



## The Effects of Scratch Software on Students' Computational Thinking Skills

Eyup YÜNKÜL<sup>1</sup>, Gürhan Durak<sup>1\*</sup>, Serkan ÇANKAYA<sup>1</sup>, Zeynel Abidin MISIRLI<sup>1</sup>

<sup>1</sup> Balıkesir University, Balıkesir/Turkey

Received : 04.10.2017

Accepted : 25.10.2017

---

*Abstract* – The aim of this study is to determine the correlation between the scores of the computational thinking skills scale (CTSS) and the achievements of students' who took the Scratch programming education given within the scope of the course, Information & Communication Technology and Software and to investigate the effect of scratch education on the CTSS scores. In this quantitative study, the CTSS scores, the cumulative grade point averages (CGPA) of the students and the exam scores of the course were used. Before the experimental process, the CGPA scores of the 69 students were compared with t-test in two secondary schools in Balıkesir. It was found that students in two different schools have similar CGPAs. Scratch education was given to the students in the first school for a semester within the scope a course. At the end of the semester the CTSS was applied in two schools and it was found that students in the first school who were given scratch education had significantly higher scores in CTSS than the scores of students in the second schools who were not given scratch education. At this point, it can be assumed that Scratch education had a positive effect in the problem solving, algorithmic thinking and creative thinking skills which were dimensions of the CTSS. Besides, exam scores and CTSS scores of students in the experimental group were compared and it was found that there was a high level correlation between them. Accordingly, it can be said that students who are good at computational thinking skills are also good at programing. When the high scores of students in both CTSS and exam of the course was considered, it is important to disseminate the scratch education in K12 level.

*Key words:* Computational thinking, algorithmic thinking, scratch

---

\* Corresponding author: Necatibey Education Faculty, Balıkesir University, Balıkesir, Turkey

Email: gurhandurak@balikesir.edu.tr

Note: This article was supported by Balıkesir University Unit of Scientific Research Projects.

### Summary

#### Introduction

Computational thinking term have become widespread in recent years. Computational thinking can be described as a process of developing algorithmic thinking skills to solve a range of problems. Description of a problem includes understanding of the problem and

presenting the algorithms for (Thomas, Odemwingie, Saunders, & Watlerd, 2015). Wing (2006) expressed that computational thinking is a necessary skill for not only computer experts but also everyone. Computational thinking includes critical thinking, algorithmic thinking, creativity, collaborative learning, use of digital tools for solving problems. Therefore skills, that every child should have, should involves the computational thinking skill (ISTE, 2015).

When considering the studies in literature, this study is important, because it investigates both computational thinking and scratch. In recent years coding education have become widespread, so it is thought that studies investigating the relationship between coding and computational thinking can contribute to the field. Besides, there are few researches about this topic in Turkey in comparison to international literature, so it is necessary to perform new researches. In this context, the aim of this study is to determine the correlation between the scores of the computational thinking skills scale (CTSS) and the achievements of students' who took the Scratch programming education given within the scope of the course, Information & Communication Technology and Software and to investigate the effect of scratch education on the CTSS scores.

### **Methodology**

In this study, experimental design with post-test and control group was used, and correlation test was used to investigate the relationship between variables. As a data collection tool, computational thinking scale and the exam of the course of Information and Communication Technologies and Software were used. Participants of the study were 69 sixth grade students from two secondary schools in Balıkesir. 36 students in one school (experimental group) used scratch, 33 students in other school (control group) did not used scratch in the course. Before the experimental process, students' CGPAs were compared with independent samples t-test for two groups. In experimental group, pearson correlation test was used to compare the scores of computational thinking scale and the exam of the course which included scratch education. It was ensured that students in both group did not get any scratch education before the experimental process. In the experimental group students got scratch education two hours a week for a semester. In scratch education, scratch application and its features, algorithmic thinking with scratch, problem solving, linear logic, using loop and conditional statements topics were taught and creative activities were performed.

In class, first the teacher performed a sample activity and then asked students to perform similar activities and solve problems by their own. After students solved a problem, the

teacher presented a harder problem to solve, so that, students were expected to use high level thinking skills. In this course student centered learning approach was adopted. The teacher mostly guided students and students participated actively. In the end of the semester, students took a final exam, prepared by the teachers in both groups, and students filled in the questionnaire containing the computational thinking scale.

### **Findings**

It was found that CGPAs of both experimental and control groups were similar. In other words, it can be said that groups were equal in respect to academic performance. It was found that students in experimental group have significantly higher scores in computational thinking scale than those in control group. Besides in experimental group, it was found that exam scores and computational thinking scale scores of students were highly correlated.

### **Discussion**

As a result of the findings it can be said that students who have high computational thinking skills are also successful at programming. When considering the relationship between computational thinking and programming, in order to improve the computational thinking skills of students, applied coding education like scratch should be made widespread. According to the findings of the study, the following suggestions were made.

- Experimental studies investigating the traditional programming course and robotic programming course can be performed.
- A measurement tool can developed to measure the students' performances of scratch education.
- A attitude scale towards the scratch education can be developed.
- Computational thinking scale scores and exam scores of students can be examined according to various variables.
- Coding education like scratch should be integrated into K12 curriculum.

