

Eđitim Teknolojisi

kuram ve uygulama

Kış 2018

Cilt 8

Sayı 1

Winter 2018

Volume 8

Issue 1

Educational Technology

theory and practice

ISSN: 2147-1908

Cilt 8, Sayı 1, Kış 2018
Volume 8, Issue 1, Winter 2018

Genel Yayın Editörü / Editor-in-Chief: **Dr. Halil İbrahim YALIN**
Editör / Editor: **Dr. Tolga GÜYER**

Sorumlu Yazı İşleri Müdürü / Publisher Editor: **Dr. Tolga GÜYER**
Redaksiyon / Redaction: **Dr. Tolga GÜYER**
Dizgi / Typographic: **Dr. Tolga GÜYER**
Sayfa Tasarımı / Page Design: **Dr. Tolga GÜYER**
Kapak Tasarımı / Cover Design: **Dr. Bilal ATASOY**
İletişim / Contact Person: **Dr. Aslıhan KOCAMAN KAROĞLU**

Dizinlenmektedir / Indexed in: **ULAKBİM Sosyal ve Beşerî Bilimler Veritabanı, Türk Eğitim İndeksi**

ETKU Dergisi **2011 yılından itibaren yılda iki defa** düzenli olarak yayınlanmaktadır.
Educational Technology Theory and Practice Journal is published regularly **twice a year since 2011.**

Editör Kurulu / Editorial Board*

Dr. Abdullah Kuzu
Dr. Ana Paula Correia
Dr. Aytekin İşman
Dr. Buket Akkoyunlu
Dr. Cem Çuhadar
Dr. Deniz Deryakulu

Dr. Deepak Subramony
Dr. Feza Orhan
Dr. H. Ferhan Odabaşı
Dr. Hafize Keser
Dr. Halil İbrahim Yalın
Dr. Hyo-Jeong So

Dr. Kyong Jee(Kj) Kim
Dr. M. Yaşar Özden
Dr. Mehmet Gürol
Dr. Özcan Erkan Akgün
Dr. S. Sadi Seferoğlu
Dr. Sandie Waters

Dr. Servet Bayram
Dr. Şirin Karadeniz
Dr. Tolga Güyer
Dr. Trena Paulus
Dr. Yavuz Akpınar
Dr. Yun-Jo An

* Liste isme göre alfabetik olarak oluşturulmuştur. / List is created in alphabetical order

Hakem Kurulu / Reviewers*

Dr. Abdullah Kuzu
Dr. Adile Aşkın Kurt
Dr. Agah Tuğrul Korucu
Dr. Arif Altun
Dr. Aslıhan İstanbullu
Dr. Aslıhan Kocaman Karoğlu
Dr. Ayça Çebi
Dr. Ayfer Alper
Dr. Aynur Kolburan Geçer
Dr. Ayşegül Bakar Çörez
Dr. Bahar Baran
Dr. Berrin Doğusoy
Dr. Bilal Atasoy
Dr. Çelebi Uluyol
Dr. Demet Somuncuoğlu Özerbaş
Dr. Deniz Atal Köysüren
Dr. Deniz Mertkan Gezin
Dr. Ebru Kılıç Çakmak
Dr. Ebru Solmaz
Dr. Ekmel Çetin
Dr. Emin İbili
Dr. Emine Cabı
Dr. Emine Şendurur
Dr. Erinç Karataş
Dr. Erhan Güneş
Dr. Erkan Çalışkan
Dr. Erkan Tekinarslan
Dr. Erman Yükseltürk
Dr. Erol Özçelik

Dr. Ertuğrul Usta
Dr. Esmâ Aybike Bayır
Dr. Fatma Bayrak
Dr. Fatma Kesinkılıç
Dr. Fezile Özdamlı
Dr. Filiz Kalelioğlu
Dr. Funda Erdoğan
Dr. Gizem Karaoğlan Yılmaz
Dr. Gökçe Becit İşıtıürk
Dr. Gökhan Akçapınar
Dr. Gökhan Dağhan
Dr. Gülfidan Can
Dr. H. Ferhan Odabaşı
Dr. Hafize Keser
Dr. Halil Ersoy
Dr. Halil İbrahim Akyüz
Dr. Halil İbrahim Yalın
Dr. Halil Yurdugül
Dr. Hasan Çakır
Dr. Hasan Karal
Dr. Hatice Durak
Dr. Hatice Sancar Tokmak
Dr. Hüseyin Bicen
Dr. Hüseyin Çakır
Dr. Hüseyin Özçınar
Dr. Hüseyin Uzunboylu
Dr. Işıl Kabakçı Yurdakul
Dr. İbrahim Arpacı
Dr. İlknur Resioğlu
Dr. Kerem Kılıçer

Dr. Kevser Hava
Dr. M. Emre Sezgin
Dr. M. Fikret Gelibolu
Dr. Mehmet Akif Ocak
Dr. Mehmet Barış Horzum
Dr. Mehmet Kokoç
Dr. Melih Engin
Dr. Meltem Kurtoğlu
Dr. Mukaddes Erdem
Dr. Mustafa Serkan Günbatır
Dr. Mutlu Tahsin Üstündağ
Dr. Nadire Çavuş
Dr. Necmi Eşgi
Dr. Nezih Önal
Dr. Nuray Gedik
Dr. Nurettin Şimşek
Dr. Onur Dönmez
Dr. Ömer Faruk İslim
Dr. Ömer Faruk Ursavaş
Dr. Ömür Akdemir
Dr. Özcan Erkan Akgün
Dr. Özden Şahin İzmirli
Dr. Özlem Çakır
Dr. Ramazan Yılmaz
Dr. Recep Çakır
Dr. Sami Acar
Dr. Sami Şahin
Dr. Selay Arkün Kocadere
Dr. Selçuk Karaman

Dr. Selçuk Özdemir
Dr. Serap Yetik
Dr. Serdar Çiftçi
Dr. Serçin Karataş
Dr. Serkan Şendağ
Dr. Serkan Yıldırım
Dr. Serpil Yalçınalp
Dr. Sibel Somyürek
Dr. Soner Yıldırım
Dr. Şafak Bayır
Dr. Şahin Gökçearslan
Dr. Şeyhmus Aydoğdu
Dr. Şirin Karadeniz
Dr. Tayfun Tanyeri
Dr. Turgay Alakurt
Dr. Tolga Güyer
Dr. Türkan Karakuş
Dr. Uğur Başarmak
Dr. Ümmühan Avcı Yücel
Dr. Ünal Çakıroğlu
Dr. Veysel Demirer
Dr. Vildan Çevik
Dr. Yalın Kılıç Türel
Dr. Yasemin Demirarslan Çevik
Dr. Yasemin Gülbahar
Dr. Yasemin Koçak Usluel
Dr. Yavuz Akbulut
Dr. Yusuf Ziya Olpak
Dr. Yüksel Göktaş

* Liste isme göre alfabetik olarak oluşturulmuştur. / List is created in alphabetical order.

