



| Araştırma Makalesi / Research Article |

YAP Programlama Öğretim Modelinin Öğrencilerin Bilgi-İşlemsel Düşünme Becerilerine Etkisi¹

The Effects of YAP Programming Teaching Model on Students' Computational Thinking Skills

Ali Kürşat Erümit², Güven Şahin³, Hasan Karal⁴

Anahtar Kelimeler

programlama öğretimi
bilgi-işlemsel düşünme
yap modeli
scratch

Keywords

programming
instruction
computational thinking
yap model
scratch

Başvuru Tarihi/Received

11.09.2019

Kabul Tarihi /Accepted

29.03.2020

Öz

Bu çalışmada YAP öğretim modeli kullanılarak işlenen programlama derslerinin öğrencilerin bilgi-işlemsel düşünme becerisine etkisini belirlemek amaçlanmıştır. Araştırma, Trabzon ilindeki bir ortaokulda 6. sınıfta öğrenim görmekte olan 38 öğrenci ile 9 hafta boyunca durum çalışması yöntemi ile yürütülmüştür. Veri toplama aracı olarak Bilgisayarca Düşünme Ölçeği kullanılmıştır. İstatistiksel analiz sonuçlarına bakıldığında öğrencilerin bilgi-işlemsel düşünme becerilerinde anlamlı fark ortaya çıkmıştır. Bu farkın ortaya çıkmasındaki etkenin Scratch aracı ile birlikte kullanılan YAP modeli olduğu düşünülmektedir. Bu nedenle programlama öğretimi sürecinde blok tabanlı görsel programlama araçları ile birlikte YAP modelinin kullanılması önerilmektedir.

Abstract

In this study, it is aimed to determine the effect of programming courses that are processed using YAP teaching model on students' computational thinking skills. The study was carried out with case study method for 9 weeks with 38 students in 6th grade in a secondary school in Trabzon. Computational Thinking Scale was used as data collection tool. When the results of statistical analysis were examined, a significant difference was found in students' computational thinking skills. It is thought that the factor that emerges this difference is the YAP model used with the Scratch tool. For this reason, it is suggested to use YAP model together with block based visual programming tools in programming process.

¹ Bu çalışma Dr. Öğr. Üyesi Ali Kürşat ERÜMİT'in danışmanlığını yaptığı Güven ŞAHİN'in yüksek lisans tezinden üretilmiştir.

² Trabzon Üniversitesi, Fatih Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü, Trabzon, TÜRKİYE; <https://orcid.org/0000-0003-4910-4989>

³ Recep Tayyip Erdoğan Üniversitesi, Rektörlük, Enformatik, Rize, TÜRKİYE; <https://orcid.org/0000-0003-3901-5819>

⁴ Trabzon Üniversitesi, Fatih Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü, Trabzon, TÜRKİYE; <https://orcid.org/0000-0002-3555-050X>

Introduction

In today's information age, the problems faced by individuals and the skills they need to overcome these problems have been differentiated. These skills required individuals to develop awareness of knowledge, skills and attitudes under the name of 21st century skills, to acquire them and to have the necessary competence and infrastructure to apply them in different areas of life (Ministry of National Education [MEB], 2017). Wagner (2008) by 21st century learners should have these skills; critical thinking and problem solving, systems and interpersonal cooperation and leadership, quick wit and adaptation, entrepreneurship and initiative, effective oral and written communication, access to and analysis of information, curiosity and imagination. Computational thinking skills, which are related to these skills, are also important for today's society.

One way to develop computational thinking skills in individuals is to teach programming. However, there are some problems and difficulties in programming teaching. These difficulties arise from many factors such as the nature of programming, features of the programming language, attitude towards programming, motivation (Gomes & Mendes, 2007; Hawi, 2010), and self-efficacy belief (Askar & Davenport, 2009). While many factors have been shown for the difficulties experienced in programming teaching, it can be said that the main factor in experiencing difficulties in this field arises from the methods used in programming teaching (Byrne & Lyons, 2001). In this case, it is due to the fact that pedagogical approaches cannot be used appropriately (Coull & Duncan, 2011; Lahtinen, Mutka, & Jarvinen, 2005).

Due to this deficiency in the literature, independent of any programming environment, developed YAP programming teaching model by Erumit et al. (2019). The YAP model is a seven-step model that includes computer-free and computerized activities. Detailed information about the YAP model and sample lesson plans are available from Şahin (2018).

The purpose of this research is to determine the effects of programming courses that are processed using YAP teaching model on students' information-processing thinking skills. In line with this aim, the research question is programming "Does programming courses using YAP teaching model have a significant effect on students' computational thinking skills?".

Method

As a research method, case study method, quasi-experimental design without control group was used. In this experimental design, measurements are made both before and after the application (Bovis, Harden, & Hotz, 2016). In the process of the implementation of the course plans, the study group consists of 14 male and 24 female sixth grade students studying in the spring term of 2016-2017 academic year in a secondary school affiliated to MoNE in Trabzon. In the study, the activities were carried out by the teacher in the Information Technologies and Software (BTY) course for 9 weeks. Computational Thinking Scale (CTS) was used as data collection tool. The scale was designed by Korkmaz, Çakır and Özden (2017). Since the language of the original scale was also Turkish, no translation step was required. Cronbach Alpha reliability coefficient of the scale was found to be 0,809. Since the Cronbach Alpha value for the total score is between 0,640 and 0,867, it can be said that the internal consistency of the scale is high enough, in other words, the scale can make reliable measurements. Paired sample t-test was used to compare the pre-test and post-test scores.

The implementation process of the lesson plans was carried out for two lessons (80 minutes) for 9 weeks. The class size is 38 people and each student sit on a computer. The computers are controlled from a central computer. In the first week, general information was given by the researcher about the application process and the importance of this study process was explained to the students. Students' computerized studies were collected by the teacher in the folder opened on the network. In the implementation of the seven-step lesson plans, the first four steps were carried out in a computer-free environment and the other three steps were computerized.

Result and Discussion

The findings show that there is a significant difference in the computational thinking skills of the students after the programming teaching done with the YAP model. The reason for this is thought to be the YAP model used in conjunction with the Scratch programming tool. The main factor in the effectiveness of the YAP model is that the pedagogical approaches it has played an important role in the development of computational thinking skills. Since these approaches are directly related to computational thinking skills and these pedagogical approaches overlap with the pedagogical approaches included in the YAP model, the YAP model was effective in making a significant difference in students' computational thinking skills. Therefore, students in the 21st century such as problem solving, abstraction, logical thinking. It is recommended to measure their effects using the YAP model in conjunction with a visual programming tool such as Scratch in order to develop the features included in the skills.

