

Flowchart Destekli Proje Tabanlı Algoritma Eğitiminin Etkililiği

Yunus Emre Üngör¹, Özgen Korkmaz*¹, Recep Çakır¹ & Feray Uğur Erdoğan¹
¹Amasya Üniversitesi, Türkiye

Gönderilme Tarihi (Received): 12/04/2020
Düzeltilme Tarihi (Revised): 23/10/2020
Kabul Tarihi (Accepted): 29/10/2020

Özet

Kodlama bilgisayar ortamında bir verinin işlenmesine yönelik kuralların komutlar şeklinde sıralı bir biçimde verildiği mantıksal yapılar olarak tanımlanabilir. Oluşturulan kodun mantıksal yapısını algoritma oluşturur. Bu nedenle kodlama süreci algoritmanın ortaya konmasıyla başlar. Bu çalışmanın amacı Flowchart 3.0 görsel programlama aracı ile proje tabanlı algoritma eğitiminin ortaokul öğrencilerinin kodlamaya dönük tutum, problem çözme becerisi ve bilgisayarca düşünme becerisi üzerindeki etkisini belirlemektir. Çalışma grubu 30 öğrenciden oluşmaktadır. Deney grubu öğrencilerine Flowchart destekli proje tabanlı algoritma eğitimi verilmiştir. Kontrol grubunda ise aynı konular kâğıt kalem etkinlikleri ile anlatılmıştır. Karma araştırma yaklaşımına göre tasarlanmış olan bu çalışmada sıralı açıklayıcı desen kullanılmıştır. Araştırma verileri bilgisayarca düşünme beceri düzeyleri ölçeği, kodlamaya yönelik tutum ölçeği, problem çözme becerilerine yönelik algı ölçeği ve görüşme formu kullanılarak toplanmıştır. Bulgulara göre, Flowchart destekli proje tabanlı algoritma eğitiminin öğrencilerin bilgisayarca düşünme becerilerine, kodlamaya yönelik tutumlarına ve problem çözme becerilerine yönelik algılarına anlamlı düzeyde daha fazla katkı sağlamadığı belirlenmiştir. Buna karşın öğrencilerin deneysel etkinliklere dönük düşüncelerinin olumlu olduğu belirlenmiştir.

Anahtar kelimeler: Algoritma, flowchart, bilgisayarca düşünme, problem çözme, kodlamaya dönük tutum

Effectiveness of Project-Based Algorithm Training with Flowchart Support

Abstract

Coding can be defined as logical structures in which rules for processing data in a computer environment are given in an ordered form in the form of commands. The algorithm creates the logical structure of the generated code. Therefore, the coding process begins with the introduction of the algorithm. The aim of this study is to determine the effect of Project-based algorithm training with Flowchart 3.0 visual programming tool on coding attitudes, problem solving skills and computer thinking skills in secondary school students. The study group consisted of 30 students. Project-based algorithm training with Flowchart support was provided to the experimental group students. In the control group, same topics are explained with paper-pencil activities. Sequential explanatory design was used in this research, which was designed according to the mixed research approach. The research data were collected using the computer thinking skill levels scale, the attitude scale for coding, the perception scale for problem solving skills, and the interview form. According to the findings, it was determined that Flowchart supported project-based algorithm training did not significantly contribute to students' perceptions about computer thinking skills, coding attitudes, and problem solving skills. Despite this, it was determined that students' thoughts about experimental activities were positive.

Keywords: Algorithm, flowchart, computer thinking, problem solving, attitude towards coding

GİRİŞ

Algoritma; 900 yıl önce yaşamış ünlü matematikçi Muhammed İbni Musa el Harezmi tarafından keşfedilmiş, adını da El Harezmi'nin Latince çevrisi olan Algoritmi'den almıştır. Güven (2018) algoritmayı; "bir problem durumunu çözmek amacıyla planlanan yol" olarak tanımlamıştır. Gülbahar, Kalelioğlu ve Karataş'a (2017) göre ise algoritma; "problemi çözmek için mantıksal sırada çözüm adımlarını oluşturmaktır". Bir programlama dilini öğrenmeden önce programlama mantığı genellikle ilk olarak algoritma ile öğrenilmektedir. Programlamaya geçmeden önce algoritma tasarlanmasının programlama sürecini kolaylaştıracağı beklenir. Gökoğlu ve Yüksel'e (2016) göre algoritma becerisi, problemlere farklı açılardan bakabilme ve en kısa çözümü bulabilme yeteneği kazandırır. Algoritmalar sorunun çözümünde adımların birbirleri ile olan ilişkilerini, akışını ve hatalarını kolaylıkla görmemizi sağlar. Programlama sürecine algoritmayla başlanıldığında algoritma adımı, programlama koduna dönüştürülerek programcının işi kolaylaştırılmaktadır (MEGEP, 2011). Millî Eğitim Bakanlığının 2023 Eğitim vizyonunda; bilgisayarsız ortamlarda algoritmik düşünce öğretimine yönelik, sınıf öğretmenlerine yüz yüze hizmet içi eğitimler verileceği ifade edilmektedir. Bu ifade algoritma eğitiminin önemine dikkat çekmektedir (MEB, 2018).

Algoritma hangi işlemin hangi sırada yapılacağını gösteren yönergeler bütünüken Akış Diyagramı ise algoritmanın şekil ve sembollerle görsel gösterimidir. Akış diyagramında hatalar daha kolay tespit edilebilmektedir. Akış diyagramını oluşturmak, program geliştirmeye geçmeden önceki son adımdır (Gülbahar, Kalelioğlu & Karataş, 2017). Her farklı hareket için farklı akış şeması kullanılır. Bu şemalar geometrik şekiller kullanılarak düzenlenir. Örneğin elips şekli Akış diyagramının başladığı ve sonlandığı yeri, Paralelkenar veri girişini, Dikdörtgen işlemlerin yapıldığı yeri, deltoid benzeri şekil mantıksal sınımayı, yuvarlak bağlantı noktalarını belirtir (Güven, 2018). Akış şemasında kullanılan şekiller evrensel olduğundan ve her adım görev türüne göre şekillerle ifade edildiğinden, algoritmaya göre daha nesnel, daha anlaşılır olması beklenebilir. Uzamsal zekâya sahip bireyler düşüncelerini şekil ve sembollerle daha kolay ifade ederler (Talu, 1999). Görsel zekâsı gelişmiş öğrencilerin akış şemaları ile algoritma mantığını daha iyi kavrayacağı öngörülebilir. Bu bağlamda algoritma eğitiminin akış şemaları ile desteklenmesinin kodlama öncesinde iyi bir hazırlık olacağı düşünülebilir. Kodlamada kullanılan kod bloklarının sıralanması ve birbirleriyle ilişkilendirilmesinin çoğunlukla algoritma mantığı ile aynı olduğu söylenebilir. Bu çerçevede algoritma ve akış diyagramı kavramları kodlama eğitimine başlangıç niteliğinde önemli kavramlardır. Kodlama eğitiminin nihai amacına ulaşması için kullanılacak uygun öğretim modellerinden birinin proje tabanlı öğretim olduğu söylenebilir.

Proje; en kısa tanımıyla gerçekleştirilmesi istenen tasarıdır (TDK, 2019). Korkmaz ve Kaptan'a (2001) göre proje tabanlı öğrenme; günlük hayattan seçilen bir problem durumunun bireysel ya da küçük gruplarla problemi çözmeye çalışan bir öğrenme yaklaşımıdır. Bu öğrenme anlayışını oluşturan tabanlı sözcüğü; projenin hedef değil bir altyapı unsuru olduğuna ve süreç yönelimli yapılanmaya işaret etmektedir. Dolayısıyla öğrenme hedeflerinin, sonuç değil; süreç becerilerini tanımlayacak biçimde oluşturulması gerekmektedir. Proje tabanlı öğrenme ortamı; yaşamın sınıf ortamına getirildiği, öğrencilerin öğrenme sürecinde aktif rol aldığı, öğrencilerin problem durumunu iş birliği ile çözmeye çalıştıkları teknoloji destekli bir öğrenme ortamıdır. Proje tabanlı öğrenme sürecinde izlenmesi gereken belli başlı adımlar vardır. Sırasıyla hedeflerin belirlenmesi, üzerinde çalışacağımız işin ya da sorunun belirlenip tanımlanması, nihai raporun özelliklerinin ve nasıl sunulacağının belirlenmesi, değerlendirme aşamasında ölçütlerin ve yeterlik seviyelerinin belirlenmesi, proje gruplarının oluşturulması, alt soruları belirleyip, bilgi toplamaya ayrılan zamanın tasarlanması, çalışma takvimine son halinin verilmesi, kontrol noktalarının tespit edilmesi, verilerin toplanması, örgütlenmesi ve raporlanması ve projenin

sunulması proje tabanlı öğrenme süreci adımları olarak ifade edilmektedir (Erdem, 2002). Öğrencilerin disiplinler arası yaklaşımla farklı disiplinlerde elde ettikleri bilgi ve becerilerini sentezlemek, ürün geliştirmek, yaşamdan kopuk olmamak amacıyla proje geliştirmeleri beklenmektedir (MEB, 2018).

Kodlama bilgisayarın kendisinden istenen görevleri yapabilmesi için ona algoritmik yönerge sunmaktır (Güven, 2018). Kodlama temelini algoritmadan almaktadır. Kodlama eğitim süreci algoritmaya başlamaktadır (Akkuş, Özhan & Kan, 2019). Algoritma eğitimi almış birinin kodlama eğitimine daha kolay uyum sağlayıp, olumlu tutumlar geliştireceği düşünülebilir. Programlama ve kodlama eş anlamlı olarak kullanılsa da programlama daha çok C, C++, Basic, Pascal gibi programlama dillerinde program tasarlamayı, kodlama ise Alice, Scratch gibi blok tabanlı editörlerle kod geliştirmeyi ifade ettiği söylenebilir. Metin tabanlı programlama araçları ile programlamaya başlayan öğrenciler; noktalı virgül, büyük küçük harf gibi söz dizimsel hatalarla çok karşılaşır. Bu durum programlama sürecinde öğrencilerin çok farklı zorluklarla karşılaşmasına neden olabilir. Öğrencilerin bu tür zorluklarla karşı karşıya kalmalarını engellemek amacıyla görsel programlama araçları üretilmiştir. Bu araçlardan birisi de flowchart'dır.