İletişim Bilgileri / Contact Information

İnternet Adresi / Web: <http://dergipark.gov.tr/etku>
E-Posta / E-Mail: tguyer@gmail.com
Telefon / Phone: +90 (312) 202 17 38
Belgegeçer / Fax: +90 (312) 202 83 87
Adres / Adress: Gazi Üniversitesi, Gazi Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü
06500 Teknikokullar - Ankara / Türkiye

Makale Geçmişi / Article History

Alındı/Received: 23.08.2017

Düzeltilme Alındı/Received in revised form: 28.11.2017

Kabul edildi/Accepted: 29.11.2017

BLOK TEMELLİ PROGRAMLAMAYA İLİŞKİN ÖZ-YETERLİK ALGISI ÖLÇEĞİ GELİŞTİRME ÇALIŞMASI: SCRATCH ÖRNEĞİ

Arif ALTUN¹, İbrahim KASALAK²

Öz

Bu çalışma kapsamında, “blok temelli programlamaya ilişkin öz-yeterlik algısı ölçeği” geliştirilmiştir. Ayrıca geliştirilen ölçme aracı ile katılımcılardan elde edilen öz-yeterlik algısı puanları çeşitli değişkenler açısından incelenmiştir. 4 farklı ilde bulunan ortaokullarda eğitim gören, daha önce blok temelli programlama eğitimi almış ve/veya halen bu eğitimi almakta olan toplamda 329 öğrenciden toplanan verilerle yapılan geçerlik ve güvenirlik çalışması sonucunda 5 maddeden oluşan “basit blok temelli programlama görevleri” ve 7 maddeden oluşan “karmaşık blok temelli programlama görevleri” şeklinde isimlendirilen 2 faktörlü 5’li Likert tipinde bir “blok temelli programlamaya ilişkin öz-yeterlik algısı ölçeği” geliştirilmiştir. Açıklayıcı faktör analizi sonunda 5 maddelik birinci faktör toplam varyansın %11.462’ini, 7 maddelik ikinci faktör ise toplam varyansın %46.763’ünü açıklamış, açıklanan toplam varyans ise %58.225 olarak hesaplanmıştır. Bu yapıya ilişkin model doğrulayıcı faktör analizi ile doğrulanarak bireylerin blok temelli programlamaya ilişkin öz-yeterlik algısını ölçen geçerli ve güvenilir bir araç elde edilmiştir.

Anahtar sözcükler: Blok temelli programlama, kodlama, öz-yeterlik algısı

PERCEIVED SELF-EFFICACY SCALE DEVELOPMENT STUDY RELATED TO BLOCK-BASED PROGRAMMING: SCRATCH CASE

Abstract

This study reports the development process of self-efficacy perception scale related to block-based programming. Self-efficacy is a well-established variable to be used in predicting programming skills. A total of 329 students from the middle school level participated in scale development study. The scale items were generated based on the literature in computer programming in general and Scratch use in particular with a Likert-type design. The findings indicated a two-factor structure for the " self-efficacy perception scale related to block-based programming ": "simple block-based programming tasks" consisting of 5 items and "complex block-based programming tasks" consisting of 7 items. At the end of the exploratory factor

¹ Prof. Dr., Hacettepe Üniversitesi, altunar@gmail.com

² Şehit Muzaffer Aydoğdu O.O., i.kasalak@gmail.com

analysis, the first factor of five items explained 11.462% of the total variance, the second factor with seven items explained 46.763% of the total variance. The model was validated by using confirmatory factor analysis.

Keywords: Blok based programming, coding, self-efficacy perception

Summary

The inclusion of block-based programming training in curriculum across the globe has increased the importance of research on coding education. Scratch software developed in the MIT media laboratories and widely used in programming education at primary education level in Turkey has been chosen as a block-based coding tool in order to promote problem solving ability, self-efficacy, motivation, interest increase.

There are studies that emphasize the relationship between perceived self-efficacy and programming performance in the literature (Ramalingam, LaBelle & Wiedenbeck, 2004; Aşkar & Davenport, 2009; Altun & Mazman, 2012; Mazman, 2013; Yükseltürk & Altıok, 2016; Altun & Mazman Akar, 2017). Altun and Mazman Akar (2017) found that there was a significant positive correlation between the programming self-efficacy perception and the programming performance in their study.

In this study, it was aimed to contribute to the literature by developing a "perceived self-efficacy scale related to block-based programming". In addition, perceived self-efficacy scores were examined in terms of various variables in order to examine the factors that affect individual self-efficacy scores related to block-based programming.

It was initially observed that existing self-efficacy scales related to computer programming were not appropriate for block-based programming tasks due to the design characteristics of block-based software. Therefore, a need for a new scale has emerged. During the development of these items, various sources were reviewed including the features related to the programming in Scratch mentioned in an article (Resnick et al., 2009) prepared by the Scratch software developers were taken as references. In addition, some items were adapted from the programming self-efficacy perception scale, developed by Altun & Mazman (2012). Once the item pool was created, face-to-face interviews with 8 students were conducted to necessary corrections were made. This process was repeated 4 times and 14-item scale was finalized.

The study group of the study consisted of 329 students who were attending to four state-funded secondary schools in Ankara, İzmir, and Uşak provinces (Girls = 163, Males = 166). The kurtosis and skewness coefficients were calculated to check the normality assumption for the items. It was observed that the coefficient of skewness of the fourth item is -3.36 and the coefficient of kurtosis is 12.001, that the coefficient of skewness of the fifth item is -3.131 and the coefficient of kurtosis is 10.424, and the items are removed from the scale.

Kaiser-Meyer-Olkin (KMO) and Barlett sphericity tests were performed to determine if the data were appropriate for factor analysis. The data were found to be appropriate for factor analysis (KMO = .887, $\chi^2 = 1835.038$ $p < 0.05$).

The determination of the discriminatory power of the items was made by the item-total test correlation coefficients. According to the analysis results, item-total correlations of the items ranged from 0.491 to 0.702. An exploratory factor analysis was performed using the

principal component analysis technique for the 12-item scale. According to the results of factor analysis; There were five items under the "simple block-based programming tasks" factor and the load values of the items ranged from 0.789 to 0.636. This factor accounted for 11.462% of variability. Under the "complex block-based programming tasks" factor, there were seven items in total, and the load values of the items ranged from 0.824 to 0.578. This factor accounted for 46.763% of variability.

First level confirmatory factor analysis was performed to confirm the structure obtained by the exploratory factor analysis. First of all, the compliance indices obtained by subjecting the model to confirmatory factor analysis were examined. After the proposed modifications were made, all values indicated a good fit index (AGFI= .92, GFI= .95, NFI= .97, CFI= .98, RMSEA= .061, S-RMR= .047). The internal consistency coefficient (Cronbach's alpha) of the scale was found to be .893. The reliability coefficients of the factors (Cronbach's alpha) were found to be .893 for factor 1 and .868 for factor 2.

When examining self-efficacy perception scores related to simple block-based programming and self-efficacy perception scores related to complex block-based programming in terms of various variables, it was first found that there was no statistically significant difference was observed between participants' perceived self-efficacy scores in two sub-factors across gender. This finding is similar to many studies in the field (see Ramalingam & Wiedenbeck, 1998, Byrne & Lyons, 2001, Pillay & Jugoo, 2005, Altun & Mazman, 2013).

There was no significant difference between participants' perceived self-efficacy scores related to simple block-based programming and whether their taking programming lessons with Scratch and writing programs with Scratch before. These findings are in line with the findings of Resnick et al. (2009) that these concepts can be learned easily without special training or support. On the other hand, there was found a significant difference between participants' scores in Simple and Complex Block-Based Programming scores and their taking programming lesson with Scratch and their previous experience with writing codes with Scratch. The difference was higher in those who had taken courses and written codes with Scratch. These findings coincide with the findings that Altun and Mazman (2013). In addition, these findings support the findings of Resnick et al. (2009) that 10th and 11th items in complex block-based programming tasks can be learned with more expert support. These significant differences were also observed in those who were currently enrolled in taking courses and writing codes in Scratch.

