddeleri arasında modele yüksek katkı yapacak 5 modifikasyon önerisi olduğu görülmüştür. İlgili maddeler aynı faktörler altında yer aldıklarından aralarında gizil bir ilişkinin olabileceği düşünülmüş ve modifikasyon önerileri dikkate alınarak analiz tekrarlanmıştır. Tekrarlanan analiz sonrasında ortaya çıkan uyum iyiliği indeksleri $\chi^2/df=2.132$, RMSEA=.081, SRMR=.0514, CFI=.941, GFI=.869, AGFI=.814 ve NFI=.896 olarak hesaplanmıştır. Ortaya çıkan uyum iyiliği değerlerinin kabul edilebilir düzeyde olduğu belirtilmektedir (Brown; 2006; Cole, 1987; Jöreskog & Sörbom, 1993; Kline, 2011; Marsh vd., 1988; Sümer, 2000). Ölçekteki maddeler ile buldukları faktörler arasındaki standartlaştırılmış faktör yük değerleri .544 ile .883 arasında

değişmektedir. Faktör yük değerlerinin .30'dan büyük olması önerilmektedir (Büyüköztürk, 2007). Maddelere ilişkin standardize çözümlene değerlerinin anlamlı olup olmadığını belirlemek amacıyla t değerleri incelenmiştir. t değerlerinin $p < .01$ anlamlılık düzeyinde 6.106 ile 11.963 arasında değiştiği belirlenmiştir. DFA analizine ilişkin nihai sonuçlar Şekil 2'de gösterilmiştir.



Şekil 2. DFA sonuçları

DFA ile doğrulanan 3 faktör arasındaki ilişki korelasyon katsayısı kullanılarak incelenmiştir. Analiz sonuçları Tablo 5'te gösterilmiştir. Tablo incelendiğinde 3 faktör arasında olumlu ve anlamlı bir korelasyon olduğu görülmüştür.

Tablo 5.

Faktörler Arası Korelasyon Değerleri

Faktörler	Programlama Yeterliği	Programlama Becerisi	Öz Düzenleme
Programlama Yeterliği	1.000	.884***	.730***
Programlama Becerisi		1.000	.701***
Öz Düzenleme			1.000

*** $p = .000$

Ölçeğin faktöriyel geçerliğinin sağlandığı görüldükten sonra güvenirliliğini belirlemek amacıyla ölçeğin geneline ve her bir alt faktöre ait α değerleri hesaplanmıştır. 16 maddenin α güvenirlilik katsayısı .936 olarak bulunmuştur. Alt faktörlere ilişkin hesaplanan iç tutarlık katsayıları; programlama yeterliği faktörü için .919, programlama becerisi faktörü için .867 ve öz düzenleme faktörü için .808 olarak belirlenmiştir. Ölçekteki maddelerin madde toplam korelasyonları .430 ile .771 arasında değişmektedir. Ölçeğin alt boyutlarına ilişkin güvenirlilik analizi sonuçları Tablo 6'da sunulmuştur.

Tablo 6.

Güvenirlilik Analizi Sonuçları

Faktör	Madde	Madde Toplam Korelasyonu	α
Programlama Yeterliği	Soru 1	.594	.919
	Soru 2	.735	
	Soru 3	.722	
	Soru 4	.741	
	Soru 5	.744	
	Soru 6	.759	
	Soru 7	.744	
Programlama Becerisi	Soru 8	.764	.867
	Soru 9	.771	
	Soru 10	.652	
	Soru 11	.706	
	Soru 12	.488	
Öz Düzenleme	Soru 13	.683	.771
	Soru 14	.567	
	Soru 15	.430	
	Soru 16	.586	

Ölçeğin geneline ve alt boyutlarına ilişkin iç tutarlılık katsayılarının kabul edilebilir değer olan .70'in üzerinde olduğu (Creswell, 2005) görülmektedir. Bu doğrultuda uyarlanan ölçeğin güvenilir olduğu söylenebilir. Madde-toplam korelasyonları .30 ve daha üzerinde olan maddelerin bireyleri ölçülen özellik bakımından iyi derecede ayırt ettiği dikkate alındığında (Büyüköztürk, 2007) ortaya çıkan değerlerin ayırt edicilik bakımından yeterli olduğu ifade edilebilir. Türkçe uyarlaması yapılan ölçek maddeleri ve uygulama yönergesi Ek-1'de sunulmuştur.

3.2. Ölçeğe ilişkin öğrenci puanları

Araştırmada çalışma grubundan elde edilen veriler öğrencilerin bilgisayar programlamaya yönelik öz yeterlik düzeylerinin incelenmesi amacıyla da kullanılmıştır. Öğrencilerin ölçekteki maddelere verdikleri yanıtlara ilişkin tanımlayıcı istatistikler Tablo 7'de gösterilmiştir.

Tablo 7.

Güvenirlilik Analizi Sonuçları

Faktörler	Madde Sayısı	En Düşük	En Yüksek	X	Sd
Programlama Yeterliği	7	1	5	3.82	.95
Programlama Becerisi	5	1	5	3.16	.94
Öz Düzenleme	4	1	5	3.70	.87

Tablo 7 incelendiğinde öğrencilerin BPÖY ölçeği alt faktörlerine ilişkin ortalama puanlarının birbirine yakın olduğu görülmektedir. Öğrenciler en yüksek puanı sırasıyla programlama yeterliği (3.82), öz düzenleme (3.70) ve programlama becerisi (3.16) faktörlerinden almışlardır. Ölçeğin 5'li Likert tipinde uygulandığı göz önüne alındığında öğrencilerin ölçeğin genelinden yüksek puan aldıkları görülmektedir ($X=3.56$). Bu doğrultuda öğrencilerin bilgisayar programlama ile ilgili öz yeterliklerinin yüksek düzeyde olduğu söylenebilir. Alt faktörler bağlamında incelendiğinde ise öğrencilerin programlama becerisi alt faktörüne ilişkin öz yeterlik algılarının diğer faktörlere göre daha düşük olduğu görülmüştür.