Flowchart görsel programlama aracında söz dizimsel hatalarla karşılaşmaz. Ayrıca akış şemalarıyla her adımın çalışması takip edilebildiği için hata ayıklama daha basittir. Flowchart genel amaçlı program geliştirmek için en iyi yoldur (Charntaweekhun & Wangsiripitak, 2006). Görsel programlama noktalı virgül koyma gibi söz dizimsel zorunlulukları ortadan kaldırdığı için motive edicidir (Koh, Basawapatna, Bennett & Repenning, 2010). Görsel kodlama metin tabanlı programlamadaki karmaşıklığın önüne geçerek öğrencilerin eğlenerek ürün geliştirmelerine fırsat sağlamaktadır. Öğrencilerin eğlenerek kendi ürünlerini geliştirmeleri kodlamaya karşı motivasyonlarını artırmaktadır (Baz, 2018). Lisans eğitiminden önce görsel programlama öğrenmek; C, Java gibi metin tabanlı programlama öğrenmeyi kolaylaştırır. Aynı zamanda yaratıcı düşünme ve analitik düşünme becerisi kazandırır (Kabak & Güneş, 2013). Okullarda Algoritma ve kodlama öğretmek öğrencilerin düşünme becerilerine katkıda bulunmalıdır (Şahin, Korkmaz, Çakır & Erdoğan, 2019). Kodlama mantıksal akıl yürütme becerilerinin bir parçası olan 21. yy. becerilerindedir. Kodlama hem öğrenciler için hem de iş dünyası için bir yetkinlik olmuştur (Sayın & Seferoğlu, 2016). Kodlama programlama bilgisi ile birlikte problem çözme becerisi de kazandırır (Baz, 2018).

Problem çözme becerisi yeterince gelişmemiş çocukların, karşılaştıkları problemleri çözmekten çok, problemleri yok sayıp geçtirdikleri ve bu çocukların kendine güven eksikliği, çaresizlik gibi çeşitli psikolojik olumsuzluklar yaşayabilecekleri belirtilmiştir (Delebe, 2018). Problem çözme becerisi sadece okul yıllarında gerekli bir beceri değil aynı zamanda tüm hayat boyu gerekli bir beceridir (Bingham, 2016). Problem çözme yeteneği edinmiş öğrencilerin özgüvenleri de yüksektir (Hidayat & Susilowati, 2020). Güven'e (2018) göre kişinin günlük yaşamında karşılaştığı, çözüm aradığı ve çözmek için bilgisini, mantığını, geçmiş deneyimlerini kullandığı durumlara problem denirken problem çözme becerisi ise bir problemi tanımlamak ve bu durumla baş etmek için gösterilen çaba olarak ifade edilmektedir. Problemin iki boyutundan biri problemi çözmek için gereksinim duymak diğeri ise çözüm için geliştirilen yollardan en uygununu seçmektir. Seçilen yol başarılı olmamışsa diğeri seçenekler denemelidir (Şahin, 2004). Problem çözme becerisi sırasıyla problem durumu ile ilgili alanda bilgili olma, çözüm için stratejiler geliştirme ve bu stratejilerden en uygununu seçip test etme şeklinde süreç odaklı bir beceridir. Eğitim kurumlarının en önemli hedef kazanımlarından birisi öğrencilere problem çözme becerisi kazandırmak olarak ifade edilmektedir (Yıldırım, 2018). Problem çözme süreci basamaklar halinde aşağıdaki gibi ifade edilebilir:

- 1-) Problemi tanımlama
- 2-) Problemi anlama
- 3-) Problemin çözmek amacıyla yollar ve yöntemler belirleme
- 4-) Farklı çözüm yollarından en uygun çözümü seçme
- 5-) Seçilen çözüm yolu ile problemi çözmek amacıyla yönergeleri tasarlama
- 6-) Çözümü değerlendirme
- 7-) Çözüm başarısızsa üçüncü adımdan döngüsel devam etme (Gülbahar, Kalelioğlu & Karataş, 2017)

Problem çözme becerisi öğrenilebilen, geliştirilebilen bir beceridir (Şahin, 2004). Problem çözme becerisi gelişmiş bir öğrencinin gerek duyuşsal, gerek bilişsel alanda kendi kendine yeten bir birey olacağı öngörülebilir. Değişen ve gelişen bir dünyada bireyin yeni durumlara en kısa sürede uyum sağlayabilmesi problem çözme becerisi ile ilişkilendirilebilir. Problem çözme becerisi; birçok beceri gerektiren bilgisayarca düşünme becerisi ile ilişkilidir (Selby & Woollard, 2013). Bilgisayarca düşünme; problem çözme sürecinde bilgisayarın kullanıldığı bir tür problem çözme becerisidir (Korkmaz, Çakır, Özden & Saroğlu, 2015).

Bilgisayarca düşünme, problemleri formüle etmek için kullanılan düşünce süreçleridir (Grover & Pea, 2013). Korkmaz, Çakır ve Özden'e (2015) göre bilgisayarca düşünme; bilgisayar biliminden yararlanılarak insan davranışlarını anlamaya dönük, 21. yy öğrencilerin kazanmaları gereken bir problem çözme becerisi olduğu ve bilgisayarca düşünmenin birçok alt beceriyi kapsadığı ifade edilmektedir. Bilgisayarca düşünmenin tam bir ortak tanımının olmadığı ancak; mantıksal düşünme, algoritmik düşünme, problem çözme, analiz, genelleme, sistem tasarımı, otomasyon gibi terimlerle ilişkili bir düşünce türü olduğu belirtilmiştir (Selby & Woollard, 2013). Bilgisayarca düşünme becerisi alt becerilerinden yaratıcı düşünme; kişinin mantığını kullanarak özgün bir ürün elde etmesidir. Eleştirel düşünme ise; önyargıları farkına varıp yeni düşünceler keşfetmedir. Öğretmenin aktardığı bilgiyi pasif olarak alan birey yerine öğrenme sürecine aktif katılan, bilgiyi seçip örgütleyen, karşılaştığı durumlara eleştirel düşünme becerisi ile yaklaşan eleştirel düşünme eğiliminde olan bireyler yetiştirilmelidir. Eleştirel düşünme; öğrencilerin bilişsel gelişimlerine katkı sağlamaktadır. Bir diğer bilgisayarca düşünme alt becerisi problem çözmedir. Bu beceri bir zorluğun üstesinden gelebilmek için o zorluğu tanımlama, zorlukla ilgili bilgileri örgütleme, gereksinim duyulacak verileri belirleme, çözümler belirleme ve en doğru çözümü bulma becerisi olarak tanımlanabilir (Seferoğlu & Akbıyık, 2006). Bilgisayarca düşünme becerisi ile ilişkilendirilebilecek işbirlikli öğrenmenin problem çözme becerisine önemli katkısı olduğu vurgulanmıştır (Genç, 2007). Üst düzey düşünme becerileri ile ilişkili olan bilgisayarca düşünmenin; okuma, yazma, dört işlem yapabilme gibi temel bir beceri olacağı söylenilmektedir (Wing, 2006).

Sonuç olarak algoritma becerilerinin geliştirilmesi ile öğrencilerin bilgisayarca düşünme ve problem çözme becerilerinin de geliştirilmesi sağlanabilir. Öte yandan öğrencilerin bu becerilerini yeterince geliştirebilmelerinin de kodlamaya dönük olumlu tutuma sahip olmalarından geçtiğini söylemek mümkündür. Bu çerçevede bu çalışmanın amacı, proje tabanlı Flowchart destekli algoritma eğitiminin bilgisayarca düşünme, problem çözme ve kodlamaya dönük tutum üzerine etkisi ve öğrencilerin bu sürece dönük görüşlerini ortaya koymak olarak belirlenmiştir.

Alt Problemler

- 1- Deneysel uygulama öncesinde öğrenciler kodlamaya dönük tutumları, problem çözme becerileri ve bilgisayarca düşünme becerileri açısından denk midir?

- 2- Flowchart destekli proje tabanlı algoritma eğitiminin öğrencilerin kodlamaya dönük tutumlarına anlamlı düzeyde katkısı var mıdır?
- 3- Flowchart destekli proje tabanlı algoritma eğitiminin öğrencilerin problem çözme becerilerine anlamlı düzeyde katkısı var mıdır?
- 4- Flowchart destekli proje tabanlı algoritma eğitiminin öğrencilerin düşünme becerilerine anlamlı düzeyde katkısı var mıdır?
- 5- Öğrencilerin flowchart destekli proje tabanlı algoritma eğitimine dönük düşünceleri nasıldır?

YÖNTEM

Araştırma Modeli

Bu araştırmanın temel aldığı felsefe karma desendir. Karma araştırma deseni; çalışmada nitel ve nicel araştırma özelliklerinin harmanlanarak veya birleştirilerek kullanılmasıdır (Fırat, Yurdakul & Ersoy, 2014). Çalışmada nicel yaklaşımların daha ağırlıklı olduğu sıralı açıklayıcı desen kullanılmıştır. Bu desen beklenilmeyen bulguları veya ilişkileri açıklamakta kullanılır (Baki & Gökçek, 2012). Bu çalışmada üç ölçekle ağırlıklı olarak nicel veriler toplanmıştır. Elde edilen nicel sonuçlara göre farklılık gösteren öğrencilerle görüşme formu kullanılarak nitel veri toplanmıştır. Çalışmanın nicel boyutunda öntest-sontest kontrol gruplu yarı deneysel desen kullanılmıştır. Yarı deneysel desenler çok denekli desenlerden biridir (Büyüköztürk, Çakmak, Akgün, Karadeniz & Demirel, 2018). Bu çalışmada bir bağımsız değişkenin üç bağımlı değişkene olan etkisi incelendiğinden tek faktörlü desen kullanılmıştır. Tek faktörlü desen; yalnızca bir bağımsız değişkenin bağımlı değişkene olan etkisinin incelendiği desendir. Deneme koşullarına göre ise gruplar arası desen kullanılmıştır.

Çalışma Grubu

Bu araştırmanın çalışma grubunu kolayda örnekleme yöntemi ile belirlenen 2019 – 2020 eğitim öğretim yılı I. döneminde Sivas il merkezinde bir ortaokulda 6 ve 7. sınıfta öğrenim görmekte olan 30 öğrenci oluşturmaktadır. Nitel verileri için çalışma grubunu ise gönüllülük esasına göre deney grubundan belirlenen 10 öğrenci oluşturmaktadır. Amaçlı örnekleme, çalışmanın amacı doğrultusunda zengin veri elde edilebilecek durum ve kişilerle yapılır (Büyüköztürk, 2018, 92-93). Bir sınıf düzeyi ile yapılacak çalışmaya yeterli sayıda öğrenci temin edilemediğinden, bu çalışma için 6 ve 7. sınıf öğrencileri birlikte dahil edilmiştir. Çalışma grubunun yarısı 6, diğer yarısı ise 7. sınıf öğrencilerinden oluşmaktadır. Öğrencilerin sınıf ve cinsiyete göre dağılımları Tablo 1’de gösterilmiştir.

Tablo 1. Çalışma grubunun cinsiyet ve sınıf değişkenlerine göre dağılımı

Sınıf			Cinsiyet		Toplam
			Erkek	Kız	
6.sınıf	Grup	Deney	3	5	8
		Kontrol	3	4	7
	Toplam		6	9	15
7.sınıf	Grup	Deney	3	4	7
		Kontrol	4	4	8
	Toplam		7	8	15

Veri Toplama Araçları

Çalışmanın nicel boyutunda Bilgisayarca Düşünme Beceri Düzeyleri Ölçeğinin (BDBD) ortaokul düzeyine uyarlanması, Ortaokul Öğrencileri için Kodlamaya Yönelik Tutum Ölçeği ve Ortaokul Öğrencileri İçin Problem Çözme Becerilerine Yönelik Algı Ölçeği kullanılmıştır.