# Scratch Yazılımının Öğrencilerin Bilgisayarca Düşünme Becerilerine Etkisi

Eyup YÜNKÜL<sup>1</sup>, Gürhan Durak<sup>2\*</sup>, Serkan ÇANKAYA<sup>3</sup>, Zeynel Abidin MISIRLI<sup>4</sup>

<sup>1,2,3,4</sup> Balıkesir Üniversitesi, Necatibey Eğitim Fakültesi

Makale Gönderme Tarihi: 04.10.2017

Makale Kabul Tarihi: 25.10.2017

*Özet* – Bu çalışmada, Scratch eğitimi alan öğrencilerin Bilgisayarca Düşünme Becerileri Ölçeği (BDBÖ) puanları ile bilişim teknolojileri ve yazılım dersine ait başarı puanları arasındaki ilişkinin belirlenmesi ve scratch eğitiminin BDBÖ puanları üzerindeki etkisinin incelenmesi amaçlanmıştır. Nicel araştırma yönteminin kullanıldığı çalışmada veri olarak BDBÖ puanları, derse ait başarı puanları ve öğrencilerin ağırlıklı genel not ortalamaları kullanılmıştır. Deney öncesi işlemde Balıkesir ilinde yer alan iki farklı okuldaki toplam 69 öğrencinin akademik başarı puanları alınmış ve t-testi ile karşılaştırılmıştır. Elde edilen bulgular bu iki sınıftaki öğrencilerin birbirine denk puanlarda olduğu yönündedir. Birinci sınıfta (deney grubu) bir dönem boyunca scratch eğitimi verilmiştir. Dönem sonunda her iki sınıfta yer alan öğrencilere BDBÖ uygulanmış ve elde edilen verilerden scratch eğitimi alan öğrencilerin daha yüksek BDBÖ puanlarına ulaştıkları görülmüştür. Bu noktadan yola çıkıldığında, Scratch kullanımının problem çözme, algoritmik düşünce ve yaratıcı düşünme becerileri üzerinde anlamlı bir etkisi olduğu sonucuna varılabilir. Aynı zamanda deney grubunda yer alan bu öğrencilerin derse ait başarı puanları ile BDBÖ'ne ait puanları arasındaki ilişki incelenmiştir. Ortaya çıkan bulgular yüksek düzeyde ilişkinin olduğu sonucunu ortaya çıkarmıştır. Dolayısıyla bilgisayarca düşünme becerileri yüksek olan öğrencilerin programlama konusunda daha başarılı oldukları söylenebilir. Deney grubunda yer alan öğrencilerin BDBÖ puanları ve derse ait başarı puanlarında olumlu sonuçlar alındığı düşünüldüğünde scratch eğitiminin K12 düzeyinde yaygınlaştırılması önem arz etmektedir.

*Anahtar kelimeler:* Bilgisayarca düşünme, algoritmik düşünme, scratch

\* Sorumlu Yazar: Necatibey Eğitim Fakültesi, Balıkesir Üniversitesi, Balıkesir, Türkiye

Eposta: gurhandurak@balikesir.edu.tr

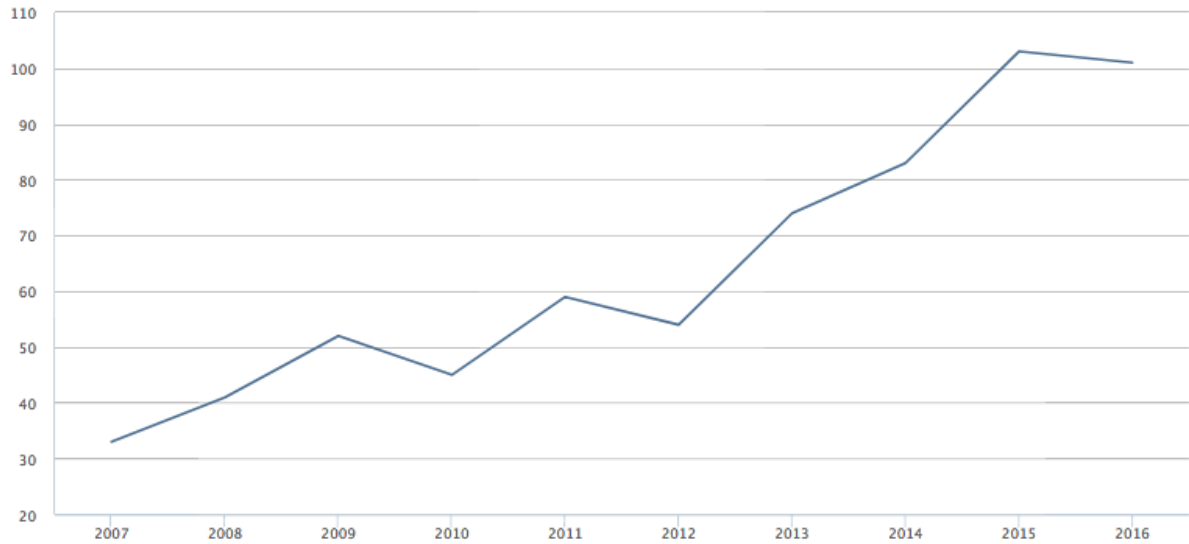
Not: Bu makale, Balıkesir Üniversitesi Bilimsel Araştırma Projeleri Birimi tarafından desteklenmiştir.

