



DOI: 10.18039/ajesi.725161

Bilgisayar Programlama Öz-Yeterlik Ölçeğinin Türkçe Formunun Geçerlik ve Güvenirlik Çalışması

Murat EKİCİ¹, Murat ÇINAR²

Geliş Tarihi: 22.04.2020

Kabul Tarihi: 16.06.2020

Makale Türü³: Araştırma Makalesi⁴

Öz

Bu araştırmanın amacı, Tsai, Wang ve Hsu (2019) tarafından geliştirilen Bilgisayar Programlama Öz-yeterlik Ölçeğinin Türkçe diline uyarlanması, geçerlik ve güvenilirlik sonuçlarının ortaya konulmasıdır. Araştırma, 297 lise ve üniversite öğrencisi ile yürütülmüştür. Orijinal ölçek 16 madde ve beş faktörden (Mantıksal Düşünme, İşbirliği, Algoritma, Kontrol ve Hata Ayıklama) oluşmaktadır. Ölçeğin yapı geçerliğinin test edilmesi için birinci ve ikinci düzey doğrulayıcı faktör analizleri yapılmıştır. Ayrıca yakınsak ve iraksak geçerlik çalışmaları yürütülmüştür. Güvenirlik analizi sonucunda iç tutarlılık, test yarılama ve bileşik güvenirlik katsayılarının kabul edilebilir düzeyde olduğu görülmüştür. Alt faktörlere ve ölçeğin geneline ilişkin Cronbach Alfa iç tutarlılık katsayıları: Mantıksal düşünme için .877, İşbirliği için .813, Algoritma için .775, Kontrol için .906, Hata ayıklama için .812 ve ölçeğin geneli için .911 olarak bulunmuştur. Madde analizi sonucunda düzeltilmiş madde toplam korelasyonlarının .41 ile .63 arasında değiştiği, %27'lik alt ve üst gruplar arasındaki ortalama farklarının anlamlı olduğu görülmüştür. Sonuç olarak ölçeğin Türkçe formunun lise ve üstü öğrenim kademesindeki öğrencilerin bilgisayar programlama öz-yeterliklerinin ölçülmesi amacıyla kullanılabilir geçerli ve güvenilir bir ölçme aracı olduğu görülmüştür.

Anahtar kelimeler: bilgisayar programlama, öz-yeterlik, bilgisayarlı düşünme

Atf: Ekici, M. ve Çınar, M. (2020). Bilgisayar programlama öz-yeterlik ölçeğinin Türkçe formunun geçerlik ve güvenilirlik çalışması. *Anadolu Journal of Educational Sciences International*, 10(2), 1017-1040. DOI: 10.18039/ajesi.725161

¹ (Sorumlu Yazar) Dr., Uşak Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü, Uşak, Türkiye, murat.ekici@usak.edu.tr, <https://orcid.org/0000-0003-2189-7294>

² Dr., Milli Eğitim Bakanlığı, Borsa İstanbul Mesleki Teknik ve Anadolu Lisesi, Adana, Türkiye, murat_cinar@rocketmail.com, <https://orcid.org/0000-0003-4012-4174>

³ Bu çalışma Uşak Üniversitesi, Sosyal ve Beşerî Bilimler Bilimsel Araştırma ve Yayın Etiği Kurulunun 13.04.2020 tarih ve 04 sayılı Etik Kurul Onayı alınarak gerçekleştirilmiştir.

⁴ Yazarlar çalışmaya eşit derecede katkı sunmuştur.



DOI: 10.18039/ajesi.725161

The Validity and Reliability Study of the Turkish Version of Computer Programming Self-Efficacy Scale

Murat EKİCİ¹, Murat ÇINAR²

Submitted by: 22.04.2020

Accepted by: 16.06.2020

Article type³: Research Paper⁴

Abstract

The purpose of this research is to adapt the Computer Programming Self-efficacy Scale developed by Tsai, Wang and Hsu (2019) to the Turkish language and to present the validity and reliability results. The research was carried out with 297 high school and university students. The original scale consists of 16 items and five factors (Logical Thinking, Cooperation, Algorithm, Control and Debugging). First and second order confirmatory factor analysis were performed to test the construct validity of the scale. Convergent and divergent validity studies were also conducted. It was determined that the internal consistency, split-half and composite reliability coefficients were within acceptable limits. Cronbach's alpha internal consistency coefficients for the sub-factors and overall scale were; .877 for Logical Thinking, .813 for Cooperation, .775 for Algorithm, .906 for Control, .812 for Debug, and .911 for overall scale. Item analysis showed that the corrected item total correlations ranged between .41 and .63 and the mean differences between top and bottom of 27% groups were significant. As a result, it has been determined that the Turkish form of the scale is a valid and reliable instrument that can be used to measure the computer programming self-efficacy of students at high school and higher levels.

Keywords: computer programming, self-efficacy, computational thinking

Cite: Ekici, M. and Çınar, M. (2020). The validity and reliability study of the Turkish version of computer programming self-efficacy scale. *Anadolu Journal of Educational Sciences International*, 10(2), 1017-1040. DOI: 10.18039/ajesi.725161

¹ (Corresponding Author) Dr., Usak University, Education Faculty, Department of Computer Education and Instructional Technologies, Turkey, murat.ekici@usak.edu.tr, <https://orcid.org/0000-0003-2189-7294>

² Dr., Ministry of National Education, Borsa İstanbul Vocational and Technical Anatolian High School, Adana, Turkey, murat_cinar@rocketmail.com, <https://orcid.org/0000-0003-4012-4174>

³ This research study was conducted with Research Ethics Committee approval of Uşak University, dated 14.04.2020 and numbered 04.

⁴ Researchers contributed equally to the study.

Giriş

Çağımızda bireysel ve toplumsal değişimlere yön veren gelişmelerin çoğunlukla teknolojik kaynaklı olduğu görülmektedir. Teknoloji, günlük ve mesleki yaşamda işlerin yapılma biçimini (örn. ileti gönderme, alışveriş yapma, eğitim alma vb.) değiştirmektedir. Özellikle dördüncü sanayi devriminin sadece ekonomik alanda değil, aynı zamanda sosyal yaşam ve eğitimde büyük dönüşümleri tetiklemesi beklenmektedir. Bu süreçte giderek önem kazanan otomasyon, makine öğrenmesi, nesnelerin interneti ve büyük veri analizi gibi dijital yetkinlikler ise yeni düşünme biçimlerinin kullanılmasını teşvik etmektedir (Diaz ve Silvain, 2020). Başta İnternet olmak üzere bilgi kaynaklarında dolaşan veri miktarının baş döndürücü bir hızda arttığı günümüz dünyasında veri işleme ve karar verme sistemlerindeki gelişim yerel ve küresel sorunların çözümü için bizi cesaretlendirmektedir. Örneğin, suç istatistiklerinin analizi güvenlik güçlerinin olası suç mahallerine ve zamanlarına odaklanarak kaynaklarını etkili bir biçimde kullanmasına yardımcı olabilir. Uygun sensör ve lens sistemleriyle donatılan bir cep telefonu uyku düzensizlikleri, kardiyovasküler aktivite bozuklukları, solunum sistemi rahatsızlıkları, göz sorunları, cilt hastalıkları ve mental bozuklukların tespiti ve takibi için uygun maliyetli, güvenilir ve pratik bir platform haline gelebilir (Majumder ve Deen, 2019). Bu değişen çevrede başarılı olmak için bireylerden problem çözen, farklı çözümler hayal eden, donanım veya yazılım geliştiren yani teknoloji ile üreten kişiler olması beklenmektedir. Eğitim kurumlarından ise öncelikli ihtiyaç alanlarına uygun olarak toplumsal değişime yön vermek adına öğrencilere bir yol haritası sunması beklenmektedir.

Günümüzde Bilgi ve İletişim Teknolojilerini (BİT) etkin ve ihtiyaçlara cevap verecek şekilde kullanma gereksiniminin yerini bu araçları ve sayısal kavramları kullanarak problemlere çözüm üretme yetkinliğine bırakmaya başlamıştır. Türkiye de dâhil olmak üzere birçok ülkede bilişim eğitimi müfredatları bilgisayarlı becerileri de kapsayacak şekilde yeniden düzenlenmeye başlamıştır (Bocconi, Chiocciariello, Dettori, Ferrari, ve Engelhardt, 2016; Jun, Han, Kim ve Lee, 2014; Yadav, Good, Voogt, ve Fisser, 2017). Bilgisayarlı okuryazarlık bilgisayar bilimleri kavramlarına dayalı olarak problemleri analiz etmek ve çözmek için ihtiyaç duyulan bir düşünme setini kavrama ve kullanma olarak tanımlanabilir. Bilgisayarlı okuryazarlığının altında yatan temel düşünce formu ise temelleri 1960'lı yıllara kadar (Logo programlama) uzanan bilgisayarlı düşünme (computational thinking) becerisidir (Feurzeig ve Papert, 2011). Papert (1980) bilgisayarların güçlü fikirlerin ve kültürel değişim tohumlarının taşıyıcısı olabileceğini, farklı disiplinleri (örneğin fen bilimleri ile sosyal bilimleri) ayıran geleneksel çizgilerin ötesinde geçerek bilgiyle yeni ilişkiler kurulmasına yardımcı olabileceğini ileri sürmüştür. Papert, bilgisayarların bilgiye erişim kalıplarını değiştirmesinin ötesinde, fiziksel varlığı olmasa dahi, düşünme süreçlerine kavramsal açıdan katkı sağlayabileceğini belirterek kendi perspektifini ortaya koymuştur (Çınar, 2019). Öğrencilerin entelektüel düşünme süreçlerinin programlama yoluyla desteklenmesi fikri vizyoner olmasına rağmen önerildiği dönemde geniş ölçüde rağbet görmemiştir. Bilgisayar programlama eğitimi literatüründen kalan köklü mirasa sahip olan bilgisayarlı düşünme (BD) ile ilgili çalışmalar ve eğitsel girişimler Wing (2006, 2008)'in çalışmalarıyla büyük bir ivme yakalamıştır. Bu ivmelenmenin altında yatan dinamik ise bilgisayar bilimi alanından türetilen bilgi ve becerilerin tüm öğrencilere faydalı olabilecek şekilde geniş yelpazedeki problemlere ve uygulamalara uygulanabileceği fikridir. Wing (2006) başlangıçta BD'nin hemen her alandaki problemlere uygulanabilecek şekilde evrensel bir bilgi, beceri ve tutum seti olduğunu belirten oldukça hipotetik (genel) bir tanımlama yapmasına rağmen, bu tanımlamasını daha sonra fikirlerin bilgi-işleme cihazları tarafından yorumlanıp, yürütülebilecek bir formda ifade edilmesi şeklinde daha operasyonel hale getirmiştir (Wing, 2008). Benzer şekilde, Uluslararası Eğitim Teknolojileri Birliği (ISTE) ve Bilgisayar Bilimi Öğretmenleri Derneği (CSTA) bilgisayar bilimi alanındaki öğretmen,

araştırmacı ve uygulayıcılardan (yaklaşık 700 kişi) anket yoluyla aldıkları geri bildirimlerden yola çıkarak BD'nin operasyonel tanımı için bir çerçeve ortaya koymuştur (ISTE ve CSTA, 2011). Buna göre BD, aşağıdaki özellikleri içeren ancak bunlarla sınırlı olmayan bir problem çözme sürecidir.