GİRİŞ

Bilgi çağını yaşadığımız günümüzde, bireylerin karşılaştıkları problemler ve bu problemlerin üstesinden gelebilmelerini sağlayan yetenekler farklılık göstermektedir. Bireylerin 21. yy. becerileri olarak ifade edilen bu bilgi, beceri ve tutumlara ilişkin farkındalık oluşturarak hayatlarının farklı noktalarında uygulayabilmeleri gerekmektedir (Milli Eğitim Bakanlığı [MEB], 2017). 21. yy. öğrenenlerinin sahip olması gereken beceriler; eleştirel düşünme ve problem çözme, sistemler ve bireyler arası iş birliği ve liderlik, kıvrak zekâ ve uyum sağlama, girişimcilik ve insiyatif alma, etkili sözlü ve yazılı iletişim, bilgiye erişebilme ve analiz edebilme, merak ve hayal gücü olarak sıralanmaktadır (Wagner, 2008). Belirtilen beceriler ile ilişkili olan bilgi-işlemsel düşünme becerisinin (computational thinking) de toplum için gerekli olduğu belirtilmektedir (Wing, 2006). Bilgi-işlemsel düşünme; problemin çözümünü adım adım gerçekleştirme (algoritma), çözümün aynı paraleldeki problemlerle nasıl ilişkilendirildiği (örüntü oluşturma), zorluk seviyesi yüksek olan problemleri yönetilebilir alt parçalara ayırma (problemin ayrıştırılması) ve bu problemlerin daha verimli bir şekilde çözülebilmesi için bilgisayarların yardımcı olma durumudur (otomasyon) (Yadav, Hong, & Stephenson, 2016).

Alanyazında yer alan çalışmalarda öğrencilerin sahip olduğu bilgi-işlemsel düşünebilme becerileri ile problem çözme ve fikir yürütme (Román-González, Pérez-González, & Jiménez-Fernández, 2017), algoritmik düşünme (Choi, Lee, & Lee, 2016), eleştirel düşünme, yaratıcı düşünme, işbirlikçi öğrenme ve iletişim becerileri (ISTE, 2016) gibi becerilerin ilişkili olduğu görülmektedir. Bunun yanı sıra bilgi-işlemsel düşünme becerisi gelişiminin, “matematik” ve “fen ve teknoloji” derslerinin öğretimi ile ilişkili olduğuna yönelik çalışmalarda mevcuttur (Calder, 2010; Lai & Lai, 2012).

Öğrencilerin bilgi-işlemsel düşünme becerilerini arttırmak amacıyla yapılan çalışmalarda programlama öğretimi kayda değer bir rol üstlenmektedir (Barut, Tuğtekin, & Kuzu, 2016; Denner, Werner, Campe, & Ortiz, 2014; Oluk & Korkmaz, 2016). Ortaokul seviyesinde, programlama öğretimi yoluyla bilgi-işlemsel düşünme, algoritmik düşünme ve problem çözme becerilerinin gelişiminin öğrencilere sağladığı farklı faydalar üzerine alanyazında yapılmış pek çok çalışma bulunmaktadır. Yapılan araştırmalara bakıldığında öğrencilerin mantıksal düşünme (Lai & Lai, 2012; Shih, 2014), problem çözme (Brown ve diğ., 2013; Çetin, 2012; Lai & Lai, 2012), bilgi-işlemsel düşünme (Sáez-López ve diğ., 2016), üst düzey düşünme becerileri (Kafai & Burke, 2014) ve eleştirel düşünme (Doğan, 2015) becerilerinde pozitif olarak ilerleme olduğu kaydedilmiştir. Bu beceriler ile birlikte programlama öğretiminin motivasyon, bağlılık, istek, matematiksel kavramlardaki başarı, kendine güven ve işbirliği özelliklerine göre değerlendirildiğinde öğrencilere pozitif olarak katkı sağladığı belirtilmektedir (Calder, 2010; Sáez-López ve diğ., 2016; Shih, 2014).

Bireylere önemli kazanımlarda bulunduğu görülen programlama öğretimine küçük yaşlarda başlanmasının gerekliliği üzerinde durulmaktadır (Kert & Uğraş, 2009). CSTA (Computer Science Teachers Association) ve ISTE (International Society for Technology in Education) (2011), tarafından gerçekleştirilen çalışmalarda öğrencilerin, okul öncesi dönemden itibaren bilişsel seviyelerine göre hazırlanacak etkinliklerle bilgi-işlemsel ve algoritmik düşünme becerilerinin geliştirilmesi önerilmektedir. Aynı zamanda bireylere erken yaşlarında programlama mantığının kavratılmasının, diğer alanlardaki çalışmalarına da pozitif yönde etkisinin olacağı düşünülmektedir (Karabak & Güneş, 2013). Nitekim öğrencilere programlama ile ilgili çalışmalar yaptırmak, öğrencilerin matematik alanına eğilimlerinin artmasını (Ke, 2014), dil öğrenimi (Moreno-León & Robles, 2015) ve sosyal bilgiler alanında da gelişme göstermelerini sağlamıştır (Navarrete, 2013).

Programlama Öğretiminde Karşılaşılan Zorluklar ve Yedi Adımda Programlama (YAP) Öğretim Modeli

Programlama öğretiminin önemi üzerinde durulmakla beraber, uygulanan öğretim yöntemi (Çatlak, Tekdal & Baz, 2015; İmal & Eser, 2009), algoritmaları kavrama ve uygulama (Futschek & Moschitz, 2010), metin tabanlı programlamada kodların söz diziminden (Özmen & Altun, 2014) kaynaklı bazı problemler yer almaktadır. Bu problemlere ilaveten programlama işleminin yapısı, dilin özellikleri, bu işleme karşı tutum, motivasyon (Gomes & Mendes, 2007; Hawi, 2010), öz yeterlik inancı (Askar & Davenport, 2009) gibi birçok hususta programlama öğretimi etkilemektedir. Programlama öğretimi sürecinde yaşanan sıkıntılar açısından birçok faktör gösterilmekle birlikte asıl nedenin bu öğretim sürecinde tercih edilen yöntemlerden kaynaklı olduğu söylenebilir (Byrne & Lyons, 2001). Bu durumun nedeninin ise pedagojik yaklaşımların öğretim sürecinde gerektiği gibi şekillendirilememesi olduğu ifade edilebilir (Coull & Duncan, 2011; Lahtinen, Mutka, & Jarvinen, 2005).

Alanyazında belirtilen problemler göz önüne alındığında programlama öğretiminin nasıl şekillendirilebileceği üzerinde fikir yürütülürken (Coull & Duncan, 2011; Lahtinen, Mutka, & Jarvinen, 2005) aynı zamanda ilgili çalışmaların azlığı dikkat çekmektedir. Alanyazında programlama öğretimi için problem çözme (Asad, Tibi, & Raiyn, 2016; Grover, Pea, & Cooper, 2016; Qian & Lehman, 2016), soyutlama (Futschek & Moschitz, 2010), akran yollu öğrenme (Denner, Werner, & Ortiz, 2012; Lewis, 2011), işbirlikçi öğrenme (Denner ve diğ., 2014), algoritma yazma (Grover, Pea, & Cooper, 2016; Milková & Hůlková, 2013), drama veya rol yapma (Karaosmanoğlu & Adıgüzel, 2017; Sarioğlu & Kartal, 2017; Weigend, 2014) pedagojik yaklaşımları kullanılmaktadır. Alan yazında belirtilen pedagojik yaklaşımlar birbirinden bağımsız olarak programlama öğretimi çalışmalarında uygulanmaktadır. Ancak, bu pedagojik yaklaşımları sınıf ve laboratuvar ortamlarında, ders sürecinde ve yaş grubuna uygun şekilde organize ederek, izlenebilecek adımlar haline getirmek ve böylelikle programlama öğretimi sistematikleştirmeye ihtiyaç vardır.

