ALGORİTMA ÖĞRETİMİNDE ÇÖZÜMLÜ ÖRNEK KULLANIMININ ÖĞRENCİ BAŞARISINA VE BİLİŞSEL YÜKE ETKİLERİ

THE EFFECTS OF USING WORKED EXAMPLE METHOD IN TEACHING ALGORITHM ON STUDENTS' ACHIEVEMENT AND COGNITIVE LOAD

Mustafa TEPGEÇ

Hacettepe Üniversitesi

Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı

Yüksek Lisans Tezi

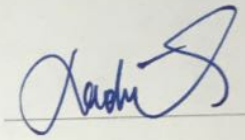
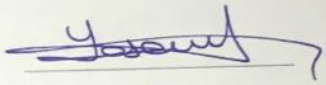
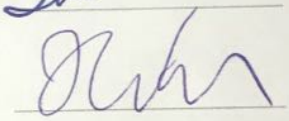
olarak hazırlanmıştır.

2017

KABUL VE ONAY

Eğitim Bilimleri Enstitüsü Müdürlüğü'ne,

Mustafa TEPGEÇ'in hazırladığı "Algoritma Öğretiminde Çözümlü Örnek Kullanımının Öğrenci Başarısına ve Bilişsel Yüke Etkileri" başlıklı bu çalışma jürimiz tarafından **Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı'nda Yüksek Lisans Tezi** olarak kabul edilmiştir.

Başkan	Prof. Dr. Süleyman Sadi SEFEROĞLU	
Üye (Danışman)	Doç. Dr. Yasemin DEMİRASLAN ÇEVİK	
Üye	Doç. Dr. Ebru KILIÇ ÇAKMAK	
Üye	Doç. Dr. Hakan TÜZÜN	
Üye	Doç. Dr. Hasan ÇAKIR	

ONAY

Bu tez Hacettepe Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği'nin ilgili maddeleri uyarınca yukarıdaki jüri üyeleri tarafından 12 / 05 / 2017 tarihinde uygun görülmüş ve Enstitü Yönetim Kurulunca ..... / ..... / ..... tarihinde kabul edilmiştir.

Prof. Dr. Ali Ekber ŞAHİN
Eğitim Bilimleri Enstitüsü Müdürü

YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kağıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe Üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.
Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alınarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

☒ **Tezimin/Raporumun tamamı dünya çapında erişime açılabilir ve bir kısmı veya tamamının fotokopisi alınabilir.**
(Bu seçenekle teziniz arama motorlarında indekslenebilecek, daha sonra tezinizin erişim statüsünün değiştirilmesini talep etseniz ve kütüphane bu talebinizi yerine getirse bile, teziniz arama motorlarının önbelleklerinde kalmaya devam edebilecektir)

☒ **Tezimin/Raporumun 18/05/2019 tarihine kadar erişime açılmasını ve fotokopi alınmasını (İç Kapak, Özet, İçindekiler ve Kaynakça hariç) istemiyorum.**
(Bu sürenin sonunda uzatma için başvuruda bulunmadığım takdirde, tezimin/raporumun tamamı her yerden erişime açılabilir, kaynak gösterilmek şartıyla bir kısmı veya tamamının fotokopisi alınabilir).

☐ **Tezimin/Raporumun ..... tarihine kadar erişime açılmasını istemiyorum ancak kaynak gösterilmek şartıyla bir kısmı veya tamamının fotokopisinin alınmasını onaylıyorum.**

☐**Serbest Seçenek/Yazarın Seçimi:** .....
.....