- Sorunları, çözümlerinde bilgisayar ve diğer bilgi işleme araçlarının kullanılabilmesini sağlayacak biçimde formüle etme,
- Verileri mantıksal olarak düzenleme ve analiz etme,
- Verileri model ve simülasyon gibi soyutlamalar yoluyla temsil etme,
- Çözümleri algoritmik düşünce yoluyla bir dizi sıralı adımda otomatikleştirme,
- Çözüm aşamalarının (adımlarının) ve kaynakların en etkili ve verimli kombinasyonunu sağlamak amacıyla olası çözümleri belirleme, analiz etme ve uygulama,
- Problem çözme sürecini genelleme ve farklı problemlere uygulama

Bu yeterlilikler, BD'nin temel boyutları olan karmaşıklıkla başa çıkma konusunda güven, zor problemle uğraşmada kararlı olma, belirsizlik karşısında tolerans, açık uçlu problemlerle baş etme, ortak bir hedef veya çözüm elde etmek için başkalarıyla iletişim kurma ve onlarla çalışabilme gibi bir dizi tutum ve eğilimlerle geliştirilebilmektedir.

BD üzerinde uzlaşılan genel bir tanımlama bulunmamaktadır (Çınar, 2019; Grover ve Pea, 2013; Haseski, Ilic ve Tuğtekin, 2018; Leonard vd., 2016). Alanyazındaki BD'ye ilişkin tanımlamalarda önceleri (2006 öncesi) düşünme kavramı ön plana çıkarken; günümüzde (2006 sonrası) problem çözme ve teknoloji ifadelerine daha çok vurgu yapıldığı görülmektedir (Haseski, Ilic, ve Tuğtekin, 2018). BD'nin çıkış noktası programcı gibi düşünme ve bu düşünme yaklaşımını bilgisayar bilimleri dışındaki problemlerin çözümü için de kullanmadır. Programcı gibi düşünme kullanıcı bakış açılarını anlamayı ve onların ihtiyaçlarını en iyi şekilde karşılamayı gerektirir. BD'ye yönelik ilk araştırmalar yoğun olarak BD'nin ne olduğuna odaklansa da son zamanlarda BD bileşenlerinin neler olduğu, bunların hangi yollarla ve nasıl destekleneceğine ağırlık verildiği görülmektedir.

Okullarda bilgisayarlı girişimlerde ortaya çıkan pratiklerin çeşitliliği (programlama/kodlama, robotik, bilgisayarlı problem çözme, oyun tasarımı/oyunlaştırma, dijital öyküleme vb.) günden güne artmaktadır. Özellikle programlama BD becerilerinin otantik olarak işe koşulduğu bir etkinliktir. Bu nedenle BD becerisinin geliştirilmesinde en sık kullanılan etkinlik programlamadır. Öte yandan, araştırmacıları BD becerilerinin hangi alt bileşenlerden oluştuğu ve BD gelişiminin nasıl ölçüleceği ya da değerlendirileceği tarzında sıkı bir dizi soru beklemektedir. Temel bir analitik yetenek olarak görülen BD için değerlendirme kritik bir faktördür. Bununla birlikte, alanyazında BD becerisinin geliştirilmesine yönelik değerlendirme yaklaşımlarının eksikliği göze çarpmaktadır (Zhong, Wang, Chen ve Li, 2016). BD'ye yönelik mevcut değerlendirme yaklaşımlarının çoğu, belirli bir programlama platformuna/aracına yönelik olarak verilen eğitim ya da gerçekleştirilen etkinlikler sonrasında öğrenci ürünlerine ya da sözdizimsel kurallara sıkı ilişkili olan test (başarı testi) performanslarına odaklanmaktadır. Bu durum, BD müfredatlarına ilişkin genel kazanımların değerlendirilmesinde kullanılabilecek bir ölçme yaklaşımını sınırlamaktadır.

Bir hedefe ulaşmak için, kendi duygu, düşünce ve davranışlarının farkında olma ve bunları yönlendirme süreci olan öz-düzenleme becerisi geliştirilebilir ve yönlendirilebilir bir beceridir (Zimmerman, Bonner ve Kovach, 1996). Öz-düzenleme becerisi gelişmiş öğrenenler öğrenme hedeflerini belirleme, öğrenme sürecinde etkin rol oynama ve bu süreci kontrol edebilme konusunda daha isteklidir (Bandura, 1986; Zimmerman, 1990a). Öz-düzenleme sürecinin sacayaklarından birisi de

şüphesiz bireyin belirli bir görevi yerine getirme konusundaki kapasitesine olan inancıdır (Li ve Zheng, 2018; Zimmerman, 2011). Bu bağlamda bireyin kendini tanıması için yapabilecekleri konusunda oluşturduğu yeterlik algısının (öz-yeterlik) ne olduğunun farkına varması üst-bilişsel gelişimi için oldukça önemlidir. Bandura'ya (1993) göre öğrencilerin kendi öğrenmelerini düzenleme ve akademik etkinliklerdeki hakimiyetleri konusundaki inançları, motivasyon ve akademik başarılarının önemli bir belirleyicisidir. Öz-yeterlikten bireyin verilen eylemleri yerine getirme kabiliyetine ilişkin kararları olarak bahseden Schunk (1991) bunun akademik motivasyon çıktılarının yordanmasında kullanılması gerektiğini belirtmiştir. Bu çalışmada bilgisayar programlama ve diğer BD etkinliklerinde kullanılmak üzere öğrenenlerin öz-yeterliklerinin belirlenmesine yönelik olarak geliştirilen bir ölçme aracının Türkçeye uyarlanması konu alınmaktadır.

Öz-yeterlik

Öz-yeterlik, “bireyin belirlenen hedeflere ulaşma ya da bir görevi yerine getirme konusunda kendisine olan inancı” olarak tanımlanmaktadır (Bandura, 1995). Öz-yeterlik, kişinin ne yapacağını bilmesinden farklı olarak beceri ve kapasitesi ışığında neyi yapmaya yeterli olup olamayacağını düşünmesidir (Schunk, 2012). Bir nevi bireyin yeteneklerine bağlı algılamalarıyla şekillenen öz-yeterlik, bu yönüyle bireyin çevresiyle olan ilişkisi ve buna bağlı yorumlamaları sonucunda farklı alanlardaki etkinliğine yönelik kollektif kavrayışlarından oluşan benlik algısından farklılaşmaktadır. Algılanan öz-yeterlik bireyin davranışlarının belirlenmesinde etkin bir rol oynamaktadır (Morrell ve Carroll, 2003; Yi ve Hwang, 2003). Öz-yeterliğin öğrenme süreçlerinde görev/kariyer tercihi, motivasyon, öğrenme için harcanan çaba, bağlılık ve başarı üzerinde etkisi bulunmaktadır (Pajares, 1996; Bandura, 1993). Ek olarak, araştırmalar öz-yeterliği yüksek olan öğrencilerin üst bilişsel stratejileri (öz-düzenleme gibi) kullanmaya daha yatkın olduğunu raporlamaktadır (Aydın, 2015; Pellas, 2014). Sosyo-bilişsel perspektiften bakıldığında öğrenme çabalarının öz-düzenlemeye dayalı olması için öğrenenin akademik amaçları doğrultusunda kendi yeterliklerine uygun stratejileri seçmesi ve uygulaması gerekmektedir (Zimmerman, 1989, 1990b). Sonuçları etkileme yeteneğine olan inanç aynı zamanda onların öngörülebilir olmasını sağlar. Amaçlı davranışların büyük bölümü bilişsel hedeflerin daha görünür olmasını sağlayan öngörü aracılığıyla düzenlenir (Bandura, 1993). Bu yönüyle öngörülebilirlik öğrenmeye hazırlıklı olmayı da kolaylaştırır (Bandura, 1995). Bireyin yeteneklerine olan güveni aynı zamanda duygu, düşünce ve eylemlerini kontrol etme yeteneğine olumlu yansımakta, bu yolla öğrenme müdahalelerinin beklenen (istenen) öğrenme sonuçlarını nasıl etkileyebileceğini anlama becerisini desteklemektedir.

Öz-yeterliğin en önemli eğitsel faydalarından birisi de öğrenenlere kendilerini dolayısıyla öğrenme süreçlerini değerlendirme fırsatlarını sağlamasıdır. Üst bilişsel nitelikli bu bağlantı sadece bireysel değil aynı zamanda sosyal/çevresel faktörlerin de içselleştirilmesini, dolayısıyla başarı için bu faktörlerin düzenleme gereksinimi (öz-etkileri) beraberinde getirmektedir. Öz-yeterlik inancının değerlendirilmesi öğretim elemanlarına öğretimsel koşulların ve stratejilerin yeniden organize edilmesi için önemli geri bildirimler sağlayabilir. Öğrenme etkinliklerini takiben öğrencilerin gelişme algısının desteklenmesi sonraki öğrenmeler için önemli pedagojik destekler (motivasyon gibi) sağlayabilir (Schunk, 2012). Sonuç olarak öz-yeterliğe ilişkin çıktılar eğitim bağlamında oldukça kullanışlıdır.

Öz-yeterlik algısının saptanması öğrenci başarılarının yanı sıra gerçekleştirilen öğretimsel uygulamaların etkililiği hakkında da önemli bilgiler sağlayabilir. Bilindiği üzere programlama öğrenciler için zorlayıcı bir aktivite olup, programlama dersleri özellikle yüksek bırakma (drop-out) oranlarıyla dikkat çekmektedir (Noone ve Mooney, 2018). Öğrencilerin özellikle ilk yıllarda BD'ye ve

programlamaya ilişkin temel kavramlar üzerinde hâkimiyet kurma mücadelelerinin bu durumun ortaya çıkmasında önemli rolü bulunmaktadır. Programlamaya giriş derslerindeki deneyimleri, öğrencilerin bilgisayar bilimlerine ilişkin ilgilerini ve bu alandaki kariyer seçimlerini büyük ölçüde etkilemektedir. Yapılan çalışmalar bilgisayar bilimleri alanının ilk yıllarında düşük öz-yeterlik algısına sahip öğrencilerin sonraki yıllarda öğrenimlerini sürdürme noktasında özellikle risk altında olduklarını raporlamaktadır (Bhardwaj, 2017). Düşük öz-yeterlik algısı, programlama deneyimi ve ön bilgi düzeyine sahip öğrenciler BD etkinliklerinde daha fazla hatayla karşılaşmakta ve hata ayıklama/giderme sürecinde daha fazla güçlük çekmektedir (Hawlitshchek, Koppen, Dietrich ve Zug, 2019). Hata giderme girişimindeki başarısızlıklar ise programlama sürecinin kaotik bir hal almasına ve öğrencilerin dersi bırakmalarına yol açabilmektedir.