Alanyazındaki bu eksiklikten dolayı herhangi bir programlama ortamından bağımsız şekilde, öğrencilerin algoritmik düşünme ve problem çözme becerilerini geliştirmek amacıyla yapılacak programlama öğretimi için Erümit ve diğ. (2019) tarafından YAP öğretim modeli geliştirilmiştir. YAP modeli bilgisayarsız ve bilgisayarlı etkinlikleri kapsayan ve yedi adımdan oluşan bir modeldir. Alanyazında programlama öğretimi için kullanılan pedagojiler belirlenerek YAP öğretim modeli içerisinde Analoji, Soyutlama,

Problem çözüme, Akran yollu öğrenme, İşbirlikçi öğrenme, Algoritma yazma ve Drama pedagojik yaklaşımları entegre edilmiştir. YAP modeli adımları şekil 1’de gösterilmektedir.



Şekil 1. Yap modeli adımları

YAP modelinde ilk adım Problemi Anlama ve Tanıma ile başlamaktadır. Bu adımda problemle ilgili sorulardan oluşan problemi anlama ve tanıma (Ek-1) formu kullanılmasıyla öğrencinin soyutlama yaparak problemi daha iyi bir şekilde anlaması sağlanır. Problemi kavramış bir şekilde ikinci adıma geçen öğrenciden problemin çözümü için strateji oluşturması beklenir. Bunun için de strateji oluşturma (Ek-2) formu kullanılır. Üçüncü adımda öğrencilerin kendi stratejilerini kullanarak ulaşılmış oldukları sonuçlar öğretmen eşliğinde tartışılarak değerlendirilir ve en kısa çözüm yolu elde edilir. Dördüncü adım problemin çözümü için elde edilen en doğru strateji kullanılarak, çözümün algoritmasının adım adım yazılması işlemini kapsamaktadır. Beşinci adımda ise öğrenciler artık kodlama adımına geçmektedir. Bu adımda öğrenciler yazmış oldukları algoritmanın bir programlama aracı ile kodlama işlemini gerçekleştirir. Altıncı adımda bu adıma kadar olan etkinliklerden bağımsız olarak öğrencilere birtakım hatalar ya da eksiklikler barındıran kodlama uygulaması verilerek hataları ya da eksiklikleri bulup düzeltmeleri istenir. Yedinci ve son adımda ise öğretmen öğrencilere konu kapsamında belirli kriterleri olan bir programlama ödevi verir. Bu ödevde öğrenciler öncelikle yapacakları kodlamanın algoritmasını yazarlar daha sonra herhangi bir programlama ortamında kodlamasını yaparlar. YAP modeli ile ilgili ayrıntı içeriğe Şahin (2018) tezinden ulaşılabilir.

Çalışmanın Amacı ve Problemi

Bu araştırmanın amacı; YAP öğretim modeli kullanılarak işlenen programlama derslerinin öğrencilerin bilgi-işlemsel düşünme becerisine etkilerini belirlemektir. Bu amaç doğrultusunda araştırma problemi; “YAP öğretim modeli kullanılarak işlenen programlama derslerinin, öğrencilerin bilgi-işlemsel düşünme becerilerine etkisi var mıdır?” şeklinde belirlenmiştir.

YÖNTEM

Araştırma Yöntemi

Araştırma yöntemi olarak, durum çalışması yöntemi kullanılmıştır. Yıldırım ve Şimşek (2011) durum çalışmasını; nasıl ve niçin sorularını temel alarak araştırmacının kontrol edemediği bir olgu ya da olayın derinlemesine incelenmesine olanak veren araştırma yöntemi olarak tanımlamıştır. Durum çalışması yönteminde, belirli bir veri toplama deseninin ve veri toplama aracının kullanılması zorunlu değildir. Araştırmacılar, incelenen duruma ilişkin derin bir anlayış geliştirebilmek adına farklı türlerden verilere ihtiyaç duyabilmektedirler (Creswell, 2012). Bu araştırma yönteminde araştırma sorusunun cevabına giden yolda bir rol üstleneceği düşünülen herhangi bir veri toplama deseni ve aracı kullanılabilir (Merriam, 1998). Dolayısıyla araştırma sorularını cevaplayabilmek için durum çalışmasında nitel ve nicel veriler kullanılabilir. Bu bağlamda çalışmada, durum çalışması yöntemi içerisinde kontrol grupsuz yarı deneysel desen ile veri toplama süreci gerçekleştirilmiştir. Bu deneysel desende hem uygulama öncesinde (öntest) hem de uygulama sonrasında (sontest) ölçümler yapılmaktadır (Bovis, Harden, & Hotz, 2016). Bu doğrultuda Bilgisayarca

Düşünme Ölçeği (BDÖ) uygulama sürecine başlamadan önce öğrencilere uygulanmış ve uygulama sürecinin tamamlanması ile birlikte bu işlem tekrarlanmıştır.

Çalışma Grubu

Uygulama Trabzon ilindeki bir ortaokulda 2016-2017 eğitim-öğretim yılı bahar döneminde öğrenim görmekte olan 14'ü erkek ve 24'ü kız toplam 38 altıncı sınıf öğrencisi ile yapılmıştır. Uygulama öncesinde öğrencilere programlama öğretimi ile ilgili deneyimleri sorulmuştur. Uygulamaya katılan öğrencilerin tamamı daha önce böyle bir eğitim almamıştır. Çalışmada etkinlikler, Bilişim Teknolojileri ve Yazılım (BTY) dersinde, ders öğretmeni tarafından 9 hafta boyunca gerçekleştirilmiştir.

Veri Toplama Araçları

Araştırmada veri toplama aracı olarak, Korkmaz, Çakır ve Özden (2017) tarafından tasarlanmış olan Bilgisayarca Düşünme Ölçeği kullanılmıştır. Ölçeğin Cronbach Alpha güvenirlik kat sayısı 0,809 olarak bulunmuştur. Toplam puana dönük Cronbach Alpha değeri 0,640 ile 0,867 arasında olması nedeniyle ölçeğin iç tutarlığının yeterince yüksek olduğu, bir başka ifadeyle ölçeğin güvenilir ölçümler yapabildiği söylenebilir.