3.3. Cinsiyet farklılıkları

Cinsiyet faktörünün bilgisayar programlamaya dair öz yeterlik algısı üzerindeki etkisini incelemek amacıyla öncelikle verilerin normal dağılım gösterip göstermediği Shapiro-Wilk testi uygulanarak test edilmiştir. Test sonuçları Tablo 8’de sunulmuştur.

Tablo 8.

Normallik Testi Sonuçları

Faktörler	İstatistik	df	p
Programlama Yeterliği	.922	174	.000
Programlama Becerisi	.983	174	.036
Öz Düzenleme	.942	174	.000

Analiz sonucunda verilerin normal dağılım göstermediği ($p < .05$) görüldüğünden cinsiyet faktörünün bilgisayar programlamaya yönelik öz yeterlik algısı üzerindeki etkisi Mann-Whitney U testi kullanılarak incelenmiştir. Analiz sonuçları Tablo 9’da gösterilmiştir.

Tablo 9.

Programlama Öz Yeterliği Cinsiyet Farklılıkları

Faktörler	Grup	N	Sıra Ortalaması	Sıra Toplamı	U	Z	p
Programlama Yeterliği	Kız	69	86.02	5935.50	3520.5	-.314	.753
	Erkek	105	88.47	9289.50			
Programlama Becerisi	Kız	69	83.96	5793.50	3378.5	-.752	.452
	Erkek	105	89.82	9431.50			
Öz Düzenleme	Kız	69	86.46	5965.50	3350.5	-.223	.824
	Erkek	105	88.19	9259.50			

Analiz sonucunda katılımcıların cinsiyetleri bakımından öz düzenleme, programlama becerileri ve programlama yeterlilikleri arasında anlamlı fark bulunamamıştır. Programlama öz yeterliğinin her bir alt faktöründeki sıra ortalamaları açısından erkek öğrencilerin kız öğrencilere göre daha yüksek puan ortalamasına sahip olduğu görülmüştür.

3.4. Programlama deneyiminin etkisi

Araştırmada öğrencilerin programlama deneyimleri ile programlamaya yönelik öz yeterlik algıları arasındaki ilişki incelenmiştir. Programlama deneyimi, öğrencilerin bilgisayar programlama ile uğraştıkları yıl sayısı üzerinden değerlendirilmiştir. Öğrencilerin programlama deneyimi düzeyleri 3 gruba ayrılarak incelenmiştir. Programlama deneyimi 1 yıl ve daha az olanlar düşük seviye ($N=81$, %46.6), 1-5 yıl arası olanlar orta seviye ($N=76$, %43.7) ve 5 yıl ve üzeri olanlar yüksek seviye ($N=17$, %9.8) olarak gruplandırılmıştır. Öğrencilerin programlama deneyimi ile bilgisayar programlama öz yeterlikleri arasındaki ilişki Kruskal-Wallis testi kullanılarak analiz edilmiştir. Analiz sonuçları Tablo 10’da sunulmuştur.

Tablo 10.

Kruskal-Wallis Testi Sonuçları

Faktörler	(1) Düşük Seviye		(2) Orta Seviye		(3) Yüksek Seviye		Sd	χ^2	p	Sonuç
	N	Sıra Ortalaması	N	Sıra Ortalaması	N	Sıra Ortalaması				
Programlama Yeterliği	81	69.41	81	92.35	81	152.03	2	39.220	.000	1 < 2 < 3
Programlama Becerisi	76	67.90	76	94.76	76	148.41	2	38.882	.000	1 < 2 < 3
Öz Düzenleme	17	72.00	17	93.69	17	133.68	2	23.350	.000	1 < 2 < 3

Tablo 10 incelendiğinde seviye grupları arasında tüm alt faktörler bakımından anlamlı farklılıklar olduğu görülmektedir. Öğrencilerin programlama deneyimi arttıkça bilgisayar programlama öz yeterlik algıları programlama yeterliği, programlama becerisi ve öz düzenleme bağlamında artış göstermektedir. Daha uzun süre programlama ile uğraşan öğrenciler programlama konusunda kendilerini daha yeterli olarak algılamaktadırlar. Sonuç olarak programlama deneyimi ile programlamaya yönelik öz yeterlik algısı arasında anlamlı ve olumlu bir ilişki olduğu söylenebilir.

4. TARTIŞMA ve SONUÇ

Bu çalışmada Tsai vd. (2019) tarafından geliştirilen bilgisayar programlama öz yeterlik ölçeği Türkçeye uyarlanmıştır. Orijinal ölçek mantıklı düşünme, iş birliği, algoritma, kontrol ve hata ayıklama şeklindeki 5 alt faktörden oluşurken Türkçeye uyarlandığında 3 faktörlü bir yapı gösterdiği belirlenmiştir. Ortaya çıkan yeni faktörler programlama yeterliği, programlama becerisi ve öz düzenleme olarak isimlendirilmiştir. Erkuş (2007), bireylerin duygu, düşünce veya davranışlarının bir kültürden diğerine farklılık gösterebileceğinden dolayı ölçülmek istenilen psikolojik değişkenlerin yapılarında da değişiklik oluşabileceğini ifade etmektedir. Dolayısıyla uyarlanan ölçeklerin madde ve faktör yapılarının değişkenlik göstermesi uyarlama çalışmalarının doğal bir sonucu olarak görülmektedir (Akbaş & Korkmaz, 2007; Erkuş, 2007). Faktör yapısının denkleştirilmesi amacıyla öncelikle uyarlanan ölçeğin yapı geçerliği 5 faktörlü olacak şekilde sınanmış, ancak sonuçta 3 faktörlü bir yapının daha geçerli ve güvenilir olduğu görülmüştür. Gökoğlu ve Çakıroğlu (2019) tarafından da vurgulandığı üzere uyarlama çalışmalarında faktör yapısı denkleştirilmeye çalışıldığında ölçeğin yapısal özellikleri zayıflayabilmektedir. Tüm bu hususlar göz önünde bulundurularak faktör yapısının eşitlenmeye çalışılmasının araştırmayı uyarlama çalışmasından uzaklaştırarak bir çeviri çalışmasına dönüştüreceği düşünülmüş ve analizler sonucunda ortaya çıkan bulgular dikkate alınarak yeni faktör yapısına karar verilmiştir.