Alanyazında programlama dâhil olmak üzere BD etkinliklerinde öz-yeterlik ile öğrenci başarısı arasındaki ilişkiye vurgu yapan çok sayıda çalışma bulunmaktadır (Gurer, Cetin ve Top, 2019; Korkmaz, Kılıç, Çakır ve Erdoğan, 2019; Mazman Akar ve Altun, 2017). Öz-düzenleyici öğrenmeye ilişkin stratejiler arasında programlama başarısı etkileyen en önemli yapının öz-yeterlik olduğu raporlanmıştır (Haslam ve Aşkar, 2007). Öz-yeterliğin öğrenmenin duyuşsal boyutu üzerindeki etkisi de göz ardı edilmemelidir. Yapılan çalışmalar öz-yeterliğin programlamaya karşı tutum üzerinde de yüksek düzeyde yordayıcı etkisi olduğunu göstermektedir (Gurer, Cetin ve Top, 2019). Benzer şekilde diğer bir çalışmada bireylerin kodlamaya yönelik öz-yeterlik algıları yükseldikçe programlama öğrenimine yönelik olumlu tutumlarının arttığı raporlanmıştır (Korkmaz, Şahin, Çakır ve Erdoğan, 2019).

Alanyazın incelendiğinde programlamaya ilişkin öz-yeterlik ölçek geliştirme ve uyarlama çabalarının bulunduğu görülmektedir (Altun ve Mazman, 2012; Altun ve Kasalak, 2018; Aşkar ve Davenport, 2009; Govender ve Basak, 2015; Ramalingam ve Wiedenbeck, 1998). Bununla birlikte, bu araçların genellikle programlama modalitesine (Altun ve Kasalak, 2018), programlama aracına/diline (Aşkar ve Davenport, 2009; Govender ve Basak, 2015; Ramalingam ve Wiedenbeck, 1998) ve müfredat çıktılarına (Kukul, Gökçeşlan, ve Günbatır, 2017) göre özelleştiği görülmektedir. Ayrıca ölçeklerin önemli bir bölümü orijinali Ramalingam ve Wiedenbeck (1998) tarafından geliştirilen ve dört faktörlü (özerklik-kararlılık, öz-düzenleme, karmaşık ve basit programlama görevlerini yerine getirme) bir yapıya sahip olan programlama öz-yeterlilik ölçeğinden uyarlandığı belirlenmiştir. Bu ölçeğin Türkçe formları basit ve karmaşık olmak üzere programlama görev zorluğu üzerine temellendirilen 2 faktörlü bir yapı üzerine kuruludur. BİT ve bilgisayar bilimleri alanında yaşanan gelişmelerin peşi sıra BD ilişkin beceri ve tutum setinde 20 yılı aşkın bir sürede gerçekleşen eğilimler göz önüne alındığında öz-yeterlilik ölçme araçlarının revize edilme gereksinimi ortaya çıkmaktadır. Dahası sadece müfredat hedefleri ya da görev türlerine/zorluğuna dayalı ölçümlerin öğrencilerin programlama dâhil olmak üzere BD'ye yönelik öz-yeterlilik inançlarını temsil yeteneği de tartışmaya değer bir diğer husustur.

Tüm dünyada olduğu gibi ülkemizde de öğrencilere bilgisayarlı bakış açılarının kazandırılmasını amacıyla revizyonu sürdürülen BİT müfredatı kapsamında yeni değerlendirme araçlarına ihtiyaç bulunmaktadır. Dünyada geliştirilen değerlendirme araçlarının Türk diline kazandırılması ve yeni araçlar geliştirilmesi bu ihtiyacın karşılanmasına yönelik çabalardır. Ölçme araçlarının başka dillerde ve kültürlerde kullanılması için sadece çeviri yapılması tek başına yeterli değildir, bu araçların sahip oldukları yapılarının da eşitliğinin sağlanması gerekir. Bu sebeple bu tür çalışmalara “uyarlama” çalışması adı verilmektedir (Hambleton, Merenda ve Spielberger, 2004). Bu kapsamda mevcut çalışmada Tsai, Wang, ve Hsu (2019) tarafından geliştirilen bilgisayar programlama öz-yeterlilik ölçeğinin Türkçeye uyarlama çalışması yapılmıştır.

Yöntem

Çalışma Grubu

Araştırma 2019-2020 Bahar döneminde farklı üniversite ve liselerde öğrenim gören 311 öğrencinin katılımıyla gerçekleştirilmiştir. Buna karşın, 14 katılımcıdan elde edilen veriler uç değer analizleri sonrası veri setinden çıkarılmıştır. Böylece çalışma grubu nihai olarak 297 öğrenciden oluşmuştur. Katılımcıların %58,6'sı (n=174) erkek; %41,4'ü (n=123) kadınlardan oluşmaktadır. Katılımcıların yaşları 16 ile 38 yaş aralığında olup yaş ortalaması 21,6'dır. Katılımcılarının %13,8'i lise; %86,2'si üniversite öğrencilerinden oluşmaktadır. Lise düzeyindeki katılımcıların tamamı bilişim teknolojileri bölümü öğrencisiyken üniversite düzeyindeki katılımcıların hemen hepsi başta Bilgisayar ve Öğretim Teknolojileri Öğretmenliği bölümü olmak üzere Bilgisayar Mühendisliği ve Yönetimi Bilişim Sistemleri gibi bilgisayar bilimleri ile ilgili farklı bölümlerde öğrenim görmektedirler (Tablo 1).

Tablo 1: Katılımcıların Öğrenim Gördükleri Okul Türü ve Bölüme Göre Dağılımı

Okul Türü	Bölüm	Frekans	Yüzde
Lise	Bilgisayar ve Öğretim Teknolojileri Eğitimi	208	70.0
Üniversite	Bilgi ve Belge Yönetimi	1	.3
	Bilgisayar Mühendisliği	21	7.1
	Bilgisayar Öğretmenliği	4	1.3
	Bilgisayar Programcılığı	1	.3
	Bilişim Teknolojileri	41	13,8
	Elektrik Elektronik Mühendisliği	1	.3
	İstatistik ve Bilgisayar Bilimleri	2	.7
	Yazılım Mühendisliği	2	.7
	Yönetim Bilişim Sistemleri	16	5.4
	Toplam		297

Katılımcıların %42'si (n=125) 10 yıl üzeri, %34'i (n=100) 7 ile 10 yıl arası, %18'i (n=54) 4 ile 6 yıl arası, %5'i (n=16) 1 ile 3 yıl arası ve geriye kalan %1'i (n=2) 1 yıldan az bilgisayar kullanım deneyimine sahiptir. Lise öğrenimini sürdüren katılımcılardan %50'si (n=20) 12. sınıf, %48,5'i (n=19) 11. sınıf ve geriye kalan %2,5'i 10. sınıfta; lisans öğrencisi katılımcılardan %50'si (n=127) 4. sınıfta, %38'i (n=97) 3. sınıfta, %10'u (n=25) 2. sınıfta ve geriye kalan %3'ü (n=7) 1. sınıfta öğrenimini sürdürmektedir. Çalışmaya katılan bir ön lisans öğrencisi ise 2. sınıfta öğrenim görmektedir.

Veri Toplama Araçları

Bilgisayar Programlama Öz-yeterlik Ölçeği

Bireyin okuryazarlık düzeyinde bilgisayar programlama yeteneğinin öz-raporlamaya dayalı olarak değerlendirmesi amacıyla geliştirilen bilgisayar programlama öz-yeterlik ölçeği Likert tipi 16

maddeden oluşan bir ölçme aracıdır. Maddelere verilecek yanıtlar 1 (hiç katılmıyorum) ile 6 arasında (tamamen katılıyorum) arasında değişmektedir. Tsai, Wang ve Hsu (2019) tarafından hazırlanan ölçeğin geliştirme sürecinde Berland ve Lee (2011) tarafından önerilen BD çerçevesinden yararlanılmıştır. Bir programlama görevini başarıyla yerine getirebilmek gerekli bileşenleri adresleyen bu çerçeve şartlı (koşullu) mantık, algoritma geliştirme, hata ayıklama/giderme, simülasyon ve dağıtık bilgisayarım (computation) bileşenlerinden oluşmaktadır. Araştırmacılar, ölçek geliştirme sürecinde bu bileşenlere ek olarak öğrenenin bir programlama ortamında gerçekleştirilebilecek işlemler üzerindeki hâkimiyeti gösteren kontrol bileşenini de dikkate almıştır. Yirmi dört maddeden oluşan ölçeğin ön formu farklı alanlarda öğrenim gören 104 üniversite öğrencisine uygulanmıştır. Faktör analizi sonucunda ölçek, mantıksal düşünme (4 madde), işbirliği (3 madde), algoritma (3 madde), kontrol (3 madde) ve hata giderme (3 madde) olmak üzere beş alt ölçek (faktör) etrafında toplanan 16 maddelik nihai formuna kavuşmuştur. Alt ölçekler düzeyinde 0,84-0,96 arasında değişmek üzere ölçeğin Cronbach alfa güvenilirlik katsayısı 0,96 olarak raporlanmıştır. Ölçeğin toplam açıklanan varyansı ise %84'tür. Faktörler arası korelasyon katsayıları 0,49-0,75 arasında değişmektedir. Ölçekten alınabilecek en düşük puan 16; en yüksek puan ise 96'dır. Alt ölçekler düzeyinde madde ortalama puanının 3,5 üzeri olması ilgili bileşene ilişkin yüksek yetkinlik algısına işaret etmektedir. Ölçeğin ortaokul üzeri eğitim kademelerinde bulunan öğrencilerin programlama deneyimlerinin değerlendirilmesi için uygun olduğu belirtilmiştir (Tsai, Wang, ve Hsu, 2019).