Giriş

Papert'in (1980) ilk çalışmalarından code.org, Scratch gibi günümüz çalışmalarına kadar olan süreçte programlama eğitiminin ilköğretim düzeyinde verilebilecek şekilde düzenlenmesi yönünde girişimler başarılı sonuçlar vermeye başlamış ve birçok ülkede programlama eğitimi öğretim programlarına dâhil edilmiştir (örn., Code.org 2016 Annual Report, 2016).

Ülkemizde programlama eğitimi kapsamında yürütülen çalışmalar neticesinde "Problem Analiz ve Çözme Yaklaşımları", "Algoritma ve Strateji Geliştirme (algoritma oluşturma mantığı, sözde kod, akış şemaları vb.)", "Programlama", "Yazılım Projesi Geliştirme, Uygulama ve Yaygınlaştırma" konu başlıkları altında programlama eğitimi 2012 yılında Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programına dâhil edilmiştir (TTKB, 2012). Lise düzeyinde ise 2016 yılı itibarıyla Bilgisayar Bilimi Dersi kapsamında "Problem Çözme ve Algoritmalar",

“Programlama”, “Web Tabanlı Programlama”, “Mobil Programlama” konu başlıklarının yer aldığı öğretim programı hazırlanmıştır (TTKB, 2016). Millî Eğitim Bakanlığı tarafından 2017-2018 eğitim-öğretim yılından itibaren kademeli olarak öncelikle 5. sınıf düzeyinde, 2018-2019 eğitim-öğretim yılından itibaren de 6. sınıf düzeyinde haftada 2 saat zorunlu olarak yürütülmesi kararı alınan bilişim teknolojileri ve yazılım dersinin 2017 yılı Temmuz ayında yayınlanan öğretim programının her iki sınıf düzeyinde de %50’si “Problem Çözme ve Programlama” ünitesi altında “Problem Çözme Kavramları ve Yaklaşımları” ve “Programlama” konu başlıklarını içermektedir (TTKB, 2017).

Blok temelli programlama eğitiminin birçok ülke ile eşzamanlı ülkemizde de öğretim programlarına dahil edilmesi, programlama eğitime ilişkin araştırmaların önemini artırmıştır. MIT medya laboratuvarlarında geliştirilen ve ülkemizde ilköğretim düzeyinde programlama eğitiminde yaygın olarak kullanılan Scratch yazılımının öğrenciler üzerinde problem çözme becerisi, öz-yeterlik, motivasyon, ilgi artışı gibi etkilerine bakan çalışmalar yapıldığı, ancak bunun sınırlı sayıda olduğu görülmüştür (örn. Genç & Karakuş, 2011; Gülbahar & Kalelioğlu, 2014; Kukul & Gökçearslan, 2014; Yükseltürk & Altıok, 2016). Alanyazında bireylerin programlama performanslarını ve öz-yeterlik algılarını yordamaya yönelik çeşitli araçlar bulunmasına rağmen A-12 düzeyinde yaygın olarak kullanılan blok temelli programlamaya ilişkin bir öz-yeterlik algısı ölçeğine rastlanılmamıştır.

İlgili alanyazında bireylerin öz-yeterlik algısı ile programlama performansı arasındaki ilişkiye vurgu yapan çalışmalar bulunmaktadır. (Ramalingam, LaBelle ve Wiedenbeck, 2004; Aşkar ve Davenport, 2009; Altun ve Mazman, 2012; Mazman, 2013; Altun ve Mazman 2013; Yükseltürk ve Altıok, 2016; Altun ve Mazman Akar, 2017). Wiedenber (2005) tarafından 120 üniversite öğrencisi ile programlamaya giriş dersi kapsamında yürütülen çalışmada, programlamaya ilişkin öz-yeterlik algısı programlama performansının anlamlı bir yordayıcısı olarak bulunmuştur. Altun ve Mazman Akar (2017), farklı üniversitelerde eğitim gören ve programlama-1 dersi almış 129 öğrenci ile yürüttükleri, “programlama performansını etkileyen faktörlerin bilişsel tabanlı bireysel farklılıklar temelinde modellenmesi” başlıklı çalışmalarında, programlamaya ilişkin öz-yeterlik algısı ile programlama performansı arasında pozitif yönde anlamlı ilişki olduğu bulgusuna ulaşmışlardır.

Alanyazında blok temelli programlamaya ilişkin bir öz-yeterlik algısı ölçeği bulunmaması nedeniyle, bu çalışma kapsamında “blok temelli programlamaya ilişkin öz-yeterlik algısı ölçeği” geliştirilerek alanyazına katkı sağlanması amaçlanmaktadır. Bunun yanında bireylerin blok temelli programlamaya ilişkin öz-yeterlik algısı puanlarını etkileyen faktörleri incelemek amacıyla çeşitli değişkenler (cinsiyet, daha önce Scratch’le programlama dersi alma, daha önce Scratch’le program yazma, halen Scratch’le programlama dersi alma, halen Scratch’le program yazma) açısından öz-yeterlik algısı puanlarının incelenmesi hedeflenmektedir.

Yöntem

Bu bölümde çalışma grubu, veri toplama araçları ve verilerin analizi ile ilgili bilgiler yer almaktadır.

Çalışma Grubu

Araştırmanın çalışma grubunu, Ankara, İzmir, Uşak illerinde bulunan 4 devlet orta okulunda eğitim gören 329 öğrenci oluşturmaktadır. Söz konusu okullarda eğitim gören

öğrencilerin Scratch'le programlama eğitimi almış ve/veya almakta olduğunun öğretmenleriyle yapılan görüşmeler neticesinde tespit edilmesi ve ilgili okullarda görev yapan bilişim teknolojileri öğretmenlerinin araştırmaya katılmaya gönüllü olmaları nedeniyle elverişli bulunmaları çalışma grubunun seçiminde belirleyici olmuştur. Bu öğrencilere ait demografik verilerden elde edilen betimsel istatistikler Tablo 1'de verilmiştir.

Tablo 1: Çalışma Grubuna Ait Betimsel İstatistikler

| | | N | % |
|---|----------|-----|-------|
| Cinsiyet | Kız | 163 | %49.5 |
| | Erkek | 166 | %50.5 |
| Sınıf Düzeyi | 5. Sınıf | 233 | %70.8 |
| | 6. Sınıf | 61 | %18.5 |
| | 7. Sınıf | 25 | %7.6 |
| | 8. Sınıf | 8 | %3.0 |
| Daha Önce Scratch'le Programlama Dersi Alma | Evet | 203 | %61.7 |
| | Hayır | 126 | %38.3 |
| Halen Scratch'le Programlama Dersi Alma | Evet | 261 | %78.6 |
| | Hayır | 68 | %20.7 |
| Daha Önce Scratch'le Program Yazma | Evet | 209 | %63.5 |
| | Hayır | 120 | %36.5 |
| Halen Scratch'le Program Yazma | Evet | 250 | %76.0 |
| | Hayır | 78 | %23.7 |

Veri Toplama Aracı

Veriler iki bölümden oluşan bir ölçme aracı ile toplanmıştır. Ölçme aracının birinci bölümünü cinsiyet, daha önce Scratch'le programlama dersi alma, daha önce Scratch'le program yazma, halen Scratch'le programlama dersi alma, halen Scratch'le program yazmaya ilişkin soruların yer aldığı demografik bilgi kısmı oluşturmuştur. İkinci bölümde ise blok temelli programlamaya ilişkin 12 maddeden oluşan sorular yer almaktadır. Ölçekte toplamda 12 madde yer almakta olup sorular 5'li Likert tipinde hazırlanmış ve 1- Hiç Güvenmiyorum, 2- Biraz Güveniyorum, 3- %50 / %50, 4- Oldukça Güveniyorum, 5- Tamamen Güveniyorum belirtecek şekilde puanlanmıştır.