Analizler sonucunda ortaya çıkan faktör yapısı incelendiğinde orijinal ölçekte mantıklı düşünme (4, 5, 6, 7) ve kontrol (1, 2, 3) faktöründe bulunan maddelerin programlama yeterliği faktörü altında birleştiği görülmüştür. Bu maddeler incelendiğinde bir programlama editörünün kullanımı, programlamanın mantıksal yapısının anlaşılması ve programın nihai sonucunun tahmin edilebilmesi gibi yeterli olma durumları sınındığından ilgili faktör programlama yeterliği olarak isimlendirilmiştir. Hata ayıklama (8, 9, 10) ve algoritma (11, 12) faktörleri altında yer alan maddeler bir araya gelerek programlama becerisi faktörünü oluşturmuştur. Bu faktör altında toplanan maddeler daha çok bir programa ilişkin işlemler, akış vb. prosedürel yapıların çözümlenmesi ve programlama sırasında karşılaşılan hataların giderilebilmesi ile ilgili becerilere odaklandığından ilgili faktör programlama becerisi olarak adlandırılmıştır. Öz düzenleme faktörü altında ise algoritma (12, 13) ve iş birliği (14, 15, 16) faktörlerinde bulunan maddeler bir araya gelmiştir. İlgili maddeler bireylerin programlama hakkındaki biliş ve davranışlarının değerlendirilmesi ve düzenlenmesine yönelik olduğundan faktöre öz düzenleme ismi verilmiştir. Benzer şekilde Ramalingam ve Wiedenbeck (1998) tarafından geliştirilen ölçekte de öz düzenleme faktörü yer almaktadır. Tsai vd. (2019) ise öz düzenlemenin orijinal ölçeğin odağını oluşturduğunu belirtmektedir. Orijinal ölçekte yer alan ve bireylerin program kontrolünü gerçekleştirme, program hatalarını düzeltmek için gerekli olan izleme, planlama, değerlendirme ve karar verme gibi unsurlara ilişkin kişisel güven düzeylerini değerlendiren maddeler öz düzenleme becerilerinin temel unsurlarını oluşturmaktadır. Dolayısıyla bu tür maddelerin bir araya gelerek oluşturdukları yeni faktör öz düzenleme olarak isimlendirilmiştir.

Öğrencilerin bilgisayar programlamaya yönelik öz yeterlik algılarını ölçmek amacıyla geliştirilen önceki ölçekler (Altun & Mazman, 2002; Askar & Davenport, 2009; Govender & Basak, 2015; Korkmaz & Altun, 2014; Ramalingam & Wiedenbeck, 1998) daha çok belirli bir programlama dilinin sözdizimi yapısına odaklanmaktadır. Altun ve Kasalak (2018) tarafından geliştirilen ölçek ise blok temelli programlama eğitimlerinde sıklıkla kullanılan Scratch uygulamasının fonksiyonlarına ve özellikleri ile ilgili öz yeterlik algısını ölçmektedir. Bu ölçekler ile sadece C++ veya Java gibi programlama dillerine veya Scratch gibi blok temelli programlama yazılımlarının fonksiyonlarına dair öz yeterlik algısı ölçülebilirken uyarlanan ölçek ile bilgisayar programlamaya yönelik öz yeterlik algısı okuryazarlık düzeyinde ve BİD becerilerini yansıttak

şekilde değerlendirilebilmektedir. Dolayısıyla uyarlanan ölçek, programlama uzmanlarının (bilgisayar mühendisleri, yazılım mühendisleri ve bilgisayar programcıları) yanı sıra temel düzeyde programlama eğitimi almış kişilerin de bilgisayar programlama hakkındaki öz yeterliklerini incelemek amacıyla kullanılabilir niteliktedir. Bu doğrultuda uyarlanan ölçeğin metin tabanlı programlama öğretiminin yanı sıra grafik temelli veya blok temelli programlama öğretimleri sırasında da öğrencilerin öz yeterliklerini değerlendirmek amacıyla kullanılabilir olduğu düşünülmektedir.

Bilgisayar programlama ile ilgili mevcut ölçekler (Altun & Mazman, 2002; Askar & Davenport, 2009; Govender & Basak, 2015; Korkmaz & Altun, 2014; Ramalingam & Wiedenbeck, 1998) üniversite düzeyinde öğrenim gören öğrenciler için geliştirilmiştir. Altun ve Kasalak (2018) ile Kukul vd. (2017) tarafından geliştirilen ölçekler ise ortaokul öğrencilerinin öz yeterlik algılarına odaklanmaktadır. Bu çalışma ile uyarlanan ölçek ise üniversite düzeyinde öğrenim gören öğrencilerden elde edilen veriler çerçevesinde geliştirilmiştir. BPÖY ölçeğinde yer alan prosedür, modül, alt program ve hata ayıklama gibi kavramların lise ve ortaokul düzeyinde öğrenim gören öğrenciler tarafından yeterince anlaşılamayacağı düşünülerek ölçek geliştirme sürecinde bu düzeylerde programlama eğitimi veren öğretmenlerin görüşleri alınmıştır. Yapılan görüşmeler sonucunda benzer düşüncenin ilgili öğretmenler tarafından da ifade edilmesi neticesinde ölçeğin üniversite öğrencilerine yönelik olarak uygulanmasına karar verilmiştir.