## Giriş

Bilgisayarca düşünme son yıllarda alanyazında giderek yaygınlaşan kavramlardan birisi haline gelmiştir. Bilgisayarca düşünme, bir dizi problemin çözümüne yönelik tasarım, uygulama ve algoritmik düşünme becerisi geliştirme süreci olarak ele alınabilir. Bir problemi tanımlama, problemi anlama ve problemin çözümüne yönelik olarak algoritmaları ortaya

koyma durumlarını içerir (Thomas, Odemwingie, Saunders, & Watlerd, 2015). Wing (2006)'ya göre bilgisayarca düşünme yalnızca bilgisayar uzmanları için değil herkes için gerekli bir beceri olarak tanımlanmıştır. Bilgisayarca düşünme, eleştirel düşünme, algoritmik düşünme, yaratıcılık, işbirlikli öğrenmenin yanında problemlerin çözümünde gerekli olan dijital araçları kullanma yöntemleri ve bunların hayata yansıtılmasını içermektedir. Bu nedenle okuma, yazma ve aritmetik işlemler gibi tüm çocukların sahip olması gereken becerilere bir çeşit analitik düşünme olan bilgisayarca düşünme de dahil edilmelidir (ISTE, 2015).

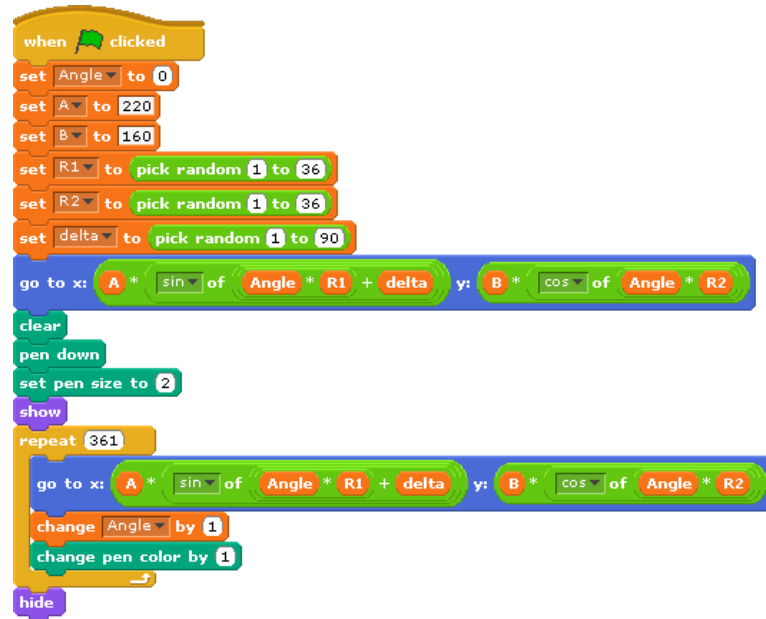
Şekil 1'de Scopus veri tabanında son 10 yılda bilgisayarca düşünme ile ilgili yapılmış çalışma sayıları yer almaktadır.



Şekil 1 incelendiğinde ilgili alandaki çalışmaların sayısı son yıllarda artış göstermiştir. Bu durum bilgisayarca düşünme kavramının popüler bir kavram olduğunu göstermektedir.

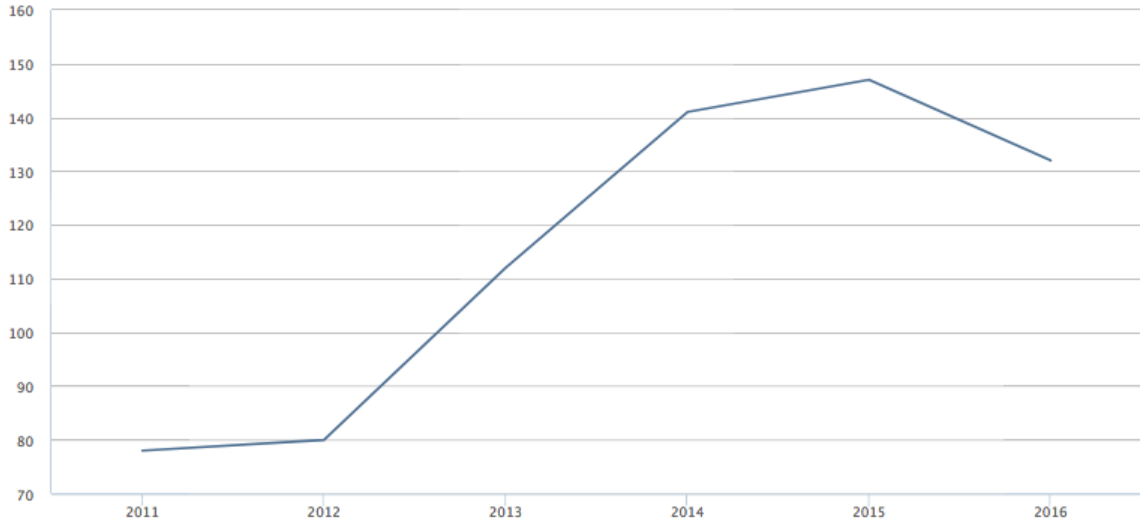
Bilgisayarca düşünme kavramının içinde yer alan algoritmik düşünmenin programlamayla yakından ilişkisi olduğu söylenebilir. Her programlama dili yapısı gereği algoritmalarından oluşmaktadır. Programlama dilleri yalnızca kod satırlarından ibaret olmayıp aynı zamanda ilgili problemin çözümüne yönelik olarak atılacak adımların mantıklı bir şekilde sıralanması sürecini içerir. Siegle (2017) ve Pearce (2013)'ye göre, bu süreçte öğrenciler problemin çözümüne yönelik yaratıcı fikirler ortaya koyarlar. Dolayısıyla, programlama, öğrencilerin bilgisayarca düşünme becerilerini geliştirmesine olanak sağlar (Lawanto, Close, Ames, & Brasiel, 2017). Bu açıdan bakıldığında programlamanın, bilgisayarca düşünmeyle birlikte ele alınması gerekir.

