

EŞLİ PROGRAMLAMA ETKİNLİĞİNİN ORTAOKUL ÖĞRENCİLERİNİN PROGRAMLAMA BAŞARILARINA ETKİSİ

Serkan Demirören
Ankara Üniversitesi

Özet

Bu çalışmanın amacı, eşli programlama uygulamasının etkililiğinin değerlendirilmesidir. Araştırmanın deneysel süreci 2014 – 2015 eğitim öğretim yılı bahar döneminde Ankara’da bulunan bir ortaokulda 6. Sınıf öğrencisi 36 öğrenci ile yürütülmüştür. Araştırmada sontest kontrol gruplu deneysel desen kullanılmıştır. Araştırmanın bağımsız değişkeni öğrencilerin tek başına ya da eşli çalışma durumudur. Her iki gruba da aynı eğitim verilip sadece bireysel ya da eşli çalışma durumunun etkisi araştırılmıştır. Araştırmanın bağımlı değişkeni ise öğrencilerin programlama başarılarıdır. Araştırmada sonuçlar üzerindeki yanlılığı azaltmak için mantıksal düşünme beceri puanları çalışma öncesinde belirlenerek kontrol değişkeni olarak kullanılmıştır. Öğrencilerin mantıksal düşünme becerilerini ölçmek için Mantıksal Düşünme Grup Testi kullanılmıştır. Programlama başarılarını belirlemek için ise bir başarı testi geliştirilmiştir. Test üç bilişim teknolojileri öğretmeninden uzman görüşü alınarak düzenlenmiştir. Ayrıca testin güvenilirlik çalışmaları farklı bir ortaokulda 65 öğrenci ile ön uygulama yapılarak gerçekleştirilmiştir. Verilerin çözümlenmesinde, başarı testinin son test olarak uygulanmasından ve Mantıksal Düşünme Grup Testi puanının kontrol değişkeni olarak kullanılmasından dolayı tek faktörlü kovaryans analizi uygulanmıştır. Bu analiz için öncelikle gerekli olan varsayımlar sınanmıştır. Elde edilen bulgulara göre, eşli olarak programlama yapan öğrenciler ile tek başına programlama yapan öğrencilerin programlama başarıları arasında anlamlı fark yoktur.

Anahtar Sözcükler

Bilgi işlemsel düşünme, Eşli programlama, Programlama öğretimi, Scratch.

THE EFFECT OF PAIR PROGRAMMING ACTIVITY ON THE PROGRAMMING SUCCESS OF THE SECONDARY SCHOOL STUDENTS

Serkan Demiroren

Ankara University

Abstract

The aim of this study is to evaluating effectiveness of pair programming. The experimental period of the research is applied to 36 sixth grade students from a secondary school in Ankara at 2014 – 2015 spring semester. The research work has been designed and realized with the posttest experimental pattern with control group. Independent variable of the research is students being pair or alone in programming sessions. The two group take the same instruction and we just look for the effect of pair working. The dependent variable of the research is programming achievement of students. The students' logical thinking abilities were used as control variable (covariate) to reduce bias on results. Group Assessment of Logical Thinking was used to measure students' logical thinking ability. A programming achievement test has been developed to measure students programming success. For the test, expert opinion was received from three information technologies teachers. The test pre examined in different secondary school with 65 students which already taken Scratch programming course. On the single factor that arises from the posttest application of the performance variable that belong to the students working as pair or alone and using Group Test of Logical Thinking results as control variable ; ANCOVA test has been applied. All assumptions has been tested for this analysis. According to the obtained findings, between the pair working students and alone working students there is no significant difference on the students' programming achievement.

Keywords

Computational thinking, Pair programming, Programming instruction, Scratch.

GİRİŞ

Bilgisayar okuryazarlığı kavramı Milli Eğitim Bakanlığı öğretim programlarında ve birçok strateji metninde sıklıkla vurgulanan kavramlardan birisidir. Uygulamalara ve tartışmalara bakıldığında ise konunun yeteri kadar anlaşılmadığı ve söz konusu bilgisayar olunca okuryazarlığın okuma kısmına odaklanıldığı dikkati çekmektedir. Bilgisayar okuryazarlığının yazma tarafı ise daha çok kelime işlemcilerde metin oluşturma ya da sunum hazırlama olarak görülmektedir (Akpınar ve Altun, 2014). Bilgisayar okuryazarlığı ya da daha geniş kapsamda dijital okuryazarlık sadece ilgili teknolojileri kullanmaktan ibaret değildir. Programlama eğitimi çocuklar ve gelecek nesiller için okuma aşamasından artık yazma sürecine geçmelerinde etkili bir basamak olabilir. Bu amaçla bilgisayar okuryazarlığı, uzun yıllardır birçok ülkenin öğretim programları kapsamına alınmıştır.

Çocuklar için programlama eğitiminin gerekliliği artık daha yüksek sesle dile getirilmeye başlanmıştır. Bununla birlikte çocuklara programlama öğretmek için uluslararası bazı organizasyonlar da ortaya çıkmıştır. “Hour Of Code” en yaygın olanlardan biridir ve “4’ten 104 yaşına kadar herkes için” sloganı ile 180 ülkede on milyonlarca öğrenciye ulaşmıştır (Hour of Code, 2015). Sonuç olarak, programlama sadece konuya yönelik eğitim almış yetişkin uzmanların yapabileceği bir iş değildir ve çocuklar programlama konusunda belli ve anlamlı bir düzeye kadar geliştirilebilir.

Programlama öğrenme, öğrencilerin algoritma öğrenmesi ve bilgisayarın sistematik işleyişini anlaması yoluyla mantıksal düşünme becerilerini geliştirebilmekte ve problemlere sistematik yöntemlerle çözüm üretmelerini sağlayabilmektedir (Yoo, Cha ve Lee, 2007). Konu çocuklar olunca özellikle vurgulanan olay problem çözme yöntem ve süreçleridir. Bu bakış açısı ile programlama becerileri problem çözme ile ilişkilendirilerek bilgi işlemsel düşünme kavramı ortaya atılmıştır.

Bilgi işlemsel düşünme kavramı bilgisayar bilimleri ile ortaya çıkmıştır ve problemleri çözmek için bilgisayar bilimleri tekniklerini kullanan bir problem çözme yöntemidir. Bilgi işlemsel düşünme süreci, problemi tanımlamak için uygun yolun seçilmesi, problemin bir modelinin oluşturulması ve sonrasında problemi çözmek için en iyi yöntemin seçilmesini kapsar (Wing, 2008).