Veri Analizi

Öğrenciler uygulama öncesinde ve sonrasında YAP modeli ile işlenen programlama derslerinin bilgi-işlemsel düşünme becerilerine etkilerini ölçmek için BDÖ'yü tamamlamıştır. Elde edilen verilerin analizi için SPSS paket programından faydalanılmıştır. Ön test ve son test verilerini karşılaştırmadan önce verilerin normal dağılım gösterip göstermedikleri kontrol edilmiştir. Örneklem sayısının 50'den küçük olması nedeniyle Shapiro-Wilk testi sonucu incelenmiştir (Büyüköztürk, 2007). Verilerin normal dağılım göstermesinden dolayı eşleştirilmiş örneklem t-testi (paired sample t test) kullanılmıştır.

Uygulama Süreci

Ders planları, Bilişim Teknolojileri ve Yazılım (BTY) dersinde bilişim teknolojileri öğretmeni tarafından haftada iki ders saati (80dk) olmak üzere 9 hafta boyunca uygulanmıştır. Uygulamada her bilgisayarı bir öğrenci kullanmıştır. İlk hafta, araştırmacının uygulama sürecinin işleyişi üzerine bilgilendirme yapması ve bu sürecin öneminden bahsetmesiyle tamamlanmıştır. Diğer haftalarda öğrencilerin bilgisayar ortamında yapmış olduğu çalışmalar öğretmenin ağ üzerinde açmış olduğu klasörde toplanmıştır. Yedi adımdan oluşan ders planlarının uygulanmasında ilk dört adım bilgisayarsız ortamda, diğer üç adım ise bilgisayarlı olarak gerçekleştirilmiştir. Haftalık yapılan işlemler Tablo 1'de gösterilmiştir.

Tablo 1. Uygulama süreci

Hafta	Konu	Ders Süreci
1	Problem Çözme Kavramları ve Yaklaşımları	1.Ders: “Çobana yardım” etkinliğinin uygulanması 2.Ders: “Şehirleri buluyorum” etkinliğinin uygulanması, verilerin toplanması.
2	Scratch Programını Tanıyalım	1.Ders: Scratch web adresinden örnek uygulamaların incelenmesi. 2.Ders: İncelenen örnek projeler doğrultusunda inceleme formunun doldurulması, verilerin toplanması.
3	Scratch Programını Tanıyalım	1.Ders: EBA’dan scratch eğitim videolarının izlenmesi. 2.Ders: Örnek bir uygulama yapılarak görev kontrol çizelgesinin doldurulması, verilerin toplanması.
4	Koşul Yapıları	1.Ders: “Ali ve ayşe tatile gidiyor” etkinliğinin uygulanması (YAP modeli ilk 4 adım). 2.Ders: “Soru-cevap” ve “Labirent harita oyunu” etkinliklerinin uygulanması (YAP modeli 5. adım), verilerin toplanması.
5	Koşul Yapıları	1.Ders: “Kod okuma ve düzenleme” etkinliğinin uygulanması. 2.Ders: Etkinliğin ders öğretmeni eşliğinde tamamlanması ve öğrencilere ödev verilmesi (YAP modeli 6. ve 7. adımları), verilerin toplanması.
6	Değişkenler	1.Ders: “Ali meyve etkinliğinde” etkinliğinin uygulanması (YAP modeli ilk 4 adım) 2.Ders: “Meyve sepeti” etkinliğinin uygulanması (YAP modeli 5. adım), verilerin toplanması.
7	Değişkenler	1.Ders: “Kod okuma ve düzenleme” etkinliğinin uygulanması. 2.Ders: Etkinliğin ders öğretmeni eşliğinde tamamlanması ve öğrencilere ödev verilmesi (YAP modeli 6. ve 7.adımları), verilerin toplanması.
8	Döngüler	1.Ders: “Ali oyun oynuyor” etkinliğinin uygulanması (YAP modeli ilk 4 adım). 2.Ders: “Yakıt miktarım” ve “Araba yarışı” etkinliklerinin uygulanması (YAP modeli 5.adım), verilerin toplanması.
9	Döngüler	1.Ders: “Kod okuma ve düzenleme” etkinliğinin uygulanması. 2.Ders: Etkinliğin ders öğretmeni eşliğinde tamamlanması ve öğrencilere ödev verilmesi (YAP modeli 6. ve 7.adımları), verilerin toplanması.

YAP öğretim modelinde öğrenciler birinci adım olan “Problemi Anlama ve Tanıma” basamağında ders planında yer alan etkinliğe göre oluşturulan soruları cevaplamışlardır. Öğrenciler bu basamakta problem durumundaki gerekli ve gereksiz verileri birbirinden ayırma işlemi yapmıştır. İkinci adım olan “Strateji Oluşturma” basamağında öğrenciler birden fazla çözümü barındıran problem durumunda sonuca ulaşmak için en iyi yolu bulmaya çalışmışlardır. Üçüncü adım olan “Strateji Karşılaştırma” basamağında öğretmen, öğrencilerin bulmuş olduğu çözümleri öğrenciler ile beraber inceleyerek en iyi sonuca ulaşmıştır. “Algoritmayı Oluşturma” olan dördüncü adımda, ilk olarak öğrenciler bulmuş oldukları çözüm yoluna göre sözde kod olarak algoritmalarını yazmışlardır. Daha sonra gönüllü öğrenciler yazmış oldukları algoritmayı drama yaparak canlandırmışlardır. Genel olarak bu dört adım bir ders saatinde gerçekleştirilebilmiştir.

“Algoritmayı Kodlama” olan beşinci adımda, öğrenciler algoritmasını oluşturdukları programın Scratch ortamında kodlamasını yapmıştır. Bu basamağın uygulama süreci, ortalama olarak iki ders saati sürmüştür. Altıncı adım olan “Farklı bir Koddaki Hatayı Belirleme ve Düzeltme” basamağında öğrenciler ders planındaki etkinlikten bağımsız olarak kod yapısı hatalı Scratch uygulamasında hataları tespit edip düzeltme işlemi yapmışlardır. Daha sonra öğretmen, akıllı tahta üzerinde öğrenciler ile birlikte hatalı kod yapısını doğru hale getirmiştir. Bu aşama ortalama olarak bir ders saati sürmüştür. Yedinci adım olan “Yeni Algoritmalar Hazırlama ve Kodlama” basamağı için ders öğretmeni öğrencilere kriterleri belirlenmiş şekilde ödevler vermiştir. Bu ödevlerde öğrencilerden ilk olarak verilen kriterlere göre algoritma yazmaları daha sonra bu algoritmayı programlama ortamında kodlamaları istenmiştir. Ders planlarının içeriği ve kullanımı Tablo 2’de verilmiştir.