30.05.2017

(İmza)

Mustafa TEPGEÇ

iii

ETİK BEYANNAMESİ

Hacettepe Üniversitesi Eğitim Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

İmza
Mustafa TEPGEÇ

iv

TEŞEKKÜR

Tez sürecimin her aşamasında önemli dönütleriyle çalışmama katkıda bulunan, iletişim kurmakta kendimi rahat hissettiğim, akademik görüşüme yön veren, “Nasıl bir akademisyen olmalıyım?” sorusunu her düşündüğümde aklıma gelecek olan, çalışkanlığını ve doğruluğunu her zaman örnek alacağım, kıymetini birkaç kelime ile açıklamakta güçlük yaşadığım değerli danışmanım, yol göstericim Doç. Dr. Yasemin DEMİRASLAN ÇEVİK’e teşekkürlerimi ve en içten saygılarımı sunarım.

Tez sürecimde kendisine gösterilmesi beklenen ilgi ve şefkati benimle istemeden de olsa paylaşan Aren Deniz ÇEVİK’e özürlerimi ve teşekkürlerimi sunarım.

Çalışmamın incelenmesinde önemli katkılar sunan Prof. Dr. Süleyman Sadi SEFEROĞLU, Doç. Dr. Hakan TÜZÜN, Doç. Dr. Ebru KILIÇ ÇAKMAK, Doç. Dr. Hasan ÇAKIR hocalarıma teşekkür ederim.

Fikirlerine, enerjisine, ışığına hayran olduğum, en zor zamanlarımda yanımda olan, bana zaman ayıran çok değerli hocam Prof. Dr. Mukaddes ERDEM’e ve neşesiyle, tatlı sert eleştirileriyle kişisel ve akademik gelişimime önemli katkılar sağlayan Prof. Dr. Süleyman Sadi SEFEROĞLU hocama ayrıca teşekkür ederim.

Yüksek lisans ders sürecimde bilimsel araştırmanın temeli olan derslerde bana katkı sağlayan Prof. Dr. Buket AKKOYUNLU, Prof. Dr. Yasemin KOÇAK USLUEL ve Prof. Dr. Halil YURDUGÜL hocalarıma teşekkürlerimi sunarım.

Lisans öğrenimimin başlarından itibaren akademisyen olma hayalimde bana yol gösteren, büyük katkılar sağlayan, benim için özel bir yere sahip olan Yrd. Doç. Dr. Fatih ÖZDİNÇ ve Yrd. Doç. Dr. Esra TELLİ hocalarıma bana olan desteklerinden ve inançlarından ötürü teşekkürlerimi sunarım.

Tez sürecimde konumla ilgili çalışmalarından sıkça istifade ettiğim ve atıf yapmakla yetinmek istemediğim John SWELLER, Jeroen J. G. VAN MERRIENBOER ve Fred PAAS’a teşekkürlerimi sunarım.

Başta uygulama sürecinde kendi çalışması gibi özen gösteren, uygulamanın yapıldığı dersi veren, test geliştirme ve değerlendirme aşamalarında görüşlerini sunan, ilkeli ve idealist Öğr. Gör. Veysel COŞĞUN ve ihtiyacım olan her anımda bana destek olan, güvenen Yrd. Doç. Dr. Yunus ŞAHİNKAYASI hocalarım olmak

üzere Mustafa Kemal Üniversitesi BÖTE bölümündeki hocalarıma ve arkadaşlarıma teşekkürlerimi sunarım.

Öğrenme ortamının geliştirilmesi sürecinde katkılarından ötürü değerli dostum Muammer Muhsin YILMAZ'a teşekkürlerimi sunarım.

Gerçekleştirilen uygulamanın bilimsel bir araştırma amacıyla yapıldığı ve herhangi bir not ile değerlendirilmeyeceklerini bilmelerine rağmen bir kişinin bile uygulamadan ayrılmayarak katkı sağladığı MKÜ BÖTE 2016-2017 Dönemi 2. sınıfta öğrenim görmüş olan kıymetli öğrencilerime teşekkürlerimi sunarım.

Çocukken kaybettiğim kişiliğiyle bana hala ışık tutan babam Hayrettin TEPGEÇ'e, beni okutmak için yaşadığı zorluklar için Ayşe TEPGEÇ'e ve bana her zaman destek olan kardeşlerim Leyla BARUT, Burak BARUT ve Mehmet TEPGEÇ'e teşekkürlerimi sunarım.