Bilgi işlemsel düşünme sürekli gelişen dijital dünya ile iletişim kurma ve bu dünya ile düşünme konusunda ne katkı sağlayabilir? Bilgisayarlar uzun zamandır okullarda olmalarına rağmen, bilgi işlemsel düşünmenin öğretim programlarının bir parçası haline gelmesi zaman almaktadır. Öğrencilere kelime işlemci ya da sunum yazılımlarının öğretilmesi onların daha yaratıcı ve derin düşüncelerini sağlamaktadır (Kafai ve Burke, 2014). Kısacası öğrencilerin her tipte probleme çözüm üretebilmeleri için daha sistematik düşünmeyi öğrenmeleri gerekmektedir.

Tüm bu gereklilikler göz önüne alınarak, 21. yüzyıl bilgi ve iletişim teknolojileri yeterliliklerinin bir gerekliliği olarak programlama eğitimi son dönemde ortaokullara ve ilkokullara kadar taşınmıştır (Kalelioğlu ve Gülbahar, 2014). Öğretmenlerin, proje tabanlı öğrenme yoluyla öğrencilerinin anlamlı ve eğlenceli oyun projeleri oluşturmalarını sağlayarak programlama öğrenmeleri konusunda onları teşvik etmesi, programlamayı eğlenceli hale getirmesi gerektiği savunulmaktadır (Calder, 2010). Programlama dillerinin çocuklara öğretilmesi ve buna bağlı olarak bilgi işlemsel düşünme becerilerinin geliştirilmesi konusunda çalışmalar son dönemde hız kazanmıştır.

Bilgisayar programlama, nesneleri programlamak ve işlemleri yürütebilmek için güçlü mantıksal düşünme becerileri gerektirir. Yapılan araştırmalarda, bilgisayar bilimleri eğitiminin mantıksal düşünme yoluyla problem çözme becerilerini geliştirmeye yardımcı olduğu fark edilmiştir (Yoo, Cha, Lee, 2007). Programlama da temelde bir problem çözme etkinliği olduğundan, programlama eğitiminin mantıksal düşünmeyi geliştirebileceği söylenebilir.

Piaget (1969) mantıksal düşünmeyi, problem çözmede kullanılan zihinsel işlemler olarak tanımlar. Mantıksal düşünme; sayıları kullanabilme, problemlere bilimsel çözümler üretebilme, hesaplama, sınıflama yapabilme gibi davranışları gösterme yeteneğidir. Mantıksal düşünme Piaget'in soyut işlemler dönemi özelliklerinden biri olarak da bilinmektedir (Bozdoğan, 2007).

Mantıksal düşünme becerisi öğrencilerin akademik başarısı ve kavramları yapılandırmaları üzerinde çok önemli bir role sahiptir (Tobin ve Capie, 1982). Benzer şekilde mantıksal düşünme becerisi daha gelişmiş olan bireyler programlama eğitiminde daha başarılı olmaktadır (Gorman ve Bourne, 1983). Programlama dilleri temel mantıksal kavramlara dayanmaktadır. Aralarındaki ilişki dikkate alındığında daha önce hiç programlama eğitimi almamış birisinin, programlama öğrenme başarısının mantıksal düşünme becerisi ile ilişkili olabileceği düşünülebilir.

Programlama eğitiminde başarı sadece bilişsel değişkenlere değil eğitim yöntemine de bağlıdır. Salomon (1993) bilginin, paylaşılan hedefler doğrultusunda ortak çabayla ya da kişilerin bakış açılarındaki farklar doğrultusunda gerçekleşen diyaloglarla sosyal olarak inşa edildiğini ifade etmiştir. Vygotsky (1980) çocuğun bilişsel gelişiminde içinde bulunduğu 'sosyal çevrenin' önemli rolü olduğunu ileri sürmüştür. Bilişsel gelişimin kaynağı, kişisel psikolojik süreçlerden önce, insanlar ve kültür arasındaki etkileşimdir. Çocuk problem çözerken, kendi bilgilerini kullanabileceği gibi, akranlarının veya yetişkinlerin yardımından da yararlanabilir. Bu doğrultuda yazılım geliştirme alanında endüstriyel bir yaklaşım olarak ortaya çıkmış olan eşli programlamanın (pair programming; Beck, 2000), programlama eğitimi için etkili bir yöntem olup olmayacağı, araştırmaya değer bir konudur.

Bilişsel kuram, eşli programlamanın tek başına programlamaya göre neden daha etkili olabileceğinin açıklanmasına yardımcı olabilir. Kaliforniya Üniversitesinde yapılan bir araştırmada Flor ve Hutchins (1991), birlikte çalışan iki deneyimli programcının çalışmalarını kayıt altına almış ve incelemişlerdir. Bu çalışmada programcıların sözel ve sözel olmayan davranışları bilinen dağıtık biliş kuramları ile ilişkilendirilmiştir. Bunlardan biri de geniş alternatifler içinden seçmedir. Bu yaklaşıma göre birden fazla kişinin katıldığı bir göreve daha fazla çeşitlilikte planların olmasının en az üç nedeni vardır. Bunlar;

- Göreve ilişkin bireylerin farklı ön deneyimleri olabilir.
- Görevle ilgili bilgiyle farklı bağlantıları olabilir.
- Kişiler kendi fonksiyonel rolleri sayesinde görevle ilgili farklı problemlerin karşısında dururlar.

Çalışmanın önemli bir sonucu olarak, programcılar anlaşılmadıklarında aralarında açıkça bir hareket planını müzakere ederler. Bu tür durumlarda bir programcının tek başına yapabileceğinden çok fazla sayıda alternatifi araştırırlar. Bu kötü bir plana odaklanma ihtimallerini azaltır (Flor ve Hutchins, 1991). Bu bulgular, eşli programlamanın, bilgi işlemsel düşünmede önemli olan “en iyi çözümün seçilmesi” için uygun bir yöntem olabileceğini akla getirmektedir.

Eşli programlama endüstride ve eğitim alanında benzer şekillerde tanımlanmıştır. Williams ve Kessler’e (2000) göre eşli programlama iki programcının bir bilgisayarla yan yana çalıştığı, aynı tasarım, algoritma, kod ya da test üzerine sürekli işbirliği yaptığı programlama yöntemidir. Beck’e (2000) göre ise eşli programlama, iki kişinin tek klavye, tek fare ve tek ekran kullanarak programlama yaptığı bir tekniktir.

Eşli programlamayı öğretim alanına aktarırken, endüstride eşli programlamanın nasıl tanımlandığı ve güncel öğretim kuramları ve yöntemleri ile nasıl ilişkilendirilebileceği araştırılmalıdır. Jehng’e (1997) göre, sınıfta akranlarla etkileşim üç temel şekilde gerçekleşir:

- Az yetenekli olan daha yetenekli tarafından yönlendirilir.
- Öğrenciler görevin değişik bölümleri üzerinde çalışırlar.
- Öğrenciler görevin genel olarak her bölümünde ortak çalışırlar.