Nitel boyutunda ise öğrencilerin flowchart destekli proje tabanlı algoritma eğitimine dönük düşünceleri betimlemek için görüşme formu hazırlanmış ve bu forma göre görüşmeler yapılmıştır. Hazırlanan görüşme formu eğitim teknolojisi alanından doktoralı iki uzmana incelenilerek, uzman görüşleri doğrultusunda gerekli düzeltmeler yapılmıştır. Böylece görüşme formuna son şekli verilmiştir. Görüşme formu yarı yapılandırılmış sorulardan oluşmaktadır. Katılımcılara açık uçlu sorular sorulmuştur.

Bilgisayarca düşünme beceri düzeyleri ölçeği

Öğrencilerin bilgisayarca düşünme beceri düzeylerini ölçmek amacıyla Korkmaz, Çakır ve Özden (2015) tarafından ortaokul düzeyine uyarlanmış Bilgisayarca Düşünme Beceri Düzeyleri Ölçeği kullanılmıştır. Ölçek beş dereceli Likert tipindedir ve 5 faktör altında toplanan 22 maddeden oluşmaktadır. Ölçeğin maksimum likelihood tekniği kullanılarak doğrulayıcı faktör analizi ve açımlayıcı faktör analizi yapılarak son hali verilmiştir. Korkmaz, Çakır ve Özden (2015) tarafından yapılan analizler çerçevesinde, ölçeğin Cronbach Alpha güvenilirlik katsayıları; dört maddeden oluşan yaratıcılık faktörü için ,640 dört maddeden oluşan algoritmik düşünme faktörü için ,762 dört maddeden oluşan işbirliklilik faktörü için ,811 dört maddeden oluşan eleştirel düşünme faktörü için ,714 altı maddeden oluşan problem çözme faktörü için ,867 ölçeğin geneli için ,809 olarak tespit edilmiştir. Ölçeğin madde ayırt ediciliği alt %27 ve üst %27'lik gruplar oluşturulup t-testi değerleri hesaplanmıştır. Çıkan sonuçlar 3,818 ile 23,287 arasındadır. Ölçeğin genelinin t değeri -37,105'dir. Bu değerlere göre ölçeğin her bir maddesinin ve genelinin ayırt ediciliğinin yüksek olduğu söylenebilir (Korkmaz, Çakır & Özden, 2015).

Ortaokul öğrencileri için kodlamaya yönelik tutum ölçeği

Öğrencilerin Kodlamaya Yönelik Tutumlarını ölçmek amacıyla Akkuş, Özhan ve Kan (2019) tarafından geliştirilmiş Ortaokul Öğrencileri için Kodlamaya Yönelik Tutum Ölçeği kullanılmıştır. Ölçek oluşturma sürecinde ilk başta alan yazın taraması yapılmıştır. Daha sonra 10 öğrenci, 2 alanında uzman ve 2 öğretmenle görüşmeler sonunda ölçeğin madde havuzu oluşturulmuştur. Ölçeğin kapsam geçerliliğini ve görünüş geçerliliğini sağlamak amacıyla dört öğretmen, iki öğretim üyesi, bir dil uzmanı ölçeği ve maddelerini değerlendirmiş ve beşli likert tipinde, tek faktörlü 20 maddeden oluşan ölçek formu haline dönüşmüştür. Akkuş, Özhan ve Kan (2019) tarafından bu ölçek formu Malatya il merkezinde iki farklı ortaokuldaki 292 6 ve 7. sınıf öğrencisine uygulanmış ve uygulamanın sonucunda ortaya çıkan veriler ışığında açımlayıcı ve doğrulayıcı faktör analizi yapılarak ölçeğin yapı geçerliliği sağlanmaya çalışılmıştır. Bu analizlerin sonucunda ölçek 10 maddeye indirilerek son halini almıştır. Kalan 10 maddenin varyans'ın %55.352'ini açıkladığı sonucuna varılmıştır. Ölçeğin iç tutarlılığı ve güvenilirlik düzeyi Cronbach Alfa güvenilirlik formülü kullanılmıştır. Ölçeğin bütünü için Cronbach Alpha güvenilirlik katsayısı .905 olarak hesaplanmıştır (Akkuş, Özhan & Kan, 2019).

Ortaokul öğrencileri için problem çözme becerilerine yönelik algı ölçeği

Öğrencilerin problem çözme becerilerine yönelik algılarını ölçmek amacıyla Ekici ve Balım (2013) tarafından geliştirilmiş Ortaokul Öğrencileri İçin Problem Çözme Becerilerine Yönelik Algı Ölçeği kullanılmıştır. Çalışmanın örneklemini İzmir ili Buca ilçesinde bulunan 9 tane ilköğretim okuludur. Ölçeğin kapsam geçerliliğini sağlamak için uzman görüşüne başvurulduğu, yapı geçerliliği için açımlayıcı ve doğrulayıcı faktör analizi yapıldığı ifade edilmiştir. Ölçek formu 894 öğrenciye uygulanmış ancak; güvenilirlik gerekçesiyle 850 öğrencinin verileri üzerinde çalışıldığı belirtilmiştir. Ölçeğin güvenilirliğini belirlemek amacıyla maddelerin toplam korelasyonları ve Cronbach Alfa değerleri hesaplanmıştır. Ölçeğin birinci faktörünün Cronbach Alpha güvenilirlik katsayısı .884, ikinci faktörünün Cronbach Alpha

güvenirlilik katsayısı .777, ölçeğin tamamı için ise Cronbach Alfa değeri .88 bulunmuştur. Ölçeğin iki faktörden oluştuğu tespit edilmiştir. Ölçeğin birinci faktörünün açıkladığı varyansın %30,239, ikinci faktörünün açıkladığı varyansın ise %9,976 olduğu söylenmiştir. Ölçeğin madde toplam korelasyonlarının .436 ile .633 arasında değiştiği belirtilmiştir. Madde ayırt edicilikleri .001 düzeyinde anlamlı olduğu ifade edilmiştir. Yapılan analizlere göre bu ölçeğin geçerli ve güvenilir olduğu saptanmıştır. 15 olumlu, 7 olumsuz olmak üzere 22 maddeden oluşmaktadır (Ekici & Balım, 2013).

Çalışmanın nitel boyutu için görüşme:

Görüşmeler; grubun genel eğiliminden farklı cevap veren öğrencilerle olmuştur. Gruptan neden farklılaştıklarını öğrenmek istenilmiştir.

Deneysel Süreç

Kontrol grubunda; mevcut öğretim programında yer aldığı şekliyle anlatım, soru cevap, tartışma yöntemleri gibi yöntemler kullanılarak kalem kâğıt ortamında algoritma eğitimi verilmiştir. Deney grubunda ise proje tabanlı öğretim modeli uyarınca ve flowchart ortamında algoritma eğitimi verilmiştir. Her iki grupta da aynı eğitim içeriği sunulmuştur. Etkinlikler yapılırken her iki grupta da ikişer kişilik çalışma grupları oluşturulmuştur. Deneysel uygulama öntest ve sontest uygulamaları dışında 4 hafta sürmüştür. Deney ve kontrol gruplarında haftalık olarak gerçekleştirilen etkinlikler şöyledir:

1. Hafta

Deney grubu: İlk hafta sırasıyla öğrencilerle birlikte proje soruları, yapılacak projenin amacı, projenin sonunda elde edilecek kazanımlar, yapılacak işlemler ve ikişer kişiden oluşan gruplar belirlenmiştir. Bu kapsamda aşağıdaki proje konuları tartışılmıştır:

- 1- Tabanı ve kuvveti verilen üslü sayıyı hesaplayan algoritmanın geliştirilmesi.
- 2- Girilen sayının faktöriyelini hesaplayan algoritmanın geliştirilmesi.
- 3- Miladi takvime göre doğum yılı girilen kişinin, Hicri ve Rumi takvime göre doğum yılını bulan algoritmanın geliştirilmesi.
- 4- Türkçesi girilen rengin İngilizcesini yazan algoritmanın geliştirilmesi
- 5- Yaş, boy, kilo, cinsiyet bilgisini alıp vücut kitle indeksini hesaplayan algoritmanın geliştirilmesi,
- 6- Bit cinsinden girilen veri depolama miktarının; bayt, kilobayt, megabayt, gigabayt cinsinden değerini yazan algoritmanın geliştirilmesi
- 7- Girilen sayının mutlak değerini hesaplayan algoritmanın geliştirilmesi
- 8- Girilen üç kelimeyi sözlük sırasına göre sıralayan algoritmanın geliştirilmesi

Öğrencilerin proje konuları disiplinler arası yaklaşıma uygun olarak diğer disiplinlerle ilişkilendirilerek belirlenmiştir. 1. proje matematik dersindeki üslü sayılar konusuyla ilişkilendirilmiştir. Kullanıcıdan taban ve üs bilgisini alıp sonucu hesaplayan akış diyagramını tasarlamaları istenmiştir. 2. proje matematik dersi faktöriyel konusu ile ilişkilendirilmiştir. Bu projede girilen sayının akış diyagramları ile hesaplanması istenmiştir. 3. proje sosyal bilgiler dersi ile ilişkilendirilmiştir Miladi takvime göre girilen bir yılın Hicri ve Rumi takvimdeki karşılıklarını akış diyagramı ile hesaplayan bir program geliştirmeleri istenmiştir. 4. proje İngilizce dersi ile ilişkilidir. Bu proje İngilizce Türkçe sözlük niteliğindedir. Kullanıcıdan rengin Türkçe karşılığını alıp daha sonra İngilizce karşılığını ekrana yazan program geliştirmeleri istenmiştir. Bu projede kullanıcı programda çıkmadığı sürece programdan çıkmayıp programın sürekli çalışması istenmiştir. 5. proje fen bilgisi dersi ile ilişkilendirilmiştir. Bu projede kullanıcıdan boy ve kilo bilgisini alıp vücut kitle indeksi

hesaplama formülüne göre vücut kitle indeksini hesaplayan akış diyagramını tasarımları istenmiştir. 6. proje bilişim teknolojileri veri depolama birimleri konusuyla ilişkilidir. Kullanıcıdan bit cinsinden girilen veri miktarının daha büyük veri birimlerindeki karşılığı ekrana yazan akış diyagramı programı istenmiştir. 7. proje matematik dersi mutlak değer konusuyla ilişkilidir. Kullanıcıyı girdiği sayının mutlak değerini ekrana yazan akış diyagramını tasarımları istenmiştir. 8. proje Türkçe dersi ile ilişkilidir. Bu projede akış diyagramları ile rastgele girilen 3 kelimenin sözcük sırasına göre dizilimini gösteren program tasarımları istenmiştir. Projeler FlowChart 3.0 görsel programlama aracılığı ile gerçekleştirilmiştir. FlowChart programı Sunu programıyla menüleri, akış şeması şekilleri, akış şemalarının kullanımları tanıtılmıştır. Algoritmanın akış diyagramına nasıl dönüştürüleceği gösterilmiştir.