Tablo 2. Ders Planlarına İlişkin Özet Tablo

Konu	Kazanımlar	Etkinlik Adı	YAP öğretim modeli adımları	Türü	Bireysel/ Grup	Süre	Hafta
Problem Çözme Kavramları ve Yaklaşımları	<ul style="list-style-type: none"> Bir problemi analiz eder. Problemlere özgün çözümler üretmenin önemini açıklar. Problemin çözümü için bir algoritma geliştirir. Problem çözme sürecini ve temel kavramları tartışır. 	Çobana yardım	-	Bilgisayarsız	Bireysel veya Grup	40dk	1
		Dokumacı kuşlar	-	Bilgisayarsız	Bireysel veya Grup	40dk	
Programlama Ortamını Tanıyalım	<ul style="list-style-type: none"> Sahneye karakter ekler. Sahne arka planını değiştirir. Karaktere hareket kazandırır. Karakter kostümü değiştirerek hareket kazandırır. Blok tabanlı programlama aracının ara yüzünü ve özelliklerini tanır. Blok tabanlı programlama aracında sunulan bir programın işlevlerini açıklar. 	Serbest zaman	-	Bilgisayarlı	Bireysel	160dk	2-3
Değişkenler	<ul style="list-style-type: none"> Bir problemi analiz eder. Problemlere özgün çözümler üretmenin önemini açıklar. Problem çözme sürecini ve temel kavramları tartışır. Problemin çözümü için bir algoritma geliştirir. Farklı algoritmaları inceleyerek en hızlı ve doğru çözümü seçer. Problemin çözümünü benzer problemler için geneller. Verileri türlerine göre sınıflandırır. Bir algoritmanın çözümünü test eder. Sabit ve değişkenleri problem çözümünde kullanır. Doğrusal mantık yapısını içeren programlar oluşturur. Blok tabanlı programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler. Blok tabanlı programlama aracında sunulan bir programın hatalarını ayıklar. Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. Bir problemi alt problemlere böler. 	"Manavdan meyve alma"	1-4	Bilgisayarsız	Bireysel veya Grup	30dk	4-5
		"Meyve seçiyorum"	4	Bilgisayarsız	Grup	10dk	
		"Meyve sepetim"	5	Bilgisayarlı	Bireysel	95dk	
		"Kod okuma ve düzenleme uygulaması"	6	Bilgisayarlı	Bireysel	20dk	
Koşul Yapıları	<ul style="list-style-type: none"> Bir problemi analiz eder. Problemlere özgün çözümler üretmenin önemini açıklar. Problem çözme sürecini ve temel kavramları tartışır. Problemin çözümü için bir algoritma geliştirir. Farklı algoritmaları inceleyerek en hızlı ve doğru çözümü seçer. Problemin çözümünü benzer problemler için geneller. Bir algoritmanın çözümünü test eder. Blok tabanlı programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler. 	"Ali ve Ayşe Tatile Gidiyor"	1-4	Bilgisayarsız	Bireysel veya Grup	35dk	6-7
		"Renkli adımlar"	4	Bilgisayarsız	Grup	10dk	
		"Soru – cevap" uygulaması	5	Bilgisayarlı	Bireysel	95dk	
		"Labirent harita oyunu"					
		"Kod okuma ve düzenleme uygulaması"	6	Bilgisayarlı	Bireysel	20dk	
Koşul Yapıları	<ul style="list-style-type: none"> Doğrusal mantık yapısını içeren programlar oluşturur. Karar yapısını içeren programlar oluşturur. Çoklu karar yapıları içeren programlar oluşturur. Blok tabanlı programlama aracında sunulan bir programın hatalarını ayıklar. Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. Karar yapısını içeren programları test ederek hatalarını ayıklar. Çoklu karar yapısını içeren programları test ederek hatalarını ayıklar. Bir problemi alt problemlere böler. 						

Konu	Kazanımlar	Etkinlik Adı	YAP öğretim modeli adımları	Türü	Bireysel/Grup	Süre	Hafta	
Döngüler	<ul style="list-style-type: none"> Bir problemi analiz eder. Problemlere özgün çözümler üretmenin önemini açıklar. Problem çözme sürecini ve temel kavramları tartışır. Problemin çözümü için bir algoritma geliştirir. Farklı algoritmaları inceleyerek en hızlı ve doğru çözümü seçer. Problemin çözümünü benzer problemler için geneller. Bir algoritmanın çözümünü test eder. Blok tabanlı programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler. Doğrusal mantık yapısını içeren programlar oluşturur. Döngü yapısını içeren programlar oluşturur. Bir algoritmayı uyarlamak için en uygun karar yapılarını seçer. Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir. Döngü yapısını içeren programları test ederek hatalarını ayıklar. Matematik ve bilgisayar bilimi arasındaki ilişkiyi tespit eder. Tüm programlama yapılarını içeren özgün bir proje oluşturur. Bir problemi alt problemlere böler. 	“Ali oyun oynuyor”	1-4	Bilgisayarsız	Bireysel veya Grup	30dk	8-9	
		“Araba yarışı”	4	Bilgisayarsız	Grup	10dk		
		Sınıf içi canlandırma						
		“Yakıt miktarım”	5	Bilgisayarlı	Bireysel	95dk		
		“Araba yarışı” uygulaması						
		“Kod okuma ve düzenleme uygulaması”	6	Bilgisayarlı	Bireysel	20dk		

Tablo 2’de YAP modelinin uygulanmasına yönelik olarak; konu, ilgili konunun kazanımları, uygulanan etkinliğin adı, YAP modelinde hangi adımları kapsadığı, etkinlik türü, uygulama şekli, etkinliğe ayrılan süre ve hafta bilgisine yer verilmiştir.

Sınırlılıklar

Araştırma, 2016-2017 bahar yarıyılında bir ortaokulun 6. sınıfında öğrenim görmekte olan 38 öğrenci ile 9 hafta sürede gerçekleştirilmiştir. Uygulama süreci kontrol grupsuz yarı deneysel desene sahiptir. Dersler “Problem çözme yaklaşımları ve kavramları”, “Programlama ortamını tanıyalım”, “Değişkenler”, “Koşul yapıları” ve “Döngüler” konuları ile sınırlıdır.

BULGULAR

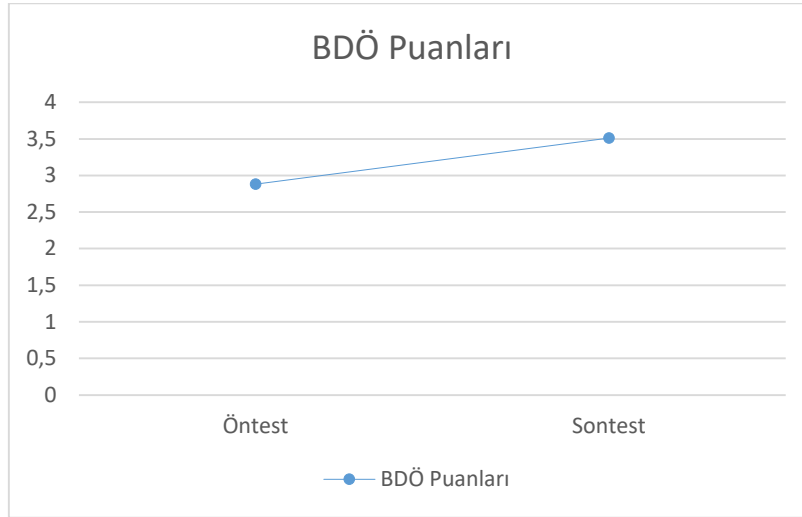
Yap Modeli Kullanılarak İşlenen Programlama Derslerinin Öğrencilerin Bilgi-İşlemsel Düşünme Becerilerine Etkileri