Eşli programlama aslında işbirlikli çalışmayı hedeflemektedir. Jehng (1997) işbirliğinin daha zengin sosyal etkileşim ve daha verimli çalışma imkânı sağlayacağını vurgulamıştır.

Nosek (1998) Temple Üniversitesinde 15 tam zamanlı ve deneyimli programcı ile bir deney yapmıştır. Deneklerin beşi tek başına geri kalanı da 5 ikili grup oluşturmuştur. Deneklerin zorlu bir problemle 45 dakika boyunca uğraşmaları sağlanmıştır. Çalışma sonunda eşli çalışan programcılar problem çözme sürecinden hoşlanmışlardır ve tek çalışanlara göre çözümlerine daha fazla

güvenmektedirler. Sonuçlar incelendiğinde eşli programlama yapan katılımcıların tek başına çalışanlara göre %40 daha hızlı olduğu ve daha etkili ve verimli kod yazdıkları ortaya konmuştur. Çalışmaya başlamadan önce programcılar eşli programlamaya şüpheci ve olumsuz bakmalarına rağmen çalışma sonunda eşli çalışan katılımcılar problem çözme sürecinden daha fazla hoşlanmış ve daha iyi performans sergilemişlerdir.

Nawrocki ve Wojciechowski (2001) eşli programlamanın etkililiğini araştırmak için Poznan Teknik Üniversitesinde 21 öğrencinin katılımı ile bir araştırma yürütmüştür. Bütün katılımcılar üniversitenin dördüncü sınıfında programlama konusunda tecrübeli öğrencilerdir. Katılımcılardan 10'u 5 ayrı ikili oluşturmuş ve diğerleri bireysel çalışmıştır. Bütün öğrencilere aynı programlama görevi verilmiş ve öğrenciler gözlenmiştir. Deney sonucunda eşli programlamanın daha önceki çalışmalarda belirtildiği kadar etkili olmadığı fakat bireysel çalışmaya göre zaman ve kod kalitesi açısından daha verimli olduğu bulunmuştur.

Thomas, Ratcliffe ve Robertson (2003) önceden programlama deneyimi olan üniversite öğrencileri ile eşli programlama üzerine bir çalışma yürütmüştür. Çalışmada öğrencilere uygulanan anket sonuçlarına göre öğrenciler kendilerine güvenen ve güvenmeyen olarak sınıflandırılmıştır. Sonraki aşamada öğrenciler benzer ve farklı öz güvene sahip ikililere ayrılmış ve programlama görevleri verilmiştir. Sonuçlar incelendiğinde;

- Öğrencilerin hepsi eşli programlamadan hoşlanmış ve eşli programlamanın kendilerinin daha iyi programlama yapmalarını sağladığını ifade etmişlerdir.
- Öz güveni düşük olan öğrenciler eşli programlamadan daha çok hoşlanmışlardır.
- Öz güveni yüksek olan öğrenciler eşli programlamadan daha az hoşlanmışlardır.
- Benzer öz güven düzeyine sahip öğrencilerden oluşan eşler daha başarılı olmuşlardır.

McDowell, Werner, Bullock ve Fernald (2006) Kaliforniya - Santa Cruz Üniversitesinde eşli programlamanın etkilerini araştırmak için bir çalışma yürütmüşlerdir. Programlamaya giriş dersini alan ve bilgisayar bilimleri ile ilgili bölümlerden olan 554 öğrenci çalışmaya katılmıştır. Veriler üç öğretim yarıyılında toplanmıştır. 2001 Bahar dönemindeki öğrenciler bütün programlama görevlerini tek başlarına yapmışlardır. Diğer iki dönemde ise öğrencilerin ikili gruplar halinde eşli çalışmaları istenmiştir. Çalışma sonunda eşli çalışan öğrencilerin istatistiksel olarak dersten geçme oranlarının daha yüksek olduğu görülmüştür. Araştırmacılar bu durumun öğrencilerin daha sonra aldığı programlama derslerinde tek başlarına çalışmaları istense de devam ettiğini görmüştür.

Konuya yönelik olarak yapılan araştırmaların sonuçları eşli programlamanın endüstride ve üniversite düzeyinde programlama öğrenme başarısını, kod kalitesini, bireylerin yaptıkları işten hoşlanma düzeyini artırma yönünde ciddi bir potansiyeli olabileceğine işaret etmektedir.

Öğrencilerin bir eşle, bilgisayar ile çalışırken nasıl problem çözdüklerine dair çok az şey bilinmektedir. Bu konuda çocuklar üzerinde yapılmış az sayıda çalışma mevcuttur (Werner ve Denning, 2009). Çocukların arkadaşları ile birlikte bilgisayar oyunu oynaması üzerine yapılan çalışmalar, aralarındaki etkili bir işbirliği ile performanslarında, özgüvenlerinde ve oyuna devam etmeye yönelik motivasyonlarında yükselme olduğunu göstermektedir (Inkpen, Booth, Klawe ve Upitis, 1995).

Özetle; son dönemde gerek uygulama gerekse akademik düzlemde programlama eğitimi, oldukça güncel bir konu haline gelmiştir. Bu çerçevede öğrencilere programlama becerileri kazandırmada kullanılabilecek yaklaşım, yöntem, teknik ya da etkinlikler de tartışma ve araştırmalara konu edilmektedir. Bunlardan birisi olarak “eşli programlama” üzerine yapılmış çeşitli araştırmalar bulunmakta ve bu araştırmaların sonuçları, zaman zaman birbirleri ile çelişse de belli durumlarda eşli programlamanın öğrenme çıktıları üzerine olumlu etkileri olabileceğine işaret etmektedir. Bu araştırmada mantıksal düşünme becerileri kontrol değişkeni olarak kullanılmak suretiyle; eşli programlamanın ortaokul öğrencilerinin programlama başarılarına etkisi test edilmiştir.

YÖNTEM

Bu araştırmada sontest kontrol gruplu deneysel desen kullanılmıştır. Araştırmanın bağımsız değişkeni öğrencilerin tek başına ya da eşli çalışma durumudur. Bu doğrultuda deney grubundaki öğrenciler eşli çalışmış, kontrol grubundaki öğrenciler ise bireysel çalışmıştır. Her iki gruba da aynı eğitim verilip sadece bireysel ya da eşli çalışma durumunun etkisi araştırılmıştır.

Araştırmanın bağımlı değişkeni ise öğrencilerin programlama başarılarıdır. Öğrencilerin bireysel ya da eşli çalışma durumunun programlama başarılarına etkisi araştırılmıştır.

Araştırmada sonuçlar üzerindeki yanlılığı azaltmak için mantıksal düşünme beceri puanları çalışma öncesinde belirlenerek kontrol değişkeni olarak kullanılmıştır.