Kontrol grubu Algoritmanın ne olduğuna ve önemine ilişkin bilgiler verilmiştir. Günlük yaşamda karşılaştıkları bir problem durumunu algoritma şeklinde ifade etmeleri istenmiştir. Ayran yapmanın algoritması gösterilmiş, araba sürmenin algoritmasını yapmaları söylenmiştir. İki sayının toplamını yapan algoritma gösterilmiş, iki sayının çarpımının ve ortalamasını yapan algoritma öğrenciler tarafından yapılmıştır. Doğum yılı alınan bir kişinin yaşını bulan algoritmayı nasıl olacağı düşündürülmüş eksiklikleri giderilmiştir. Öğrencinin iki sınav notunun ortalamasını alıp eğer 50'den küçükse başarısız değilse başarılı şeklinde yazan algoritma ödev verilmiştir.

2. hafta

Deney grubu: Sonuç raporunun özellikleri, sunuş biçimi, değerlendirme ölçütleri, alt sorular, kontrol noktaları, çalışma takvimi belirlenmiş. Bilgi toplama süreci başlatılmıştır. İnternette, kitaplardan, bilgiler edinilmeye çalışılmıştır.

Kontrol grubu: Hazır algoritmalar sınıfa getirilmiştir. Algoritmaları test edip verilen değerlere göre nasıl sonuçlar vereceğinin çıktıları istenmiştir. Hatalı algoritmalar gösterilmiş buluş yolu stratejisiyle hataları bulup düzeltmeleri sağlanmıştır.

3. hafta

Deney grubu: Proje soruları ile ilgili bilgileri öğrenciler ders kitaplarından internette araştırmışlardır. Konu ile ilgili branş öğretmenlerinden bilgiler almışlardır. Bilgileri organize ettikten sonra proje sorularının algoritmaları yapılmıştır. Bilgisayar laboratuvarında öğrenciler FlowChart programı ile projelerini çalışma takvimine göre yürütmeye başlamışlardır.

Kontrol grubu: Ekrana 0 ile yüz arasındaki 10'un katlarını yazan algoritma döngü yöntemiyle yani akışı daha önceki adıma gönderilmesi yöntemiyle anlatılmıştır. Hatalı algoritma yapan öğrencilerin hataları hep birlikte sınıfta düzeltilmiş döngü mantığı öğretilmiştir. Sıfır ile yüz arasındaki çift sayıları yazan algoritma öğrencilere ödev olarak verilmiştir.

4. hafta

Deney grubu: Öğrenciler projelerini öğretmen rehberliğinde tamamlamaya çalışmışlardır. Son kontroller yapılmıştır. Proje raporlanmıştır ve arkadaşlarına projelerini sunmuşlardır.

Kontrol grubu: Değişken kavramı anlatılmıştır. Algoritma tasarlarken değişkenler kullanılmıştır. Sınıf beş gruba ayrılmış her bir gruba bir soru verilmiştir. Gruplar yaptıkları algoritmaları arkadaşları ile paylaşmışlardır. Arkadaşları tarafından eleştirilmişlerdir ve arkadaşları önerilerini söylemişlerdir. Deney grubunda uygulanan proje tabanlı etkinliklere ilişkin hazırlanan örnek plan Tablo2'de sunulmuştur.

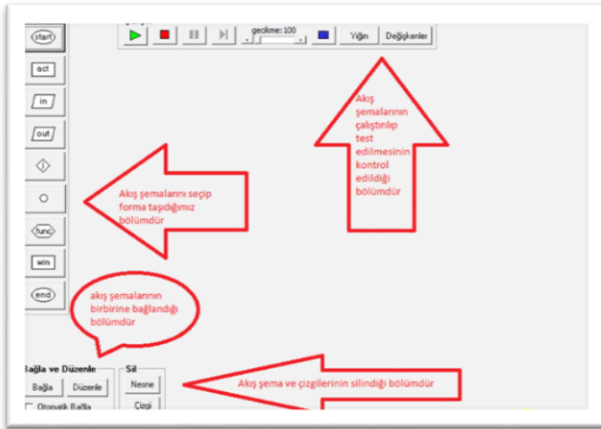
Tablo 2. Proje tabanlı öğrenme uygulaması örnek

Proje sorusu:	Bugün kullandığımız Miladi takvimdeki bir tarihi, diğer takvim sistemlerindeki karşılığını nasıl buluruz
Projenin Amacı:	Miladi takvimdeki bir yılı hicri ve rumi takvime algoritma ve akış diyagramı kullanarak çevirme
Hedeflerin belirlenmesi	1- Farklı takvim sistemlerini öğrenir 2- Tarihleri takvimlere göre dönüştürür 3- Karşılaştığı problemleri algoritma ile çözümler 4- Karşılaştığı problemi akış diyagramı ile ifade eder.
Yapılacak işin belirlenmesi	Miladi, Hicri, Rumi takvimin özellikleri öğrenilip flowchart programı ile miladi takvime göre girilen bir yıl hicri ve rumi takvimde karşılık gelen yıla çevrilecektir.
Sonuç raporunun ve Sunuş Biçiminin belirlenmesi	Grup flowchart programında yaptığı projeyi tanıtacak, raporlarını görsel materyallerle sunacaktır.
Değerlendirme ölçütlerinin belirlenmesi	Temel Soruları Belirleme ve iş bölümü %5 Araştırma, Bilgi toplama %30 Bilgiyi örgütleme ve Projeyi sonuçlandırma %25 Rapor Yazma %10 Sunu %20 İşbirlikli çalışabilme %10
Grupların oluşturulması	Gruplar ikişer kişiden oluşacaktır.
Alt soruların belirlenmesi bilgi toplama sürecinin başlaması	1- İnsanlar neden farklı takvimler kullanmışlardır 2- Takvimleri dönüştürme formülleri nelerdir ve Algoritmayla nasıl yapılandırırız 3- Tasarladığımız algoritmayı flowchart'a nasıl aktarırız
Çalışma takviminin oluşturulması	Alt soruların, Bilgi kaynaklarının belirlenmesi 2 gün Bilgi toplama sürecinde iş bölümünün yapılması 1 gün Kütüphane internet taramasının yapılması 2 gün Öğretmenlerden ve ilgili kişilerden bilgi alma 1 gün Toplanan bilgilerin denetlenmesi alt sorulara cevap olacak şekilde düzenlenmesi 2 gün Flowchart programının özelliklerinin öğrenilmesi 1 gün İşbirlikli çalışma sürecinin başlatılması 7 gün Rapor oluşturma 2 gün Projenin sunulması 1 gün
Kontrol noktalarının belirlenmesi	1-) Toplanan bilgilerin güvenilirliğin denetlenmesi ve alt sorulara cevap olacak şekilde düzenlenmesi 2- Flowchart programında proje yapım aşamalarının kontrol edilmesi
Bilgilerin toplanması	Okul Kütüphanesi, Ders kitapları, Öğretmenlerden bilgi edinme, Eba, İnternet siteleri
Bilgilerin organize edilip raporlanması	Çalışma takvimine uyulup uyulmadığına bakarak, raporlaştırmanın yapılması
Projenin Sunulması	Projenin görsel materyallerle sunulması

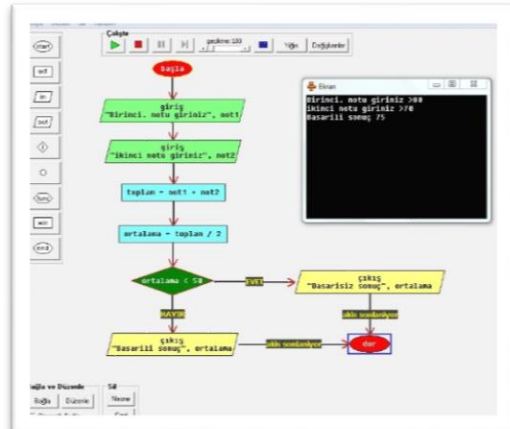
Öğretim aracı

Bu çalışmada öğretim aracı olarak deney grubunda Flowchart Visual Programming Language 3.01 (akış şeması görsel programlama dili) kullanılmıştır. Bu yazılım aracılığıyla akış diyagramları ile görsel programlama yapılabilmektedir. Bu programlama ortamında komutlar akış şemaları ile oluşturulmaktadır. Akış şemaları anında test edilebilmektedir. Hatalar kolaylıkla tespit edilebilmektedir. Kelime işlemci programlarında oluşturulan akış şemalarını bilgisayar aracılığı ile test etme seçeneğimiz yoktur. Programda akış şeması şekilleri genellikle evrensel olarak kullanılan akış şemaları şekilleri ile aynıdır. Farklı olarak giriş ve çıkış akış şemaları için paralel kenar kullanılmaktadır. Ancak çıkış birimi akış şemalarında altı dalgalandırılmış dikdörtgenle ifade edilmektedir. Altıgen işareti fonksiyon oluşturma anlamındayken akış şemaları genel kullanımında altıgen döngüyü ifade etmektedir. Şekil 1'de yazılıma ilişkin çalışma ekranı görüntülenmektedir.

Şekil 1. Flowchart'ın Çalışma Ekranı



Şekil 2. Flowchart'da örnek bir uygulama



Şekil 1’de Flowchart ara yüzü tanıtılmıştır. Sol tarafta akış şemaları, üst kısımda programı kontrol edilen bölümü mevcuttur. Gecikme: 100 skalasını ileri geri yaparak program akış hızını ayarlanabilmektedir. Hız yavaş olduğunda diyagramdaki akışı daha net görebiliriz. Değişkenler butonunda değişkenler ve değerlerini görülebilir. Sol alt kısımda akış şemalarını birbirine bağlama ve silme bölümleri mevcuttur. Şekil 2’de FlowChart programında kullanıcıdan iki sayı alan, aldığı sayıların ortalamasını hesaplayan eğer sayı elliden küçükse başarısız değilse başarılı yazan akış şeması ve derlenmiş hali olan siyah ekran mevcuttur.

Verilerin Analizi:

Deney ve kontrol grupları öğrenci sayıları 20’den az olduğu için elde edilen verilere parametrik olmayan testler kullanılarak analiz edilmiştir. Deney ve kontrol grubu arasında anlamlı bir farklılık olup olmadığını tespit etmek için Mann Whitney U testi kullanılmıştır (Büyüköztürk, 2019, 39-40). Bu çalışmada görüşme formu ile elde edilen nitel verilerin analizi için NVIVO 10.0 programından yararlanılmıştır. Görüşmeler deney grubundaki 10 öğrenci ile yapılmıştır. Bu 10 öğrenciden daha çok veri toplanacağı düşünülmüştür. Görüşmeler gönüllük esasına göre yürütülmüştür.

BULGULAR

Grupların Bilgisayarca Düşünme Beceri Düzeyleri, Kodlamaya Yönelik Tutumları, Problem Çözme Becerilerine Yönelik Algılarının denkliliğine ilişkin bulgular Tablo 3’te özetlenmiştir. Deney ve kontrol grubu öğrencilerinin bilgisayarca düşünme, kodlamaya dönük tutum, problem çözme becerilerine yönelik algı ön test puanları arasında anlamlı bir farklılık olup olmadığını belirlemek için Mann Whitney-U testi uygulanmıştır. Tablo 3’e göre yaratıcılık, işbirlikçilik, eleştirel düşünme, problem çözme, kodlamaya dönük tutum (KYT), problem çözme becerilerine yönelik algı alt faktörü, kararlılık ve isteklilik algısı ve problem çözme becerilerine yönelik algıları (PÇBYA) açısından kontrol ve deney grupları arasında anlamlı bir farklılık bulunamamıştır. Bilgisayarca düşünme becerileri düzeyi ve algoritmik düşünme alt faktörü açısından deney grubu lehine anlamlı bir farklılık bulunmuştur. Bu farklılaşmaların kontrol altına alınabilmesi amacıyla deneysel uygulamanın etkililiğine dönük yapılan analizlerde sontest-öntest fark puanları kullanılmıştır.