Veri Analizi

Verilerin analizinde öncelikle, normallik testleri (Q-Q ve P-P grafikleri) ve uç değerlerin belirlenmesi işlemleri yapılmıştır. Uç değer analizinde ölçek toplam puanı hesaplanmış, +3 ve -3 standart sapma aralığı dışında kalan gözlemler tespit edilerek analiz dışında bırakılmıştır (Alpar, 2011). Çok değişkenli uç değerlerin tespit edilmesi için Mahalanobis uzaklıklarına bakılmış ve .001 altında kalan iki gözlem analiz dışında bırakılmıştır. Grafik testleri incelendiğinde verilerin normal dağılıma uygun olarak diyagonal bir hat boyunca doğrusal olarak ilerlediği görülmüştür. Gözlenen değişkenlerin dağılımının normalliğinin sınanması için her değişkenin basıklık ve çarpıklık değerlerine bakılmış ve standartlaştırılmış basıklık ya da çarpıklık değerlerinin 3.75'ten küçük olduğu görülmüştür (Tabachnick ve Fidell, 2012). Ayrıca ölçek toplam puanı dağılımında basıklık katsayısı -.437 ve çarpıklık katsayısı -.235 olarak hesaplanmıştır. Basıklık ve çarpıklık katsayılarının +1 ile -1 aralığında olması sebebiyle verilerin normal dağıldığı söylenilebilir (Hair, Black, Babin ve Anderson, 2019). Ek olarak ölçek toplam puanları Z puanlarına dönüştürülmüş ve örneklem büyüklüğü göz önünde bulundurulduğunda puanların 3.29 altında kaldığı gözlenmiştir (Mayers, 2013; West, Finch ve Curran, 1995). Normallik varsayımın sağlanmasını takiben ölçeğin Türkçe formunun yapı geçerliği için birinci ve ikinci düzey doğrulayıcı faktör analizleri, yakınsak ve ıraksak geçerlik hesaplamaları gerçekleştirilmiştir. Güvenirlik analizleri için Cronbach alfa, McDonald omega ve bileşik güvenilirlik katsayılarından yararlanılmıştır. Ayrıca test yarılama yönteminden elde edilen korelasyon katsayısı incelenmiştir. Verilerin analizinde IBM SPSS 23 ve Lisrel 8.71 programları kullanılmıştır.

Ölçeğin Türkçeye Çeviri Çalışmaları

Ölçeğin Türkçeye uyarlanması sürecinde öncelikle Men-Jung Tsai ile elektronik posta yoluyla iletişime geçilerek ölçeğin uyarlanması için gerekli izin alınmıştır. Daha sonra araştırmanın etik ilkelere uygunluğunun değerlendirilmesi için Uşak Üniversitesi Sosyal ve Beşeri Bilimler Bilimsel Araştırma

ve Yayın Etiği başvurulmuş ve 13.04.2020 tarihinde 04 sayılı etik kurul izni alınmıştır. Ölçeklerin kaynak dillerden hedef dillere uyarlanmasında yargılayıcı ve istatistiksel ileri ya da geri çevirme yöntemleri kullanılmaktadır. Yargılayıcı ileri çeviri yöntemi, ölçek maddelerinin bir çevirmen ya da bir grup tarafından kaynak dilden hedef dile çevrilmesi ve başka bir grup tarafından dilsel eşdeğerliklerinin değerlendirilmesine dayanmaktadır. Yargılayıcı geri çevirme yönteminde ise kaynak dilden hedef dile çeviri yapıldıktan sonra elde edilen form başka bir grup çevirici tarafından hedef dilden tekrar kaynak dile çevrilir. Daha sonra elde edilen form ile orijinal formun dilsel eşdeğerliği test edilir (Hambleton, Merenda ve Spielberger, 2004). Bu araştırmada yargılayıcı ileri çeviri yöntemi kullanılmıştır. Orijinal ölçek öğretim teknolojileri alanından iki uzman ve bir yabancı dil uzmanı tarafından Türkçeye çevrilerek 3 farklı çeviri formu elde edilmiştir. Her bir çeviri formu ölçek maddelerine göre bir araya getirilerek uzman görüşü formu hazırlanmıştır. Uzman görüşü formunda her bir madde çevirisinin İngilizce-Türkçe uyumluluğu açısından değerlendirilmesi sağlanmıştır. Çeviri maddelerinin dil ve ifade yönünden Türkçeye uygunluğu hakkındaki uzman görüşleri 3'lü derecelendirme ölçeği (0-Uygun değil, 1-Kısmen uygun, 2-Uygun) kullanılarak alınmıştır. Uzman görüşü formuna her maddeye ilişkin en uygun çevirinin seçilmesi için bir madde eklenmiştir. Ek olarak, uzmanların madde çevirilerine ilişkin görüş ve önerilerinin alınması için forma açık uçlu bir madde de eklenmiştir. Uzman görüşü formu beş farklı devlet üniversitesinden BÖTE alanı uzmanı toplam 9 kişiye uygulanmıştır. Uzmanların 6'sı (%67) doktora, 3'ü (%33) yüksek lisans derecesine sahiptir. Cinsiyet özellikleri açısından incelendiğinde 6'sı erkek (%67), 3'ü kadın (%33) olan uzmanların hepsi 10 yıl ve üzeri alan deneyimine sahiptir. Uzman görüşleri doğrultusunda derecelendirme puanı en yüksek maddeler bir araya getirilerek ölçeğin Türkçe formu hazırlanmıştır. Uzmanlar arası uzlaşma oranı %70,4 olarak hesaplanmıştır. Daha sonra bu form Türk Dili ve Edebiyatı alanında bir uzman tarafından Türkçe dil bilgisi, ifadesel açıklık ve anlaşılabilirlik açısından kontrol edilmiştir. Böylece ölçeğin Türkçe aday formu uygulama öncesi nihai haline kavuşmuştur.

Bulgular

Geçerliğe İlişkin Bulgular

Ölçeğin yapı geçerliğinin incelenmesi amacıyla doğrulayıcı faktör analizi (DFA), yakınsak ve iraksak geçerlik çalışmaları gerçekleştirilmiştir.

Yapı Geçerliği

Ölçeğin Türk kültürüne uyumunun ne ölçüde sağlandığının belirlenmesi için DFA'dan yararlanılmıştır. Ölçme araçlarının kültürler arası uyumunun sağlanmasında faktör yapılarının doğrulanması söz konusu olduğu için DFA'nın kullanılması önerilmektedir (Watkins, 1989). DFA ile testin orijinal versiyonunu ile hedef versiyonunun model uyumuna bakılmaktadır (Hedrih, 2019).

Birinci Düzey Doğrulayıcı Faktör Analizi Sonuçları

Orijinal ölçeği oluşturan yapının doğrulanması amacıyla birinci düzey DFA gerçekleştirilmiştir. DFA ve yapısal eşitlik modellerinde modelin veriye uyumunun değerlendirilmesinde genellikle χ^2 (ki-kare), RMSEA (Root Mean Square Error of Approximation), RMSR (Standardized Root Mean

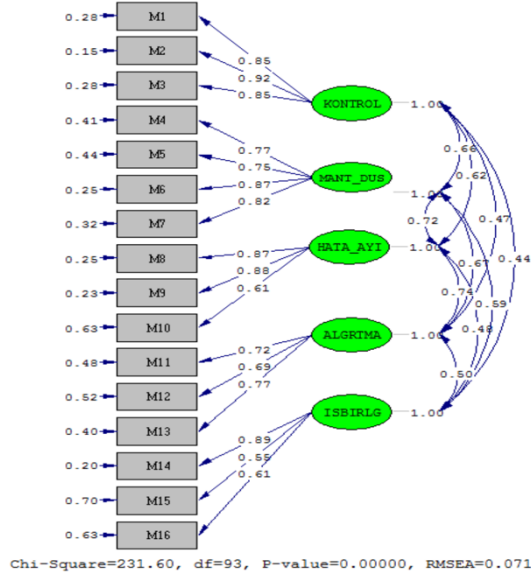
Squared Residual), GFI (Goodness of Fit Index), AGFI (Adjusted Goodness of Fit Index), CFI (Comparative Fit Index) ve NNFI (Non-Normed Fit Index) uyum indeksleri kullanılmaktadır (Çelik ve Yılmaz, 2016; Sümer, 2000). Bu değerlerden χ^2/sd oranının 5'ten küçük, RMSEA'nın .08'den küçük, SRMR değerinin .10'dan küçük, GFI değerinin .90'dan büyük, AGFI değerinin .80'den büyük, CFI değerinin .90'dan büyük ve NNFI değerinin .90'dan büyük olması modelin veriye uyumunun iyi olduğuna işaret etmektedir (Çokluk, Şekercioğlu ve Büyüköztürk, 2011; Meydan ve Şeşen, 2011; Schermelleh-Engel, Moosbrugger ve Müller, 2003). Bilgisayar programlama öz yeterlik ölçeğinin birinci düzey DFA sonucunda hesaplanan uyum indeksleri Tablo 2'de verilmiştir.

Tablo 2: Ölçeğin Birinci Düzey Doğrulayıcı Faktör Analizi Uyum İndeksleri

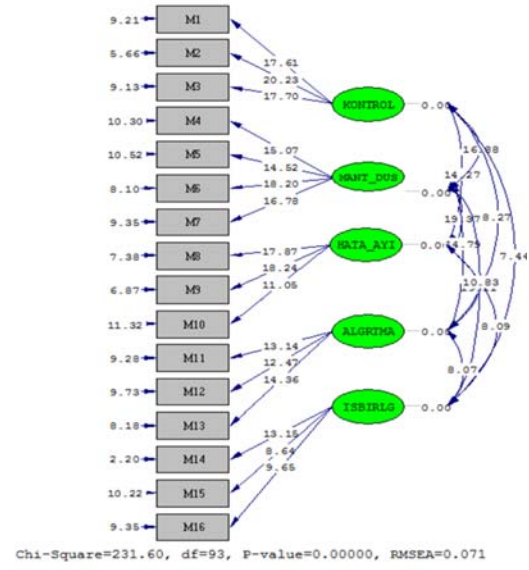
Uyum İndeksi	Kabul Edilebilir Değer	Model Değeri
χ^2/sd	<5	2.49
RMSEA	< .08	0.071
SRMR	<.10	.053
GFI	\geq .90	0.90
AGFI	\geq .80	0.87
CFI	\geq .90	0.98
NNFI	\geq .90	0.97

Tablo 2'de görüleceği üzere bilgisayar programlama öz-yeterlik ölçeğinin birinci düzey DFA uyum indeksi değerleri [$\chi^2/sd=2.49$, $n=297$, $\chi^2=231.60$, $p<.000$, $RMSEA=.071$, $SRMR=0.053$, $GFI=0.90$, $AGFI=0.85$, $CFI=0.98$, $NNFI=0.97$] olarak hesaplanmıştır. Bu değerler tüm uyum ölçütlerine göre modelin istatistiksel olarak kabul edilebilir uyuma sahip olduğunu göstermektedir.