Çalışma Grubu

Bu araştırmada çalışma grubu Ankara'nın Çankaya ilçesindeki bir ortaokulda öğrenim gören 36 altıncı sınıf öğrencisinden oluşmaktadır. Çalışma haftada iki saat olan Bilişim Teknolojileri dersinde 5 hafta boyunca yürütülmüştür. Denekler kontrol ve deney gruplarına yansız olarak atanmıştır. Çalışmadaki her grup 18

öğrenciden oluşmuştur. Deney grubunda öğrenciler her bilgisayarda eşli olarak çalışmış, kontrol grubunda ise tek başlarına çalışmıştır.

Öğretim Materyali ve Uygulama

Bu çalışmada özel bir öğretim materyali kullanılmamış, ancak hem deney hem kontrol grubunda aynı materyal kullanılmıştır. Ders öğretmen tarafından anlatılmış ve uygulama haftada iki ders saati olarak gerçekleştirilmiştir.

Uygulama öncesi öğrenciler eşli programlamanın işleyişi hakkında bilgilendirilmiştir. Eşli çalışan öğrencilerden biri klavye ve fareyi kullanarak bilgisayarın kontrolünü almış, diğeri ise gözlemci ve yönlendirici olarak yanında yer almıştır. Eşler istediklerinde rollerini değiştirebilmiştir.

Uygulamanın ilk haftasında Scratch programının genel özellikleri tanıtılmış, karakterin kostümünü değiştirme ve karakter hareket ettirme konuları işlenmiştir. Programlama görevi olarak kostüm değiştirerek karakteri gerçekçi yürütmeleri istenmiştir.

İkinci hafta çizgi çizme, karakteri döndürme ve döngü konuları işlenmiştir. Programlama görevi olarak ise öğrencilerden çeşitli geometrik şekilleri (kare, üçgen daire, dikdörtgen) kod yazarak çizdirmeleri istenmiştir.

Üçüncü hafta fonksiyon oluşturma konusu işlenmiştir. Programlama görevi olarak öğrencilerden öncelikle bir önceki hafta oluşturdıkları geometrik şekil çizme kodlarını fonksiyona dönüştürmeleri istenmiş, bir sonraki görevde ise şekillerin kenar uzunluklarının dinamik olarak atanabilmesi için fonksiyonları parametre alır hale getirmeleri istenmiştir.

Dördüncü hafta klavye ve fareden girdi alma ve koşullar (eğer - değilse) konuları işlenmiştir. Programlama görevi olarak öğrencilerden bir önceki hafta fonksiyon olarak tanımladıkları geometrik şekil çizme programını kullanıcının seçimine göre şekil çizen programa dönüştürmeleri istenmiştir.

Uygulamanın son haftasında birden fazla karakter kullanma, karakter kodlarının haberleştirilmesi, karakter kodları arasındaki akışın sıralanması ve kodun bekletilmesi konuları işlenmiştir. Programlama görevi olarak öğrencilerden karşılıklı iki karakteri konuşturmaları ve bu konuşmanın görsel olarak uygun hızda ve karakterlerin diyaloglarının doğru sıralamada olmasının sağlanması istenmiştir.

Veri Toplama Araçları

Öğrencilerin çalışma öncesi mantıksal düşünme becerileri belirlenerek kontrol değişkeni olarak kullanılmıştır. Bu sayede araştırma sonuçlarında öğrencilerin mevcut mantıksal düşünme becerilerinden kaynaklanabilecek yanlışlık azaltılmıştır. Öğrencilerin mantıksal düşünme becerilerini belirleyebilmek için Roadrangka, Yeany ve Padilla (1983) tarafından geliştirilen ve Türkçeye çevirisi

Aksu, Berberoğlu, Martin ve Paykoç (1990) tarafından yapılan Mantıksal Düşünme Grup Testi (MDGT) kullanılmıştır. MDGT araştırmanın ilk haftasında konu anlatımı başlamadan önce uygulanmıştır. MDGT uygulanan hafta ders işlenmemiştir. Sonrasında uygulama 5 hafta sürmüştür.

Öğrencilerin programlama başarılarını belirleyebilmek için başarı testi geliştirilmiştir. Başarı testi öğretim süreci tamamlandıktan sonra uygulanmıştır.

Mantıksal Düşünme Grup Testi (MDGT): Toplam 21 maddeden oluşan bu test, daha önce bu alanda geliştirilmiş olan farklı muhakeme yeteneklerini ölçen testlerden (Lawson's Classroom Test of Formal Operation 1978; Burney, 1974; Akney ve Joyce 1974 ve Longeol 1968; akt: Aksu, Berberoğlu, Martin ve Paykoç, 1990) geçerliği ve güvenilirliği yüksek maddeler seçilerek oluşturulmuştur.

Roadrangka, Yeany ve Padilla (1983) testin, 6.sınıftan üniversiteye kadar öğrencilerin mantıksal düşünme yeteneklerini ölçebilecek geçerlik ve güvenilirliğe sahip olduğunu ifade etmişlerdir. Testin güvenilirliği (Cronbach's alpha) $\alpha = 0.85$ olarak bulunmuştur. Altıncı sınıftan lise son sınıfa kadar, ayrıca lisans ve lisansüstü öğrencilere uygulanan testin sonuçları, bu testin soyut muhakeme yeteneğini ölçer nitelikte olduğunu göstermiştir.

Mantıksal Düşünme Grup Testi, 'Türkçe' ye Aksu, Berberoğlu ve Paykoç tarafından çevrilmiş ve testin güvenirlik katsayısı ITEMAN programı kullanılarak 0.88 olarak bulunmuştur (Aksu, Berberoğlu, Martin ve Paykoç, 1990). Yapılan çalışmalara göre test, Türkiye'de orta öğretim düzeyinden itibaren mantıklı düşünmeyi ölçebilecek niteliktedir.

Başarı Testi: Araştırmada öğrencilerin başarı seviyelerini belirlemek için çoktan seçmeli başarı testi uygulanmıştır. Bu kapsamda çoktan seçmeli türde ve hazırlanan belirtke tablosu çerçevesinde 12 soru hazırlanmıştır.

Hazırlanan testin uzman görüşleri, hazırlanan belirtke tablosu, test soruları ve uzman görüş formundan oluşan bir belge Ankara il sınırları içinde görev yapan üç bilgisayar öğretmenine e-posta ile gönderilerek alınmıştır. Alınan Uzman görüşleri sonucunda soruların uygun olduğuna karar verilmiştir.

Uzman görüşleri alındıktan sonra ön uygulama için araştırmaya katılmayan, bir önceki dönem Scratch eğitimi almış 65 altıncı sınıf öğrencisine test uygulanmış ve madde analizleri yapılmıştır. Başarı testinin madde ayırt ediciliğine ilişkin gerçekleştirilen alt ve üst %27'lik gruplar için bağımsız t-testi sonuçları ve madde (madde-toplam ve madde-kalan) analizleri Çizelge 1'de gösterilmiştir.