Başta İngiltere olmak üzere birçok ülke ilköğretim düzeyinden başlayarak öğrencilerin bilgisayarca düşünme becerilerini geliştirmek amacıyla eğitim programlarında bilgisayar programlamaya daha çok yer vermeye başlamışlardır (Bargury et al., 2012; Jones et al., 2014; Kalelioğlu, 2015; Kalelioğlu & Gülbahar, 2014; Morrison, 2013). İlköğretimde yer alan öğrencilerin programlamayla tanışmaları ve problem çözmeye yönelik kod blokları oluşturmaları açısından programlama dilleri dışında daha çok algoritma yapısının kullanıldığı diller mevcuttur. Bunlardan birisi de günümüzde K12 düzeyinde programlama öğretiminde kullanılan ve oldukça yaygın olan Scratch yazılımıdır (Yükseltürk & Altıok, 2016a). Bloklarla programlama imkânı veren Scratch (Şekil 1), sunduğu bu görsel kodlama ortamı ile düşük seviyeden yüksek seviyelere kadar program yazmak isteyenler için tercih sebebi olabilmektedir (Grover & Pea, 2013a). Sürükle bırak tarzı uygulamalar içeren Scratch ile küçük yaştaki öğrenciler sadece programlama öğrenmekle kalmamakta aynı zamanda yüksek düzeyli düşünme becerilerini de geliştirmektedirler (Zhang, Yang, Luan, Yang, & Chua, 2014).



Şekil 1. Scratch ile hazırlanmış kod bloğu

Alanyazında scratch ile ilgili oldukça fazla çalışmanın yapıldığı görülmektedir. Şekil 3'te Scopus veri tabanında 2011- 2016 yılları arasında yapılmış çalışma sayıları yer almaktadır.



Şekil 3. 2011 yılından itibaren Scratch ile ilgili yapılan çalışmaların sayısı

Şekil 3 incelendiğinde ilgili alandaki çalışmaların sayısı son yıllarda artış göstermiştir. Bu durum scratch eğitiminin popüler bir kavram olduğunu göstermektedir. Marcelino, Pessoa, Viera, Salvador ve Mendes (2017)'ye göre de scratch'in alanyazında K-12 düzeyinde en fazla kullanılan program olduğu sonucu bu grafiği destekler niteliktedir.

Alanyazında programlama öğretiminde scratch kullanımının öğrenci başarısını (Giordano & Maiorana, 2015; Taylor, Harlow, & Forret, 2010; Wang, Huang, & Hwang, 2014), programlamaya yönelik tutumunu (Malan & Leitner, 2007; Nikou & Economides, 2014; Resnick et al., 2009; Ruf, Mühling, & Hubwieser, 2014) ve motivasyonlarını (Yükseltürk & Altıok, 2016b) olumlu yönde etkilediği belirtilen çalışmalar vardır. Bununla birlikte scratch kullanımının, bilgisayarca düşünme ve bilgisayarca düşünmenin alt kategorileri olan algoritmik düşünme, problem çözme ve yaratıcılık ile ilişkisine yönelik çalışmalar yer almaktadır (Çatlak, Tekdal, & Baz, 2015; Olabe, Olabe, Basogain, Maiz, & Castaño, 2011).

K-12 seviyesinde programlama derslerinde scratch'in kullanıldığı ve bilgisayarca düşünme altında yer alan problem çözme becerilerine olumlu katkı sağladığına yönelik sonuçları olan çalışmalar (Begosso & Silva, 2013; Brown et al., 2008; O. Korkmaz, 2016; Ö. Korkmaz, Çakır, Özden, Oluk, & Sarioğlu, 2015; Lai & Yang, 2011; Nam, Kim, & Lee, 2010; Shin & Park, 2014) yer almaktadır.

Ancak Kalelioğlu ve Gülbahar (2014)'ın yaptığı deneysel çalışmada, Scratch kullanan öğrenenlerin problem çözme yetenekleri ile Scratch kullanmayan öğrenenlerin problem çözme yetenekleri arasında anlamlı bir farkın olmadığı sonucu ortaya çıkmıştır. Scratch ile

programlama öğretiminin İlköğretim öğrencilerinin işbirlikli öğrenme (Theodorou & Kordaki, 2010) ve yaratıcılıkları üzerine etkisinin araştırıldığı bazı çalışmalarda sonuçların olumlu olduğu belirtilmiştir (Kobsiripat, 2015; Pinto & Escudeiro, 2014). Grover & Pea (2013b)'nin, karma yöntem kullandıkları çalışmalarında scratch'in öğrenenlerin algoritmik düşünme becerilerine olumlu etkisi olduğu ortaya çıkmıştır.

Gerek bilgisayarca düşünme becerisi (BDB) kavramı gerekse de Scratch kavramlarıyla ilgili alanyazında yapılmış çalışma sayıları göz önüne alındığında bu iki kavramı bir arada inceleyen bu çalışma önem arz etmektedir. Özellikle kodlama eğitiminin ülkemizde giderek yaygınlaşmasıyla birlikte, bilgisayarca düşünme kavramı ile kodlama eğitimi arasındaki ilişkiyi araştıran bu tür çalışmalar alanyazına katkı sunabileceği düşünülmektedir. Bununla birlikte Türkiye'de bu alanla ilgili çok az sayıda çalışma olduğu ve ilgili kavramların uluslararası alanyazındaki önemi göz önünde bulundurulduğunda ülkemizde de daha fazla çalışma yapılması gerekmektedir. Bu doğrultuda çalışmada, Scratch eğitimi alan öğrencilerin Bilgisayarca Düşünme Becerileri Ölçeği (BDBÖ) puanları ile bilişim teknolojileri ve yazılım dersine ait başarı puanları arasındaki ilişkinin belirlenmesi ve scratch eğitiminin BDBÖ puanları üzerindeki etkisinin incelenmesi amaçlanmıştır.