Birinci Düzey DFA'nın bağlantı diyagramı ve hesaplanan standart katsayılar Şekil 1'de ve t-testi değerlerini gösteren diyagram Şekil 2'de yer almaktadır. Şekil 1'de görüleceği üzere ölçeğin DFA sonucunda her bir maddenin standartlaştırılmış çözümlene katsayıları .61 ile .92 arasında değişmektedir. Standartlaştırılmış çözümlene katsayıları +1 ile -1 arasında değer almaktadır ve bu değer 0'a yaklaştıkça zayıf bir ilişkiyi temsil etmektedir (Hair, Hult, Ringle ve Sarstedt, 2016).



Şekil 1: Standart Katsayılar



Şekil 2: T Değerleri

DFA sonucunda her bir maddeye ait standartlaştırılmış çözümlenme değerlerinin anlamlı olup olmadığının belirlenmesi için hesaplanan t-testi değerleri 11.05 ile 20.23 arasında değişmektedir (Şekil 2). Hesaplanan t-testi değerleri de tüm maddeler için anlamlıdır ($t > 2.57$, $p < .01$) (Hair vd., 2016).

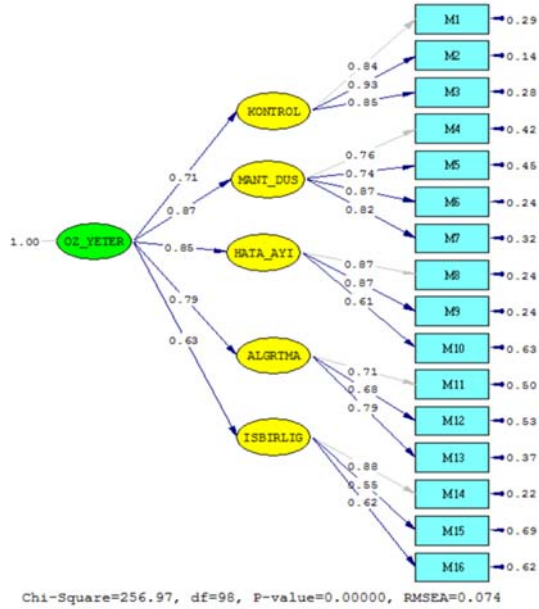
İkinci Düzey Doğrulayıcı Faktör Analizi Sonuçları

Ölçeğin birinci düzey DFA ile elde edilen boyutlarının, bilgisayar programlama öz-yeterliği yapısını hangi derecede temsil ettiğinin belirlenmesi için beş boyutun bir üst boyutu olan bilgisayar programlama öz-yeterliği ikinci düzey değişkeni ile olan yapısal ilişkilerini ortaya koyan ikinci düzey doğrulayıcı faktör modeli oluşturulmuştur. Bu çalışma kapsamında 5 gizil ve 16 gösterge değişken ile test edilen ikinci düzey faktör modelinin test edilmesi sonucunda hesaplanan uyum iyiliği değerleri Tablo 3'te verilmiştir.

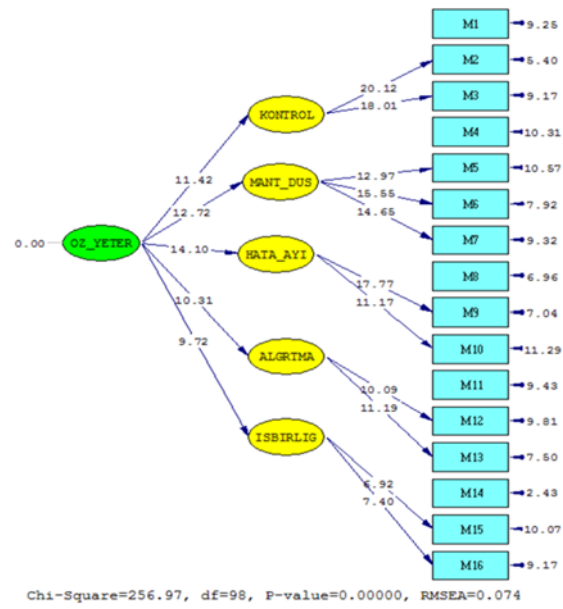
Tablo 3: Ölçeğin İkinci Düzey Doğrulayıcı Faktör Analizi Uyum İndeksleri

Uyum İndeksi	Kabul Edilebilir Değer	Model Değeri
χ^2/sd	< 5	2.62
RMSEA	< .08	0.074
SRMR	< .10	0.058
GFI	$\geq .90$	0.90
AGFI	$\geq .80$	0.86
CFI	$\geq .90$	0.98
NNFI	$\geq .90$	0.97

Tablo 3'te görülebileceği üzere bilgisayar programlama öz yeterlik ölçeğinin ikinci düzey DFA'dan elde edilen uyum indeksi değerleri [$\chi^2/sd=2.62$, $n=297$, $\chi^2=256.97$, $p<.000$, $RMSEA=.074$, $GFI= 0.90$, $AGFI=0.86$, $CFI=0.98$, $NNFI=0.97$] olarak hesaplanmıştır. Bu değerler kabul edilebilir düzeyde uyum olduğunu göstermektedir. İkinci düzey DFA'nın bağlantı diyagramı standart katsayılar Şekil 3'te ve buna ilişkin t-testi değerlerini gösteren diyagram Şekil 4'te verilmiştir.



Şekil 3: Standart Katsayılar



Şekil 4: T Değerleri

Yakınsak ve İraksak Geçerlik

Araştırmada ölçeğin yapı geçerliliğinin incelenmesi için ayrıca yakınsak ve ıraksak geçerlilik hesaplamaları yapılmıştır. DFA'dan elde edilen faktör yüklerinin .5'den büyük olması yakınsak geçerlik için kanıt sağlamaktadır (Hair vd., 2019). Ek olarak, Açıklanan Ortalama Varyans (AOV) değerleri hesaplanmıştır. Bu değerlerin de .5 ve üzerinde olması beklenmektedir (Fornell ve Larcker, 1981). AOV değerleri DFA'dan elde edilen faktör yüklerinin karesi alınıp elde edilen değerlerin aritmetik ortalaması ile hesaplanmaktadır. Tablo 4'te alt ölçeklerin her biri için hesaplanan AOV değerleri verilmiştir.

Tablo 4: AOV Değerleri

Alt Ölçekler	AOV değeri
Kontrol	.707
Mantıksal Düşünme	.559
Hata Ayıklama	.518
Algoritma	.560
İşbirliği	.665

Tablo 4'te yer alan veriler incelendiğinde AOV değerlerinin .52 ile .71 arasında değiştiği ve .5 kriterinin (Fornell ve Larcker,1981) sağladığı görülmektedir. Bu durumda ölçeğin yakınsak geçerliğinin sağlandığı söylenebilir. İraksak geçerlilik ya da ayırma geçerliği ise AOV değerlerinin kareköklerinin faktörler arasındaki ilişki değerlerinden büyük olmasına dayanmaktadır. Hesaplanan verilerden hareketle iraksak geçerliğin sağlandığı söylenebilir.

Tablo 5: Faktörler Arası Korelasyon Matrisi

	Kontrol	Hata Ayıklama	Mantıksal Düşünme	Algoritma	İşbirliği
Kontrol	(.84)				
Hata Ayıklama	0.62	(.75)			
Mantıksal düşünme	0.66	0.72	(.72)		
Algoritma	0.47	0.74	0.67	(.75)	
İşbirliği	0.44	0.48	0.59	0.50	(.86)

Not: Parantezler içerisindeki değerler AOV değerlerinin kare kökünü, diğer değerler ise alt ölçekler arasındaki korelasyon katsayısını belirtmektedir.

Güvenirlğe İlişkin Bulgular

Ölçeğin güvenirlüğünün belirlenmesi amacıyla iç tutarlık katsayısı Cronbach'ın alfa değeri, McDonald'ın omega değeri ile Bileşik Güvenirlilik (Composite Reliability, CR) değerleri ve madde toplam korelasyonu hesaplanmıştır. Cronbach'ın alfa katsayısı $\alpha = .911$ ve McDonald'ın omega katsayısı $\omega = .917$ ve bileşik güvenirlilik $CR = .962$ olarak hesaplanmıştır. Cronbach alfa güvenirlilik katsayıları; Kontrol alt ölçeği için .906, Hata Ayıklama alt ölçeği için .812, Mantıksal Düşünme alt ölçeği için .877, Algoritma alt ölçeği için .775 ve İşbirliği alt ölçeği için .813 olarak bulunmuştur. Bilgisayar programlama öz yeterlik ölçeğinde yer alan alt ölçekler ve bu ölçekler için hesaplanan güvenirlilik katsayıları ayrıca Tablo 6'da verilmiştir.

Tablo 6: Ölçekte Yer Alan Alt Ölçekler ve Hesaplanan Güvenirlilik Katsayıları

Alt Ölçekler	$\bar{\alpha}$	ω	Test Yarılama	CR
Kontrol	.906	.906	.789	.906
Mantıksal Düşünme	.877	.878	.849	.879
Hata Ayıklama	.812	.824	.799	.835
Algoritma	.775	.778	.741	.771
İşbirliği	.813	.825	.787	.733
Ölçeğin tümü	.911	.917	.823	.962

α : Cronbach'ın iç tutarlık katsayısı, ω : McDonald'ın iç tutarlılık katsayısı, CR: Bileşik güvenirlilik.

Cronbach'ın alfa katsayısı 0 ile 1 arasında değer almaktadır ve bir ölçeğin güvenilir sayılabilmesi için .7 ve üzerinde değer alması beklenmektedir (de Vaus, 2002). McDonald'ın omega katsayısı Cronbach'ın alfa katsayısının bir alternatifidir (Peters, 2014) ve alfa'ya göre daha doğru sonuçlar ürettiği iddia edilmektedir (Revelle ve Zinbarg, 2009). Hesaplanan her iki katsayıdan hareketle ölçeğin genelini ve alt ölçeklerin güvenilir olduğu söylenebilir. Test yarılama yöntemi ile elde edilen güvenilirlik katsayıları ölçeğin geneli için .82 ve alt ölçekler için; Kontrol .79, Mantıksal Düşünme .84, Hata Ayıklama .80, Algoritma .74, İşbirliği .79 olarak hesaplanmıştır. Bileşik güvenilirlik katsayıları birinci düzey DFA'dan elde edilen faktör yükleri ve standart hata değerleri kullanılarak hesaplanmıştır. Hair ve diğerleri (2019) 0.7 ve üzerinde bileşik güvenilirlik katsayısının yüksek düzeyde güvenilirliğe işaret ettiğini ve bütün maddelerin aynı gizil yapıyı temsil ettiğini ifade etmektedir. Bileşik güvenilirlik yöntemi ile elde edilen güvenilirlik katsayıları alt ölçekler için; Kontrol .91, Mantıksal Düşünme .88, Hata Ayıklama .84, Algoritma .77, İşbirliği .73 olarak hesaplanmıştır. Ölçek alt faktörlerinin güvenilirliğinin yüksek düzeyde olduğu söylenebilir.