Ön Uygulama ve Madde Havuzunun Oluşturulması

Madde havuzunun oluşturulması amacıyla ilgili alanyazında bulunan Altun ve Mazman (2012) tarafından Türkçeye uyarlaması yapılmış olan programlamaya ilişkin öz-yeterlik algısı ölçeğinin blok temelli programlamaya ilişkin öz-yeterlik algısı ölçeği olarak uyarlama

çalışmasının yapılmasına karar verilmiştir. Ancak, blok temelli programlama dillerinin programlama eğitiminin genç insanlara verilmesinin önündeki engelleri kaldırmaya yönelik özel tasarımları nedeniyle, ölçek maddelerinin büyük çoğunluğunun blok temelli programlamaya yönelik uyarlanma çalışmasının mümkün olmadığı görülmüştür. Maddelerin geliştirilmesi sürecinde, blok temelli programlama dillerinden birisi olan Scratch yazılımını geliştiren MIT Media Lab tarafından hazırlanan bir makalede (Resnick ve diğerleri, 2009) belirtilen blok temelli programlamaya ilişkin özellikler referans alınmıştır. Bunun yanı sıra Altun ve Mazman (2012) tarafından Türkçeye uyarlaması yapılmış olan programlamaya ilişkin öz-yeterlik algısı ölçeğinde bulunan ve uyarlaması yapılabilen beş maddede hazırlanan ölçeğe dâhil edilmiştir. Madde geliştirilme sürecine dahil edilecek programlamaya ilişkin ifadelerin tespitinde blok temelli programlama dillerinin ortak özelliklerinin dikkate alınmasına özen gösterilmiştir. MIT Media Lab tarafından hazırlanan makalede (Resnick ve diğerleri, 2009) belirtilen özelliklerden Scratch'te bulunan ancak blok temelli programlama dillerinin ortak özelliklerinden olmayan çizim, ses kaydı, sosyal paylaşım gibi özellikler çalışmaya dahil edilmemiştir. Hazırlanan bu öz-yeterlik algısı ölçeği araştırmacı bilişim teknolojileri öğretmeninin görev yaptığı devlet okulunda eğitim gören, blok temelli programlama becerisinin iyi düzeyde olduğu gözlemlenen 2 erkek - 2 kız öğrenciye ve blok temelli programlama becerisinin düşük düzeyde olduğu gözlemlenen 2 erkek - 2 kız öğrenciye olmak üzere toplam 8 öğrenciye uygulanmıştır. Uygulamanın ardından öğrencilerle birebir görüşme tekniği ile soruları anlayıp anlamadıklarını tespit etmeye yönelik görüşmeler gerçekleştirilmiştir. Anlaşılmayan, net olmayan ya da eksik ifade edilen maddelere yönelik yapılan görüşmelerle elde edilen veriler neticesinde ölçekte düzeltmelere gidilmiştir. Öğrencilerin blok temelli programlama, programlama, kodlama gibi ifadeleri anlamakta güçlük çektikleri tespit edilmiştir. Bu nedenle birçok ülkede olduğu gibi ülkemizde de yaygın olarak kullanılan blok temelli programlama dili olan Scratch'in, blok temelli programlama ifadesi yerine kullanılmasına karar verilmiştir. Bu şekilde ortaokul düzeyindeki öğrencilerin, ön pilot uygulamalarında, maddeleri daha anlaşılır bulduğu görülmüştür. Yeniden düzenlenen sorularla aynı çalışma farklı 8 öğrenci ile tekrarlanmıştır. Alınan dönütler neticesinde tekrar düzeltmeler yapılmıştır. Bu süreç 4 defa tekrarlanmış ve ön çalışma neticesinde ölçeğe 14 maddelik son şekli verilmiştir.

Verilerin Toplanması

Ölçek maddeleri oluşturulduktan sonra, ölçek çevrimiçi erişim için hazır hale getirilmiştir. Öğrencilerin Scratch'le programlama eğitimi almış veya almakta olduğunun tespit edilmesi ve ilgili okullarda görev yapan bilişim teknolojileri öğretmenlerinin araştırmaya gönüllü olmaları nedeniyle elverişli bulunan Ankara, İzmir, Uşak illerindeki 4 devlet okulunda eğitim gören 329 ortaokul öğrencisi ile ölçek uygulanmıştır. "Google Formlar" ile hazırlanan çevrimiçi ölçek kullanılarak toplanan veriler SPSS 24 ve Lisrel 8.7 programlarında analiz edilmek üzere Microsoft Excel (.xls) formatında yerel bilgisayarın sabit diskine kaydedilmiştir.

Verilerin Analizi

Verilerin analizine geçilmeden önce kayıp verilen tespit çalışması yapılmıştır. Öğrencilerin ölçeklere verdikleri yanıtlarda kayıp verilerin ilgili maddelere verilen toplam yanıtlar içerisindeki oranı %5'ten küçük olduğu tespit edilmiştir. Bu doğrultuda, tespit edilen eksik veriler o maddeye ilişkin ortalamayı değiştirmemesi nedeniyle ilişkili maddenin ortalamasıyla doldurulmuştur.

Maddelere ilişkin normallik varsayımını kontrol etmek için basıklık ve çarpıklık katsayıları hesaplanmıştır. 4. maddenin çarpıklık katsayı değerinin -3.36 ve basıklık katsayı değerinin 12.001 olduğu, 5. maddenin çarpıklık değerinin -3.131 ve basıklık katsayı değerinin 10.424 olduğu görülmüştür ve söz konusu maddeler ölçekten çıkartılmıştır.

Verilerin faktör analizi için uygun olup olmadığının tespit edilmesi amacıyla Kaiser-Meyer-Olkin (KMO) ve Barlett Küresellik testleri yapılmıştır. KMO örneklem yeterlilik indeksi, %60'dan yüksek olması ve Barlett küresellik testinin anlamlı çıkması (< 0.05) verilerin faktör analizi için uygun olduğunu göstermektedir (Büyüköztürk, 2007).

Bu çalışmadaki verilerin KMO ve Bartlett küresellik testi sonuçları Tablo 2'de gösterilmektedir.

Tablo 2: KMO and Bartlett Test Sonuçları

| | | |
|--|------------------|----------|
| Kaiser-Meyer-Olkin Örneklem Yeterliği Ölçümü | | .887 |
| ωBartlett Küresellik Testi | Yaklaşık Ki-Kare | 1835.038 |
| | df | 66 |
| | p | .000 |

Ölçeğin yapı geçerliliği için açımlayıcı ve doğrulayıcı faktör analizleri yapılmıştır. Doğrulayıcı faktör analizinde kurulan modelin iyilik uyumunun değerlendirilmesinde iyilik uyum indekslerinden AGFI, IFI, GFI, NNFI, NFI, CFI, RMSEA ve S-RMR değerleri esas alınmıştır. Güvenirlik çalışması için ölçeğin iç tutarlılık katsayısı (Cronbach alpha) hesaplanmıştır. Grupların çeşitli değişkenler açısından bağımlı değişken verileri üzerinden karşılaştırıldığı analizler için bağımsız-örneklem t testi yapılmıştır.