## **Yöntem**

Bu bölüm; araştırma modeli, veri toplama araçları, katılımcılar, veri analizi, geçerlik ve güvenirlik ile uygulama süreci başlıklarından oluşmaktadır.

## **Araştırma Deseni**

Sontest kontrol gruplu deneysel desen olarak yapılandırılmış bu çalışmada aynı zamanda değişkenler arasında ki ilişkiyi belirlemek amacıyla korelasyon yönteminden de yararlanılmıştır.

## **Veri Toplama Araçları**

Çalışma kapsamında veriler, Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği ve dönem sonu sınavı aracılığıyla toplanmıştır.

### **1. Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği**



Araştırmada öğrencilerin Bilgisayarca düşünme becerileri düzeylerini ölçmek için kullanılan BDBÖ, Korkmaz, Çakır ve Özden (2015) tarafından geliştirilmiştir. 5'li likert tipi ölçekte 22 maddeli ve 5 faktörlü bir yapı yer almaktadır. İki tane olumsuz maddenin yer aldığı ölçekte maddeler; hiçbir zaman (1), nadiren (2), bazen (3), genellikle (4), her zaman (5) şeklinde ölçeklendirilmiştir. Ölçekte yaratıcılık, algoritmik düşünme, işbirlik, eleştirel düşünme, problem çözme olmak üzere 5 faktörlü bir yapı mevcuttur. Ölçeğe tamamına ait Cronbach Alpha değeri 0.809'dur.

## **2. Bilişim Teknolojileri ve Yazılım Dersi dönem sonu sınavı**

Deney grubundan yer alan scratch eğitimi alan öğrencilerin Bilgisayarca düşünme becerileri düzeyleri ile ders notları arasındaki korelasyonun belirlenmesi amacıyla Bilişim Teknolojileri ve Yazılım Dersi dönemsonu sınavı kullanılmıştır. Sınav, dersi veren öğretmen tarafından algoritmik düşünme, problem çözme ve yaratıcılığı ölçecek biçimde hazırlanmıştır. Örneğin bir problem belirlenmiş ve bu problemin çözümüne yönelik olarak adımların sıralanması istenmiştir. Ayrıca yine bir problemin çözüm aşamalarından bazıları eksik olarak verilmiş ve öğrencilerden bu eksikliklerin yerine gelmesi gereken kod blokları istenmiştir.

### **Katılımcılar**

Çalışmanın katılımcıları Balıkesir ilindeki, iki farklı okulda 6. sınıflarda yer alan toplam 69 öğrenciden oluşmaktadır. Bu öğrencilerden bir okulda (deney grubu) Bilişim Teknolojileri ve Yazılım Dersinde Scratch ile öğretim yapılırken diğer okulda (kontrol grubu) ise derslerde Scratch kullanılmamıştır. Deney grubunda 36, kontrol grubunda ise 33 öğrenci yer almaktadır.

### **Veri Analizi**

Araştırmada deneysel işlem öncesinde grupların denkliliğini belirlemek amacıyla öğrencilerin akademik ortalamaları bağımsız gruplar t-testi ile analiz edilmiştir. Deneysel işlem sonrasında deney ve kontrol grupları arasında bilgisayarca düşünme becerileri düzeyi açısından anlamlı fark olup olmadığını belirlemek amacıyla yine bağımsız gruplar t-testi kullanılmıştır. Yapılan Kolmogrov-Smirnov ve Shapiro-Wilk testlerinde verilerin normal dağıldığı görülmüştür ( $p > .05$ ). Deney grubunda yer alan öğrencilerin bilgisayarca düşünme becerileri düzeyi ve Bilişim Teknolojileri ve Yazılım dersinden (Scratch konusu) aldıkları not arasındaki korelasyonu belirlemek amacıyla Pearson Correlation testi yapılmıştır.

BDBÖ'ye ait güvenilirlik çalışmasında tüm ölçeğe ait iç tutarlılık değerini gösteren Cronbach Alpha değeri .88 bulunmuştur. Ortaya çıkan bu güvenilirlik değerinden ölçeğin “oldukça güvenilir” düzeyde (Can, 2013) bir ölçek olduğu sonucuna varılabilir.

### Uygulama Süreci

DeneySEL işlem öncesi hem deney grubu hem de kontrol grubu öğrencilerinin daha önce scratch yazılımı hakkında bilgi sahibi olmadıkları belirlenmiştir. Deney grubuna yapılan scratch öğretimi derin öğretmeni tarafından 1 yarıyıl boyunca haftada 2 saat süreyle sürdürülmüştür. Öğrencilere derste scratch uygulaması ve özellikleri, scratch ile algoritmik düşünme, problem çözme, doğrusal mantık, döngü mantık ve karar mantık yapılarını öğrenerek yaratıcı etkinlikler oluşturma eğitimi verilmiştir.

Derslerde öncelikle örnek bir etkinlik öğretmen tarafından gerçekleştirilmiş, ardından öğrencilere benzer etkinlikler verilerek ilgili problemleri çözmeleri istenmiştir. Öğrenciler problemi çözdükçe problem biraz daha zorlaştırılarak ileri seviyelere taşınmış, böylece üst düzey düşünmelerine imkan sunulmuştur. Bu uygulamada daha çok öğrenci merkezli bir öğretim yöntemi benimsenmiştir. Öğretmen daha çok rehberlik etmiş ve öğrenciler ise aktif bir şekilde derse katılım göstermişlerdir. Dönem sonunda deney grubu ve kontrol grubunda yer alan öğrencilere dersin öğretmeni ve araştırmacılar tarafından hazırlanan dönemsonu sınavı ve BDBÖ uygulanmıştır.