Tablo 3. Grupların Denkliğine İlişkin Bulgular

	Ölçekler ve alt boyutlar	Grup	N	S.T.	S.O.	U	z	p
<i>Bilgisayarca Düşünme</i>	Yaratıcılık	Deney	15	17.6	265			
		Kontrol	15	13.3	200	80	200	,175
	Algoritmik Düşünce	Deney	15	18.86	283			
		Kontrol	15	12.13	182	62	182	,035
	İşbirliklilik	Deney	15	17.03	255.5			
		Kontrol	15	13.96	209.5	89,5	209,5	,336
	Eleştirel Düşünme	Deney	15	16.53	248			
		Kontrol	15	14.46	217	97	217	,518
Problem Çözme	Deney	15	18.26	274				
	Kontrol	15	12.73	191	71	191	,084	
<i>Toplam Puan</i>	Deney	15	19.3	289.5				
	Kontrol	15	11.7	175.5	55	175,5	,018	
<i>Kodlama Dönük Tutum</i>	Deney	15	15.76	236.5				
	Kontrol	15	15.23	228.5	108,5	228,5	,868	
<i>Problem Çözme Algı ölçeği</i>	Problem çözme Algı faktörü	Deney	15	16.56	248.5			
		Kontrol	15	14.43	216.5	96,5	216,5	,506
	Kararlılık ve İsteklilik	Deney	15	16.03	240.5			
		Kontrol	15	14.96	224.5	104,5	224,5	,739
	<i>Toplam Puan</i>	Deney	15	16.3	244.9			
Kontrol	15	14.6	220	100	220	,604		

Deneyisel uygulamanın BDBD ve alt boyutları, KDT, PÇBYA ve alt boyutlarına ilişkin farklılaşmaya dönük sontest-öntest fark puanlarına ilişkin yapılan analizler Tablo 4’de özetlenmiştir.

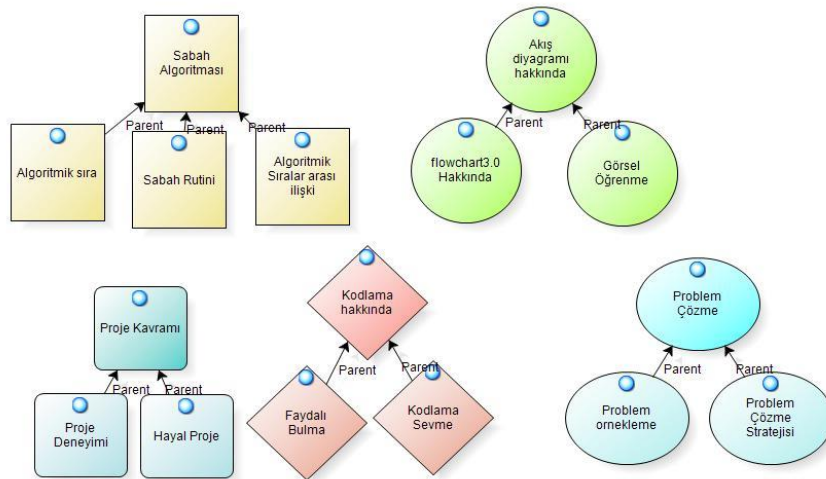
Tablo 4. Deneyisel uygulamanın etkililiğine ilişkin bulgular

	Ölçekler ve alt boyutlar	Grup	N	S.T.	S.O.	U	z	p
<i>Bilgisayarca Düşünme</i>	Yaratıcılık	Deney	15	14.77	221.5	101.5	-.461	.645
		Kontrol	15	16.23	243.5			
	Algoritmik Düşünce	Deney	15	11.97	179.5	59.5	-2.222	.026
		Kontrol	15	19.03	285.5			
	İşbirliklilik	Deney	15	17.93	269	76	-1.534	.125
		Kontrol	15	13.07	196			
	Eleştirel Düşünme	Deney	15	15.33	230	110	-.104	.917
		Kontrol	15	15.67	235			
Problem Çözme	Deney	15	12.77	191.5	71.5	-1.704	.088	
	Kontrol	15	18.23	273.5				
<i>Toplam Puan</i>	Deney	15	12.67	190	70	-1.766	.077	
	Kontrol	15	18.23	275				
<i>Kodlama dönük tutum</i>	Deney	15	15.27	229	109	-.145	.884	
	Kontrol	15	15.73	236				
<i>Problem çözme Algı ölçeği</i>	Problem çözme Algı faktörü	Deney	15	16	240	105	.302	.755
		Kontrol	15	15	225			
	Kararlılık ve İsteklilik	Deney	15	14.63	219.5	99.5	-.540	.589
		Kontrol	15	16.37	245.5			
	<i>Toplam Puan</i>	Deney	15	15.47	232	112	-.021	.983
Kontrol	15	15.53	233					

Uygulama sonrasında gruplar arasında Bilgisayarca Düşünme Beceri Düzeyleri (BDBD) yaratıcılık, işbirliklilik, eleştirel düşünme, problem çözme alt boyutları ve Kodlamaya Dönük

Tutum (KDT), Problem Çözme Becerilerine Yönelik Algı (PÇBYA) ve alt boyutlarına ilişkin fark puanları arasında anlamlı bir farklılık olup olmadığını anlamak için puan farklarına Non-Parametrik Mann Whitney-U testi uygulanmıştır. Tablo 7’de verilen ölçekler ve alt boyutlarının yaratıcılık ($z=-.461$, $p>05$), işbirlikçilik ($z=-1.534$, $p>05$), eleştirel düşünme ($z=-.104$, $p>05$), problem çözme ($z=-1.704$, $p>05$), bilgisayarca düşünme becerileri düzeyi(BDBD) ($z=-1.766$, $p>05$) kodlamaya dönük tutum ($z=-.145$, $p>05$), problem çözme becerisine yönelik algı faktörü ($z=.302$, $p>05$), kararlılık ve isteklilik algı ($z=-.540$, $p>05$), problem çözme becerisine yönelik algıları ($z=-.021$, $p>05$) açısından gruplar arasında anlamlı bir farklılaşma olmadığı görülmektedir. Buna karşın algoritmik düşünce alt boyutu ($z=-2.222$, $p<05$) açısından ise kontrol grubu lehine anlamlı bir farklılaşma olduğu görülmektedir. Buna göre deneysel uygulamanın öğrencilerin tüm becerilerine kontrol grubundaki uygulamaya göre anlamlı düzeyde daha fazla katkı sağlamadığı, dahası öğrencilerin algoritmik düşünme becerilerine kontrol grubundaki uygulamadan anlamlı düzeyde daha az katkı sağladığı söylenebilir. Öğrencilerin deneysel uygulamaya dönük düşüncelerine ilişkin yapılan içerik analizi sonuçları aşağıda özetlenmiştir.

Şekil 3. Dersin kavramlarına dönük tema haritası



Şekil 3’te öğrencilerin dersin temel kavramlarına dönük görüşleri çerçevesinde elde edilen temalar incelendiğinde algoritma, akış diyagramı, proje, kodlama, problem çözme olmak üzere beş tane ana temanın ortaya çıktığı görülmektedir. Algoritma temasının; algoritmik sıra, sabah rutini ve algoritmik sıralar arası ilişki olmak üzere üç alt kategoriye ayrıldığı görülmektedir. Algoritma teması kapsamında “Sabah uyanıp okula gelene kadar olan süreçteki yaptıklarınızı bahseder misiniz?” sorusuna öğrencilerin genel olarak mantıklı ve uygun sıra ile cevap verdikleri görülmüştür.

“Uyanıyorum. Üzerimi giyiyorum. Elimi yüzümü yıkıyorum. Yemek yiyorum, ders programımı ayarlıyorum. Saçımı yapıyorum. Hırkamı giyiyorum. Ayakkabılarımı giyiyorum. Servise gidiyorum” (Ö4).

“İlk önce yüzümü yıkayıp, saçlarımı örüp, üstümü giyip, kahvaltımı yapıyorum dışarı kıyafetlerimi giyip dışarı çıkıyorum yolda biriyle karşılaşırsam arkadaşımınla konuşuyorum okula geliyorum” (Ö5).

Algoritma teması, algoritmik sıralar alt kategorisi kapsamındaki “Bu yaptıklarınızın belli bir sırası var mıdır?” sorusuna görüşme yapılan on öğrencinin altısı Sabah yaptıklarının belirli bir

sirasının olduğu ifade etmiştir. Algoritmik Sıralar Arası İlişki alt kategorisi ilişkili olan “Yaptıklarınızın sırasını değiştirmek mümkün müdür?” sorusuna on öğrencinin yedisi sabah yaptıklarının sırasının değişebileceğini söylemiştir. Öğrencilerin görüşleri incelendiğinde aslında sıralamanın değişebileceği veya değişmeyeceği görüşünde olan öğrencilerin fikirlerinin aynı doğrultuda olduğu söylenebilir. Öğrenciler temel adımların değişmeyeceği ama öncelik sonralık ilişkisi olmayan adımların değişebileceğini söylemişlerdir. Sürecin sonucunu etkilemeyecek, olmazsa olmaz olmayan adımların değişken olabileceğini söylemişlerdir. Bu görüşlerinin algoritmik düşünceye aykırı olmadığı sonucuna varılabilir. Bazı öğrencilerin görüşleri aşağıdaki gibidir.

“Hayır mümkün değildir. Bazen geç kaldığımda değişebiliyor.” (Ö2).

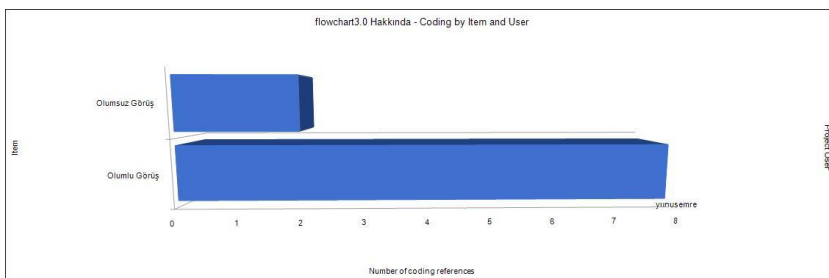
“Evet birkaç tanesi değişebilir Ders programını önce ayarlayabilirim” (Ö4)

Algoritma teması Sabah rutini alt kategorisine yönelik “Bu süreçte yaptıklarınız hemen hemen her gün aynı mıdır?” sorusuna on öğrencinin onu da sabah yaptıklarının aynı olduğunu ifade etmiştir. Alınan cevaplara göre öğrencilerin algoritma ile ifade edilecek durumlar ile algoritma ile ifade edilemeyecek durumları ayırt edebildikleri görülmüştür. Öğrencilerin verdikleri cevaplardan algoritma ile ifade edilen süreçlerin çok fazla değişken olmadığını düşündükleri söylenebilir. Bir öğrencinin; bu süreçte yaptıklarınız hemen hemen her gün aynı mıdır? Sorusuna verdiği cevap şöyledir;

“Aynıdır ama bazen geç kaldığımda aynı olmuyor. Bazen kahvaltı yapamıyorum geç kaldığım için” (Görüşme Kaydı: Ö2).