Bulgular

Madde Geçerliliğine İlişkin Bulgular

Maddelerinin ayırt edicilik gücünün belirlenmesi, madde toplam test korelasyon katsayıları ile yapılmıştır. Bu yöntem ile testin her bir maddesinden alınan puan ile testin tamamından alınan puan arasındaki ilişki gösterilmektedir. Madde toplam korelasyonlarının 0.3'ten küçük olması o maddenin diğer maddelerle aynı yapıyı ölçmediğini, yani maddenin ölçeğin geri kalanı korelasyonunun zayıf olduğunu gösterir (Field, 2005).

Analiz sonucunda elde edilen verilere göre ölçeğe ilişkin hiçbir maddenin madde toplam korelasyonu 0.3'ün altında olmayıp, toplamda 12 maddenin madde toplam korelasyonları 0.491 - 0.702 arasında değişmektedir.

Yapı Geçerliliğine İlişkin Bulgular

İlk olarak ölçeğin tümüne toplamda 12 maddenin faktör yüklerini ve faktör yapılarını belirlemek amacıyla açımlayıcı faktör analizi yapılmıştır. Analiz sonucunda faktörler altındaki yük değerleri incelenmiş, ortaya çıkan faktörler yüklerine göre 12 maddenin öz değeri 1'den büyük olan iki faktör altında toplandığı doğrulanmıştır. 3., 6., 7., 8. ve 9. maddeler birinci faktörde toplanırken 1., 2., 10., 11., 12., 13. ve 14. maddeler ikinci faktörde toplanmıştır.

Bu çalışma kapsamında geliştirilen blok temelli programlamaya ilişkin öz-yeterlik algısı ölçeğinde, ölçek maddelerinin altında toplandığı iki faktörün isimlendirilmesinde, Resnick ve diğerleri'nin (2009) çalışmasında blok temelli programlamaya yönelik kavramların isimlendirmesi ve Altun ve Mazman (2012) tarafından yapılan çalışmada kullanılan faktör isimlendirmesi referans alınmıştır. Açıklayıcı faktör analizinden elde edilen 1. faktör altında toplanan maddeler Resnick ve diğerleri'nin (2009) makalesinde "temel programlama kavramlarına" yönelik görevlere karşılık gelmektedir. 2. faktör altında toplanan maddelerden değişken oluşturma ve kullanmaya yönelik görevleri içeren 10. ve 11. maddeler "daha fazla uzman desteği ile öğrenilebilecek kavramlara" yönelik görevlere karşılık gelmekte iken 1., 2., 12., 13. ve 14. maddeler ise Altun ve Mazman (2012) tarafından Türkçeye uyarlanan programlamaya ilişkin öz-yeterlik algısı ölçeğindeki "karmaşık programlama görevlerine" ilişkin faktör altında toplanan maddelerin blok temelli programlamaya yönelik küçük yaş grubunun anlayabileceği ifadelerle sahip uyarlaması şeklindedir.

Yapılan değerlendirme sonucunda, blok temelli programlama ilişkin öz-yeterlik algısı ölçeğine ait iki faktörün, 5 maddeden oluşan (3., 6., 7., 8. ve 9.) birinci faktörü "basit blok temelli programlama görevleri" olarak, 7 maddeden oluşan (1., 2., 10., 11., 12., 13. ve 14.) ikinci faktörü ise "karmaşık blok temelli programlama görevleri" olarak isimlendirilmiştir. Ölçek 5'li likert tipi olduğundan basit blok temelli programlamaya ilişkin öz-yeterlik algısı için alınabilecek maksimum puan 25, minimum puan ise 5, karmaşık blok temelli programlamaya yönelik öz-yeterlik algısı için alınabilecek maksimum puan 35, minimum puan ise 5'tir.

Açıklayıcı Faktör Analizine İlişkin Bulgular

Blok temelli programlamaya ilişkin öz-yeterlik algısı ölçeğinin son hali verilen 12 maddelik formuna boyutların belirlenmesi amacıyla açıklayıcı faktör analizi uygulanmış; faktör çıkarım tekniği olarak ise temel bileşenler analizi tekniği kullanılmıştır. Analiz sonucunda faktörler altındaki yük değerleri incelenmiş, ortaya çıkan faktörler yüklerine göre öz değeri (eigenvalue) 1'den büyük olan (5.612, 1.375) iki faktör altında toplandığı bulunmuştur.

Programlamaya ilişkin öz-yeterlik algısı ölçeğinin temel bileşenler analizi sonuçları Tablo 3'te verilmektedir.

Tablo 3: Programlamaya İlişkin Öz-Yeterlik Algısı Ölçeği Temel Bileşenler Analizi Sonuçları

| Faktör | Madde | Faktör Yükleri | |
|---|--|----------------|--------------|
| | | 1 | 2 |
| Basit Blok Temelli Programlama Görevleri | M3. Bir karaktere herhangi bir hareket vermek istediğimde, Scratch'te bunu nereden yapabileceğimi bilirim. | 0.636 | 0.203 |
| | M6. Sahnedeki karakteri istediğim hızda hareket ettirebilirim. | 0.753 | 0.165 |
| | M7. Sahnedeki karakteri sürekli hareket ettirebilirim. | 0.789 | 0.189 |
| | M8. Scratch'te bir karakterin görünümünü (kostüm, renk, boyut, konuşma gibi) bir koşula bağlı olarak (örneğin: eğer ise) değiştirebilirim. | 0.769 | 0.234 |
| | M9. Scratch'te bir karakterin hareketini (hızı, yönü, konumu gibi) bir koşula bağlı olarak (örneğin: eğer ise) değiştirebilirim. | 0.690 | 0.349 |
| Karmaşık Blok Temelli Programlama Görevleri | M1. Scratch'te yazılmış bir program (yazılar) gördüğümde, çalıştırıldığında neler olacağını söyleyebilirim. | 0.459 | 0.535 |
| | M2. Başkası tarafından hazırlanan bir programı (yazılarını) okuyup anlayabilirim. | 0.423 | 0.578 |
| | M10. Bir oyunda kullanıcının elde ettiği puan değerinin tutulacağı bir değişken oluşturabilirim. | 0.164 | 0.725 |
| | M11. Bir oyunda istenilenler başarıldıkça "Puan" veya "Skor" değerinin arttığı veya azaldığı bir program hazırlayabilirim. | 0.142 | 0.824 |
| | M12. İstenilenler açıkça tanımlandığında oldukça karmaşık ve uzun kodlardan (yazılardan) oluşan bir oyun hazırlayabilirim. | 0.277 | 0.767 |
| | M13. Scratch'te hazırlanan bir programdaki hataları bulabilirim. | 0.176 | 0.726 |
| | M14. Scratch'te hazırlanan bir programdaki hataları düzeltip çalışabilir hale getirebilirim. | 0.330 | 0.670 |

Tablo 3'te görüldüğü üzere faktör analizi sonuçlarına göre; ilk faktörün altında 5 madde yer almakta ve maddelerin yük değerleri 0.789 ile 0.636 arasında değişmektedir. Faktör yapısında yer alan maddelerin içeriğine göre bu faktör "basit blok temelli programlama görevleri" olarak adlandırılmıştır. Bu faktör değişkenliğinin %11.462'ini açıklamaktadır. Diğer faktörde ise toplamda 7 madde yer almakta, maddelerin yük değerleri 0.824 ile 0.578 arasında değişmektedir. Bu faktör ise "Karmaşık blok temelli programlama görevleri" şeklinde adlandırılmıştır. Bu faktör değişkenliğinin %46.763'ünü açıklamaktadır. Açıklanan toplam varyans ise %58.225 olarak hesaplanmıştır.