Araştırmada “YAP öğretim modeli kullanılarak işlenen programlama derslerinin, öğrencilerin bilgi-işlemsel düşünme becerilerine etkisi var mıdır?” sorusuna yanıt bulmak amacıyla BDÖ’den elde edilen veriler SPSS programı kullanılarak analiz edilmiştir. Uygulanacak analize karar verebilmek amacıyla öncelikle verilerin normallik durumları kontrol edilmiştir. Örneklem sayısının 50’den küçük olması nedeniyle Shapiro-Wilk testi sonucu incelenmiştir (Büyüköztürk, 2007). Verilerin normal dağılım göstermesinden dolayı ($p > .05$) eşleştirilmiş örneklem t-testi kullanılmıştır. Grubun ortalamaları ve standart sapmaları Tablo 3’de gösterilmektedir. (ayrıca bkz. şekil 2).

Tablo 3. Grubun BDÖ puanları t testi değerleri

	Grup	n	Öntest		Sontest		p	d
			M	SD	M	SD		
BDBÖ	S1	38	2.88	.43	3.51	.32	.000	1.64

$P < .05$



Şekil 2. Grubun öntest-sontest BDÖ puanları arasındaki gelişme

Öğrencilerin öntest-sontest gelişimlerinin anlamlı olup olmadığını belirlemek için eşleştirilmiş örneklem t-testi uygulanmıştır. Test sonucunda S1 grubu sontest puanlarının öntest puanlarından anlamlı derecede daha yüksek olduğu görülmektedir ($t(38) = -7.337$, $p = .000$, $d=1.64$). Ayrıca hesaplanan etki büyüklüğü ise Cohen (1988) tarafından tanımlandığı gibi büyük etki boyutunu ($.50 > d \geq .20$ küçük etki boyutu, $0.80 > d \geq .50$ orta etki boyutu ve $d \geq .80$ büyük etki boyutu) ifade etmektedir. Bu bulgular öğrencilerin YAP modeliyle yapılan programlama öğretileri sonrasında bilgi-işlemsel düşünme becerilerinde anlamlı bir farklılık ortaya çıktığını göstermektedir.

TARTIŞMA

Araştırmada YAP modeli kullanılarak işlenen programlama derslerinin öğrencilerin bilgi-işlemsel düşünme becerisine etkilerini belirlemek amaçlanmıştır. Elde edilen bulgulara göre, öğrencilerin bilgi-işlemsel düşünme becerilerinde olumlu yönde anlamlı bir değişim olduğu görülmüştür.

Alanyazında programlama öğretiminde farklı pedagojik yaklaşımların kullanıldığı çalışmalar bulunmaktadır. Qian ve J.D. Lehman (2016) tarafından yapılan çalışmada programlama öğretimi için problem çözme yaklaşımı kullanılmıştır. 14 hafta boyunca süren çalışmada SangTian Programming Learning Environment (SPLE) ortamı üzerinde öğrenciler etkinlikler yapmışlardır. Bu ortam programlama problemlerini barındıran ve öğrencilerin problemlerin çözümleri için kısa olarak kod yazmaları gereken bir ortamdır. Çalışma sonucunda öğrencilerin matematik ve İngilizce yetenekleri ile programlama becerileri arasında önemli bir ilişki olduğu bulunmuştur. Asad, Tibi ve Raiyn (2016) tarafından yapılan çalışmada da, problem çözme yaklaşımı kullanılmıştır ve programlama öğretimine yönelik tutumun araştırılması amaçlanmıştır. Çalışmada mantıksal oyunlar, blok tabanlı programlama ve metin tabanlı programlama olmak üzere üç farklı ders tasarlanmıştır. Sonuçlar genel olarak öğrencilerin programlamaya yönelik tutumlarının kurslara katıldıktan sonra olumlu olduğunu ortaya koymuştur. Ayrıca programlamanın öğrencilerin problem çözme becerilerini de geliştirdiği tespit edilmiştir. Yapılan çalışmalara benzer şekilde YAP modeli ile gerçekleştirilen uygulama sürecinde de problem çözme yaklaşımı kullanılmıştır.

Milková ve Hülkova (2013) tarafından yapılan çalışmada, öğrencilerin algoritmik düşünme becerilerini geliştirmek için problem çözme yaklaşımı kullanılmıştır. Bu yaklaşımda bulmacanın farklı biçimlerini içeren etkinlikler yer almaktadır. Futschek ve Moschitz (2011) ise çalışmalarında algoritma öğretiminde farklı seviyelerde, eğlenceli ve bilgisayarsız etkinliklere yer vermiştir. Kaučič ve Asič (2011) tarafından yapılan çalışmada, programlama öğretimi için problem çözme yaklaşımı kullanılmıştır. Uygulama sürecinin son 2 saatinde öğrenciler zorluk derecesi olarak 3 farklı düzeyde hazırlanan testlerdeki görevleri çözmüş ve anketi cevaplamışlardır. Birçok öğrenci için Scratch ile çalışmak ilk deneyimleri olduğundan bu kodlama ortamı hakkında oldukça olumlu izlenimler göstermişlerdir. Benzer şekilde bu çalışmada da problem çözme yaklaşımıyla bilgisayarsız etkinlikler kullanılmıştır.

Bir başka çalışma da Grover, S., Pea, R. ve Cooper, S. (2016) tarafından yapılmış ve ortaokul öğrencilerine bilgisayar bilimlerini öğretmek için algoritmik düşünme ve programlama konularına yönelik 7 haftalık bir müfredat tasarlamışlardır. Eğitim için problem çözme ve algoritma yazma yaklaşımını kullanmışlardır. Çalışma sonunda öğrencilerin algoritmik düşünme becerilerinde önemli kazanımlar elde ettikleri görülmüştür. YAP modeli ile yapılan bu çalışma da problem çözme ve algoritma yazma yaklaşımı kullanılmıştır. Elde edilen sonuçlar alanyazındaki çalışmalarla benzerlik göstermektedir.

Alan yazında belirli bir pedagojik yaklaşım kullanılmasına rağmen öğrencilerin beceri gelişimine anlamlı düzeyde etkisi olmayan çalışmalar da yer almaktadır. Kalelioğlu ve Gülbahar (2014), tarafından yapılan çalışmada programlama öğretimi için problem çözme yaklaşımı kullanılmıştır. Scratch ile yapılan programlama etkinliklerinin 5.sınıf öğrencilerinin problem çözme becerileri üzerindeki etkisi araştırılmıştır. Sonuç olarak öğrencilerin problem çözme becerilerinde anlamlı farklılık olmadığı görülmüştür. Öğrencilerin problem çözme becerilerinin gelişmesine yardımcı olmak için üst düzey düşünme gerektiren farklı uygulama ve etkinliklerle desteklenmesi gerektiği belirtilmiştir.