Araştırmada öğrencilerin programlamaya ilişkin öz yeterliklerinin yüksek düzeyde olduğu bulunmuştur. Benzer şekilde programlamaya yönelik öz yeterlik algısını ölçmek amacıyla ölçek uyarlama çalışması yürüten Korkmaz ve Altun (2014) ise araştırmalarında öğrencilerin öz yeterliklerinin orta düzeyde olduğunu ortaya koymuştur. İlgili araştırmada kullanılan ölçek C++ programlama diline yönelik olarak uyarlanmıştır. Araştırma örneklemini mühendislik öğrencilerinden oluşmaktadır. Bu araştırmada ise programlamaya yönelik öz yeterlik düzeyi okuryazarlık seviyesinde ve BİD becerilerini yansıtabilecek şekilde ölçülmüştür. Dolayısıyla öğrencilerin öz yeterliklerinin yüksek seviyede bulunmasının ölçeğin maddelerinin programlamaya yönelik daha genel bir yapıyı ölçmeye yönelik olduğundan kaynaklandığı düşünülmektedir. Öte yandan öğrencilerin programlama becerisi alt faktörüne ilişkin öz yeterlik algıları diğer faktörlere göre daha düşük çıkmıştır. Bu sonuç öğrencilerin programlamaya yönelik kendilerini bilişsel açıdan yeterli gördükleri, ancak uygulama konusunda kendilerini nispeten daha az yeterli gördükleri şeklinde yorumlanabilir.

Cinsiyet farklılığı, bilgisayar bilimleri alanında yürütülen araştırmalarda etkisi incelenen genel değişkenler arasında yer almaktadır. Özellikle bilgisayar programlama becerisi, BİD becerisi, programlamaya yönelik öz yeterlik algısı; cinsiyet faktörünün etkisinin sıklıkla incelendiği konu alanlarından (Askar & Davenport, 2009; Altun & Mazman, 2012; Gezgin & Adnan, 2016; Karaçaltı vd., 2018; Korkmaz, 2013; Korkmaz & Altun, 2014; Özyurt & Özyurt, 2015; Ramalingam & Wiedenbeck, 1998). Öğrencilerin bilgisayar programlamaya yönelik öz yeterliklerinin cinsiyet açısından farklılaşıp farklılaşmadığını inceleyen araştırmaların çoğunda kız ve erkek öğrenciler arasında herhangi bir anlamlı farklılık bulunmadığı raporlaştırılmıştır (Altun & Kasalak, 2018; Altun & Mazman, 2012; Korkmaz, 2013; Korkmaz & Altun, 2014; Ramalingam & Wiedenbeck, 1998). Öte yandan az sayıda araştırmada programlamaya yönelik öz yeterlik algısının cinsiyet faktörüne göre anlamlı bir şekilde farklılaştığı ve erkek öğrencilerin kız öğrencilere göre daha yüksek öz yeterliğe sahip olduğu ortaya konulmuştur (Askar & Davenport, 2009; Özyurt & Özyurt, 2015). Bu araştırmada ise öğrencilerin bilgisayar programlamaya yönelik öz yeterlik algıları cinsiyete göre anlamlı bir farklılık göstermemiştir. Ancak ölçeğin alt faktörlerinden elde edilen ortalama puanlar incelendiğinde erkek öğrencilerin kız öğrencilere göre daha yüksek öz yeterlik algısına sahip olduğu görülmüştür. Bu sonuç öğrencilerin bilgisayar programlamaya yönelik öz yeterliklerinin gerek Java, C++ ve blok temelli programlama gibi belirli bir programlama dilinin sözdizimi yapısına yönelik olarak gerekse okuryazarlık düzeyinde ve BİD becerileri temele alınarak incelendiğinde cinsiyete göre anlamlı bir farklılık oluşturmadığını ortaya koyması bakımından önemli görülmektedir. Benzer şekilde Karaçaltı vd. (2018) de cinsiyete göre programlama becerisi ve BİD becerilerinde farklılaşma bulunmadığını vurgulamaktadır.

Yaptığı araştırmada cinsiyetin programlamaya yönelik öz yeterlik üzerinde etkili olduğu bulgusuna ulaşan Askar ve Davenport (2009), bu farklılığın örneklem grubunun programlama dersini alma zorunluluğu bulunmasından ve bu öğrencilerin üniversite giriş sınavında çok yüksek puanlar alarak ilgili bölümlere yerleşmelerine sebep olan yüksek matematiksel ve analitik becerilere sahip olmalarından kaynaklanabileceğini belirtmektedir.

Programlama deneyiminin bilgisayar programlamaya yönelik öz yeterlik algısı üzerindeki etkisi ile ilgili olarak bu araştırmada, daha yüksek programlama deneyimine sahip olan öğrencilerin daha yüksek öz yeterliğe sahip oldukları belirlenmiştir. Bu sonuç literatürdeki araştırma bulguları ile paralellik göstermektedir (Altun & Mazman, 2012; Askar & Davenport, 2009; Jegede, 2009; Korkmaz & Altun, 2014; Özyurt & Özyurt, 2015; Ramalingam vd., 2004; Ramalingam & Wiedenbeck, 1998). İlgili araştırmalar sınıf düzeyinin, öğrenim görülen bölümün, bilgisayar programlamaya yönelik alınan ders sayısının, bilgisayarla ilgili zihinsel altyapının, dolayısıyla tüm bu bileşenlerin işaret ettiği programlama deneyiminin öz yeterlik üzerinde etkili olduğunu vurgulamaktadır. Bandura (1997), bireylerin öz yeterliklerinin önceki bireysel deneyimlerinden kaynaklandığının altını çizmektedir. Dolayısıyla öğrencilerin bilgisayar programlamaya yönelik yaşadıkları deneyim arttıkça öz yeterlikleri de artacaktır. Bu bağlamda bilgisayar mühendisliğinde öğrenim gören öğrencilerin diğer mühendislik dallarında öğrenim gören öğrencilere göre programlamaya yönelik daha yüksek öz yeterlik algısına sahip olması, bilgisayar programlama ağırlıklı bölümlerde öğrenim gören öğrencilerin sınıf seviyesi yükseldikçe öz yeterliklerinin de artması doğal bir sonuç olarak karşılanabilir. Nitekim bu araştırmada da yüksek seviyede programlama deneyimine sahip olan öğrenciler orta ve düşük seviyede programlama deneyimine sahip olan öğrencilere göre, programlama yeterliği, programlama becerisi ve öz düzenleme bakımından daha yüksek puan almışlardır.