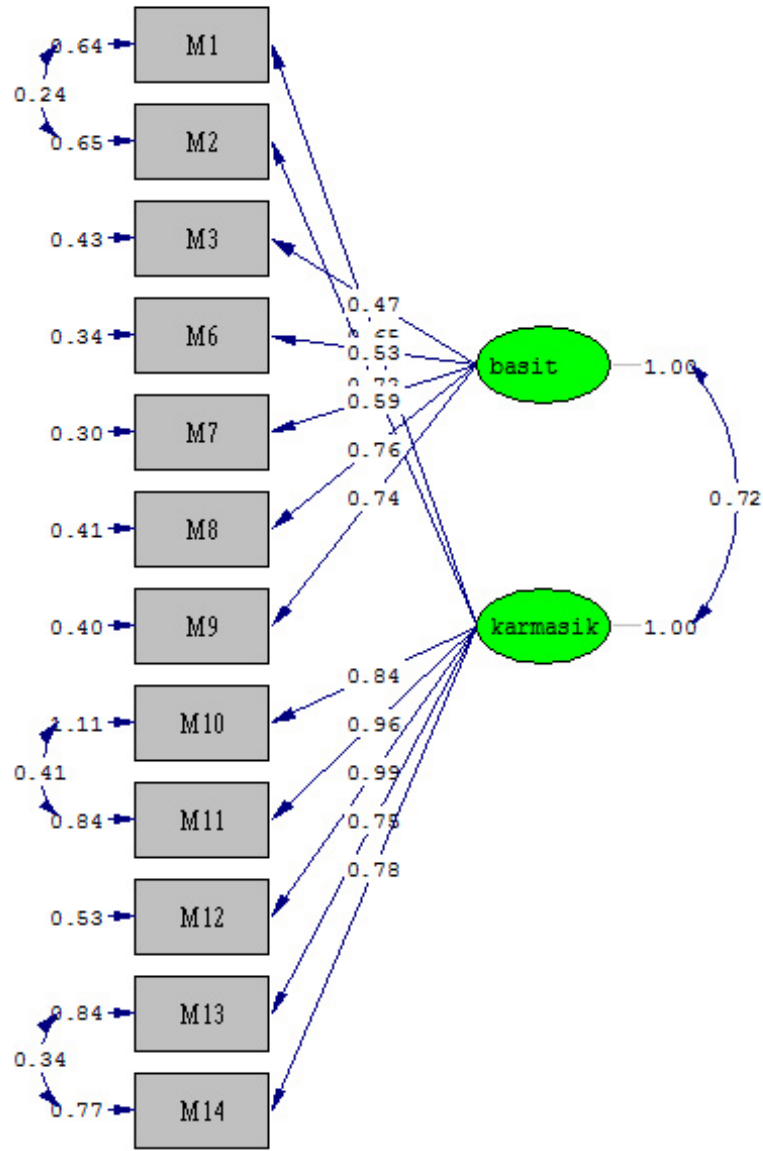
Doğrulayıcı Faktör Analizine İlişkin Bulgular

Açımlayıcı faktör analizi ile iki faktör (“basit blok temelli programlama görevleri” ve “karmaşık blok temelli programlama görevleri”) ve 12 maddeden (gösterge değişken) elde edilen yapının doğrulanması amacıyla Lisrel 8.70 programı kullanılarak birinci düzey doğrulayıcı faktör analizi yapılmıştır. İlk olarak iki gizil değişken (“basit blok temelli programlama görevleri ve “karmaşık blok temelli programlama görevleri”) ve 12 gösterge değişken (M1, M2.....M12) ile oluşturan model doğrulayıcı faktör analizine tabi tutularak elde edilen uyum indeksleri incelenmiştir. Elde edilen iyilik uyum indeksleri uygun değer aralığında bulunmadığından, önerilen modifikasyon indeksleri incelenmiş ve aynı faktörün açıkladığı maddeler olan “M1 ve M2”, “M10 ve M11” ve “M13 ve M14” arasındaki hata kovaryansları serbest bırakılmıştır. Modifikasyonlar yapıldıktan sonra program tekrar çalıştırılarak analiz edilmiştir. Analiz sonucunda elde edilen değerler Tablo 4’te gösterilmiştir.

Tablo 4: Birinci Düzey Doğrulayıcı Faktör Modeli Uyum İndeksleri

| Uyum Ölçütleri | Mükemmel Uyum | Kabul Edilebilir Uyum | Model Değeri |
|----------------|--------------------------|-----------------------|--------------|
| RMSEA | $0 < RMSEA < 0.05$ | $0.05 < RMSEA < 0.08$ | 0.061 |
| S-RMR | $0 \leq S-RMR \leq 0.05$ | $0.05 < S-RMR < 0.1$ | 0.047 |
| NNFI | $0.97 \leq NNFI \leq 1$ | $0.95 < NNFI < 0.97$ | 0.98 |
| NFI | $0.97 \leq NFI \leq 1$ | $0.95 < NFI < 0.97$ | 0.97 |
| CFI | $0.97 \leq CFI \leq 1$ | $0.95 < CFI < 0.97$ | 0.98 |
| GFI | $0.95 \leq GFI \leq 1$ | $0.90 < GFI < 0.95$ | 0.95 |
| AGFI | $0.90 \leq AGFI \leq 1$ | $0.85 < AGFI < 0.90$ | 0.92 |
| IFI | $0.95 \leq IFI \leq 1$ | $0.90 < IFI < 0.95$ | 0.98 |

Tablo 4’te görüldüğü üzere modelin tüm değerlerinin oldukça iyi uyum gösterdiği bulunmuştur. Birinci düzey doğrulayıcı faktör analizinin bağlantı diyagramı Şekil 1’de yer almaktadır.



Şekil 1: Birinci Düzey Doğrulayıcı Faktör Analizi Bağlantı Diyagramı (standart katsayılar)
Güvenirlilik Analizine İlişkin Bulgular

Ölçeğin tümünün iç tutarlık katsayısı (Cronbach's alpha) .893 olarak bulunmuştur. Her bir faktör için hesaplanan güvenirlilik katsayıları Tablo 5'te gösterilmektedir.

Tablo 5: Faktörlere Ait Güvenirlik Katsayıları

| Faktör | Madde | α |
|---|-------|----------|
| Basit Blok Temelli Programlama Görevleri | M3 | 0.828 |
| | M6 | |
| | M7 | |
| | M8 | |
| | M9 | |
| Karmaşık Blok Temelli Programlama Görevleri | M1 | 0.868 |
| | M2 | |
| | M10 | |
| | M11 | |
| | M12 | |
| | M13 | |
| | M14 | |

Tablo 5'te görüldüğü gibi faktörlere ait güvenilirlik katsayılarının tümü (Cronbach's alpha) 0.8'in üzerinde olup bu sonuç ölçme aracının güvenilirliğinin oldukça iyi olduğunu göstermektedir (George ve Mallery, 2003).

Ölçeğin Uygulanmasından Elde Edilen Puanlara İlişkin Çeşitli Değişkenler Açısından Bulgular

Bireylerin programlamaya ilişkin öz-yeterlik algısı puanlarını etkileyen faktörleri belirlemek amacıyla çeşitli değişkenler (cinsiyet, daha önce Scratch'le programlama dersi alma, daha önce Scratch'le program yazma, halen Scratch'le programlama dersi alma, halen Scratch'le program yazma) açısından öz-yeterlik algısı puanları incelenmiştir.

Bireylerin cinsiyet, daha önce Scratch'le programlama dersi alma, daha önce Scratch'le program yazma, halen Scratch'le programlama dersi alma, halen Scratch'le program yazmaya ilişkin ölçek puanlarının ortalamaları ve standart sapmaları Tablo 6'da verilmiştir.