### Bulgular ve Yorumlar

Çalışmanın bu bölümünde elde edilen bulgular tablolar halinde sunulmuş ve alanyazındaki diğer çalışmaların bulgularıyla karşılaştırmalı olarak yorumlanmıştır.

Çalışmada yer alan deney ve kontrol gruplarının akademik başarı düzeyleri açısından farklı olup olmadıkları öğrencilerin akademik ortalamaları karşılaştırılarak belirlenmiştir. Deney ve kontrol grubu öğrencilerinin akademik ortalamaları bağımsız gruplar t testi ile incelenmiş ve sonuçlar Tablo 1’de verilmiştir.

Tablo 1. Deney ve Kontrol Gruplarının akademik ortalamalarının karşılaştırılması için t-testi

	N	X	SS	t	df	p
Deney Grubu	36	79,77	6,79	1,245	67	,217*
Kontrol Grubu	33	77,69	7,08			

\*p>.05

Tablo 1 incelendiğinde, deney ve kontrol gruplarının akademik ortalamalarının birbirine yakın olduğu görülmektedir (Deney grubu  $X=79.77$ , kontrol grubu  $X=77.69$ ). Bu puanların arasında istatistiksel olarak fark olmadığı görülmüştür ( $t_{(67)}=1.245$ ,  $p>.05$ ). Diğer bir deyişle deney ve kontrol gruplarının akademik başarı açısından birbirine denk olduğu söylenebilir.

Deneyel işlem sonrasında deney ve kontrol gruplarının BDBÖ ölçeğinden toplanan verilere göre hesaplanan ortalama puanlar bağımsız gruplar t testi ile karşılaştırılmıştır (Tablo 2).

Tablo 2. Deney ve Kontrol Gruplarının BDB düzeylerinin karşılaştırılması için t-testi

	N	X	SS	t	df	p
Deney Grubu	36	3,86	,57	3,05	67	,003*
Kontrol Grubu	33	3,50	,38			

\* $p<.05$

Tablo 2'ye göre deney grubu öğrencilerinin BDB düzeyi puanları kontrol grubu öğrencilerinin puanlarına göre anlamlı bir şekilde farklıdır ( $t_{(67)}=3,05$ ;  $p<.05$ ). Scratch eğitimi alan deney grubu öğrencilerinin BDB puanlarının ortalamasının kontrol grubu öğrencilerine göre daha yüksek olduğu görülmektedir. Ortaya çıkan bu sonucu alanyazında yer alan çalışmalar (Begosso & Silva, 2013; Brown et al., 2008; Ö. Korkmaz, Çakır, Özden, et al., 2015; Lai & Yang, 2011; Nam et al., 2010; Oluk & Korkmaz, 2016; Shin & Park, 2014) destekler niteliktedir. Bu sonucun aksine alanyazında Scratch kullanımının problem çözme becerileri üzerinde anlamlı bir etkisi olmadığına yönelik yapılmış çalışmalar da (Kalelioğlu & Gülbahar, 2014) mevcuttur. Ortaya çıkan farklı sonuçların nedenleri arasında verilen eğitimin niteliği, öğretmen faktörü ve kullanılan deneysel yöntemin etkili olabileceği düşünülmektedir. Deneyel işlem sonrasında deney grubuna uygulanan dönem sonu sınavı puanları ile BDB ölçeğinden elde edilen puanlar arasındaki korelasyona bakılmıştır. Elde edilen veriler Tablo 3'te verilmiştir.

Tablo 3. Deney grubunda yer alan öğrencilerin dönem sonu sınavı puanları ile BDBÖ puanları arasındaki korelasyon tablosu

		BDB Düzeyi	Ders Notu
BDB Düzeyi	Pearson Correlation	1	,765
	Sig. (2-tailed)		,000
	N	36	36
Ders Notu	Pearson Correlation	,765**	1
	Sig. (2-tailed)	,000	
	N	36	36

Tablo 3'e göre deney grubuna ait dönem sonu sınavı puanları ile yine deney grubuna ait BDB ölçeğinden elde edilen puanlar arasında yüksek korelasyon olduğu görülmüştür ( $r=,765;p<.01$ ). Bu bulguyu destekler nitelikte yapılan bir çalışmada (Oluk & Korkmaz, 2016) BDB düzeyleri ile programlama becerisi arasında yüksek düzeyde ilişki olduğu sonucu yer almaktadır. Bu durumdan yola çıkılarak, bilgisayarca düşünme becerileri yüksek olan öğrencilerin programlama konusunda da başarılı oldukları sonucuna varılabilir.