Çizelge 1. Başarı testinin madde analizi sonuçları

Madde No	Madde Toplam	Madde Kalan	Alt %27 ve Üst % 27	
	r	r	t	p
m1	.087	.074	1.000	.331
m2	.226	.201	3.688	.002
m3	.434	.497	5.464	.000
m4	.393	.346	7.331	.000
m5	-.201	.176	.329	.744
m6	.508	.456	5.831	.000
m7	.490	.276	7.714	.000
m8	.282	.311	3.431	.002
m9	.332	.356	3.554	.001
m10	.339	.218	4.605	.000
m11	.038	.085	1.000	.331
m12	.160	.124	4.043	.000

$$n = 65, n1(Alt \%27) = 18, n2(Üst \%27) = 18, p < .01$$

Bağımsız iki grubun karşılaştırıldığı t-testinin hesaplanmasında gruplar arasında varyans farkının olup olmadığına dikkat edilmesi gerekmektedir (Büyüköztürk, 2010). Bu sebeple, bu çalışmada t-testi hesaplamasında öncelikle varyansların eşit olup olmadığı Levene testi ile değerlendirilmiştir. Levene testi anlamlılık düzeyi .05 değerinin altındaysa buradan iki grubun varyanslarının farklı olduğu anlaşılmaktadır (Büyüköztürk, 2010). Farklı varyanslar olduğunda SPSS veri sayfasında “Equal variances not assumed (Varyansların eşitliğinin sağlanmadığı)” satırındaki değer dikkate alınmıştır.

Yukarıda Çizelge 1’deki bulgular dikkate alınarak maddeler değerlendirilmiştir. Öncelikle ayırt ediciliği düşük m5, m1 ve m11 maddeleri sırası ile testten çıkarılarak analizler tekrarlanmıştır. Bu aşamada m12 maddesinin ayırt ediciliği yüksek olduğu fakat madde toplam ve madde kalan korelasyonlarının düşük olduğu görülmüş ve testten çıkarılmasına karar verilmiştir.

Sonuçta testte 8 soru kalmıştır. Testin iç güvenilirliğini ölçmek için KR-20 (Kuder-Richardson) değeri hesaplanmıştır. Hesaplanan güvenilirlik katsayısının 0.70 ve daha yüksek olması test puanlarının güvenilirliği için genel olarak yeterli görülmektedir (Büyüköztürk, 2010). Yapılan analiz sonucu testin iç güvenilirlik

katsayısı 0.71 olarak hesaplanmıştır. Elde edilen sonuçlar, testin yeterli bir güvenilirliğe sahip olduğunu göstermiştir.

Verilerin Çözümlemesi ve Yorumlanması

Araştırma kapsamında toplanan verilerin analizlerinde ilk olarak parametrik analiz yöntemlerinin kullanılmasının uygun olup olmadığının tespit edilmesi amacıyla ortalama, ortanca, mod, standart sapma, basıklık ve çarpıklık değerleri hesaplaması ve Shapiro-Wilks Normallik testi yapılarak elde edilen dağılımda normallik varsayımının karşılanıp karşılanmadığı belirlenmiştir.

Öğrencilerin eşli ve ya tek çalışma durumlarına göre tüm alt gruplar için ortak ve bağımlı değişkenlerin normallikleri test edilmiştir. Grup büyüklükleri $n < 50$ olduğu için Shapiro-Wilks testi sonuçlarına bakılmıştır. Test sonucu $p > .05$ olan gruplar normal kabul edilmiştir. Diğer gruplar için çarpıklık katsayısı incelenmiştir. Grupların çarpıklık katsayısını +1, -1 aralığında (Büyüköztürk, 2010) olduğu ya da bu aralıktan çok uzaklaşmadığı ve grup büyüklüklerinin yeterli olduğu ($n > 15$) (Green, Salkind ve Akey, 1997; akt: Büyüköztürk, 1998) görüldüğünden normallik varsayımı kabul edilmiştir.

Programlama başarısı değişkeninin son test olarak uygulanmasından ve MDGT puanının kontrol değişkeni olarak kullanılmasından dolayı tek faktörlü ANCOVA testi (kovaryans analizi) uygulanması öngörülmüştür. Bu analiz için öncelikle gerekli olan varsayımlar olan (Büyüköztürk, 2010); bağımlı değişkenle kontrol değişkeni arasındaki doğrusal ilişki, varyansların homojenliği için Levene Testi ve gruplar içi regresyon eğimleri (regresyon katsayıları) eşitliği test edilmiştir.

Bağımlı değişkenle kontrol değişkeni arasında doğrusal ilişki olup olmadığını incelemek amacı ile saçılma diyagramı incelendiğinde bağımlı değişken (başarı puanı) ve ortak değişken (mantık puanı) arasında doğrusal bir ilişki olduğu görülmüştür. Bağımlı değişkenin (başarı puanları) her bir grup için varyanslarının homojen olması varsayımını test etmek için Levene testi sonuçlarına bakılmıştır. Sonuç incelendiğinde varyansların homojen olduğu görülmüştür, $F(1, 34) = 0.800$, $p = .377$ ($p > .05$). Regresyon doğrularının eğiminin eşitliği için ANOVA sonuçlarına bakılmıştır. Başarı üzerinde grup x MDGT ortak etkisinin anlamsız olduğu görülmektedir, $[F(1, 32) = 0.009, p = .926 (p > .05)]$. Regresyon doğrularının eğimi eşittir.

ANCOVA testi için bütün varsayımlar sağlanmıştır. ANCOVA testinin sonuçları Tip-1 hata yapma ihtimalini azaltmak için .01 manidarlık düzeyinde yordanmıştır.

BULGULAR

Yapılan ölçümlerde öğrencilerin MDGT puanlarının aritmetik ortalaması; tekli çalışan öğrenciler için 2.33 iken eşli olarak çalışan öğrenciler de 3.22 olduğu

görülmüştür. İki grup kıyaslandığında MDGT puanlarının aritmetik ortalamalarının eşli çalışan grup lehine yüksek olduğu gözlenmektedir. Bu durum eşli çalışan grubun çalışma başında mantıksal düşünme becerileri açısından avantajlı olduğu şeklinde yorumlanmıştır. Çizelge 2’de öğrencilerin MDGT puanlarının betimsel istatistikleri verilmiştir.