Akış diyagramı temasının iki alt kategorisi vardır. Bunlar Flowchart programı hakkındaki görüşler ve görsel öğrenmedir. Görsel öğrenme alt kategorisi ile alakalı “Algoritmayı Akış Diyagramı ile ifade etme konusunda görüşleriniz nelerdir?” sorusuna Öğrenciler; algoritmayı akış diyagramı ile ifade ederken hangi algoritma adımını hangi akış şeması şekli ile ifade edeceklerini karıştırdıklarını ve bu yüzden biraz zorlandıkları belirtmişlerdir. Ancak; öğrenciler genel olarak algoritmanın akış diyagramı ile gösterilmesinin kolay olduğu görüşündedir. Flowchart Visual Programing 3.0 programı hakkında görüşler alt kategorisi ile ilgili “Flowchart Programı hakkında ne düşünüyorsunuz?” sorusu için on öğrencinin sekizinin görüşü olumludur. Aşağıdaki grafikte bu durum gösterilmiştir.

Şekil 4. Flowchart 3.0 görsel programlama aracına yönelik tutum grafiği



Bazı öğrencilerin algoritmayı akış diyagramı ile göstermek ile ilgili görüşleri aşağıdaki gibidir.

“Bazen zorlanıyorum. Yani dikdörtgen filan var ya algoritma ile ifade etmek daha iyi.” (Ö2)

“Bence yapması birazcık daha zor oluyor ama fiziksel olarak yaptığımız için daha işlevsel oluyor yani ne yaptığımızı görebiliyoruz.” (Ö6)

Alınan cevaplara göre FlowChart 3.0 programının algoritma geliştirmeye olumlu katkıda bulunduğu söylenebilir. FlowChart Programında girdi ve çıktı işlevleri için kullanılan şekil

genel kullanımın aksine her ikisinin de paralelkenarla ifade edilmesi öğrencilerde kafa karışıklığına yol açtığı anlaşılmıştır. Oysaki akış diyagramı sembollerinden çıkış işlevi altı dalgalı dikdörtgenle ifade edilmektedir. FlowChart Visual Programing 3.0 programı; tasarlanan akış şemasının ilerleyişini ve çıktısını göstermesi öğrenciler için dönüt niteliğinde olduğu söylenebilir. FlowChart Visual Programing 3.0 programı hakkındaki öğrenci görüşlerini örnekleyecek olursak;

“Bence gayet güzel bir program. Çocukların kullanabileceği bir şey ve yapılan işlerde gayet güzel olabiliyor.” (Ö6).

“Çok iyi program herkesin denemesini tavsiye ederim.” (Ö8).

Akış diyagramı hakkındaki görüşler teması Görselle öğrenme alt faktörüne ait “Görsel (Tablo, şekil, grafik) anlatım öğrenmeye etkisi var mıdır?” sorusuna on öğrencinin dokuzu olumlu etkisi olduğunu söylemiştir. Öğrencilerin görüşleri Görsel içerikle öğrenmenin daha kalıcı olduğu, öğrenmeye olumlu etkisi olduğu, ilgi çekici olduğu, eğlenceli olduğu yönündedir. Edgar Dale'nin öğrenme piramidine göre Öğrenme çoğunlukla görme duyumuzla olur (Büyükaslan, 2007). Öğrencilerin verdikleri cevapların bazıları şöyledir;

“Vardır. mesela bir şeyi göstererek, çizerek anlatılırsa benim daha çok ilgimi çekiyor, ilgimi çekince o konuyu daha da iyi anlıyorum mesela Sosyal bilgiler dersinde orda slaytlarda gördüğümüz adamları, resimleri çizince o benim daha iyi öğrenmeme yardım ediyor. Görseller ilgimi çekiyor” (Ö5).

“Vardır Tek yazı yazarken anlaşılıyor da onları gösterince daha iyi anlaşılıyor.” (Görüşme Kaydı: Ö9).

Proje kavramı temasının iki alt kategorisi vardır. Biri proje deneyimi bir diğeri ise hayal edilen projedir. Proje kavramı teması yönelik “Proje denilince aklınıza neler gelmektedir?” sorusuna karşılık her öğrencinin proje kavramının bir yönüne değindiği fark edilmiştir. Her birinin cevabı birlikte bütün olarak değerlendirildiğinde proje kavramı hakkında yapılan tanım ve açıklamalara örtüştüğü anlaşılmıştır. Öğrencilerin proje hakkında söylediklerini şöyle özetlenebilir; proje bir sorunu çözmek veya insan yararına iş yapmak, bir şeyler icat etmek, ihtiyaçları gidermek amacıyla grup ya da bireysel olarak öğretmen rehberliğinde ortaya konulan maket yani üründür. Bazı öğrencilerin proje kavramı ile söyledikleri aşağıdaki gibidir;

“Algoritma yani bilişim dersiyse algoritma, böyle akış diyagramı böyle kartona yapabileceğimiz şeyler.” (Ö2).

“Proje bence bir sorunu çözmek için yapılan iştir.” (Ö6).

Proje deneyimi alt faktörü ile alakalı “daha önce bir projede görev aldınız mı?” sorusuna on öğrencinin altısı daha önce projede görev aldığını söylemiştir. Öğrencilerin proje diye bahsettiklerinin aslında birer uygulama veya grup çalışmasından ibaret olduğu anlaşılmaktadır. Söyledikleri projeler; proje tabanlı öğrenme sürecine göre yürütülmemiş projeler oldukları söylenebilir. Öğrencilerin proje tabanlı eğitim ile bu çalışmayla tanıştıkları söylenebilir. Bazı öğrencilerin cevapları aşağıdaki gibidir.

“Teknoloji tasarım dersinde ışık pervaneli ışık yaptık. Pervaneye üflediğimde ışık yanıyordu evde.” (Ö10).

“Fabl yaptık. Türkçe dersinde fabl Hayvanları konuşurmak. Mesela bir grup oluyoruz. Bu gruplarda istediğin hayvanı oluyorsun. Böyle konuşmalar filan yapıyorsun buna fabl diyoruz. İsteddiğimiz hayvanı canlandırdık” (Ö7).

Hayal edilen proje alt faktörü kapsamında “İleride yapmayı düşündüğünüz proje var mıdır?” sorusu yöneltilmiştir. Öğrenciler yapmayı hayâl ettikleri projeleri net bir şekilde ifade

edemeseler dahi birçok değişik proje fikri söylemişlerdir. Öğrencilerin hayal ettikleri projelerden bazıları şöyledir.

“Hastaların işini kolaylaştıracak robot. Hastalar için ilaç verecek bir robot.” (Ö10).

“Uçan araba yapmak istiyorum. Ülkemizi daha geliştirecek şeyler yapmak istiyorum. Mesela su israf etmemek için ya da arabaların zararlı çıkardığı gazları önleme gibi.” (Ö4).

Kodlama hakkındaki görüşler temasının iki alt faktörü vardır. Bunlar kodlamayı faydalı bulma ve kodlamayı sevmedir. Kodlamaya hakkındaki görüşler temasına yönelik “Kodlama ile ilgili düşünceniz nedir? Anlatabilir misiniz?” sorusuna verilen cevaplara göre on öğrencinin dokuzu kodlamayı faydalı bulmuştur. Öğrencilerin kodlamaya karşı algı ve tutumlarının olumlu olduğu ve bunda öğrencilerin kodlamayla Scratch programı ile tanışmalarının da etkisi olduğu düşünülmektedir. Scratch programı sade, anlaşılır, kod yazmadan kodlamayı, proje yapmayı öğreten kolay, keyifli bir programdır (Çatlak, Tekdal ve Baz, 2015). Kodlama eğitiminde Flowchart 3.0 gibi görsel programların özellikle ortaokul öğrencilerinde motivasyonu artırması beklenebilir. Bazı öğrencilerin kodlamaya karşı görüşleri aşağıda örneklenmiştir.

“Gelecekte işime yarayacağını düşünüyorum. Seviyorum. Bunda Scratch programının çok etkili olduğunu düşünüyorum çünkü ben kodlamayı ilk önce düz sayılarla yapacağımızı düşünüyordum. Ve biraz korkuyordum biz daha çocukuz biz bunu nasıl yapacağız diye ama bu süreçte daha basit olduğunu öğrendim” (Ö5).

“Bence gayet hayırlı bir şey ileride gelecek kodlamayla çalışacağı için öğrenilmesi gereken bir şey” (Ö6).

Problem çözme becerisi temasının iki alt kategorisi vardır. Bunlar örnek problem durumu anlatma ve öğrencilerin problem çözme stratejisidir. Problem çözme becerisi teması ile ilgili “Problemlerle karşılaştığınızda neler yaparsınız?” sorusuna karşılık öğrenciler ilk önce kendilerinin problemi çözmeleri gerektiğini söylemişlerdir. Öğrencilerin problemi çözmek için problemin anlaşılması gerektiğini belirtmişlerdir. Problem çözme sürecinin adım adım ilerleyeceğini ve her zaman ilk seferde problemin çözülemeyeceğini ifade etmişlerdir. Öğrencilerin görüşlerinin bazıları aşağıda örneklenmiştir.

“O problemi çözmeye çalışırım. İlk önce elimde olanları kullanmaya çalışırım. Elimden neler gelebilir veya kimden yardım alabilirim. Kiminle yapabilirim, ilk önce neler yapabileceğimi araştırıp sıraya düzene koyarım ve bu liste plan üzerinden ilerlerim.” (Ö5).

“Önce problemin neden kaynaklandığını çözmeye çalışırım. Ondan sonrada elimden geleniyle de çözmeye çalışırım.” (Ö6).

Problem durumunu anlatma alt faktörüyle ilişkili “Bir problem örneği verebilir misiniz?” sorusuna verdikleri cevaplardan öğrencilerin problem çözerken başaramadıklarında farklı yollar deneyerek problemi çözmeye çalıştıkları anlaşılmaktadır.

“Maket yaparken defalarca yapmam rağmen bozuldu ne kadar yaparsam yapıyım hiç yapamadım. Ama en sonunda bir liste düzene tekrar koydum. Listede hata yapmışım o listede, listeyi düzelttiğim zaman maketimi daha düzgün yaptım.” (Ö5).

“Mesela biz Flowchart’ta o programı yaparken çalışmıyordu oradaki problemi belirledik. Ondan sonra farklı bir yol deneyerek yaptık.” (Ö 6).

“Bu problemlerin üstesinden nasıl geliyorsunuz?” sorusu Problem çözme stratejisi alt temasına yöneliktir. Öğrencilerin verdikleri cevaplar göre birinci aşamada problemi anlaşılmalı ikinci aşamada çözüm yollarından en iyi olanı seçmeye çalışılmalıdır. Bu soruya öğrencilerden ikisinin verdiği cevaplar şöyledir.