Madde Analizi

Tablo 7: Madde Analizi Sonuçları

Alt boyut	Madde No	Ortalama	SS	Madde çıkarıldığında ölçek alfa değeri	Düzeltilmiş madde toplam korelasyonu	T değeri
Kontrol	M1	5.276	.910	.907	.562	11.43**
	M2	5.067	.949	.904	.671	14.54**
	M3	5.128	.928	.904	.662	12.99**
Mantıksal Düşünme	M4	4.838	.990	.903	.684	14,68**
	M5	5.296	.908	.906	.624	13.26**
	M6	4.828	1.04	.903	.710	16,38**
	M7	4.838	.976	.902	.727	16,27**
Hata Ayıklama	M8	4.461	1.78	.902	.712	16,26**
	M9	4.519	1.75	.903	.685	15,56**
	M10	4.640	1.31	.908	.550	11,83**
Algoritma	M11	3.953	1.91	.909	.541	12,58**
	M12	3.667	1.95	.911	.467	10,57**
	M13	4.444	1.56	.904	.667	13,72**
İşbirliği	M14	4.805	.967	.907	.561	11,69**
	M15	4.993	.959	.911	.411	7,52**
	M16	5.020	.962	.911	.435	8,94**

** sd=158, $p < .001$

Tablo 7’de yer alan bulgular incelendiğinde %27’lik alt ve üst grupların madde puanlarının farklarına ilişkin t değerlerinin; Kontrol alt ölçeği için 11,43 ile 14,54 arasında (sd=158, p<.001), Mantıksal Düşünme alt ölçeği için 13,26 ile 16,27 arasında (sd=158, p<.001), Hata Ayıklama alt ölçeği için 11,83 ile 16,26 (sd=158, p<.001), Algoritma alt ölçeği için 10,57 ile 13,72 arasında (sd=158, p<.001) ve İşbirliği alt ölçeği için 7,52 ile 11,69 arasında (sd=158, p<.001) değiştiği görülmektedir. Madde toplam korelasyonuna ilişkin sonuçlar; Kontrol alt ölçeği için .56 ile .67 arasında, Mantıksal Düşünme alt ölçeği için .62 ile .73 arasında, Hata Ayıklama alt ölçeği için .55 ile .71 arasında, Algoritma alt ölçeği için .47 ile .67 arasında ve İşbirliği alt ölçeği için .41 ile .56 arasında değişmektedir. Düzeltilmiş madde toplam korelasyonu her bir madde puanının ölçek toplam puanı ile ilişkisini ortaya koymaktadır. Güvenilir bir ölçekte her bir maddenin ölçek toplam puanı ile ilişkili olması beklenir ve .3 üzerindeki değerler iyi ilişkiye işaret etmektedir (Field, 2018). Ayrıca ölçek maddelerinin ayırt ediciliğinin test edilmesi amacıyla %27’lik alt ve üst gruplar arası farka ilişkin t değerlerinin anlamlı olması maddenin ayırt ediciliğine ve iç tutarlılığa kanıt olarak gösterilmektedir (Büyüköztürk, 2011; Tezbaşaran, 1996). Ölçek maddelerinin tamamının ayırt ediciliğinin yüksek olduğu söylenilebilir.

Tartışma, Sonuçlar ve Öneriler

Bu araştırma kapsamında Tsai, Wang ve Hsu (2019) tarafından geliştirilen bilgisayar okur-yazarlığı eğitiminde Bilgisayar Programlama Öz-yeterlik Ölçeğinin Türk kültürüne uyarlama çalışması yürütülmüştür. Gerçekleştirilen birinci ve ikinci düzey DFA’lar sonucunda ölçeğin Türkçe formunun geçerli ve güvenilir olduğu sonucuna ulaşılmıştır. Ölçek maddeleri Ek1’de verilmiştir.

Öğrencilerin söz konusu ölçek sayesinde bilgisayar programlamaya karşı kendi öğrenmeleri hakkındaki algılarının farkına varmalarını beklenmektedir. Programlama BD becerilerinin sergilenmesi ve geliştirilmesi için oldukça otantik bir deneyimdir. BD soyutlama ve ayrıştırma başta olmak karmaşık bir beceri ve tutum setini barındıran bir problem çözme sürecidir. Dolayısıyla özünde BD süreçlerinin gerektiren programlamayı ve buna ilişkin psikolojik yapıları tek bir monolitik yapı düzeyinde incelemek makul gözükmemektedir. Diğer birçok programlama öz-yeterlik ölçeğinden farklı olarak Türkçe uyarlaması yapılan mevcut ölçeğin faktör yapıları BD’ye ilişkin farklı beceri alanlarını kapsamaktadır. Bunlar kontrol, mantıksal düşünme, hata ayıklama, algoritma ve işbirliğidir. Kontrol alt ölçeği herhangi bir programlama platformunun kullanımına ilişkin yetkinlik algısına yöneliktir. Mantıksal düşünme alt ölçeği öğrenenlerin koşullu bağlantılar oluşturmak suretiyle program akışının mantıksal kontrolüne odaklanmaktadır. Bu nedenle bu faktör maddeleri programlama yaparken koşul yapılarını oluşturma yeteneğine ilişkin algıları ölçmektedir. Hata giderme, adından da anlaşılacağı gibi, öğrenenlerin program hatalarını giderme yeteneklerine ilişkin algılarını ölçer. Algoritma alt ölçeği, öğrencilerin bir problemin çözümü için programlama yaparken başkalarından bağımsız olarak algoritma oluşturma yetenekleri hakkındaki algılarını belirlemektedir. İşbirliği alt ölçeği programlamanın işbirlikli görev yapısına ait algıların değerlendirilmesini amaçlamaktadır. Ölçeğin orijinal formunun geliştirilme sürecinde faktöriyel geçerlik için sadece açımlayıcı faktör analizinden yararlanılmıştır. Mevcut çalışma ise test maddeleriyle (gösterge) faktör yapıları (gizil) arasındaki ilişkilerin ortaya çıkarılması ve ölçeği altında yatan modelinin test edilmesi amacıyla DFA kullanılmıştır. Faktörler arası en yüksek ilişkilerin hata ayıklama-algoritma ve hata ayıklama-mantıksal düşünme arasında olduğunu belirlenmiştir. Ayrıca mantıksal düşünmenin programlama öz-yeterliğini yordayan en önemli bileşen görülmektedir.

Ölçeğe ilişkin toplam madde puanları ($X=75,77$) göz önünde bulundurulduğunda katılımcıların genel olarak yüksek bir programlama öz-yeterlik algısına sahip olduğu söylenebilir. Bu durumun katılımcıların bilgisayar bilimleri alanındaki eğitim almış olmalarından kaynaklandığı düşünülmektedir.

Madde ortalama puanları incelendiğinde en yüksek değerin kontrol alt ölçeğine ($X=5,15$), en düşük değerin algoritma alt ölçeğine ($X=4,02$) ait olduğu saptanmıştır. Bu sonuçlar orijinal ölçeğin geliştirildiği çalışmanın bulguları ile örtüşmektedir (Tsai, Wang ve Hsu, 2019). Bilgisayar bilimleri ile ilgili bölümlerde öğrenim görmelerine rağmen öğrencilerin algoritma geliştirme konusundaki yeteneklerine ilişkin güvenlerinin yüksek olmadığı görülmektedir. Bu durumun algoritmik düşünme yeteneğinin karmaşık bilişsel işlemler (soyutlama, modelleme, analitik ve mantıksal düşünme gibi) kadar yüksek düzey üst bilişsel işlemleri (planlama, değerlendirme, karar verme gibi) gerektirmesinden kaynaklanabileceği düşünülmektedir (Çınar, 2019; Katai, 2015; Tsai, Wang ve Hsu, 2019). Algoritmik düşünme bir problemin belirli girdileri göz önünde bulundurarak bunlara uygun çıktılar üretmek üzere çözüm adımların aşamalı bir düzende yapılandırıldığı zihinsel bir yönelimdir (Denning, 2009). Algoritma geliştirme genellikle soyutlama, mantıksal bağlantılar kurma, düşünceleri kurallı yapılar içinde temsil etme gibi özelliklerin bir araya getirilmesini gerektirmektedir. Bununla birlikte, öğrenciler genellikle gerçek hayatta algoritmalarla ilişkilendirebilecekleri referansları bulmakta güçlük çekmektedir (Ramadhan, 2000). Ek olarak, yüksek ders bırakma oranı, düşük akademik başarı, motivasyon ve kariyer sürdürme eğilimi dahil olmak programlama derslerinin yaşanan sıkıntıların önemli bir kısmı geleneksel programlama öğretimi yaklaşımı ve söz dizim temelli görev yapısından kaynaklanmaktadır (diSessa, 2001; Park, Kim, Oh, Jang ve Lim, 2015). Geleneksel programlama görevleri sıklıkla sınırlı bir talimat setine dayanılarak hazırlanmış dar kapsamlı algoritmaları içermektedir. Bu nedenle bilgisayarlı problemlerin ve görevlerin sözdizimine dayalı mekanik bir rutinin ötesinde yüksek düzeyli düşünme becerilerin sergilenmesini gerektiren algoritmaların geliştirilmesine olanak tanıyacak şekilde tasarlanması önerilmektedir.

Disessa (2001) bilgisayarlı okuryazarlığın materyal, sosyal ve bilişsel olmak üzere üç boyutuna dikkati çekmiştir. Materyal boyutu bilgisayarlı bir şeyler ortaya çıkarma, sosyal boyut kişiler arasında bilgisayarlı ürünlerden ne beklenildiğinin ifade edilmesi, tartışılması ve ortak anlayışların geliştirilmesi olarak tanımlanmaktadır. Bilişsel boyut ise bilgisayarlı ne yapılabileceğine, nerede ve niçin kullanılabileceğine ve bir araç olarak nasıl kullanılabileceğine ilişkin olarak problemler, bağlamlar ve otantik dünya düzenlemeleri hakkında düşünme yeteneğini tanımlar (Berland, 2016). Buna göre belirli bir pratiğin okuryazarlık olarak nitelendirilmesi büyük ölçüde sosyal olarak inşa edilmiş bağlamlara dayanmaktadır. Alanyazın incelendiğinde mevcut programlama öz-yeterlik ölçeklerinin genellikle belirli bir özelliği (görev zorluğu, müfredat kazanımları gibi) merkeze alan tek tipli bir yapı arz ettiği ve programlamanın sosyal boyutunu yeterince adreslemediği görülmektedir. Programlamanın kişilerin arası etkileşim gerektiren bir BD etkinliği olduğu düşünüldüğünde programcıların verimli ve etkili olabilmesi aynı zamanda işbirlikli çalışma yetkinliklerine bağlıdır.