Çizelge 2. MDGT ölçümlerinin betimsel istatistikleri

Öğrenme Etkinliği	N	Ortalama	SS
Tekli	18	2.33	0.396
Eşli	18	3.22	0.689

Aşağıda Çizelge 3’te öğrencilerin başlangıçtaki MDGT puanlarının sonuca etkisi istatistiksel olarak kontrol edildiğinde ortaya çıkan başarı puanlarına ait betimsel istatistikler verilmiştir. Buna göre eşli çalışan öğrencilerin başarı testi puan ortalamaları 7.11, tekli çalışan öğrencilerin başarı testi puan ortalamaları 6.72 olarak hesaplanmıştır. Grupların MDGT puanları istatistiksel olarak kontrol edildiğinde başarı testi ortalamalarında değişimler olduğu görülmektedir. Başarı testi düzeltilmiş ortalama puanları eşli çalışanlar için 7.04, tekli çalışanlar için 6.80 olarak hesaplanmıştır.

Çizelge 3. MDGT puanları kontrol edildiğinde başarı puanlarına ait betimsel istatistikler

Öğretim tekniği	N	Ortalama	SS	Düzeltilmiş ortalama	Düzeltilmiş SS
Tekli	5	6.72	1.36	6.80	0.275
Eşli	5	7.11	1.02	7.04	0.275

Düzeltilmiş başarı testi ortalama puanları arasında eşli çalışan grup lehine gözlenen farkın anlamlı olup olmadığına ilişkin yapılan ANCOVA sonuçları Çizelge 4’de verilmiştir.

Çizelge 4. Sayısal yetenek testine göre düzeltilmiş başarı testi ANCOVA sonuçları.

Varyansın Kaynağı	Kareler		Kareler		p	Kısmi eta-kare
	Toplamı	sd	Ortalaması	F		
MDGT	5.377	1	5.377	4.031	.053	0.109
Grup (Eşli-Tek)	0.502	1	0.502	0.377	.544	0.011
Hata	44.012	33	1.334			
Toplam	1773.000	36				

Eşli ya da tekli çalışarak ders işleyen öğrencilerin başarı testi puan ortalamalarının MDGT puanlarına göre düzeltilmiş ortalama puanları arasında anlamlı bir fark yoktur, $[F(1, 33) = 0.377, p > .01]$. Öğrencilerin programlama başarılarının eşli ya da tekli çalışmaları ile ilişkisinin olmadığı görülmüştür. Kısmi eta-kare (0.011) sonuçları incelendiğinde de eşli çalışmanın başarı üzerinde tekli çalışmaya göre etkisinin olmadığı görülmüştür.

Bu bulguya dayanarak, eşli çalışan öğrenciler ile tek başına çalışan öğrencilerin programlama başarılarının benzer olduğu söylenebilir. Bu durum, programlama eğitiminde öğrencilerin bir bilgisayar başında tek başına çalışmaları ile iki kişi çalışmaları arasında başarı açısından bir fark yaratmadığı şeklinde yorumlanabilir. Ayrıca bu durum bakış açısına göre bir bilgisayarda iki öğrencinin çalışmasının programlama başarısı açısından bir olumsuzluğa sebep olmadığı şeklinde de yorumlanabilir.

Alanyazına bakıldığında eşli programlama yöntemi, programlama öğretiminde kullanıldığında programlama başarısının arttığını gösteren çok sayıda çalışma bulunmaktadır (Cockburn ve Williams, 2000; Flor ve Hutchins, 1991; Inkpen, Booth, Klawe ve Uptis, 1995; McDowell, Werner, Bullock ve Fernald, 2006; Nosek, 1998; Thomas, Ratcliffe, ve Robertson, 2003; Williams, Wiebe, Yang, Ferzlive Miller, 2002). Alanyazında eşli programlama eğitiminin tek başına çalışmaya göre avantajlı olduğunu ortaya koyan bu çalışmaların aksine bu araştırmada grupların başarı puanları arasında anlamlı fark bulunmamıştır.

Benzer şekilde Nawrocki ve Wojciechowski (2001)'de eşli programlamanın alanyazında belirtildiği kadar etkili olmadığını ortaya koymuştur. Bununla birlikte eşli programlamanın avantajlarını ortaya koyan tüm çalışmalar lisans seviyesi ve üstü ya da profesyonel programcılar ile yapılmıştır. Bu durum dikkate alındığında araştırmanın bulguları ortaokul seviyesinde eşli programlamanın etkilerini ortaya koyan az sayıda çalışmadan biridir. Bu çalışmada denekler arasında fark

olmamasının nedeni ortaokul seviyesindeki öğrencilerin birlikte çalışma kültürlerinin yetişkinlerden farklı olması olarak yorumlanmıştır. Yetişkinlerin eşli çalışmada daha iyi iş paylaşımı yapıp daha sistemli çalışabileceği düşünülebilir. Ortaokul seviyesindeki öğrencilerde ise eşli çalışmada iş paylaşımında çıkan sorunlar ve işbirliği kültürünün eksikliğinin bu sonuca sebep olduğu düşünülebilir. Bu bulgulara farklı bir bakış açısı ile baktığımızda ise bilgisayar başına iki öğrenci düşmesinin, bilgisayar başına tek öğrenci düşmesi durumuna göre programlama başarısında bir olumsuzluğa sebep olmayacağı şeklinde de yorumlanabilir.

SONUÇ VE ÖNERİLER

Araştırmada elde edilen bulguların ortaya koyduğu sonuçlara göre, eşli programlama eğitiminin tek başına çalışmaya göre öğrencilerin programlama başarıları üzerinde anlamlı farka neden olmadığı tespit edilmiştir. Programlama öğretiminde öğrencilerin bir bilgisayar başında eşli ya da tekli çalışmalarının benzer etkilerinin olduğu görülmüştür. Bununla birlikte alanyazında eşli programlamanın, tek başına programlamaya göre daha etkili olduğunu ortaya koyan çok sayıda çalışma vardır. Alanyazındaki çalışmaların aksine bu çalışmada elde edilen bulgu Nawrocki ve Wojciechowski (2001)'nin ortaya koydukları, eşli programlamanın alanyazında belirtildiği kadar etkili olmadığı sonucu ile benzerlik göstermektedir.

Thomas, Ratcliffe ve Robertson (2003) benzer öz güven düzeyine sahip öğrencilerden oluşan eşler ile eşli programlamanın daha etkili olduğunu, eşler arasındaki özgüven farkının yüksek olduğu durumlarda bu etkinin azaldığını ortaya koymuştur. Eşli programlamanın etkisinin eşler arasındaki bireysel farklılıklardan etkilenebileceği görülmektedir. Eşli programlamada eşlere bağlı bireysel farklılıkların, grup çalışmasına etkisinin daha detaylı incelenmesi gerekmektedir. Bu açıdan eşli programlamadan beklentiler konusunda daha temkinli olunması gerektiği söylenebilir.