“İlk başta düşüncelerimi sıralarım. En mantıklı olanı seçer sonrada uygulardım.” (Ö4).

“İlk önce o problemle ilgili araştırma yapmam gerekir ki o problemin neden olduğunu öğreneyim. Mesela sıra arkadaşım bana neden küstü diyemem. Neden bana küstüğünü öğrenmem gerekir. Ondan sonra onun gönlünü almaya çalışırım. Ki tekrar benimle barışsın” (Ö5).

TARTIŞMA VE SONUÇ

Deneysel uygulamanın öğrencilerin tüm becerilerine kontrol grubundaki uygulamaya göre anlamlı düzeyde daha fazla katkı sağlamadığı, dahası öğrencilerin algoritmik düşünme becerilerine kontrol grubundaki uygulamadan anlamlı düzeyde daha az katkı sağladığı belirlenmiştir. Çalışmanın nitel boyutunda deney grubundan 10 öğrenci ile görüşme yapılmış ve görüşmeler sonucunda görüşmenin 5 temel teması olduğu görülmüştür. Bunlar sabah rutini algoritması, kodlamaya yönelik tutum, proje kavramı, Flowchart hakkındaki görüşler ve problem çözmeye yönelik algıdır. Yapılan görüşmeler sonucunda öğrencilerin genellikle kalabalık ailelerde yaşadıkları, evlerinde bilgisayar olmadığı, bilgisayar bilgilerinin bilişim teknolojileri ve yazılım dersindeki öğrendikleri ile sınırlı olduğu, daha önce proje tabanlı bir eğitim almadıkları anlaşılmıştır. Bu eğitim sonunda öğrencilerin Proje Tabanlı Öğretim modelini daha iyi kavradıkları söylenebilir. Öğrencilerin FlowChart görsel programlama aracına ve kodlamaya dair genellikle olumlu tutumlar geliştirdikleri görülmüştür. Öğrencilerin problem çözme kavramını ve sürecini kavradıkları belirlenmiştir.

Algoritma eğitimi konu alan bu çalışmada öğretim aracı olarak akış diyagramları ile kodlama yapmaya fırsat veren Flowchart 3.0 görsel programlama aracı kullanılmıştır. Öğretim yöntemi olarak proje tabanlı öğretim modeli esas alınmıştır. Kâğıt kalem etkinliğine dayalı algoritma eğitimi ile Flowchart destekli proje tabanlı algoritma eğitimi arasında bilgisayarca düşünme becerisi, kodlamaya dönük tutum ve problem çözme becerisine yönelik algı açısından anlamlı farklılık bulunmamıştır. Alanyazında görsel programlama araçları kullanılarak kodlama eğitiminin yararlarına ilişkin pek çok kanıt rastlamak mümkündür. Örneğin, Arslan Namlı ve Şahin (2017) tarafından yapılan çalışmada algoritma eğitiminin problem çözme becerisine anlamlı katkı sağladığı belirtilmiştir. Küçük çocuklara verilen programlama eğitiminin bilişsel becerilere olan etkisi hep araştırılmıştır (Akpınar & Altun, 2014). Programlama eğitimine küçük yaşlarda başlamak; analitik düşünme ve algoritmik çözüm yapabilme açısından önemlidir (Alkan, 2019). Programlama öğretimine katı kurallı, karmaşık yapı, hata ayıklama sistemi gelişmiş programlama dilleri ile başlamak öğrencinin motivasyonunu düşürebilir (Ersoy, Madran & Gülbahar, 2011). Programlama dilinin söz dizim kurallarının karmaşık olması öğrenmesi zor olan programlamayı daha da zorlaştırmaktadır (Saygıner & Tüzün, 2017). Küçük yaşta öğrencilerin ilgisini çekmek için görsel ortamlarla programlamaya başlanmalıdır (Alkan, 2019). Görsel programlama araçlarının öğrencilerin motivasyonlarını artırdığı (Erümit, Karal, Şahin, Aksoy & Benzer, 2018), yaratıcı düşünme ve analitik düşünme becerisi kazandırmaya da yardımcı olduğu söylenebilir (Kabak & Güneş, 2013). Görsel programlama noktalı virgül koyma gibi söz dizimsel zorunluluklardan kurtardığı için motive edicidir (Koh, Basawapatna, Bennett & Repenning, 2010).

Görsel programlama ve geleneksel programlama eğitimin karşılaştırıldığı üniversite öğrencilerine yönelik yüksek lisans tez çalışmasında deney grubu öğrencilerinin daha başarılı olduğu fakat kodlamaya dönük tutumları arasında anlamlı bir farklılık olmadığı görülmüştür (Yiğit, 2016). Yapılan bu çalışmada da gruplar arasında kodlamaya yönelik anlamlı bir farklılık bulunmadığı belirtilmiştir.

Uslu, Mumcu ve Eğin, (2018) tarafından yapılan düşünme becerilerine etkisinin araştırıldığı çalışmanın bulgularına göre görsel programlama etkinliklerinin bilgi işlemsel düşünme

becerisine anlamlı bir etkisi bulunmadığı ancak öğrencilerin hayal güçlerini geliştirdiği ve bilgisayar bilimine karşı farkındalık oluşturduğu sonucuna ulaşılmıştır. Araştırmada elde edilen bu sonuç bu çalışmanın bulgularını destekler niteliktedir. Flowchart Görsel Programlama Eğitiminin problem çözme becerisine olan etkisinin incelendiği bu çalışmanın nicel bulgularına göre öğrencilerin problem çözme becerilerine karşı anlamlı düzeyde katkısı olmadığı bulunmuştur. Ancak çalışmanın nitel boyutunda öğrencilerle yapılan görüşmeler sonucunda bu çalışmanın öğrencilerin hayal güçlerine ve kodlamaya yönelik tutumlarına dönük olumlu katkılarının olduğu görülmüştür. Beşinci sınıf öğrencileriyle yapılan bir başka çalışmada görsel programlamanın öğrencilerin problem çözme becerilerine anlamlı bir etkisi olmadığı bulunmuştur (Kalelioğlu & Gülbahar, 2014). Bu deneysel çalışma dört hafta sürmüştür. Eğitimin kısa süreli olması problem çözme ve bilgisayarca düşünme becerileri üzerindeki katkısının görülememesinin bir nedeni olabilir. Problem çözme gibi üst düzey becerilerin kısa süreli çalışmalarda gelişim göstermesi beklenemez (Kukul & Gökçearslan, 2014). Flowchart Destekli Proje Tabanlı Algoritma Eğitimi ile Geleneksel algoritma eğitimi arasında Kodlamaya dönük tutumları açısından anlamlı bir farklılık bulunamamıştır. Erken yaşta kodlama eğitimi ile ilgili çalışmalarda genellikle nitel araştırma yöntemleri kullanıldığı tespit edilmiştir (Durak, Karaoğlu-Yılmaz, Yılmaz-R. & Seferoğlu, 2017). Çalışmanın nitel boyutunda yapılan görüşmeler sonucunda öğrencilerin proje tabanlı eğitim tecrübelerinin olmadığı saptanmıştır. Proje tabanlı eğitim ile bu çalışmayla tanıştıkları düşünülmektedir. Öğrenciler bu deneysel çalışma sürecinde proje tabanlı eğitim ve problem çözme sürecini öğrendikleri görülmüştür. Alice Görsel Programlama Eğitimi alan öğrencilerin Alice programı hakkındaki görüşlerinin alındığı nitel bir çalışmada öğrenciler Alice programında görsel kod kullanılmasının kalıcılığı artırdığını ve ilgilerini çektiğini söylemişlerdir (Kaya & Çakır, 2018). Yapılan bu çalışmada da öğrenciler flowchart'ta yapılan uygulamalarla öğrendiklerinin daha kalıcı olduğunu belirtmişlerdir. Bununla birlikte görsel temalarla öğrenmenin daha ilgi çekici olduğunu söylemişlerdir. Alice Görsel programlama ortamlarından bilgi aktarma üzerine yapılan bir başka araştırma da görsel programlamaya alışan öğrencilerin Java, C++ gibi metin tabanlı programlama eğitimine geçtiklerinde hayal kırıklığına uğradıklarını ve söz dizimsel hatalardan bunaldıkları belirtilmiştir (Kaya & Çakır, 2018). Programlamanın nihai hedefinin C++ gibi metin tabanlı programlama dillerinde program geliştirmek olduğu ifade edilmiştir (Parsons & Haden, 2007). Öğrencilerin görüşmelerde verdikleri cevaplar incelendiğinde proje tabanlı eğitim basamakları ile problem çözme adımlarını kavradıkları düşünülmektedir.

Her ne kadar kâğıt kalem etkinliğine dayalı algoritma eğitimi ile Flowchart destekli proje tabanlı algoritma eğitimi arasında bilgisayarca düşünme becerisi, kodlamaya dönük tutum ve problem çözme becerisine yönelik algı açısından anlamlı bir farklılık oluşturmamış olsa da, öğrencilerini bu tür etkinlikleri daha eğlenceli buldukları göz önünde bulundurulduğunda, programlama eğitiminde bu tür etkinliklere yer verilmesi, ancak aynı kazanımlara kâğıt kalem etkinlikleri ile de ulaşılabileceği göz önünde bulundurulması önerilebilir.

KAYNAKÇA

- Akkuş, İ., Özhan, U. & Kan, A. (2019). Ortaokul öğrencileri için kodlamaya yönelik tutum ölçeği. *İlköğretim Online*, 18(2), 837-851
- Akpınar, Y. & Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *İlköğretim Online*, 13(1), 1-4.
- Alkan, A. (2019). Özel yetenekli öğrencilerin bilgisayar oyunları destekli kodlama öğrenimine yönelik tutumları. *Milli Eğitim Dergisi*, 48(223), 113-128.
- Baz, F. Ç. (2018). Çocuklar için kodlama yazılımları üzerine karşılaştırmalı bir inceleme. *Current Research in Education*, 4(1), 36-47.