Tablo 6: Demografik Verilere İlişkin Ortalama ve Standart Sapmalar

| | | Basit Blok Temelli Programlamaya İlişkin Öz-Yeterlik Algısı (En yüksek 25) | | | Karmaşık Blok Temelli Programlamaya İlişkin Öz-Yeterlik Algısı (En yüksek 35) | |
|--|-------|--|----------|----------------|---|----------------|
| | | N | Ortalama | Standart Sapma | Ortalama | Standart Sapma |
| Cinsiyet | Kız | 163 | 22.349 | 3.630 | 26.276 | 6.139 |
| | Erkek | 166 | 22.410 | 3.131 | 26.405 | 6.408 |
| Daha Önce Scratch'le Programlama Dersi Aldım | Evet | 203 | 22.393 | 3.449 | 26.955 | 6.301 |
| | Hayır | 126 | 22.359 | 3.287 | 25.353 | 6.107 |
| Daha Önce Scratch'le Program Yazdım | Evet | 209 | 22.548 | 3.288 | 27.193 | 6.071 |
| | Hayır | 120 | 22.088 | 3.536 | 24.858 | 6.351 |
| Halen Scratch'le Programlama Dersi Alıyorum | Evet | 261 | 22.621 | 2.968 | 26.983 | 5.778 |
| | Hayır | 68 | 21.46 | 4.55 | 23.88 | 7.42 |
| Halen Scratch'le Program Yazıyorum | Evet | 250 | 22.811 | 2.788 | 27.340 | 5.790 |
| | Hayır | 79 | 21.02 | 4.56 | 23.18 | 6.69 |

İlk olarak bireylerin basit blok temelli programlamaya ilişkin öz-yeterlik algısı puanlarının ve karmaşık blok temelli programlamaya ilişkin öz-yeterlik algısı puanlarının cinsiyete göre farklılaşıp farklılaşmadığını ortaya koymak amacıyla bağımsız örneklem t testi yapılmıştır. Bağımsız örneklem t testi sonuçlarına göre bireylerin basit blok temelli programlamaya ilişkin öz-yeterlik algısı puanlarının ($\bar{X}_{kız} = 22.349$; $\bar{X}_{erkek} = 22.410$; $t = 0.164$, $p = 0.870$) ve karmaşık blok temelli programlamaya ilişkin öz-yeterlik algısı puanlarının ($\bar{X}_{kız} = 26.276$; $\bar{X}_{erkek} = 26.405$; $t = 0.187$, $p = 0.852$) cinsiyete göre farklılaşmadığı bulunmuştur.

Daha önce Scratch'le programlama dersi alan ve almayan öğrenciler arasında basit blok temelli programlamaya ilişkin öz-yeterlik algısı puanlarının anlamlı derecede farklılaşmadığı bulunmuştur ($\bar{X}_{dersalan} = 22.393$; $\bar{X}_{dersalmayan} = 22.359$; $t = 0.87$, $p = 0.931$). Daha önce Scratch'le programlama dersi alan öğrencilerin karmaşık blok temelli programlamaya ilişkin öz-yeterlik algısı puanları daha önce Scratch'le programlama dersi almayan öğrencilere göre anlamlı derecede yüksek olarak bulunmuştur ($\bar{X}_{dersalan} = 26.955$; $\bar{X}_{dersalmayan} = 25.353$; $t = 2.268$, $p = 0.024$).

Daha önce Scratch'le program yazan ve yazmayan öğrenciler arasında basit blok temelli programlamaya ilişkin öz-yeterlik algısı puanlarının anlamlı derecede farklılaşmadığı bulunmuştur ($\bar{X}_{prgyazan} = 22.548$; $\bar{X}_{prgyazmayan} = 22.088$; $t = 1.188$, $p = 0.236$). Daha önce Scratch'le program yazan öğrencilerin karmaşık blok temelli programlamaya ilişkin öz-yeterlik algısı

puanları daha önce Scratch'le program yazmayan öğrencilere göre anlamlı derecede yüksek olarak bulunmuştur ($\bar{X}_{\text{prgyazan}}=27.193$; $\bar{X}_{\text{prgyazmayan}}=24.858$; $t=3.301$, $p=0.01$).

Halen Scratch'le programlama dersi alan öğrencilerin hem basit hem de karmaşık blok temelli programlamaya ilişkin öz-yeterlik algısı puanları halen Scratch'le programlama dersi almayan öğrencilere göre anlamlı derecede yüksek olarak bulunmuştur (basit: $\bar{X}_{\text{halendersalan}}=22.621$; $\bar{X}_{\text{halendersalmayan}}=21.46$; $t=2.551$, $p=0.011$; karmaşık: $\bar{X}_{\text{halendersalan}}=26.983$; $\bar{X}_{\text{halendersalmayan}}=23.88$; $t=3.705$, $p=0.00$).

Halen Scratch'le program yazan öğrencilerin hem basit hem de karmaşık blok temelli programlamaya ilişkin öz-yeterlik algısı puanları halen Scratch'le program yazmayan öğrencilere göre anlamlı derecede yüksek olarak bulunmuştur (basit: $\bar{X}_{\text{halenprgyazan}}=22.811$; $\bar{X}_{\text{halenprgyazmayan}}=21.02$; $t=4.218$, $p=0.00$; karmaşık: $\bar{X}_{\text{halenprgyazan}}=27.340$; $\bar{X}_{\text{halenprgyazmayan}}=23.18$; $t=5.355$, $p=0.00$).

Sonuç ve Tartışma

A-12 düzeyinde programlama eğitiminin öğretim programlarına dahil edilmesi kapsamında birçok ülkede yürütülen çalışmalarla eşzamanlı olarak ülkemizde de programlama eğitimi öğretim programlarına dâhil edilmiştir (bkz. TTKB, 2012; TTKB, 2016; TTKB, 2017). Alanyazın taramasında A-12 düzeyinde yaygın olarak kullanılan blok temelli programlamaya ilişkin bir öz-yeterlik algısı ölçeğine rastlanılmamış ve bu çalışma kapsamında geliştirilen blok temelli programlamaya ilişkin öz-yeterlik algısı ölçeği ile alanyazına katkı sağlanması amaçlanmıştır.

Ayrıca geliştirilen ölçme aracı ile çeşitli illerde bulunan 4 okuldan elde edilen blok temelli programlamaya ilişkin öz-yeterlik algısı puanlarının çeşitli değişkenler açısından (cinsiyet, daha önce Scratch'le programlama dersi alma, daha önce Scratch'le program yazma, halen Scratch'le programlama dersi alma, halen Scratch'le program yazma) farklılaşıp farklılaşmadığı ortaya konulmuştur.

Basit blok temelli programlamaya ilişkin öz-yeterlik algısı puanları ve karmaşık blok temelli programlamaya ilişkin öz-yeterlik algısı puanları çeşitli değişkenler açısından incelendiğinde, ilk olarak cinsiyet bakımından istatistiksel olarak anlamlı bir fark olmadığı ortaya çıkmıştır. Bu bulgu ilgili alan yazındaki birçok araştırma ile benzerlik göstermektedir (Ramalingam & Wiedenbeck, 1998; Byrne & Lyons, 2001; Pillay & Jugoo, 2005; Altun ve Mazman, 2013).