Bu araştırma kapsamında uyarlanan BPÖY ölçeği ile öğrencilerin bilgisayar programlamaya yönelik öz yeterlik algıları BİD becerileri temele alınarak okuryazarlık düzeyinde incelenebilecektir. 21. yüzyıl öğrenen becerileri olarak ifade edilen standartlar göz önünde bulundurulduğunda, okuma ve yazma gibi temel becerilerin yanına programlama becerisinin de eklenebileceği öngörülebilir. Dolayısıyla bilgisayar programlamaya yönelik öz yeterlik algısını ölçmek için kullanılacak ölçme aracının bilgisayar bilimlerine yönelik bölümlerde öğrenim gören öğrencilerin yanı sıra diğer tüm bölümlerde öğrenim gören öğrenciler için de kullanılabilir nitelikte olması faydalı olacaktır. Gerek programlama dillerinin çeşitliliği gerekse farklı programlama yöntemlerinin bulunmasından dolayı ortak bir öz yeterlik algısı ölçme aracının oluşturulması zorlaşmaktadır. Bu araştırma ile uyarlanan ölçeğin bu sorunun çözümüne katkıda bulunabileceği düşünülmektedir.

Kaynakça / Reference

- Akbaş, G., & Korkmaz, L. (2007). Ölçek uyarlaması (Adaptasyon). *Türk Psikoloji Bülteni*, 13(40), 15-16.
- Alsancak Sırakaya, D. (2019). Programlama öğretiminin bilgi işlemsel düşünme becerisine etkisi. *Türkiye Sosyal Araştırmalar Dergisi*, 23(2), 575-590.
- Altun, A., & Kasalak, İ. (2018). Blok temelli programlamaya ilişkin öz-yeterlik algısı ölçeği geliştirme çalışması: Scratch örneği. *Eğitim Teknolojisi Kuram ve Uygulama*, 8(1), 209-225. <https://doi.org/10.17943/etku.335916>
- Altun, A., & Mazman, S. G. (2012). Programlamaya ilişkin öz yeterlilik algısı ölçeğinin Türkçe formunun geçerlilik ve güvenilirlik 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), 26-32.
- Ater-Kranov, A., Bryant, R., Orr, G., Wallace, S., & Zhang, M. (2010, October 7-9). *Developing a community definition and teaching modules for computational thinking: Accomplishments and challenges* [Paper presentation]. 2010 ACM Conference on Information Technology Education, USA.
- Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661-670. <https://doi.org/10.1016/j.robot.2015.10.008>
- Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 84(2), 191-215. <https://doi.org/10.1037/0033-295X.84.2.191>
- Bergey, B. W., Ketelhut, D. J., Liang, S., Natarajan, U., & Karakus, M. (2015). Scientific inquiry self-efficacy and computer game self-efficacy as predictors and outcomes of middle school boys' and girls' performance in a science assessment in a virtual environment. *Journal of Science Education and Technology*, 24(5), 696-708. <https://doi.org/10.1007/s10956-015-9558-4>
- Berland, M., & Lee, V. R. (2011). Collaborative strategic board games as a site for distributed computational thinking. *International Journal of Game-Based Learning*, 1(2), 65-81. <https://doi.org/10.4018/ijgbl.2011040105>
- Brennan, K., & Resnick, M. (2012, April 13-17). *New frameworks for studying and assessing the development of computational thinking* [Paper presentation]. Annual American Educational Research Association Meeting, Canada.
- Brown, T. A. (2006). *Confirmatory factor analysis: For applied research*. Guilford Press.
- Büyüköztürk, Ş. (2007). *Sosyal bilimler için veri analizi el kitabı* (8. Baskı). Pegem Akademi Yayıncılık.
- Cole, D. A. (1987). Utility of confirmatory factor analysis in test validation research. *Journal of Consulting and Clinical Psychology*, 55, 1019-1031. <https://doi.org/10.1037/0022-006X.55.4.584>
- Creswell, J. W. (2005). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research* (2nd ed.). Pearson Merrill Prentice Hall.
- Çankaya, S., Durak, G., & Yüncül, E. (2017). Education on programming with robots: Examining students' experiences and views. *Turkish Online Journal of Qualitative Inquiry*, 8(4), 428-445. <https://doi.org/10.17569/tojqi.343218>
- Çınar, M., & Tüzün, H. (2017, Mayıs 24-26). *Eğitimde bilgisayarlı düşünme uygulamalarına ilişkin bir alanyazın incelemesi* [Sözlü bildiri]. 11. Uluslararası Bilgisayar ve Öğretim Teknolojileri Eğitimi Sempozyumu, Türkiye.
- Çokluk, Ö., Şekercioğlu, G., & Büyüköztürk, Ş. (2016). *Sosyal bilimler için çok değişkenli istatistik: SPSS ve LISREL uygulamaları*. Pegem Akademi.
- Davidsson, K., Larzon, L., & Ljunggren, K. (2010). *Self-efficacy in programming among STS students* (Technical Report). Computer Science Education Course of Upssala University. <http://www.it.uu.se/edu/course/homepage/datadidaktik/ht10/reports/Self-Efficacy.pdf>