## Sonuç ve Tartışma

Scratch eğitiminin BDB düzeylerine üzerine etkisinin araştırıldığı bu çalışmada verilen eğitimin olumlu sonuçlar ortaya çıkardığı söylenebilir. Deney öncesi işlemde her iki grupta yer alan öğrencilerin akademik ortalamalarının birbirine denk olduğu sonucuna varılabilir. Dönem sonunda her iki gruba da uygulanan BDBÖ verilerine göre scratch eğitimi alan deney grubu öğrenenlerin diğer gruba göre daha yüksek puanlar elde ettiği görülmüştür. Dolayısıyla Scratch kullanımının problem çözme, algoritmik düşünce ve yaratıcı düşünme becerileri üzerinde anlamlı bir etkisi olduğu sonucuna varılabilir. Deney grubu öğrencileri üzerinde yapılan korelasyon testi sonucunda dönem sonu sınavı puanları ile BDBÖ'den elde edilen puanlar arasında yüksek düzeyde ilişkinin olduğu görülmüştür. Bu sonuçlar doğrultusunda yüksek bilgisayarca düşünme becerileri olan öğrencilerin programlama konusunda da başarılı oldukları söylenebilir. Öğrencilerin BDB puanları ve derse ait başarı puanlarının scratch eğitimi sonrasında olumlu sonuçlar verdiği düşünüldüğünde bu tür uygulamalı eğitimlerin yaygınlaştırılması önem arz etmektedir. Elde edilen bulgular ışığında aşağıdaki önerilere yer verilmiştir.

- Geleneksel programlama eğitimi ile robotlarla verilen programlama eğitimi arasındaki farklılıkları ortaya çıkarmak üzere deneysel çalışmalar yapılabilir.
- Scratch eğitimi sonrasında öğrencilerin başarılarını ölçmek amacıyla bir değerlendirme aracı geliştirilebilir.
- Scratch eğitimi sonrasında öğrencilerin tutumlarını belirleyebilmek amacıyla bir tutum ölçeği geliştirilebilir.
- BDBÖ verileri ile derse ait başarı puanları çeşitli değişkenler açısından incelenebilir.
- Ortaya çıkan olumlu bulgular doğrultusunda programlama eğitiminde özellikle K-12 düzeyinde scratch eğitiminin yaygınlaştırılması ve öğretim programlarında daha fazla yer verilmesi önem arz etmektedir.

- Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümlerinde bu tür programların öğretimini teşvik edilmesi önerilmektedir.

### Kaynakça

- Bargury, I. Zur, Haberman, B., Cohen, A., Muller, O., Zohar, D., Levy, D., & Hotoveli, R. (2012). Implementing a new Computer Science Curriculum for middle school in Israel. *2012 Frontiers in Education Conference Proceedings*, 1–6. <http://doi.org/10.1109/FIE.2012.6462365>
- Begosso, L. C., & Silva, P. (2013). Teaching computer programming : a practical review. In *IEEE Frontiers in Education Conference* (pp. 1–3).
- Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E., & Fontecchio, A. (2008). Computer aided instruction as a vehicle for problem solving: Scratch boards in the middle years classroom. *ASEE Annual Conference and Exposition, Conference Proceedings*. Retrieved from <http://www.scopus.com/inward/record.url?eid=2-s2.0-56749091148&partnerID=tZOtx3y1>
- Can, A. (2013). *SPSS ile Bilimsel Araştırma Sürecinde Nicel Veri Analizi* (1st ed.). Pegem Akademi.
- Çatlak, Ş., Tekdal, M., & Baz, F. (2015). Scratch Yazılımı İle Programlama Öğretiminin Durumu: Bir Doküman İnceleme Çalışması. *Journal of Instructional Technologies &*, 4(3), 13–25. Retrieved from <http://www.jitte.org/article/view/5000163313>
- Giordano, D., & Maiorana, F. (2015). Teaching algorithms: Visual language vs flowchart vs textual language. In *2015 IEEE Global Engineering Education Conference (EDUCON)* (pp. 499–504). IEEE. <http://doi.org/10.1109/EDUCON.2015.7096016>
- Grover, S., & Pea, R. (2013a). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43. <http://doi.org/10.3102/0013189X12463051>
- Grover, S., & Pea, R. (2013b). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43. <http://doi.org/10.3102/0013189X12463051>
- ISTE. (2015). *CT Leadership toolkit*. Retrieved from <http://www.iste.org/docs/ct-documents/ctleadershipt-toolkit.pdf?sfvrsn=4>.
- Jones, S. P., Liu, C. C., Cheng, Y. B., Huang, C. W., Kalelioğlu, F., Bers, M. U., ... Houlden, N. (2014). Computing at school in the UK : from guerrilla to gorilla. *Computers and*

- Education*, 72(3), 1–13. <http://doi.org/10.1016/j.compedu.2013.10.020>
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, 200–210. <http://doi.org/10.1016/j.chb.2015.05.047>
- Kalelioğlu, F., & Gülbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills : A Discussion from Learners ' Perspective. *Informatics in Education*, 13(1), 33–50.
- Kobsiripat, W. (2015). Effects of the media to promote the scratch programming capabilities creativity of elementary school students. *Procedia-Social and Behavioral Sciences*, (174), 227–232.
- Korkmaz, O. (2016). The effect of Lego Mindstorms Ev3 based design activities on students' attitudes towards learning computer programming, self-efficacy beliefs and levels of academic achievement. *Baltic Journal of Modern Computing*, 4(4), 994–1007. <http://doi.org/10.22364/bjmc.2016.4.4.24>
- Korkmaz, Ö., Çakır, R., & Özden, M. Y. (2015). ( BDBD ) ORTAOKUL DÜZEYİNE UYARLANMASI ADAPTATION FOR SECONDARY SCHOOL LEVEL. *Gazi Eğitim Bilimleri Dergisi*, 1(2), 143–462.
- Korkmaz, Ö., Çakır, R., Özden, M. Y., Oluk, A., & Sarıoğlu, S. (2015). Bireylerin Bilgisayarca Düşünme Becerilerinin Farklı Değişkenler Açısından İncelenmesi. *Ondokuz Mayıs Üniversitesi Eğitim Fakültesi Dergisi*, 34(2), 68–87. <http://doi.org/10.7822/omuefd.34.2.5>
- Lai, A.-F., & Yang, S.-M. (2011). The learning effect of visualized programming learning on 6th graders' problem solving and logical reasoning abilities. In *2011 International Conference on Electrical and Control Engineering* (pp. 6940–6944). IEEE. <http://doi.org/10.1109/ICECENG.2011.6056908>
- Lawanto, K., Close, K., Ames, C., & Brasiel, S. (2017). Exploring Strengths and Weaknesses in Middle School Students' Computational Thinking in Scratch. In *Emerging Research, Practice, and Policy on Computational Thinking* (pp. 307–326). Cham: Springer International Publishing. [http://doi.org/10.1007/978-3-319-52691-1\\_19](http://doi.org/10.1007/978-3-319-52691-1_19)
- Malan, D. J., & Leitner, H. H. (2007). Scratch for Budding Computer Scientists. *Sigcse 2007: Proceedings of the Thirty-Eighth Sigcse Technical Symposium on Computer Science*