İlkeli bir bilim insanı olma yolunda en büyük gücü aldığım, kendisini anlamaya çalıştığım ancak hiçbir zaman tam olarak anlayamayacağım ulu önder Mustafa Kemal ATATÜRK'e teşekkürlerimi sunarım.

Tezi hazırlamamda büyük fedakârlık gösteren, ilgisini, sevgisini ve desteğini her anımda hissettiren, moral ve motivasyon kaynağım, huzurum, sevgili eşim Tuğçe TEPGEÇ'e sevgilerimi ve teşekkürlerimi sunarım.

ALGORİTMA ÖĞRETİMİNDE ÇÖZÜMLÜ ÖRNEK KULLANIMININ ÖĞRENCİ BAŞARISINA VE BİLİŞSEL YÜKE ETKİLERİ

Mustafa TEPGEÇ

ÖZ

Bu çalışmanın amacı, algoritma öğretiminde kullanılan geleneksel çözümlü örnek ve karartılmış çözümlü örnek yöntemlerinin üniversite öğrencilerinin başarılarına ve bilişsel yüklerine etkilerini incelemektir.

Bu çalışmada, ön test – son test kontrol gruplu gerçek deneysel desen kullanılmıştır. Araştırmanın çalışma grubu, bir devlet üniversitesinde Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü 2. sınıfında öğrenim gören 24 kadın, 9 erkek olmak üzere toplam 33 öğrenciden oluşmaktadır.

Çalışma için geliştirilen web tabanlı öğrenme ortamı aracılığıyla deney grubuna (n=17) karartılmış çözümlü örnekler sunulurken, kontrol grubuna (n=16) geleneksel çözümlü örnekler sunulmuştur. Öğrenme ortamının geliştirilmesinde çözümlü örnek alanyazınında etkililiği sınanmış tasarım ilkeleri kullanılmıştır. Web tabanlı öğrenme ortamında uygulama 2 hafta sürmüştür. İlk hafta sözde kod uygulaması gerçekleştirilirken, ikinci hafta akış şemaları uygulaması gerçekleştirilmiştir.

Araştırmanın verileri, öğrencilerin algoritma oluşturma performansını ölçmek için geliştirilen ön test - son test puanları, bilişsel yüklenme düzeylerini ölçmek için uygulanan bilişsel yük derecelendirme ölçeği puanları, uygulama ortamındaki süreç performans puanları ve öğrencilerin Programlama Dilleri I dersinden aldıkları final puanlarından oluşmuştur. Uygulanan yöntemleri öğretim verimliliği açısından karşılaştırmak için öğrencilerin ön test-son test puanları ve bilişsel yüklenme düzeyleri dikkate alınmıştır.

Sonuç olarak, algoritma öğretiminde kullanılan karartılmış çözümlü örnek yönteminin geleneksel çözümlü örnek yöntemine göre başarı açısından daha etkili olduğu belirlenmiştir. Deney ve kontrol grubundaki öğrenenlerin bilişsel yüklenme düzeyleri arasında anlamlı bir farklılık olmadığı görülürken, öğretim verimliliği açısından bakıldığında karartılmış çözümlü örneklerin daha verimli bir strateji

olduğu ortaya konulmuştur. Buna ek olarak dersin final sınavı puanları dikkate alınarak programlama performansının deney grubunda daha yüksek olduğu görülmüştür.

Anahtar sözcükler: Çözümlü örnek yöntemi, karartılmış çözümlü örnekler, algoritma öğretimi, programlama öğretimi, bilişsel yük, öğretim verimliliği.