- Demir, Ö., & Seferoğlu, S. S. (2017). Yeni kavramlar, farklı kullanımlar: Bilgi-işlemsel düşünmeyle ilgili bir değerlendirme. H. F. Odabaşı, B. Akkoyunlu ve A. İşman (Eds.), *Eğitim teknolojileri okumaları 2017* içinde (ss. 468-483). Sakarya Üniversitesi.
- Erkuş, A. (2007). Ölçek geliştirme ve uyarlama çalışmalarında karşılaşılan sorunlar. *Türk Psikoloji Bülteni*, 13(43), 17-25.
- Fang, X. (2012). Application of the participatory method to the computer fundamentals course. *Affective Computing and Intelligent Interaction*, 137, 185-189. https://doi.org/10.1007/978-3-642-27866-2_23
- Field, A. P. (2000). *Discovering statistics using SPSS for Windows: Advanced techniques for the beginner*. Sage Publications.
- Garner, S. (2009, July 15-17). *Learning to program from Scratch* [Paper presentation]. 9th IEEE International Conference on Advanced Learning Technologies, Latvia.
- Genç, Z., & Karakuş, S. (2011, Ekim, 11-13). *Tasarımla öğrenme: Eğitsel bilgisayar oyunları tasarımında Scratch kullanımı* [Sözlü bildiri]. 5. Uluslararası Bilgisayar ve Öğretim Teknolojileri Eğitimi Sempozyumu, Türkiye.
- Gezgin, D. M., & Adnan, M. (2016). Makine mühendisliği ve ekonometri öğrencilerinin programlamaya ilişkin öz yeterlik algılarının incelenmesi. *Ahi Evran Üniversitesi Kırşehir Eğitim Fakültesi Dergisi*, 17(2), 509-525.
- Girasoli, A. J., & Hannafin, R. D. (2008). Using asynchronous AV communication tools to increase academic self-efficacy. *Computers & Education*, 51, 1676-1682. <https://doi.org/10.1016/j.compedu.2008.04.005>
- Gomes, A. ve Mendes, A. J. (2007, September 3-7). *Learning to program-difficulties and solutions* [Paper presentation]. International Conference on Engineering Education, Portugal.
- Govender, D. W., & Basak, S. K. (2015). An investigation of factors related to self-efficacy for java programming among computer science education students. *Journal of Governance and Regulation*, 4(4), 612-619. https://doi.org/10.22495/jgr_v4_i4_c5_p6
- Gökoğlu, S., & Çakıroğlu, Ü. (2019). Sanal gerçeklik temelli öğrenme ortamlarında bulunuşluk hissini ölçülmesi: Bulunuşluk ölçeğinin Türkçe'ye uyarlanması. *Eğitim Teknolojisi Kuram ve Uygulama*, 9(1), 169-188. <https://doi.org/10.17943/etku.441497>
- Hair, J., Anderson, R., Tatham, R., & Black, W. (1998). *Multivariate data analysis* (5th ed.). Prentice Hall.
- Ho, R. (2006). *Handbook of univariate and multivariate data analysis and interpretation with SPSS*. Chapman & Hall/CRC.
- Hoffman, B., & Spatariu, A. (2008). The influence of self-efficacy and metacognitive prompting on math problem-solving efficiency. *Contemporary Educational Psychology*, 33(4), 875-893.
- ISTE (2015). Computational thinking leadership toolkit (First Edition). <https://www.iste.org/explore/articleDetail?articleid=152&category=Solutions&article=Computational-thinking-for-all>
- ISTE (2016). ISTE standards for students. <https://www.iste.org/standards/standards-for-students-2016>
- Jöreskog, K. G., & Sörbom, D. (1993). *LISREL 8: Structural equation modeling with the simplis command language*. Scientific Software International Inc.
- Kalaycı, Ş. (Ed.). (2010). *SPSS uygulamalı çok değişkenli istatistik teknikleri* (5. baskı). Asil Yayıncılık.
- Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, 4(3), 583-596.
- Karaçaltı, C., Korkmaz, Ö., & Çakır, R. (2018). Öğrencilerin programlama başarılarının bilgisayarca-eleştirel düşünme ile problem çözme becerileri çerçevesinde incelenmesi. *Amasya Üniversitesi Eğitim Fakültesi Dergisi*, 7(2), 343-370.
- Karsten, R., Mitra, A., & Schmidt, D. (2012). Computer self-efficacy : A meta-analysis. *Journal of Organizational Ans End User Computing*, 24(4), 54-80.
- Kazimoğlu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2012). Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science*, 9, 522-531. <https://doi.org/10.1016/j.procs.2012.04.056>
- Kline, R. B. (2011). *Principles and practice of structural equation modeling* (3rd ed.). The Guilford Press.

- Korkmaz, Ö. (2013). Prospective CITE teachers' self-efficacy perceptions on programming. *Procedia - Social and Behavioral Sciences* 83, 639-643. <https://doi.org/10.1016/j.sbspro.2013.06.121>
- Korkmaz, Ö., & Altun, H. (2014). Adapting computer programming self-efficacy scale and engineering students' self-efficacy perceptions. *Participatory Educational Research*, 1(1), 20-31. <https://doi.org/10.17275/per.14.02.1.1>
- Korkmaz, Ö., Çakır, R., & Özden, M. Y. (2017). A validity and reliability study of the Computational Thinking Scales (CTS). *Computers in Human Behavior*, 72, 558-569. <https://doi.org/10.1016/j.chb.2017.01.005>
- Korkut, E., & Akkoyunlu, B. (2008). Yabancı dil öğretmen adaylarının bilgi ve bilgisayar okuryazarlık öz-yeterlikleri. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 34, 178-188.
- Kukul, V., Gökçearslan, Ş., & Günbatır, M. S. (2017). Computer programming self-efficacy scale (cpses) for secondary school students: Development, validation and reliability. *Eğitim Teknolojisi Kuram ve Uygulama*, 7(1), 158-179. <https://doi.org/10.17943/etku.288493>
- Kurbanoglu, S., & Akkoyunlu, B. (2002). Öğretmen adaylarına uygulanan bilgi okuryazarlığı programının etkililiği ve bilgi okuryazarlığı becerileri ile bilgisayar öz-yeterlik algısı arasındaki ilişki. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 22, 98-105.
- Leech, N. L., Barrett, K. C., & Morgan, G. A. (2005). *SPSS for intermediate statistics, use and interpretation* (2nd ed.). Mahwah.
- Marsh, H.W., Balla, J. R., & McDonald, R. P. (1988). Goodness-of-fit indexes in confirmatory factor analysis: The effect of sample size. *Psychological Bulletin*, 103, 391-410.
- Moos, D. C., & Azevedo, R. (2009). Learning with computer-based learning environments: A literature review of computer self-efficacy. *Review of Educational Research*, 79(2), 576-600. <https://doi.org/10.3102/0034654308326083>
- Nilsen, H., & Larsen, A. (2011, November 21-23) *Using the personalized system of instruction in an introductory programming course* [Paper presentation]. NOKOBIT, Norveç.
- Oluk, A., & Korkmaz, Ö. (2016). Comparing students' scratch skills with their computational thinking skills in terms of different variables. *International Journal of Modern Education and Computer Science*, 11, 1-7. <https://doi.org/10.5815/ijmecs.2016.11.01>
- Oluk, A., Korkmaz, Ö., & Oluk, H. A. (2016). Scratch'ın 5. sınıf öğrencilerinin algoritma geliştirme ve bilişimsel düşünme becerilerine etkisi. *Turkish Journal of Computer and Mathematics Education*, 9(1), 54-71. <https://doi.org/10.16949/turkbilmat.399588>
- Özmen, B., & Altun, A. (2014). Undergraduate students' experiences in programming: Difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry*, 5(3), 9-27.
- Özyurt, Ö., & Özyurt, H. (2015). A study for determining computer programming students' attitudes towards programming and their programming self-efficacy. *Journal of Theory and Practice in Education*, 11(1), 51-67.
- Pedhazur, E. J., & Schmelkin, L. P. (1991). *Measurement, design, and analysis: An integrated approach*. Lawrence Erlbaum.
- Pillay, N., & Jugoo, V. R. (2005). An investigation into student characteristics affecting novice programming performance. *SIGCSE Bulletin* 37(4), 107-110. <https://doi.org/10.1145/1113847.1113888>
- Ramalingam, V., LaBelle, D., & Wiedenbeck, S. (2004, June 28-30). *Self-efficacy and mental models in learning to program*. 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, United Kingdom.
- Ramalingam, 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. <https://doi.org/10.2190/C670-Y3C8-LTJ1-CT3P>
- Sümer, N. (2000). Yapısal eşitlik modelleri: Temel kavramlar ve örnek uygulamalar. *Türk Psikoloji Yazıları*, 3(6), 49-74.
- Tabachnick, B. G., & Fidell, L. S. (2007). *Using multivariate statistics* (5th ed.). Allyn & Bacon.