Ülkemizde okullardaki bilgisayar sınıflarının yetersizlikleri sonuçları farklı bir bakış açısı ile değerlendirme ihtiyacı doğurmaktadır. Okullarda bilişim teknolojileri sınıflarında sıklıkla bilgisayar başına birden fazla öğrenci düştüğü ve her öğrenciye ayrı bilgisayar düşecek şekilde bu durumun geliştirilmesi önerilmektedir (Durdukoca ve Arıbaş, 2011; Şerefoğlu Henkoğlu ve Yıldırım, 2012; Gülcü, Aydın ve Aydın, 2013). Bilişim teknolojileri sınıflarındaki öğrenci başına düşen bilgisayar sayısındaki yetersizlikler yadsınamaz bir dezavantaj olmakla birlikte, bu çalışma bize bilgisayar başında iki öğrenci çalışmasının, tek öğrenci çalışmasına göre programlama eğitiminde başarı açısından olumsuz bir etkisi olmadığını göstermiştir. Programlama eğitimi bağlamında eşli programlama uygulaması, bilişim teknolojileri sınıflarında bilgisayar sayısının yetersiz olduğu durumlar için önerilebilir.

Araştırma öğrencileri tek bir bilgisayar başında ikili olarak gruplamanın negatif bir etkisinin olmadığını ortaya koymuştur. Buradan alınan cesaretle ileriye dönük olarak eşli programlama eğitiminin okul ortamında çeşitli boyutlarda derinlemesine çalışılması önerilebilir. Bu boyutların bazıları aşağıda listelenmiştir.

- Öğrenciler eşleştirilirken bireysel farklılıkları (konuya ilgi, motivasyon, öz yeterlik vb.) dikkate alınarak, bu farkların etkisi derinlemesine analiz edilebilir.
- Eşleştirilirken öğrencilerin programlama ön bilgi seviyelerinin durumunun etkisi derinlemesine analiz edilebilir. Yüksek bilgi seviyesi ile yüksek, düşükle düşük, yüksek ile düşük bilgi seviyelerinden öğrenciler eşleştirilerek sonuçları değerlendirilebilir.
- Programlama becerisi ile ilişkili olduğu alanyazında ifade edilen problem çözme becerisi, matematik başarısı gibi farklı değişkenler kovaryans olarak kullanılarak eşli programlama uygulamasının programlama öğrenmeye etkisi incelenebilir.

KAYNAKLAR

Abazaoğlu, Akpınar, Y. ve Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *İlköğretim Online*, 13(1), 1-4.

Aksu, M., Berberoğlu, G., Martın, G. ve Paykoç, F. (1990). Problem çözme becerilerinin ölçülmesi: GALT'ın Türkiye'deki uygulamasına ilişkin bir ön çalışma. *Ankara: I.Ulusal Eğitim Bilimleri Kongresi*.

Beck, K. (2000). *Extreme programming explained: Embrace change*. Addison-Wesley Professional.

Bozdoğan, A., (2007). Fen Bilgisi Öğretiminde Çalışma Yaprakları ile Öğretimin Öğrencilerin Fen Bilgisi Tutumuna ve Mantıksal Düşünme Becerilerine Etkisi (Yayınlanmamış Yüksek Lisans Tezi). Çukurova Üniversitesi.

Büyüköztürk, Ş. (2010). *Sosyal Bilimler İçin Veri Analizi El Kitabı*. Pegem Yayınevi, 12. Baskı.

Büyüköztürk, Ş. (1998). Kovaryans Analizi: Varyans Analizi İle Karşılaştırmalı Bir İnceleme. *Ankara Üniversitesi Eğitim Bilimleri Fakültesi Dergisi*, 31(1), 91-105.

Calder, N. (2010). Using scratch: An integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom*, 15(4), 9-14.

Cockburn, A., and Williams, L. (2000). The Costs And Benefits Of Pair Programming. *Extreme programming examined*, 223-247.

Durdukoca, Ş. F. ve Arıbaş, S. (2011). İlköğretim seçmeli bilişim teknolojileri dersi 5. basamak öğretim programının öğretmen görüşleri doğrultusunda değerlendirilmesi, (Malatya İli Örneği). *Yüzüncü Yıl Üniversitesi Eğitim Fakültesi Dergisi*, 8(1), 140-168.

Gorman, H., and Bourne, L. E. (1983). Learning to think by learning LOGO: Rule learning in third-grade computer programmers. *Bulletin of the Psychonomic Society*, 21(3), 165-167.

- Gülcü, A., Aydın, S. ve Aydın, Ş. (2013). İlköğretim okullarında bilişim teknolojileri dersi yeni öğretim programının öğretmen görüşlerine göre değerlendirilmesi. *Karadeniz Sosyal Bilimler Dergisi*, 5(8), 73-92.
- Flor, N. V., and Hutchins, E. L. (1991). A case study of team programming during perfective software maintenance. In *Empirical studies of programmers: Fourth workshop* (p. 36). Intellect Books.
- Hour Of Code. (2015). 11 Mart 2015 tarihinde <http://hourofcode.com/tr> adresinden erişildi.
- Inkpen, K., Booth, K. S., Klawe, M., and Uptis, R. (1995, October). Playing together beats playing apart, especially for girls. In *The first international conference on Computer support for collaborative learning* (pp. 177-181). L. Erlbaum Associates Inc.
- Jehng, J. C. J. (1997). The psycho-social processes and cognitive effects of peer-based collaborative interactions with computers. *Journal of Educational Computing Research*, 17(1), 19-46.
- Kafai, Y. B., and Burke, Q. (2014). Computer programming goes back to school. *Education Week*, 61-65.
- Gülbahar, Y., and Kalelioğlu, F. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education-An International Journal*, 13(1), 33-50.
- McDowell, C., Werner, L., Bullock, H. E., and Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Communications of the ACM*, 49(8), 90-95.
- Nawrocki, J., and Wojciechowski, A. (2001). Experimental evaluation of pair programming. *European Software Control and Metrics (Eseom)*, 99-101.
- Nosek, J. T. (1998). The case for collaborative programming. *Communications of the ACM*, 41(3), 105-108.
- Roadrangka, V., Yeany, R. H., and Padilla, M. J. (1983, April). The construction of a group assessment of logical thinking (GALT). In *th annual meeting of the National Association for Research in Science Teaching*. Dallas, Texas, April (pp. 5-8).
- Salomon, G. (Ed.). (1993). *Distributed cognitions: Psychological and educational considerations* (Vol. 11, No. 9). Cambridge: Cambridge University Press.
- Şerefoglu Henkoğlu, H , Yıldırım, S . (2012). Türkiye'deki İlköğretim Okullarında Bilgisayar Eğitimi: Kuram ve Uygulamadaki Farklılıklar. *Ankara Üniversitesi Eğitim Bilimleri Fakültesi Dergisi*, 45 (1), 23-62.
- Thomas, L., Ratcliffe, M., and Robertson, A. (2003, February). Code warriors and code-a-phobes: a study in attitude and pair programming. In *ACM SIGCSE Bulletin* (Vol. 35, No. 1, pp. 363-367). ACM.
- Tobin, K. G., and Capie, W. (1982). Relationships between formal reasoning ability, locus of control, academic engagement and integrated process skill achievement. *Journal of Research in Science Teaching*, 19(2), 113-121.
- Vygotsky, L. S. (1980). *Mind in society: The development of higher psychological processes*. Harvard University Press.