Danışman: Doç. Dr. Yasemin DEMİRASLAN ÇEVİK, Hacettepe Üniversitesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı

THE EFFECTS OF USING WORKED EXAMPLE METHOD IN TEACHING ALGORITHM ON STUDENTS' ACHIEVEMENT AND COGNITIVE LOAD

Mustafa TEPGEÇ

ABSTRACT

The aim of the study is to examine the effects of two instructional strategies (traditional and faded worked example) in teaching algorithm on university students' achievement and cognitive load.

In this study, randomized pretest-posttest control group research design was used. Participants of the study consisted of 33 (24 female, 9 male) students studying at Department of Computer Education and Instructional Technologies.

While faded worked examples presented to experimental group (n=17) through web based learning environment which was developed for this study, traditional worked examples presented to control group (n=16). In the development of learning environment, design principles which had been tested in worked example literature was used. The implementation lasted for 2 weeks in the web based learning environment. During first week the pseudo-code application was performed, while the flow chart application was performed in the second week.

Research data consisted of pretest-posttest scores, which were developed to measure students' algorithm performances, cognitive rating scale, which was implemented to measure students' cognitive load, task performance scores in the learning environment, and final exam scores. Pre-test and post-test scores and cognitive load levels of the students were taken into account in order to determine the instructional efficiency of the methods applied.

As a result, faded worked example strategy used in teaching algorithm to university students found to be more effective in terms of achievement than the traditional worked example strategy. While there was no significant difference between the cognitive load levels of the learners in the experimental and control groups, faded worked example strategy was more efficient. In addition, the results indicated that the final programming scores of the experimental group were significantly higher than those of control group.

Keywords: Worked examples, faded worked examples, teaching algorithm, programming instruction, cognitive load, instructional efficiency.

Advisor: Assoc. Prof. Dr. Yasemin DEMİRASLAN ÇEVİK, Hacettepe University, Department of Computer Education & Instructional Technology

İÇİNDEKİLER

TABLOLAR DİZİNİ

ŞEKİLLER DİZİNİ

SİMGELER VE KISALTMALAR DİZİNİ

ANCOVA: Kovaryans analizi

ANOVA: Varyans analizi

BÖTE: Bilgisayar ve Öğretim Teknolojileri Eğitimi

BYK: Bilişsel Yük Kuramı

n: Katılımcı sayısı

ss: Standart sapma

p: Anlamlılık düzeyi

pj: Madde güçlük indeksi

rjx: Madde ayırt edicilik indeksi

1. GİRİŞ

1.1. Problem Durumu

Programlamayla ilk kez tanışan öğrenenler, program yazmayı oldukça zor ve karmaşık bir süreç olarak görmektedir. Programlama sürecinde öğrenenlerin program yazmayı gerçek yaşam durumlarıyla ilişkilendirememesi, yani değişkenler, veri türleri, döngüler, karar yapıları ve fonksiyonlar gibi soyut kavramların özümsenememesi programlama mantığının kavranmasını zorlaştırmaktadır (Dunican, 2002). Ayrıca programlama öğrenirken yeterli deneyimi olmayan öğrenenler yoğun bilişsel çaba sarf etmektedir (Lister, 2011). Bunun nedeni olarak programlama sürecinde öğrenenlerin hem programlama dilini yani sözdizimi kurallarını öğrenirken hem de problemlerin çözümü için algoritma adımlarını öğrenmek ve uygulamak durumunda kalması gösterilebilir.

Algoritma, herhangi bir problemin çözümünün adım adım gösterimi olarak tanımlanır (Eker, 2011). Algoritmalar yalnızca bilgisayar bilimleri alanında değil yaşamın her alanında kullanılmaktadır. Algoritmalar sözde kodlar halinde metinsel olarak ifade edilebileceği gibi akış şemaları halinde de ifade edilebilir. Akış şemalarında problemin çözüm adımlarının gösterimi için geometrik şekiller kullanılır.