- Baki, A. & Gökçek, T. (2012). Karma yöntem araştırmalarına genel bir bakış. *Elektronik Sosyal Bilimler Dergisi*, 11(42), 1-21.
- Bingham, A. (2016). *Çocuklarda problem çözme yeteneklerinin geliştirilmesi*, (Çev.:A.F. Oğuzkan), 11-14. Ankara: M.E.B
- Büyükaslan, A. (2007). Yabancı dil Türkçenin öğretilmesinde yeni yöntemler: Bilişim uygulamaları, çözüm önerileri. Department D'études Turques Turcologue u-strasbourg, Strasbourg, 30(05), 2015.
- Büyüköztürk, Ş. (2019). *Sosyal bilimler veri analizi el kitabı*, 25, 42-43, Ankara: Pegem Akademi
- Büyüköztürk, Ş., Çakmak, E., Akgün, Ö. E., Karadeniz, Ş. & Demirel, F. (2018). *Eğitimde bilimsel araştırma yöntemleri*. Ankara: Pegem Akademi
- Charntaweekhun, K. & Wangsiripitak, S. (2006). Visual programming using flowchart. In *2006 International Symposium on Communications and Information Technologies*, 1062-1065
- Çatlak, Ş., Tekdal, M. & Baz, F. (2015). Scratch yazılımı ile programlama öğretiminin durumu: Bir doküman inceleme çalışması. *Journal of Instructional Technologies & Teacher Education*, 4(3), 13-25.
- Delebe, E. (2018). *5. ve 6. sınıf bilişim teknolojileri ve yazılım dersi kodlama kılavuzu* 12-14. Ankara: Yenilik ve Eğitim Teknolojileri Genel Müdürlüğü
- Durak, H., Karaoğlan-Yılmaz, G., Yılmaz-R. & Seferoğlu, S. S. (2017). Erken yaşta programlama eğitimi: Araştırmalardaki güncel eğilimlerle ilgili bir inceleme. H. F. Odabaşı, B. Akkoyunlu ve A. İşman (Ed). *Eğitim teknolojileri okumaları 2017*, 12. Bölüm, 205-236).
- Ekici, D. İ. & Balım, A. G. (2013). Ortaokul öğrencileri için problem çözme becerilerine yönelik algı ölçeği: Geçerlik ve güvenilirlik çalışması. *YYÜ Eğitim Fakültesi Dergisi (YYU Journal of Education Faculty)*, 10(1), 67-86.
- Erdem, M. (2002). Proje tabanlı öğrenme. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 22, 172-179.
- Ersoy, H. Madran, R. O. & Gülbahar, Y. (2011). Programlama dilleri öğretimine bir model önerisi: robot programlama. *Akademik bilişim*, 11.
- Erümit, K. A., Karal, H., Şahin, G., Aksoy, D. A., Aksoy, A. & Benzer, A. İ. (2018). Programlama öğretimi için bir model önerisi: Yedi adımda programlama. *Eğitim ve Bilim*, 44(197).
- Fırat, M., Kabakçı Yurdakul, I. & Ersoy, A. (2014). Bir eğitim teknolojisi araştırmasına dayalı olarak karma yöntem araştırması deneyimi. *Eğitimde Nitel Araştırmalar Dergisi - Journal of Qualitative Research in Education*, 2(1), 65-86. [Online]: www.enadonline.com, doi: 10.14689/issn.2148-2624.1.2s3m
- Genç, M. (2007). İşbirlikli öğrenmenin problem çözmeye ve başarıya etkisi. Yayımlanmamış Doktora Tezi. Marmara Üniversitesi. Eğitim Bilimleri Enstitüsü.
- Güven, Y. (2018) *5. sınıf bilişim teknolojileri ve yazılım dersi öğretmen rehber kitabı*, 1, 163-164 Ankara: M.E.B.
- Gökoğlu, S. & Yüksel, D. (2016). Bilgisayar programcılığı öğrencilerinin algoritma ile ilgili metaforlar. *10th International Computer and Instructional Technologies Symposium (ICITS)*
- Grover, S. & Pea, R. (2013). Computational thinking in K-12: A Review of the State of the Field. *Educational Researcher* 42(1), 38-43.
- Gülbahar, Y., Kalelioğlu, F. & Karataş, E. (2017). *Orta öğretim bilgisayar bilimi kitabı kurl* 239-240. Ankara: M.E.B.
- Güven, Y. (2018). *5. Sınıf bilişim teknolojileri ve yazılım dersi öğretmen rehber kitabı* 167-231. Ankara: M.E.B.
- Hidayat, P. W. & Susilowati S. M. E. (2020). Analysis of problem solving abilities of elementary school students through problem-based learning model based on self-confidence, *Journal of Primary Education* 9(1), 8-15.
- Kabak, D. & Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 2(3).
- 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.

- Korkmaz, F. & Kaptan, F. (2001). Fen eğitiminde proje tabanlı öğrenme yaklaşımı. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 20, 193-200.
- Korkmaz, Ö., Çakır, R. & Özden, M. Y. (2015). Bilgisayarca Düşünme Beceri Düzeyleri Ölçeğinin (Bdbd) Ortaokul düzeyine uyarlanması *Gazi Eğitim Bilimleri Dergisi*, 1(2), 143-162.
- Koh, K. H. Basawapatna, A. Bennett V. & Repenning, A. (2010) Towards the Automatic Recognition of Computational Thinking for Adaptive Visual Language Learning 2010 IEEE Symposium on Visual Languages and Human-Centric Computing
- Kukul, V. & Gökçearslan, Ş. (2014). Scratch ile programlama eğitimi alan öğrencilerin problem çözme becerilerinin incelenmesi. *International Computer & Instructional Technologies Symposium in Trakya University Edirne*, 8, 58-63.
- MEGEP. (2011). *Kodlamaya hazırlık*, 13-14. Ankara: M.E.B
- MEB (2018). 2023 Vizyonu. http://2023vizyonu.meb.gov.tr/doc/2023_EGITIM_VIZYONU,75-76.21.01.2020 tarihinde erişilmiştir.
- Parsons D. & Haden, P. (2007). Programming osmosis: Knowledge transfer from imperative to visual programming environments. In *Proceedings of The Twentieth Annual NACCCQ Conference*, 209-215
- Saygıner, Ş. & Tüzün, H. (2017). Programlama eğitiminde yaşanan zorluklar ve çözüm önerileri. 1. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu.
- Sayın, Z. & Seferoğlu, S. S. (2016). Yeni bir 21. yüzyıl becerisi olarak kodlama eğitimi ve kodlamanın eğitim politikalarına etkisi. *Akademik Bilişim Konferansı*, 3-5.
- Seferoğlu, S. S. & Akbıyık, C. (2006). Eleştirel düşünme ve öğretimi. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 30(30), 193-200.
- Selby, Cynthia C. & Woollard, J. (2013). Computational thinking: the developing definition. *University of Southampton (E-prints)*.
- Şahin, Ç. (2004). Problem Çözme Becerisinin Temel Felsefesi *Kazım Karabekir Eğitim Fakültesi Dergisi*, 10, 160-171
- Şahin, H., Korkmaz, Ö., Çakır, R. & Erdoğan, F. U. (2019). Bilişim teknolojileri öğretmenlerinin kodlamaya dönük tutumları ve öz-yeterlilikleri. 1. *Uluslararası Çağdaş Eğitim ve Sosyal Bilimler Sempozyumu*, 16.
- Arslan Namlı, N. & Şahin, M.C. (2017). Algoritma eğitiminin problem çözme becerisi üzerine etkisi. *Recep Tayyip Erdoğan Üniversitesi Sosyal Bilimler Dergisi* 5: 135-153
- Talu, N. (1999). Çoklu zeka kuramı ve eğitime yansımaları. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 15, 164-172
- TDK. (2019). Türk Dil Kurumu Güncel Türkçe Sözlük http://tdk.gov.tr/index.php?option=com_gts&arama=gts&guid=TDK.GTS.5da0bf5ab0d747.51299676 11.10.2019 tarihinde erişilmiştir.
- Uslu, N. Mumcu, F. & Eğin F. (2018). Görsel programlama etkinliklerinin ortaokul öğrencilerinin bilişimsel düşünme becerilerine etkisi. *Ege Eğitim Teknolojileri Dergisi*, 2(1), 19-31
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Yıldırım, H.İ. (2018). Bilim şenliklerinin ortaokul 6. sınıf öğrencilerinin problem çözme becerilerine etkisi. *Trakya Üniversitesi Eğitim Fakültesi Dergisi*, 8(2), 390-409
- Yiğit, M. F. (2016). *Görsel programlama ortamı ile öğretimin öğrencilerin bilgisayar programlamayı öğrenmesine ve programlamaya karşı tutumlarına etkisinin incelenmesi*. Yayımlanmamış yüksek lisans tezi. Ondokuz Mayıs Üniversitesi, Eğitim Bilimleri Enstitüsü, Samsun.

Kaynak gösterimi için (for cite in):

Üngör, Y.E., Korkmaz, Ö., Çakır, R. & Erdoğan, F.U. (2020). Flowchart destekli proje tabanlı algoritma eğitiminin etkililiği. *Turkish Journal of Primary Education (TUJPED)*, 5(2), 98-118.

EXTENDED ABSTRACT

Introduction

Programming training usually begins with algorithm training. Flowchart can be included in the algorithm training process. The flowchart is a visual representation of the algorithm with shapes and symbols, while the algorithm is a complete set of instructions that show which operation is performed in which order. Project based learning model can be used in the training of algorithms and programming. Project-based learning is a learning approach that attempts to solve a problem situation chosen from daily life with individual or small groups. Coding provides algorithmic instruction to the computer so that the computer can perform the tasks requested from it. Visual programming gives opportunity developing of product by enjoying of students by prevent text-based programming's complex. Debugging is simpler as flowchart steps can be followed in the Flowchart visual programming tool. Coding improves problem solving skill along with programming knowledge. Problem solving skill is an effort that is shown to deal with the situation and defining a problem. The skill to solve problems is an effort to identify and deal with a problem. Computational thinking is related to problem solving skill, Algorithmic Thinking, Creative thinking and critical thinking. Computational thinking is thought processes used to formulate problems. In this context, the aim of this study was to determine the effect of Project-Based Flowchart supported algorithm training on Computational thinking, problem solving and attitude toward coding.

Method

In this research, quasi experimental with pretest posttest control group, sequential explanatory, mixed design was used. In the qualitative part of study, 10 students were interviewed by using semi-structured interview form with. The groups were composed of 30 students studying in 6th and 7th grades with convenience sampling. In this study, quantitative data were collected with three scales in five grade Likert types. the computational thinking skill levels scale consists of 22 items under a factor of 5. he Cronbach alpha reliability coefficient of the scale was 0.809 and the substance differentiation t value was found -37,105. the attitude scale towards coding is consisted of one factor and 10 substances. The Cronbach alpha reliability coefficient of the scale .905 was found. The perception scale for problem solving skills consists of 2 factors and 22 substances, 15 positives and 7 negatives. The Cronbach Alpha reliability value of this scale .88 was found. The substance discrimination of this scale is significant at the .001 level. This experimental study took 4 weeks.

Findings

The Mann Whitney-U test was applied to the scores to determine whether there was a significant difference between the pretest scores of the experimental and control group There isn't a significant difference other than computational thinking. In order to control these differences, the last test-pre-test difference scores were used in the analysis of the effectiveness of the experimental application. It can be said that experimental practice doesn't contribute significantly more to all students ' skills than control group. five main themes emerged in the content analysis of students' thoughts on experimental practice. These are morning routine algorithm, attitude towards coding, project concept, opinions about flowchart and problem-solving perception.

Conclusion

There couldn't determine significant difference between paper-pen-based algorithm training and flowchart-supported project-based algorithm training. There is a lot of evidence in the literature that visual coding training contributes to thinking skills. For young students, visual programming is motivating because it saves them from syntactic errors. This study took four weeks. However, high-level skills such as problem solving are expected to develop over a longer period of time. In the field has been found that more qualitative research has been done in research conducted with younger ages. In interviews with students, students' attitudes towards project-based learning, coding and the flowchart 3.0 visual programming tool were positive. Although there is no significant difference between the groups, given that the students find visual programming activities more enjoyable, it is advisable to include such activities in programming education, but to consider that the same gains can be achieved through paper and pencil activities.