Son yıllarda BD'nin yaygınlaştırılması ve diğer disiplinlerle birleştirilmesine yönelik çabaların ardında 21. yüzyılın zorluklarıyla daha etkin bir yolla başa çıkmak üzere öğrencilerin bilgisayarlı esaslara dayalı problem çözme stratejileri geliştirilmesi ve kullanmasının teşvik edilmesi bulunmaktadır. Çoğu ülkede BD'nin öğrencilerin dijital yetkinliklerine eklenmesi için okul müfredatlarında yer almasına yönelik girişimler hız kazanmıştır. Bununla birlikte, BD yeterliklerinin kazandırılması için uygun öğrenme deneyimleri oluşturma ve bu deneyimlerin etkinliğini değerlendirme konusunda eğitimcileri bir dizi zorluk beklemektedir. Özellikle, BD beceri ve yetkinliklerin değerlendirilmesine yönelik girişimler halen beklentilerin altındadır (Angeli ve Giannakos, 2020). Alanyazın incelendiğinde BD'nin geliştirilmesi adına çok sayıda etkinlik türü olmasına rağmen en baskın stratejinin programlama olduğu görülmektedir. Mevcut çalışmada öğrencilerin programlama öz-yeterliklerinin değerlendirilmesi için geçerliği ve güvenilirliği kanıtlanmış bir ölçme aracının Türkçeye uyarlama çalışması gerçekleştirilmiştir. Ülkemizde K-12 veya yükseköğretim kademelerinde çok çeşitli programlama ders

ve etkinlikleri (blok-tabanlı programlama, robotik, nesne yönelimli ve metin temelli programlama gibi) yürütülmektedir. Uygun bir değerlendirmeye tabi tutulmaksızın gerçekleştirilen eğitsel etkinliklerin amacına uygunluğuna ilişkin saptamaların varsayımdan öteye geçmesi beklenemez. Türkçe formu oluşturulan mevcut ölçeğin öğrencilerin programlama yeterliklerine ilişkin algılarının değerlendirilmesi ve böylece öğretimsel uygulamalara ilişkin çok yönlü çıkarımların elde edilmesine katkı sağlaması beklenilmektedir. Ölçek aynı zamanda programlama/kodlama görevlerini (fiziksel programlama) içeren Fen, Teknoloji, Mühendislik, Sanat, Matematik (STEAM) eğitimi gibi disiplinlerarası entegre müfredatlara da uygulanabilir. Gelecekteki çalışmalarda öğrencilerin programlama öz-yeterlikleri ile cinsiyet, branş, sınıf düzeyi, programlama deneyimi gibi çeşitli değişkenler arasındaki ilişkilerin incelenmesi önerilmektedir.

Kaynaklar

- Alpar, R. (2011). *Uygulamalı çok değişkenli istatistiksel yöntemler (3. baskı)*. Ankara: Detay Yayıncılık.
- Altun, A. ve Mazman, S. G. (2012). Programlamaya ilişkin öz-yeterlik algısı ölçeğinin Türkçe formunun geçerlilik ve güvenilirlik çalışması. *Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi*, 3(2), 297-308.
- Altun, A. ve Kasalak, İ. (2018). Blok temelli programlamaya ilişkin öz-yeterlik algısı ölçeği geliştirme çalışması: Scratch örneği. *Educational Technology Theory and Practice*, 8(1), 209-225.
- Angeli, C. and Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, 105(3), doi:10.1016/j.chb.2019.106185
- Aşkar, P. and Davenport, D. (2009). An investigation of factors related to self-efficacy for java programming among engineering students. *Turkish Online Journal of Educational Technology*, 8(1), 26-32.
- Aydın, S. (2015). An analysis of the relationship between high school students' self-efficacy, metacognitive strategy use and their academic motivation for learning biology. *Journal of Education and Training Studies*, 4(2), 53-59.
- Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Englewood Cliffs, NJ, US: Prentice-Hall.
- Bandura, A. (1993). Perceived self-efficacy in cognitive development and functioning. *Educational Psychologist*, 28(2), 117-148. doi:10.1207/s15326985ep2802_3
- Bandura, A. (1995). Exercise of personal and collective efficacy in changing societies. In A. Bandura (Ed.), *Self-efficacy in changing societies* (pp. 1-45). Cambridge, UK: Cambridge University Press.
- Berland, M. (2016). Making, tinkering, and computational literacy. K. Peppler, E. R. Halverson, & Y. B. (Eds.). Kafai, *Makeology: Makers as learners* (Vol. 2, pp. 196-205). NY: Routledge.
- Berland, M. and 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.
- Bhardwaj, J. (2017). In search of self-efficacy: development of a new instrument for first year Computer Science students. *Computer Science Education*, 27(2), 79-99.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A. and Engelhardt, K. (2016). *Developing computational thinking in compulsory education – Implications for policy and practice* (EUR 28295 EN). European Commission, Joint Research Centre. doi:10.2791/792158
- Büyüköztürk, Ş. (2011). *Sosyal bilimler için veri analizi el kitabı*. Ankara: Pegem Akademi Yayıncılık.
- Çelik, H. E. ve Yılmaz, V. (2016). *LISREL 9.1 ile yapısal eşitlik modellemesi: Temel kavramlar-uygulamalar-programlama*. Ankara: Anı Yayıncılık.
- Çınar, M. (2019). *Nesneye yönelik ve robot programlamanın öğrenci başarısına, soyutlamaya, problem çözmeye ve motivasyona etkilerinin incelenmesi*. (Yayımlanmamış doktora tezi). Hacettepe Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara
- Çokluk, Ö., Şekercioğlu, G. ve Büyüköztürk, Ş. (2012). *Sosyal bilimler için çok değişkenli istatistik: SPSS ve LISREL uygulamaları*. Ankara: Pegem Akademi Yayıncılık.
- De Vaus, D. A. (2002). *Surveys in Social Research*. Routledge.
- Denning, P. J. (2009). The profession of IT beyond computational thinking. *Communications of the ACM*, 52(6), 28–30. doi:10.1145/1516046.1516054

- Diaz, E. C. and Silvain, G. L. (2020). Computational Thinking. New challenges for 21st century education. *Virtualidad Educacion Y Ciencia, 11(20)*, 115-137.
- diSessa, A. A. (2001). *Changing minds: Computers, learning, and literacy* (1st ed.). Cambridge, MA, US: The MIT Press.
- Field A. (2018). *Discovering statistics using ibm spss statistics (5th Edition)*. SAGE Publications.
- Feurzeig, W. and Papert, S. A. (2011). Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments, 19(5)*, 487–501.
- Fornell, C. and Larcker, D. F. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research, 18(1)*, 39-50.
- Govender, D. W. and 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.
- Grover, S. and Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher, 42(1)*, 38-43.
- Gurer, M. D., Cetin, I. and Top, E. (2019). Factors affecting students' attitudes toward computer programming. *Informatics in Education, 18(2)*, 281-296. doi:10.15388/infedu.2019.13
- Hair, J. F., Hult, G. T. M., Ringle, C. and Sarstedt, M. (2016). *A primer on partial least squares structural equation modeling (PLS-SEM)*. Sage publications.
- Hair, J. F., Black, W. C., Babin, B. J. and Anderson, R. E. (2019). *Multivariate data analysis (8th Edition)*. Boston: Cengage.
- Hambleton, R. K., Merenda, P. F. and Spielberger, C. D. (2004). Issues, designs, and technical guidelines for adapting tests into multiple languages and cultures. In R.K. Hambleton, P. F. Marenada ve C. D. Spielberger (Eds.), *Adapting educational and psychological tests for cross-cultural assessment* (pp. 15-50). Psychology Press.
- Haseski, H. İ., Ilic, U. and Tugtekin, U. (2018). Defining a New 21st Century Skill-Computational Thinking: Concepts and Trends. *International Education Studies, 11(4)*, 29-42.
- Haslaman, T. and Askar, P. (2007). Investigating the relationship between self-regulated learning strategies and achievement in a programming course. *Hacettepe Universitesi Egitim Fakultesi Dergisi, (32)*, 110-122.
- Hawlitshchek, A., Koppen, V., Dietrich, A. and Zug, S. (2019). Drop-out in programming courses - prediction and prevention. *Journal of Applied Research in Higher Education, 12(1)*, 124-136. doi:10.1108/jarhe-02-2019-0035
- Hedrih, V. (2019). *Adapting psychological tests and measurement instruments for cross-cultural research: an introduction*. Routledge.
- International Society for Technology in Education (ISTE) and Computer Science Teachers Association (CSTA). (2011). Operational Definition of Computational Thinking for K–12 Education. Erişim adresi: <https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2>
- Jun, S., Han, S., Kim, H. and Lee, W. (2014). Assessing the computational literacy of elementary students on a national level in Korea. *Educational Assessment Evaluation and Accountability, 26(4)*, 319-332. doi:10.1007/s11092-013-9185-7
- Katai, Z. (2015). The challenge of promoting algorithmic thinking of both sciences- and humanities-oriented learners. *Journal of Computer Assisted Learning, 31(4)*, 287-299. doi:10.1111/jcal.12070