Algoritmaların ifade edilmesi kullanılan programlama diline göre değişkenlik göstermemektedir (Durak, 2009). Diğer bir deyişle, hangi programlama dili kullanılırsa kullanılsın programlama mantığı aynıdır ve algoritmalar programlama mantığının oluşturulması için kullanılmaktadır (Begosso, Begosso, Gonçalves ve Gonçalves, 2012). Bu nedenle bir programlama platformuna geçiş yapılmadan önce “Programlamaya Giriş” ve muadili derslerde öğrenenlerde programlama mantığının oturması sağlanmalıdır. Bu derslerde programlama mantığı oluşturulmadan doğrudan programlama dili öğretimine geçilmesi sık karşılaşılan bir durumdur (Durak, 2009). Bu durumda öğrenenler yeterli bilişsel şemaları oluşturmakta güçlük çekmekte ve programlama öğrenmede başarısız olabilmektedir (Lister, 2011).

Yeterli deneyime sahip olmayan programlama eğitimcilerinin algoritma öğretiminin önemini yeterince benimsememesi sonucu programlama derslerinde algoritma

oluşturma uygulamaları ya yüzeysel olarak yapılmakta ya da hiç yapılmadan programlama dili öğretimine geçiş yapılmaktadır (Durak, 2009; Gal-Ezer vd., 2010). Gal-Ezer ve arkadaşları (2010) bütün yazılım sistemlerinin en iyi performansı gösterebilmeleri için nitelikli bir algoritma oluşturulmasının gerekli olduğunu vurgulamışlardır. Dolayısıyla programlama eğitimcileri için algoritma öğretimi oldukça önemli görülmektedir. Hubalovsky (2012), algoritma geliştirme ve programlama öğretimi için programlama eğitimcilerinin en çok üzerinde durması gereken nokta olarak uygun öğretim yöntemlerinin seçilmesi gerekliliğini belirtmiştir. Benzer şekilde Begosso ve arkadaşları (2012) programlama derslerinde başarısızlık oranının oldukça yüksek olmasının sebebinin öğrencilerin bu dersin zorluğuyla ilgili yaşadıkları kaygının yanı sıra programlama eğitimcilerinin algoritma öğretiminde sınırlı yöntem kullanımı ve yeni yöntemlerin denenmesindeki eksiklerden kaynaklanabileceğini belirtmişlerdir. Dolayısıyla, algoritma geliştirme ve programlama öğretiminin daha açık ve anlaşılır olması için etkili öğretim yöntemlerinin uygulanması öncelikli bir konu olmuştur (Bucks, 2010; Goldenson, 1996; Kert ve Kurt, 2012; May ve Dhillon, 2009; Renumol, Janakiram ve Jayaprakash, 2010). Morrison'a (2013) göre programlama eğitimcileri programlamayı nasıl daha verimli öğretebilirim sorusunu göz önünde bulundurarak etkili yöntem ve stratejileri uygulayabilirlerse öğrenenlerin bu süreç içerisinde gereksiz yere saatlerini harcamaları ve stres yapmaları önlenebilir.

Programlama öğretiminde kullanılan yöntem ve araçların devingen olduğu günümüzde temel amaç "açık" ve "etkili" bir öğrenme-öğretme süreci tasarlanmasıdır. Öğretimin açıklığından kasıt öğrenenler için dışsal kaynakları azaltıp var olan bilişsel kaynaklarını daha iyi kullanmalarına olanak tanınmasıdır. Öğretimin etkililiğinden kasıt ise öğrenenlerin daha sonraki problem durumları ile başa çıkabilmeleri için bilişsel şemalarının anlamlı bir şekilde yapılandırılmasına katkı sağlanmasıdır. Öğretimin açık ve etkili bir şekilde tasarlanması için de uygun öğretim yöntemlerinin kullanılması önem arz etmektedir (Abdul-rahman ve Boulay, 2014; Begosso vd., 2010). Alanyazın incelendiğinde programlama öğretiminde yapılan çalışmalarda genellikle geleneksel problem çözme yöntemlerinin etkililiğinin sınandığı ve alternatif yöntemlerin kullanımını ele alan yeterince çalışma olmadığı görülmüştür. Hubalovsky (2012) de programlama öğretimi konusunda sayısız çalışma yapılmasına rağmen alternatif yöntemlerin etkililiğinin

sınandığı yeterince çalışma olmadığını vurgulamıştır. Hubalovsky (2012) ayrıca geleneksel yöntemlerle programlama öğretiminin yeterli deneyimi olmayan öğrenenlerin derse karşı tutumunu olumsuz yönde etkileyebileceğini belirtmiştir.