- Tavşancıl, E. (2010). *Tutumların ölçülmesi ve SPSS ile veri analizi* (4. baskı). Nobel Yayın Dağıtım.
- Tsai, M.-J., Wang, C.-Y., & Hsu, P.-F. (2019). Developing the computer programming self-efficacy scale (CPSES) for computer literacy education. *Journal of Educational Computing Research*, 56(8), 1345-1360. <https://doi.org/10.1177/0735633117746747>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of The Royal Society*, 366, 3717-3725. <https://doi.org/10.1098/rsta.2008.0118>

EKLER

Ek-1. Bilgisayar okuryazarlığı eğitimi için bilgisayar programlama öz yeterlik ölçeği

Lütfen aşağıdaki soruları kendi durumunuza göre 1 - 5 arasında puanlayınız.

- 1 = Kesinlikle Güvenmiyorum (Yapamam)
2 = Güvenmiyorum
3 = Kararsızım
4 = Güveniyorum
5 = Kesinlikle Güveniyorum (Yapabilirim)

1) Program editöründe bir programı açabilir ve kaydedebilirim.	(1)	(2)	(3)	(4)	(5)
2) Program editöründe bir programı gözden geçirebilir ve düzenleyebilirim.	(1)	(2)	(3)	(4)	(5)
3) Program editöründe bir programı çalıştırabilir ve test edebilirim.	(1)	(2)	(3)	(4)	(5)
4) Bir programın temel mantıksal yapısını anlayabilirim.	(1)	(2)	(3)	(4)	(5)
5) Eğer... ise... (if... else...) gibi bir koşul ifadesini anlayabilirim.	(1)	(2)	(3)	(4)	(5)
6) Mantıksal koşullar içeren bir programın nihai sonucunu tahmin edebilirim.	(1)	(2)	(3)	(4)	(5)
7) Girdi değerleri verildiğinde bir programın sonucunu tahmin edebilirim.	(1)	(2)	(3)	(4)	(5)
8) Bir programı test ederken karşılaştığım bir hatanın kaynağını bulabilirim.	(1)	(2)	(3)	(4)	(5)
9) Bir programı test ederken karşılaştığım bir hatayı düzeltebilirim.	(1)	(2)	(3)	(4)	(5)
10) Hata ayıklama (debugging) işlemi aracılığıyla programlama hakkında daha çok şey öğrenebilirim.	(1)	(2)	(3)	(4)	(5)
11) Bir örnek olmadan program prosedürlerini (işlemler, akış vb.) anlayabilirim.	(1)	(2)	(3)	(4)	(5)
12) Bir program oluşturmak için başkalarının yardımına ihtiyaç duymam.	(1)	(2)	(3)	(4)	(5)
13) Bir problemi çözmek için programlamadan faydalanabilirim.	(1)	(2)	(3)	(4)	(5)
14) Programlama işinin insanlar için alt görevlere bölünebileceğini bilirim.	(1)	(2)	(3)	(4)	(5)
15) Bir programı yazarken başkaları ile çalışabilirim.	(1)	(2)	(3)	(4)	(5)
16) Programlama verimliliğini artırmak için bölümlerden (modül, alt program vb.) faydalanabilirim.	(1)	(2)	(3)	(4)	(5)

EXTENDED ABSTRACT

1. INTRODUCTION

The importance of computer programming education has been increasingly emphasized in recent years in terms of having students gain technological literacy skills which are accepted as one of 21st-century learner competencies. There are many studies in the literature that highlight that computer programming enables the development of various cognitive skills. Also, programming is considered as an assistant tool that develops one's thinking and problem-solving skills at a high level (Özmen & Altun, 2014). Another skill is computational thinking (CT). CT is a basic skill that enables individuals to think and design by making use of fundamental programming concepts (Wing, 2006). CT will constitute a new literacy skill in the digital world of the future, because it teaches individuals how to process the building of information and solve problems. CT skills will be a necessity not only for today's programmers (who are engaged in computer science), but also for all future individuals (Tsai et al., 2019). There are various methods that focus on developing students' CT skills. Programming education is one of these methods and a popular choice. Self-efficacy – an important factor associated with programming education (Tsai et al., 2019) – is a psychological concept that can be used to evaluate the individual beliefs as well as psychological states of individuals under different conditions and their performances (Bandura, 1997). When evaluated in the context of computer literacy, students' self-efficacy is related to their learning performance in computer-based environments (Moos & Azevedo, 2009). Self-efficacy, which is also an important component of programming education (Ramalingam & Wiedenbeck, 1998), can positively predict students' performance in computer programming (Ramalingam et al, 2004).