Daha önce Scratch'le programlama dersi alma ve daha önce Scratch'le program yazma durumlarına göre öğrencilerin basit blok temelli programlamaya ilişkin öz-yeterlik algılarında anlamlı bir fark bulunmamıştır. Basit blok temelli programlama görevlerine ilişkin maddelerde yer alan kavramlara ilişkin bu bulgular, Resnick ve diğerlerinin (2009) çalışmasında bu kavramların özel bir eğitim veya desteğe gerek kalmadan kolayca öğrenilebileceği yönündeki bulguları ile paralellik göstermektedir. Daha önce Scratch'le programlama dersi alan öğrencilerin karmaşık blok temelli programlamaya ilişkin öz-yeterlik algısı puanları daha önce Scratch'le programlama dersi almayan öğrencilere göre, daha önce Scratch'le program yazan öğrencilerin karmaşık blok temelli programlamaya ilişkin öz-yeterlik algısı puanları daha önce Scratch'le program yazmayan öğrencilere göre anlamlı derecede yüksek olarak bulunmuştur. Bu bulgular, Altun ve Mazman'ın (2013) öz-yeterlik algısı puanlarına yönelik yaptıkları çalışmada, programlama deneyiminin artışı ile öz-yeterlik puanının anlamlı olarak arttığı yönünde elde ettikleri bulgularla örtüşmektedir. Ayrıca bu bulgular, karmaşık blok temelli

programlama görevleri arasında yer alan 10. ve 11. maddelerin, Resnick ve diğerleri (2009) tarafından daha fazla uzman desteği ile öğrenilebilecek nitelikte olduğu yönündeki tespitlerini de desteklemektedir.

Halen Scratch'le programlama dersi alan öğrencilerin halen Scratch'le programlama dersi almayan öğrencilere göre, halen Scratch'le program yazan öğrencilerin halen Scratch'le program yazmayan öğrencilere göre hem basit hem de karmaşık blok temelli programlamaya ilişkin öz-yeterlik algısı puanları anlamlı derecede yüksek olarak bulunmuştur. Bu bulgular, programlama dersi alma ile doğru orantılı olarak bireylerin programlamaya ilişkin öz-yeterlik algılarının da arttığı yönündeki bulguları (bkz. Ramalingam ve Wiedenbeck, 1998; Jegede, 2009; Altun ve Mazman, 2013) destekler niteliktedir.

Ölçek maddelerinin geliştirme sürecinde ortaokul düzeyindeki öğrencilerin blok temelli programlama, programlama, kodlama gibi ifadeleri anlamakta güçlük çektikleri tespit edilmiştir ve bu ifadelerin yerine blok temelli programlama eğitiminde yaygın olarak kullanılan "Scratch" ifadesinin kullanılması tercih edilmiştir. Ölçek geliştirme çalışmasının yürütüldüğü 2015-2016 eğitim öğretim yılı itibarıyla 2013 yılında yürürlüğe giren bilişim teknolojileri ve yazılım dersi öğretim programı yürürlükteydi. Söz konusu öğretim programında programlamaya ilişkin ünitenin ancak öğretim programının sonunda kısa bir ünite olarak yer alması "programlama" ifadesinin öğrenciler tarafından anlaşılmasında etkili olabilir. 2017-2018 eğitim-öğretim yılında yürürlüğe giren öğretim programının %50'sinin "Problem Çözme ve Programlama" ünitesine ayrılmış olması nedeniyle blok temelli programlama eğitimi almış olan ortaokul öğrencisi sayısının artacağı düşünüldüğünde, daha sonraki çalışmalarda daha geniş bir örnekleme ulaşma imkanının mümkün olabileceği öngörülmektedir.

Kaynakça

- 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.
- Altun, A., & Mazman, S. G. (2013). Programlama-1 dersinin BÖTE bölümü öğrencilerinin programlamaya ilişkin öz yeterlilik algıları üzerine etkisi. *Journal of Instructional Technologies & Teacher Education*, 2(3), 24-29.
- Aşkar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for java programming. *The Turkish Online Journal of Educational Technology – TOJET January*, 8(1).
- Byrne, P., & Lyons, G. (2001). The effect of student attributes on success in programming. *Proceedings of ITICSE*. 49-52.
- Code.org 2016 Annual Report. (2016). *Global computer science education*. Kasım 21, 2017 tarihinde <https://goo.gl/bHifH0> adresinden alındı
- Field, A. (2005). *Discovering statistics using SPSS* (2 b.). London: Sage.
- Genç, Z., & Karakuş, S. (2011). Tasarımla öğrenme: Eğitsel bilgisayar oyunları tasarımında Scratch kullanımı. *5th International Computer & Instructional Technologies Symposium (ICITS)*. Elazığ.
- George, D., & Mallery, P. (2003). *SPSS for Windows step by step: A simple guide and reference. 11.0 update* (4 b.). Boston: Allyn & Bacon.
- Gülbahar, Y., & Kalelioğlu, F. (2014). The effects of teaching programming. *Informatics in Education-An International Journal*, 13(1), 33-50.
- Kline, R. (2005). *Structural Equation Modeling, 2nd ed.* New York, NY: The Guilford Press.
- Kukul, V., & Gökçearslan, Ş. (2014). Scratch ile programlama eğitimi alan öğrencilerin problem çözme becerilerinin incelenmesi. *8. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu*, (s. 58-63). Edirne.
- Mazman Akar, S. G., & Altun, A. (2017). Individual differences in learning computer programming: A social cognitive approach. *Contemporary Educational Technology*, 8(3), s. 195-213.
- Mazman, S. G. (2013). *Programlama performansını etkileyen faktörlerin bilişsel tabanlı bireysel farklılıklar temelinde modellenmesi*. Ankara: Hacettepe Üniversitesi.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Inc., Publishers.
- Pillay, N., & Jugoo, V. (2005). An investigation into student characteristics affecting novice programming performance. *ACM SIGCSE Bulletin*, 37(4). doi:10.1145/1113847.1113888

- 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), 365-379.
- Ramalingam, V., LaBelle, D., & Wiedenbeck, S. (2004). Self-efficacy and mental models in learning to program. *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science*. Leeds, United Kingdom.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., . . . Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11).
- Rusk, N., Resnick, M., Berg, R., & Pezalla-Granlund, M. (2008). New pathways into robotics: Strategies for broadening participation. *Journal of Science Education and Technology*, 17, 59-69.
- Schermelleh-Engel, K., & Moosbrugger, H. (2003). Evaluating the fit of structural equation models: Test of significance and descriptive goodness-of-fit measures. *Methods of Psychological Research Online*, 8(2), 23-74.
- Sümer, N. (2000). Yapısal eşitlik modelleri: Temel kavramlar ve örnek uygulamalar, 3(6). 49-74. Aralık. *Türk Psikoloji Yazıları.*, 3(6), 49-74.
- TTKB. (2012). *Bilişim teknolojileri ve yazılım dersi (5, 6, 7 ve 8. sınıflar) öğretim programı*. Ankara: MEB.
- TTKB. (2016). *Ortaöğretim bilgisayar bilimi dersi (kur 1, kur 2) öğretim programı*. Ankara: MEB.
- TTKB. (2017). *Bilişim teknolojileri ve yazılım dersi öğretim programı*. Ankara: MEB.
- Wiedenbeck, S. (2005). Factors Affecting the Success of Non-Majors in Learning to Program. *First international workshop on Computing education research*. Seattle, WA, USA.
- Yükseltürk, E., & Altıok, S. (2016). An investigation of the effects of programming with scratch. *British Journal of Educational Technology*. doi:10.1111/bjet.12453
- Yükseltürk, E., & Altıok, S. (2016). Investigation of pre-service information technology teachers' game projects prepared with Scratch. *SDU International Journal of Educational Studies*, 3(1), 59-66.