- Korkmaz, Ö., Kılıç, F. N., Çakır, R. ve Erdoğan, F. U. (2019). Meslek lisesi bilişim teknolojileri öğrencilerinin programlama öz-yeterlilikleri, STEM ve bilgisayarca düşünme becerilerine yönelik algıları. *Gazi Eğitim Bilimleri Dergisi*, 5(2019), 196-218.
- Korkmaz, Ö., Şahin, H., Çakır, R. ve Erdoğan, F. U. (2019). Bilişim teknolojileri öğretmenlerinin kodlamaya dönük tutumları, öz-yeterlilikleri ve kodlama öğretimi için kullandıkları yöntemler. *Ondokuz Mayıs Üniversitesi Eğitim Fakültesi Dergisi*, 38(2), 1-16. doi:10.7822/omuefd.612449
- Kukul, V., Gökçearslan, Ş. and Günbatar, M. S. (2017). Computer programming self-efficacy scale (CPSES) for secondary school students: Development, validation and reliability. *Educational Technology Theory and Practice*, 7(1), 158-179.
- Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T. and Almuğhyrah, S. (2016). Using robotics and game design to enhance children's self-efficacy, STEM attitudes, and computational thinking skills. *Journal of Science Education and Technology*, 25(6), 860-876. doi:10.1007/s10956-016-9628-2
- Li, S. and Zheng, J. (2018). The relationship between self-efficacy and self-regulated learning in one-to-one computing environment: the mediated role of task values. *The Asia-Pacific Education Researcher*, 27(6), 455-463. doi:10.1007/s40299-018-0405-2
- Majumder, S. and Deen, M. J. (2019). Smartphone Sensors for Health Monitoring and Diagnosis. *Sensors (Basel)*, 19(9), 1-45. doi:10.3390/s19092164
- Mayers, A. (2013). *Introduction to statistics and SPSS in psychology*. Pearson Higher Ed.
- Mazman Akar, S. G. and Altun, A. (2017). Individual differences in learning computer programming: A social cognitive approach. *Contemporary Educational Technology*, 8(3), 195-213.
- Meydan, C. H. ve Şeşen, H. (2011). *Yapısal eşitlik modellemesi AMOS uygulamaları*. Ankara: Detay Yayıncılık.
- Morrell, P. D. and Carroll, J. B. (2003). An extended examination of preservice elementary teachers' science teaching self-efficacy. *School Science and Mathematics*, 103(5), 246-251.
- Noone, M. and Mooney, A. (2018). Visual and textual programming languages: a systematic review of the literature. *Journal of Computers in Education*, 5(2), 149-174.
- Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas (1st ed.)*. NY, USA: Basic Books.
- Pajares, F. (1996). Self-efficacy beliefs in academic settings. *Review of Educational Research*, 66(4), 543-578.
- Park, I., Kim, D., Oh, J., Jang, Y. and Lim, K. (2015). Learning effects of pedagogical robots with programming in elementary school environments in Korea. *Indian Journal of Science and Technology*, 8(26), 1-5. doi:10.17485/ijst/2015/v8i26/80723
- 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), 157-170.
- Peters, G. Y. (2014). The alpha and the omega of scale reliability and validity: why and how to abandon Cronbach's Alpha. *European Health Psychologist*, 16(5), 576.
- Ramadhan, H. A. (2000). Programming by discovery. *Journal of Computer Assisted Learning*, 16(1), 83-93.
- Ramalingam, V. and 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.
- Revelle, W. and Zinbarg, R. E. (2009). Coefficients Alpha, Beta, Omega, and the glb: comments on Sijtsma. *Psychometrika*, 74(1), 145-154.

- Schermelleh-Engel, K., Moosbrugger, H. and Müller, H. (2003). Evaluating the fit of structural equation models: Tests of significance and descriptive goodness-of-fit measures. *Methods of psychological research online*, 8(2), 23-74.
- Schunk, D. H. (1991). Self-efficacy and academic motivation. *Educational Psychologist*, 26(3-4), 207-231. doi:10.1207/s15326985ep2603&4_2
- Schunk, D. H. (2012). *Learning theories: an educational perspective (6th Edition)*. Boston, MA: Prentice Hall.
- Sümer, N. (2000). *Yapısal eşitlik modelleri: temel kavramlar ve örnek uygulamalar*. Türk Psikoloji Yazıları.
- Tabachnick, B. G. and Fidell, L. S. (2012). *Using multivariate statistics (Vol. 6)*. Boston, MA: Pearson.
- Tezbaşaran, A. (1996). *Likert tipi ölçek geliştirme kılavuzu*. Ankara: Psikologlar Derneği Yayınları.
- Tsai, M. J., Wang, C. Y. and Hsu, P. F. (2019). Developing the computer programming self-efficacy scale for computer literacy education. *Journal of Educational Computing Research*, 56(8), 1345-1360.
- Watkins, D. (1989). The role of confirmatory factor analysis in cross-cultural research. *International journal of psychology*, 24(6), 685-701
- West, SG, Finch. J. F. and Curran, P. J. (1995). Structural equation models with non-normal variables: problems and remedies. In RH Hoyle (Ed.). *Structural equation modeling: Concepts, issues and applications*. Newbery Park, CA: Sage; p56-75.
- 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(1881), 3717-3725.
- Yadav, A., Good, J., Voogt, J. and Fisser, P. (2017). Computational thinking as an emerging competence domain. In M. Mulder (Ed.), *Competence-based Vocational and Professional Education: Bridging the Worlds of Work and Education* (pp. 1051-1067). Switzerland: Springer International Publishing.
- Yi, M. Y. and Hwang, Y. (2003). Predicting the use of web-based information systems: self-efficacy, enjoyment, learning goal orientation, and the technology acceptance model. *International Journal of Human-Computer Studies*, 59, 431-449.
- Zhong, B. C., Wang, Q. Y., Chen, J. and Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research*, 53(4), 562-590. doi:10.1177/0735633115608444
- Zimmerman, B. J. (1989). A social cognitive view of self-regulated academic learning. *Journal of Educational Psychology*, 81(3), 329-339.
- Zimmerman, B. J. (1990a). Self-regulated learning and academic achievement: An overview. *Educational Psychologist*, 25(1), 3-17.
- Zimmerman, B. J. (1990b). Self-regulating academic learning and achievement: The emergence of a social cognitive perspective. *Educational Psychology Review*, 2(2), 173-201. doi:10.1007/BF01322178
- Zimmerman, B. J., Bonner, S. and Kovach, R. (1996). *Developing self-regulated learners: Beyond achievement to self-efficacy*. Washington, DC: American Psychological Association.
- Zimmerman, B. J. (2011). Motivational sources and outcomes of self-regulated learning and performance. In *Handbook of self-regulation of learning and performance*. (pp. 49-64). New York, NY, US: Routledge.

Extended Abstract

Today, computers have become an essential part of our lives almost in every aspect, including professional activities and daily routines. Contrary to the past, the ability to produce solutions to problems by using Information and Communication Technologies (ICT) has become more important than the ability to use them effectively and responsibly. To be successful in this ever-changing environment, individuals are expected to solve problems, generate different solutions, develop hardware or software. In other words, they are expected to produce by using technology. Educational institutions, on the other hand, are expected to provide students with a roadmap to guide social change in line with primary needs of the society. In this regard, in many countries, including Turkey, the need for rearranging national ICT education curriculums in such a way to include computational skills has emerged (Bocconi, Chiocciariello, Dettori, Ferrari, & Engelhardt, 2016; Jun, Han, Kim, & Lee, 2014; Yadav, Good, Voogt, & Fisser, 2017). Computer programming education is considered as an essential way of developing computational skills (Tsai, Wang & Hsu, 2019). Research showed that there is a strong relationship between self-efficacy and student achievement in computational thinking (CT) activities, including programming (Gurer, Cetin & Top, 2019; Haslaman & Aşkar, 2007; Hawlitschek, Koppen, Dietrich, & Zug, 2019; Korkmaz, Kılıç, Çakır & Erdoğan, 2019; Mazman Akar & Altun, 2017). Students' self-efficacy in programming can provide valuable information about students' CT skills as well as the effectiveness of instructional practices.

The existing programming self-efficacy scales in the literature generally have a monolithic structure based on a particular feature such as task difficulty, programming language/modality, curriculum achievements, and they do not adequately address the social aspect of programming. But programming is a CT activity that often requires interpersonal interaction. Therefore, the effectiveness and efficiency of programmers also depend highly on their collaboration skills. Moreover, most of these scales were adapted from the programming self-efficacy scale, originally developed by Ramalingam and Wiedenbeck (1998). Considering the developments in the field of ICT and computer science, as well as the changes in the current understanding of CT over the past 20 years, the need to revisit self-efficacy measurement tools becomes more evident. Hence, this study aims to adapt the Computer Programming Self-efficacy Scale developed by Tsai, Wang, and Hsu (2019) to the Turkish language and to examine its psychometric properties.

Research data were collected from 297 participants consisting of high school and university students in 2019. In the adaption of the scale into Turkish, the forward judgmental design translation method was used (Hambleton, Merenda & Spielberger, 2004). To ensure the compatibility of measurement tools across cultures, the use of confirmatory factor analysis (CFA) is recommended since factor structures need to be verified (Hedrih, 2019; Watkins, 1989). In addition to CFA, convergent and divergent validity measurements were also used to test the construct validity of the Turkish form of the scale.

According to first-order ($\chi^2/df = 2.49$, $n = 297$, $\chi^2 = 231.60$, $p < .000$, $RMSEA = .071$, $SRMR = .053$, $GFI = .90$, $AGFI = .85$, $CFI = .98$, $NNFI = .97$) and second-order ($\chi^2/df = 2.62$, $n = 297$, $\chi^2 = 256.97$, $p < .000$, $RMSEA = .074$, $GFI = .90$, $AGFI = .86$, $CFI = .98$, $NNFI = .97$) CFA results, data compatibility of the model was seen to be acceptable level and it confirmed the factor structure in the original scale. Average Variance Extracted (AVE) values calculated for convergent validity, varies between .52 and .71, which can be considered acceptable (Fornell & Larcker, 1981). The AVE square root values calculated for divergent validity are within the acceptable range. To determine reliability, Cronbach alpha ($\alpha = .911$), McDonald's omega ($\omega = .917$), split-half ($r_{SH} = .823$), and composite

reliability coefficients ($\alpha = .962$) were calculated, and it was concluded that the Turkish form of the scale was reliable. As a result, it has been determined that the Turkish form of the scale is a valid and reliable measurement tool that can be used to measure the computer programming self-efficacy of students at high school or higher education levels.

Ek 1. Türkçe Ölçek Maddeleri

1. Bir programın kodlarını programlama editöründe (düzenleyicisinde) açabilir ve kaydedebilirim.
2. Bir programın kodlarını programlama editöründe düzenleyebilir ve değiştirebilirim.
3. Bir programın kodlarını programlama editöründe çalıştırabilir ve test edebilirim.
4. Bir programın temel mantıksal yapısını anlayabilirim
5. “if . . . else . . . (eğer . . . değilse . . .)” gibi koşullu bir ifadeyi anlayabilirim.
6. Mantıksal sınamalar içeren bir programın en nihai sonucunu (çıktısını) tahmin edebilirim.
7. Girdi değerleri verilen bir programın sonucunu (çıktısını) tahmin edebilirim.
8. Bir programı test ederken oluşan bir hatanın kaynağını bulabilirim.
9. Bir programı test ederken oluşan bir hatayı düzeltebilirim.
10. Hata ayıklama (debugging) aracılığıyla programlama hakkında daha fazla şey öğrenebilirim.
11. Program prosedürlerini (fonksiyonlarını) bir örnekten faydalanmadan anlayabilirim.
12. Bir program oluşturmak için başkalarının yardımına ihtiyaç duymam.
13. Bir problemi çözmek için programlamayı kullanabilirim.
14. Programlama çalışmasının farklı kişilerce tamamlanabilen alt görevlere ayrılabilceğini biliyorum.
15. Bir programı yazarken başkalarıyla çalışabilirim.
16. Programlama verimini artırmak için iş bölümünden yararlanabilirim.