Upon emphasizing the importance of computer programming in computer literacy education, there are various scales that have been developed to evaluate students' self-efficacy about programming (Altun & Kasalak, 2018; Kukul, Gökçearsan, & Günbatır, 2017; Ramalingam & Wiedenbeck, 1998). These scales have been adapted into different programming languages (Askar & Davenport, 2009; Govender & Basak, 2015) and into Turkish (Altun & Mazman, 2002; Korkmaz & Altun, 2014). An examination made on these scales reveals that they focus on the syntax structure of a particular programming language rather than on individual's general CT skills when it comes to problem-solving related to computer programming. While such scales may be seen as appropriate to assess the self-efficacy perceptions of future programming experts, they are not sufficient to assess CT skills which are considered to be a requirement for all students and are among 21st-century learner standards. Hence, Tsai et al., (2019) developed a general self-efficacy scale based on CT skills to examine students' self-efficacy about computer programming. In this study, the computer programming self-efficacy scale (CPSES), developed by Tsai et al. (2019), was adapted into Turkish and its validity and reliability analyses were performed. Also, self-efficacy scores obtained from the scale were examined in terms of various variables such as gender and programming experience.

2. METHOD

The sample consisted of 174 university students (69 females, 39.7% and 105 males, 60.3%). 47.1% of the students were studying at a vocational high school (N=82) and 52.9% of them were studying at a faculty (N=92). 46.6% of the participants (N=81) had one year or less of programming experience. 20.7% (N=36) one to three years, 23% (N=40) three to five years, 7.5% (N=13) five to seven years, and 2.3% (N=4) seven years or more.

The validity of the Turkish version of the scale was analyzed using factor analysis. EFA was applied to examine the construct validity of the scale. Here, it is worth noting that principal component analysis is used as a factorization technique to subtract the maximum variance from the data during the EFA and to reduce a large number of variables into a smaller number of components (Tabachnick & Fidell, 2007).

Cronbach's alpha (α) reliability coefficient and item-total correlations were calculated to determine the reliability of the scale and the levels of discrimination of each scale item.

3. FINDINGS, DISCUSSION, AND RESULTS

The convenience of the data and the principal components analysis were examined by applying a KMO coefficient and Bartlett's test. It was found that the KMO coefficient was .919 and Bartlett's test was significant ($\chi^2=1895.592$, $p=.000$). The EFA's results revealed a three-factor structure obtained that accounted for 68.853% of the total variance. These factors were programming competence (Factor 1), programming skill (Factor 2), and self-regulation (Factor 3). The programming competence factor consists of seven items that assess the students' perceptions when it comes to predicting structure and the results of the program by using a program editor. The programming skill factor consists of five items which measure the students' perceptions about program procedures as well as measure their ability to overcome programming problems. The self-regulation factor includes four items that reveal the students' perceptions about how they regulate their feelings, thoughts, and behaviors towards programming..

After the EFA, the CFA was applied to evaluate the construct validity of the three-factor model. The goodness of fit indices were calculated as follows: $\chi^2/df=2.132$, RMSEA=.081, SRMR=.0514, CFI=.941, GFI=.869, AGFI=.814, and NFI=.896. The standardized factor load values between the items and the factors vary between .544 and .883. The t values of the items were between 6.106 and 11.963 at a significance level of $p<.01$.

The Cronbach's alpha (α) reliability coefficient of all sixteen items was .936. The Cronbach's alpha (α) coefficients of the factors were found to be .919, .867, and .808 for the programming competence factor, the programming skill factor, and the self-regulation factor, respectively. The Item-total correlations of the items in the scale ranged between .430 and .771.

The data obtained from the students included in the sample group were also used to examine their self-efficacy levels for computer programming. The mean scores of the factors were observed to be similar to one another. The students obtained higher scores from the programming competence (3.82), self-regulation (3.70), and programming skills (3.16), respectively. Therefore, it can be said that the students had a high self-efficacy about computer programming. The Shapiro-Wilk test was used to examine the effects of gender on students' self-efficacy perception about computer programming. No significant difference was found between their self-regulation, programming skills, or programming competencies in terms of gender ($p > .05$). This study examined the correlation between students' programming experiences and self-efficacy perceptions about programming. The students were grouped as low level (N=81, 46.6%) for one year or less of programming experience, medium level (N=76, 43.7%) for one to five years of experience, and high level (N=17, 9.8%) for five years or more of experience. The correlation between their programming experience and computer programming self-efficacy was analyzed using Kruskal-Wallis test. The more programming experience students had, the higher their computer programming self-efficacy perceptions were – in the context of programming competence, programming skills, and self-regulation. Students, who have been involved in programming for a long time, feel more competent about it. Consequently, it can be asserted that there was a significant and positive correlation between the students' programming experience and self-efficacy perception about programming.

ARAŞTIRMANIN ETİK İZİNİ

Bu çalışmada “Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesi” kapsamında uyulması gerektiği belirtilen tüm kurallara uyulmuştur. Yönergenin ikinci bölümü olan “Bilimsel Araştırma ve Yayın Etiğine Aykırı Eylemler” başlığı altında belirtilen eylemlerden hiçbiri gerçekleştirilmemiştir.

ARAŞTIRMACILARIN KATKI ORANI

Araştırma tek yazarlı olduğu için yazarın katkı oranı %100'dür.

ÇATIŞMA BEYANI

Araştırmada herhangi bir kişi ya da kurum ile finansal ya da kişisel yönden bağlantı yoktur. Araştırmada herhangi bir potansiyel çıkar çatışması bulunmamaktadır.