- Education*, 223–227. <http://doi.org/10.1145/1227504.1227388>
- Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T., & Mendes, A. J. (2017). Learning Computational Thinking and Scratch at Distance. *Computers in Human Behavior*. <http://doi.org/10.1016/j.chb.2017.09.025>
- Morrison, N. (2013). Teach kids how to code and you give them a skill for life. *Forbes*.
- Nam, D., Kim, Y., & Lee, T. (2010). The Effects of Scaffolding-Based Courseware for The Scratch Programming Learning on Student Problem Solving Skill. In *ICCE2010* (pp. 723–727).
- Nikou, S. A., & Economides, A. A. (2014). Transition in student motivation during a scratch and an app inventor course. In *2014 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1042–1045). IEEE. <http://doi.org/10.1109/EDUCON.2014.6826234>
- Olabe, J. C., Olabe, M. A., Basogain, X., Maiz, I., & Castaño, C. (2011). Programming and Robotics with Scratch in Primary Education. *Education in a Technological World: Communicating Current and Emerging Research and Technological Efforts*, (July 2016), 356–363.
- Oluk, A., & Korkmaz, Ö. (2016). Comparing Students ' Scratch Skills with Their Computational Thinking Skills in Terms of Different Variables. *I.J. Modern Education and Computer Science*, 11(November), 1–7. <http://doi.org/10.5815/ijmecs.2016.11.01>
- Pearce, K. (2013). Why you should learn to code (and how to actually do it). *DIYGenius*.
- Pinto, A., & Escudeiro, P. (2014). The use of Scratch for the development of 21st century learning skills in ICT. In *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1–4). IEEE. <http://doi.org/10.1109/CISTI.2014.6877061>
- Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., ... Silver, J. (2009). Scratch. *Communications of the ACM*, 52(11), 60. <http://doi.org/10.1145/1592761.1592779>
- Ruf, A., Mühlring, A., & Hubwieser, P. (2014). Scratch vs. Karel. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education on - WiPSCE '14* (pp. 50–59). New York, New York, USA: ACM Press. <http://doi.org/10.1145/2670757.2670772>
- Shin, S., & Park, P. (2014). A Study on the Effect affecting Problem Solving Ability of Primary Students through the Scratch Programming Research method. *Advanced Science*

- and Technology Letters*, 59(Education), 117–120.
- Siegle, D. (2017). Encouraging Creativity and Problem Solving Through Coding. *Gifted Child Today*, 40(2), 117–123. <http://doi.org/10.1177/1076217517690861>
- Taylor, M., Harlow, A., & Forret, M. (2010). Using a Computer Programming Environment and an Interactive Whiteboard to Investigate Some Mathematical Thinking. *Procedia - Social and Behavioral Sciences*, 8, 561–570.
- Theodorou, C., & Kordaki, M. (2010). Super mario: A collaborative game for the learning of variables in programming. *International Journal of Academic Research*, 2(4).
- Thomas, J. O., Odemwingie, O. C., Saunders, Q., & Watlerd, M. (2015). While Enacting Computational Algorithmic Thinking in the Context of Game Design. *Journal of Computer Science and Information Technology*, 3(1). <http://doi.org/10.15640/jcsit.v3n1a2>
- Wang, H. Y., Huang, I., & Hwang, G. J. (2014). Effects of an Integrated Scratch and Project-Based Learning Approach on the Learning Achievements of Gifted Students in Computer Courses. In *2014 IIAI 3rd International Conference on Advanced Applied Informatics* (pp. 382–387). IEEE. <http://doi.org/10.1109/IIAI-AAI.2014.85>
- Yükseltürk, E., & Altıok, S. (2016a). Bilişim Teknolojileri Öğretmen Adaylarının Programlama Öğretiminde Scratch Aracının Kullanımına İlişkin Algıları. *Mersin Üniversitesi Eğitim Fakültesi Dergisi*, 12(1). <http://doi.org/10.17860/efd.94270>
- Yükseltürk, E., & Altıok, S. (2016b). Bilişim Teknolojileri Öğretmen Adaylarının Programlama Öğretiminde Scratch Aracının Kullanımına İlişkin Algıları. *Mersin Üniversitesi Eğitim Fakültesi Dergisi*, 12(1). <http://doi.org/10.17860/efd.94270>
- Zhang, H., Yang, Y., Luan, H., Yang, S., & Chua, T.-S. (2014). Start from Scratch. In *Proceedings of the ACM International Conference on Multimedia - MM '14* (pp. 187–196). New York, New York, USA: ACM Press. <http://doi.org/10.1145/2647868.2654915>