- Werner, L., and Denning, J. (2009). Pair programming in middle school: What does it look like? *Journal of Research on Technology in Education*, 42(1), 29-49.
- Williams, L. A., and Kessler, R. R. (2000). All I really need to know about pair programming I learned in kindergarten. *Communications of the ACM*, 43(5), 108-114.
- Williams, L., Wiebe, E., Yang, K., Ferzli, M., and Miller, C. (2002). In support of pair programming in the introductory computer science course. *Computer Science Education*, 12(3), 197-212.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725.
- Yoo, S., Cha, S., and Lee, W. (2007). Improving K-12's Logical Thinking Abilities using Educational Programming Language "Dolittle". *Wseas Transactions On Advances In Engineering Education*, 4, 251-258.

EXTENDED ABSTRACT

The necessity of programming education for children has now become more pronounced. To this end, international organizations have been introduced to teach programming to children. The issue highlighted in this regard for children is problem solving methods and processes. From this point of view, the concept of computational thinking was introduced by associating programming skills with problem solving.

The process of computational thinking is to choose the appropriate path to define the problem, to create a model of the problem and then to choose the best method to solve the problem (Wing, 2008). Although computers have been in schools for a long time, computational thinking has not been part of the curriculum. Teaching word processing or presentation software to students does not allow them to think more creatively and deeply (Kafai & Burke, 2014). In short, students should learn to think more systematically to produce solutions to all types of problems.

Researches have shown that computer science education helps to develop problem-solving skills through logical thinking. (Yoo, Cha & Lee, 2007). Since programming is a problem solving activity, it can be said that programming education is related to logical thinking.

Success in programming education depends not only on cognitive variables but also on educational method. Salomon (1993) stated that information was socially constructed by means of shared efforts in the direction of shared goals or through dialogues in the direction of the differences in people's perspectives. Vygotsky (1980) argued that the social environment in which the child is involved plays an important role in cognitive development. While solving the problem, the child can use his / her own knowledge and also benefit from the help of his / her peers or adults. Pair programming (Beck, 2000), which has emerged as an industrial approach in the field of software development, can be an effective method for programming education.

The results of the researches conducted on the subject shows that pair programming has a significant potential to increase the learning success of programming, the quality of the code, and the level of enjoyment of the individuals for the industry and the university level.

In this study, secondary school students were given programming training by using Scratch software. When the logical thinking skill is controlled, the effect of pair programming activities on the success of the programming constitutes the problem of this research.

In this research, experimental design with posttest control group was used. The independent variable of the study is the individual or paired working status of

the students. The dependent variable is the students' programming success. The effect of individual or paired working status on the success of the students was investigated. In order to decrease the bias on the results, logical thinking skill scores were determined before the study and used as control variable.

The study group is 36 sixth grade students studying in a secondary school in Ankara. The study was conducted for five weeks in the Information Technologies course, which is two hours per week. In order to determine the students' logical thinking skills, Group Assessment of Logical Thinking (GALT) developed by Roadrangka, Yeany and Padilla (1983) and translated into Turkish by Aksu, Berberoğlu, Martin and Paykoç (1990) was used. Programming achievement test was developed in order to determine the students' programming success. The test was applied at the end of the teaching process.

Since the programming success variable was applied as a final test and the MDGT score was used as the control variable, it was proposed that the single factor ANCOVA test should be applied. All the assumptions for the ANCOVA test were met.

When the results were examined, there was no significant difference between the mean scores of the achievement test scores of the students who worked in a paired or a single according to the GALT scores., $F(1, 33) = 0.377, p > .01$. It was seen that the students' programming success was not related to paired or single working. When the results of partial eta-square (0.011) were examined, it was seen that the paired working had no effect on the programming success.

Looking at the literature, there are many studies showing that the programming success increases when the pair programming method is used in programming instruction (Cockburn & Williams, 2000; Flor & Hutchins, 1991; Inkpen, Booth, Klawe & Uptis, 1995; McDowell, Werner, Bullock & Fernald, 2006; Nosek, 1998; Thomas, Ratchiffe, & Robertson, 2003; Williams, Wiebe, Yang, Ferzli & Miller, 2002). Contrary to the literature, there was no significant difference between the achievement scores of the groups in this study. Similarly, Nawrocki and Wojciechowski (2001) also demonstrated that pair programming is not as effective as it is stated in the literature.

However, most of the studies that demonstrate the advantages of pair programming have been done with bachelor level or higher or professional programmers. The findings of the study are one of the few studies that show the effects of the pair programming at the secondary school level.

The reason why there was no difference between the subjects in this study was interpreted as that the study cultures of the students in the secondary school were different from the adults. It may be thought that adults can work better in paired work and work more systematically. In secondary school students, it can be thought that the problems of sharing work and lack of cooperation culture cause

this result. When we look at these findings from a different point of view, it can be interpreted that two student per computer will not cause any negation in the success of the programming according to the situation of single student per computer.

YAZAR HAKKINDA

2003 yılında İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği Bölümünden mezun olan Serkan DEMİRÖREN, 2013 yılında Ankara Üniversitesi Eğitim Teknolojisi Programında yüksek lisansını tamamlamıştır. Halen Özel sektörde Proje Yöneticisi olarak çalışma hayatını sürdüren yazar, yazılım uzmanı, sistem mimarı ve proje yöneticisi pozisyonlarında toplam oniki yıllık bir çalışma tecrübesine sahiptir. Yazar aynı zamanda Ankara Üniversitesi Eğitim Teknolojisi Programında doktora eğitimine devam etmektedir. /İletişim Adresi: Gazi Üniversitesi Gölbaşı Yerleşkesi Technopark Binası C Blok Kat : 2 TÜRKSAT A.Ş. Gölbaşı, Ankara, Türkiye. Eposta: demirorens@gmail.com.

ABOUT THE AUTHOR

Serkan DEMİRÖREN graduated from İstanbul Technical University Computer Engineering Department in 2003. In 2013 he completed his master degree at Ankara University in Educational Technology Program. He is still working in private sector as a project manager, and he has a total of twelve years of working experience in software specialist, system architect and project manager positions. He is also a PhD student at Ankara University Educational Technology Program. /Correspondence Address: Gazi University Gölbaşı Campus Technopark Building C Block Floor : 2 TÜRKSAT A.Ş. Gölbaşı, Ankara, Türkiye. Email: demirorens@gmail.com