Mayer (2013) ve Skinner (2016) programlama öğretimi ile ilgili yapılan çalışmaların programlama öğretiminde kullanılan ortamlara veya programlara odaklanılmasının doğru bir yaklaşım olmadığını ve "ne" sorusundan ziyade "nasıl" sorusuna cevap aranması gerektiğini belirtmişlerdir. Son yıllarda programlama öğretiminde öğrenene rehberlik sağlayacak yöntem ve stratejilerin etkililiğinin sınandığı birçok çalışma bulunmaktadır (Abdul-rahman ve Boulay, 2014; Lee, 2013; Margulieux, Catrambone ve Guzdial, 2016; Margulieux ve Catrambone, 2016; Si, Kim ve Na, 2014). Mayer (2013) programlama öğretiminde keşife dayalı öğretim stratejilerinden ziyade öğrenenlere rehberlik sağlayacak yeni stratejilerin uygulanması gerektiğini belirtmiştir. Dolayısıyla yeni yöntemlerin programlama öğretiminde etkililiğin sınanması önemli görülmektedir.

Programlama öğretiminin niteliğini arttırmak için benzer süreçleri içeren alanlarda, etkililiği kanıtlanmış yöntem ve stratejilerin uygulanması önemli görülmektedir. Bu çalışma kapsamında belirli bir algoritmik çözüm yolunun bulunduğu problemler üzerinde durulması, sınırlı sayıda kavram ve ilkelerin bulunması gibi programlama öğretimine benzer süreçleri içeren Matematik ve Fizik öğretiminde sıkça kullanılan yöntemlerden biri olan çözümlü örnek yöntemi ele alınmıştır. Van Merrienboer ve Paas (1990) çözümlü örneklerin programlama öğretimindeki etkililiğinin sınanmasının göz ardı edildiğini belirtmiş olsa da alanyazın incelendiğinde programlama öğretiminde çözümlü örneklerin kullanımının son yıllarda artış gösterdiği görülmektedir (Abdul-rahman ve Boulay, 2014; Lee, 2013; Margulieux, Catrambone ve Guzdial, 2016; Margulieux ve Catrambone, 2016; Si, Kim ve Na, 2014). Ancak yine de bu çalışmaların oldukça sınırlı olduğu görülmüştür. Bununla ilgili ayrıntılı bilgiye ilgili araştırmalar bölümünde yer verilmiştir.

Çözümlü örnek yöntemi, bir sorunun nasıl çözüldüğünü ya da verilen görevin nasıl gerçekleştirildiğini öğretebilmek amacıyla çözüm yolunun öğrenene adım adım verildiği bir öğretim yöntemidir (Clark, Nyugen, Sweller ve Baddeley, 2006). Çözümlü örnek kullanımında temel amaç, konu hakkında yeterli deneyimi olmayan öğrenene uzman bakış açısıyla çözüm yolunu göstererek konuyla ilgili kural ve ilkelerin öğrenilmesini sağlamaktır. Çözümlü örneklerde problem ile birlikte çözüm

adımları verilir ve öğrenenden problemin çözümünün nasıl gerçekleştirildiğini görmesi için örnekler üzerinde çalışması beklenir. Konu hakkında yeterince çözümlü örnek sunduktan sonra öğrenciye problem sunularak öğrenciden problemi çözmesi beklenir. Bu şekilde çözümlü örnekler aracılığıyla edinilen şemaların güçlendirilmesi hedeflenir.