Uslu, Mumcu ve Eğin (2018) tarafından yapılan çalışmada, ortaokul düzeyinde görsel programlama etkinliklerinin öğrencilerin bilgi-işlemsel düşünme becerileri üzerindeki etkisi araştırılmıştır. Çalışma 2 aşamadan oluşan 12 haftalık bir süreçte gerçekleştirilmiştir. Çalışmada eşli kodlama ve problem çözme yaklaşımları kullanılmıştır. Araştırma sonunda görsel programlama etkinliklerinin öğrencilerin bilgi-işlemsel düşünme becerilerinde anlamlı bir etki yaratmadığı belirtilmiştir. Ayrıca çalışmada bilgisayarsız bilgisayar bilimi etkinliklerine yer verilmesinin öğrencilerin bilgi-işlemsel düşünme becerilerine olumlu yönde etkisinin olabileceği ileri sürülmüştür. Lai ve Yang (2011) tarafından yapılan çalışmada da programlama öğretimi için problem çözme yaklaşımı kullanılmıştır. Çalışma ortaokul öğrencileri ile birlikte yürütülmüş olup kontrol gruplu yarı deneysel desen kullanılmıştır. Araştırma sonunda görsel programlamanın öğrencilerin mantıksal düşünme becerileri üzerinde anlamlı bir etkisinin olmadığı belirtilmiştir. Buna karşın, YAP modeli ile gerçekleştirilen uygulamada bilgisayarlı etkinliklerin yanı sıra bilgisayarsız etkinliklerin de bulunmasının, bilgi-işlemsel düşünme becerilerinin gelişmesine katkı sağladığı görülmüştür. Bu durum uygulanan yöntemin ve etkinliklerin önemini ortaya çıkarmaktadır. Alanyazından elde edilen bulgular da bunu desteklemektedir (Kalelioğlu & Gülbahar, 2014; Uslu, Mumcu, & Eğin, 2018).

SONUÇ VE ÖNERİLER

Sonuç olarak alanyazında programlama öğretimi için, problem çözme yaklaşımı, algoritma yazma, eşli kodlama, bilgisayarsız etkinlikler gibi yöntemlerin kullanıldığı görülmüştür. Ancak belirtilen yaklaşımların tek başlarına kullanılmasının bazı çalışmalarda olumlu yönde etki ederken bazılarında ise olumlu yönde herhangi bir etkisinin olmadığı görülmüştür. Bu nedenle programlama öğretiminde kullanılan araçlar kadar hazırlanan etkinlikler ve bu etkinliklerin nasıl uygulanacağı yani yöntem de önemli görülmektedir. Yöntem, öğretim için gerekli pedagojileri içermeli ve etkinliklerde bu pedagojileri doğru şekilde sunacak şekilde hazırlanmalıdır. YAP modeli ile yürütülen bu çalışma sonucunda öğrencilerin bilgi-işlemsel düşünme becerilerinde olumlu yönde ve anlamlı düzeyde gelişme olmasının, modelin barındırdığı pedagojiler ve bu pedagojilere uygun etkinliklerden kaynaklandığı düşünülmektedir.

YAP modelinin öğrencilerin problem çözme, soyutlama, mantıksal düşünme gibi diğer bilişsel becerileri üzerindeki etkileri ölçülebilir. YAP modelinin farklı yaş gruplarındaki öğrencilerin bilişsel becerileri ve akademik başarıları üzerindeki etkileri ölçülebilir. YAP modelinin farklı programlama ortamlarında kullanımıyla sonuçlarının araştırılması sağlanabilir.

KAYNAKÇA

- Askar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for Java programming among engineering students. *The Turkish Online Journal of Educational Technology*. 8(1), 26-27.
- Asad, K., Tibi, M., & Raiyn, J. (2016). Primary school pupils' attitudes toward learning programming through visual interactive environments. *World Journal Of Education*. 6(5), 20.
- Barut, E., Tuğtekin, U., & Kuzu, A. (2016). Robot uygulamalar ile bilgi işlemsel düşünme becerilerine bakış. *In 4th Instructional Technologies & Teacher Education Symposium* (210-214).
- Bovis, S. E., Harden, T., & Hotz, G. (2016). Pilot study: a pediatric pedestrian safety curriculum for preschool children. *Journal of Trauma Nursing*. 23(5), 247-256.
- Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E., & Fontecchio, A. (2013). Computer aided instruction as a vehicle for problem solving: Scratch programming environment in the middle years classroom. http://www.pages.drexel.edu/~dmk25/ASEE_08.pdf
- Byrne, P., & Lyons, G. (2001). The effect of student attributes on success in programming. *ACM SIGCSE Bulletin*. 33(3), 49-52.
- Calder, N. (2010). Using Scratch: An integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom*. 15(4), 9-14.
- Choi, J. Lee, Y., & Lee, E. (2016). Puzzle based algorithm learning for cultivating computational thinking. *Wireless Personal Communications*. 93(1), 131-145.
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: L. Erlbaum Associates.
- Coull, N. J., & Duncan, I. M. (2011). Emergent requirements for supporting introductory programming. *Innovation in Teaching and Learning in Information and Computer Sciences*. 10(1), 78-85.
- Creswell, J.W. (2012). *Educational research: planning, conducting, and evaluating quantitative research* (4. Baskı). Boston: Pearson Education Inc.
- CSTA & ISTE (2011). *Computational Thinking Teacher Resources*, Second Edition. http://www.iste.org/docs/ct-documents/ct-teacher-resources_2ed-pdf.pdf?sfvrsn=2
- Çatlak, Ş., Tekdal, M., & Baz, F. Ç. (2015). Scratch yazılımı ile programlama öğretiminin durumu: Bir döküman inceleme çalışması. *Journal of Instructional Technologies & Teacher Education*. 4(3), 13-25.
- Çetin, E. (2012). *Bilgisayar programlama eğitiminin çocukların problem çözme becerileri üzerine etkisi*. Yayımlanmamış Yüksek Lisans Tezi, Gazi Üniversitesi Eğitim Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalı, Ankara.
- Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students?. *Journal of Research on Technology in Education*. 46(3), 277-296.

- Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?. *Computers & Education*. 58(1), 240-249.
- Doğan, U. (2015). *Ortaokul öğrencilerinde bilgisayar oyunu geliştirme sürecinin öğrencilerin eleştirel düşünme becerilerine ve algoritma başarılarına etkisi*. Yayınlanmamış Yüksek Lisans Tezi. Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalı, İstanbul.
- Erümit, K. A., Karal, H., Şahin, G., Aksoy, D. A., Aksoy, A., & Benzer, A. İ. (2019). Programlama öğretimi için bir model önerisi: yedi adımda programlama. *Eğitim ve Bilim*. 44(197), 155-183.
- Futschek, G. (2006). Algorithmic thinking: The key for understanding computer science. In: Mittermeir, R. (Ed.), *Informatics Education The Bridge between Using and Understanding Computers* (159 – 168).
- Futschek, G., & Moschits, J. (2010). Developing algorithmic thinking by inventing and playing algorithms. https://publik.tuwien.ac.at/files/PubDat_187461.pdf
- Gomes, A., & Mendes, A. J. (2007). Learning to program - difficulties and solutions. In *International Conference on Engineering Education-ICEE*.
- Grover, S., Pea, R., & Cooper, S. (2016). Factors influencing computer science learning in middle school. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (552-557).
- Gülbahar, Y. (2017). *Bilgi işlemsel düşünmeden programlamaya* (1. Baskı) Ankara: Pegem Akademi.
- Hawi, N. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. *Computers & Education*. 54(4), 1127-1136.
- ISTE, (2016). The ISTE National Educational Technology Standards (NETS) and Performance Indicators for Students. <http://www.iste.org/standards/nets-for-students>
- İmal, N., & Eser, M. (2009). Programlama dili öğrenmedeki zorluklar ve çözüm yaklaşımları. http://www.emo.org.tr/ekler/8bd988bd20804a2_ek.pdf
- Kafai, Y. B., & Q. Burke. (2014). *Connected code: Why children need to learn programming*. MIT Press.
- Karabak, D., & Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*. 2(3),163-169.
- Karaosmanoğlu, G., & Adıgüzel, Ö. (2017). Yaratıcı drama yönteminin 6. sınıf bilişim teknolojileri ve yazılım dersi alan öğrencilerin akademik başarılarına etkisi. *İlköğretim Online*. 16(2), 693-712.
- Karasar, N. (2014). *Bilimsel araştırma yöntemi "kavramlar ilkeler teknikler"*(26. Baskı) Ankara: Nobel.
- Kaučič, B., & Asič, T. (2011). Improving introductory programming with scratch?. In 2011 Proceedings of the 34th International Convention MIPRO. (pp. 1095-1100).
- Kalelioğlu, F., & Gülbahar, Y. (2014). The effects of teaching programming via Scratch on problem solving skills: a discussion from learners' perspective. *Informatics in Education*. 13(1), 33-50.
- Kert, S. B., & Uğraş, T. (2009). Programlama eğitiminde sadelik ve eğlence: Scratch örneği. In *The First International Congress of Educational Research*. Çanakkale, Türkiye.
- Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*. 73, 26-39.
- Korkmaz, Ö., Çakır, R., & Özden, M. Y. (2017). A validity and reliability study of the Computational Thinking Scales (CTS). *Computers in Human Behavior*, 72, 558-569.
- Kubica, M., & Radoszewski, J. (2010). Algorithms without programming. *Olympiads in Informatics*. 4, 52-66.
- Lai, C. S., & Lai, M. H. (2012). Using computer programming to enhance science learning for 5th graders in Taipei. *International Symposium on Computer, Consumer and Control* (146-148).
- Lai, A., & Yang, S. (2011). The learning effect of visualized programming learning on 6th graders' problem solving and logical reasoning abilities. *International Conference on Electrical and Control Engineering* (6940-6944).
- Lahtinen, E., Ala-Mutka, K., & Jarvinen, H. (2005) A study of difficulties of novice programmers. In *Acm Sigcse Bulletin, ACM*. 37(3), 14-18.
- Lewis, O. (2011). *The children of sanchez: Autobiography of a mexican family*. Vintage.
- Merriam, S.B. (1998). *Qualitative research and case study applications in education*. California: Jossey-Bass.
- Milková, E., & Hulkova, A. (2013). Algorithmic and logical thinking development: base of programming skills. *WSEAS Transactions on Computers*. 12(2), 41-51.
- Milli Eğitim Bakanlığı [MEB]. (2017). *Bilişim teknolojileri ve yazılım dergisi öğretim programı, 2016-2017*. Ankara: Milli Eğitim Basımevi.
- Moreno-León, J., & Robles, G. (2015). Computer programming as an educational tool in the English classroom a preliminary study. In *Global Engineering Education Conference* (961-966).
- Navarrete, C. C. (2013). Creative thinking in digital game design and development: A case study. *Computers & Education*. 69, 320-331.
- Oluk, A., & Korkmaz, Ö. (2016). Comparing student's scratch skills with their computational thinking skills in terms of different variables. *Modern Education and Computer Science*. 11, 1-7.
- Özmen, B., & Altun, A. (2014). "Undergraduate students' experiences in programming: difficulties and obstacles." *Turkish Online Journal of Qualitative Inquiry*. 5(3), 1-27.
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior*. 72, 678-691.

- Sariođlu, T., & Kartal, G. (2017). Bir yöntem olarak drama bilişim teknolojileri öğretiminde iyi bir seçenek olabilir mi?. *İlköğretim Online*. 16(1), 366-376.
- Shih, I.-J. (2014). *The effect of scratch programming on the seventh graders' mathematics abilities and problem solving attitudes*. Unpublished master's thesis, Taipei University, Taiwan.
- Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*. 97, 129-141.
- Şahin, G.(2018). *Ortaokul seviyesinde programlama öğretimi için bir yöntem önerisi*. Yayınlanmamış Yüksek Lisans Tezi, Karadeniz Teknik Üniversitesi Eğitim Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalı, Trabzon.
- Uslu, N. A., Mumcu, F., & Eğin, F (2018). Görsel programlama etkinliklerinin ortaokul öğrencilerinin bilgi-işlemsel düşünme becerilerine etkisi. *Ege Eğitim Teknolojileri Dergisi*. 2(1), 19-31.
- Wagner, T. (2008). *The global achievement gap: Why even our best schools don't teach the new survival skills our children need-and what we can do about it*. Basic Books.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*. 49(3), 33–35.
- Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for All: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*. 60(6), 565–568.
- Yıldırım, A. ve Şimşek, H. (2011). *Sosyal bilimlerde nitel araştırma yöntemleri* (8. Baskı). Ankara: Seçkin Yayıncılık.
- Qian, Y., & Lehman, J. D. (2016). Correlates of success in introductory programming: A study with middle school students. *Journal of Education and Learning*. 5(2), 73.
- Weigend, M. (2014). The digital woodlouse--scaffolding in science-related scratch projects. *Informatics in Education*. 13(2), 293-305.

EKLER

Ek-1. Problemi Anlama ve Tanıma Tablosu (PAT)

#	Sorular	Cevaplar
1.	Oyundaki parkur tipleri nelerdir?	
2.	Oyundaki araç renkleri nelerdir?	
3.	Oyun kaç tur sonunda bitmektedir?	
4.	Oyunun amacı nedir?	

Ek-2. Strateji Oluşturma Tablosu (SOT)

Yol	Araçın Rengi	Yolda Geçen Süre	Araç Değişimi	Araç Değişimi Süresi
Toprak				
Asfalt				
Kum				
Çakıl				
				
Gidiş Süresi:		Değişim Süresi:		
Tur Süresi:		Toplam Süre:		