Son yıllarda özellikle çözümlü örneklerin farklı biçimlerde kullanımı dikkat çekmektedir (Bkz. Renkl, 2014). Geleneksel problem çözme yönteminde, geleneksel çözümlü örneklere göre öğrenenler daha aktif bir süreç geçirmektedir (van Merrienboer, 1990). Ancak yeterli deneyimi olmayan öğrenenlerde bir yapı iskelesi oluşturmadan problem çözümüne geçilmesi aşırı bilişsel yüklenmeye sebep olmakta ve öğrenmeyi olumsuz etkilemektedir (Renkl, Atkinson ve Große, 2004). Dolayısıyla yeterli deneyimi olmayan öğrenenler için çözümlü örneklerin problem çözme yöntemlerine göre daha verimli olduğu birçok çalışmada ifade edilmiştir (Kalyuga vd., 2001; Lee, 2013; Renkl ve Atkinson, 2003). Bu nedenle son yıllarda yapılan çalışmalar çözümlü örnek yöntemi ile problem çözme yönteminin etkililiğini karşılaştırmak yerine, farklı çözümlü örnek kullanım biçimlerinin etkililiğini karşılaştırmaktadır (Abdul-rahman ve Boulay, 2014; Catrambone ve Guzdial, 2016; Lee, 2013; Margulieux, Catrambone ve Guzdial, 2016; Si, Kim ve Na, 2014). Yapılan çalışmalarda geleneksel çözümlü örnekler ile alt hedeflerin vurgulandığı çözümlü örnekler (Margulieux, Catrambone ve Guzdial, 2013; Margulieux ve Catrambone, 2016), tamamlamalı çözümlü örnekler (Abdul-rahman ve Boulay, 2014; Harms, Rowlett ve Kelleher, 2015), öz açıklamalı çözümlü örnekler (Abdul-rahman ve Boulay, 2014) gibi yeni yöntemler karşılaştırılmıştır.

Tamamlamalı çözümlü örnek yöntemi geleneksel çözümlü örnek yöntemine bir alternatif oluşturmuştur. Geleneksel çözümlü örneklerde öğrenenlerin süreç içerisinde aktif olamaması ve örneklerin altında yatan mantığı kavramak yerine öğrenenlerin tekrar tekrar okuma gibi istenmeyen davranışlarda bulunması bu yöntemin dezavantajlarıdır (Lee, 2013; van Merrienboer, 1990). Tamamlamalı çözümlü örneklerde öğrenenler örneklerde boş bırakılan çözüm adımlarını doldurmakta ve problem çözümü öncesi daha aktif bir süreç geçirmektedir. Ancak tamamlamalı çözümlü örneklere çalışırken öğrenenler daha yoğun bilişsel çaba harcamakta ve örnekler iyi yapılandırılmadığı takdirde aşırı bilişsel yüklenme

oluşabilmektedir (Renkl, Atkinson ve Große, 2004). Bir tamamlamalı çözümlü örnek stratejisi olan karartılmış örneklerde örnek adımlarındaki boşluklar aşamalı bir şekilde arttırılır ve örneklerden problem çözmeye daha yumuşak bir geçiş sağlanmış olur. Karartılmış çözümlü örnekler geleneksel çözümlü örnek yöntemine göre daha etkili bir stratejidir (Kissane, Kalyuga, Chandler ve Sweller, 2008; Renkl ve Atkinson, 2003). Karartılmış çözümlü örnekler aynı zamanda giderek artan problem çözme becerisinin uygulanmasından dolayı uzmanlığın ters tepme etkisine karşı etkili bir yöntemdir (Kalyuga, 2007). Karartılmış çözümlü örnekler ayrıca şema edinimi ve kural otomatikleştirmeye katkı sağlayan bir stratejidir (Große ve Renkl, 2007).

Şema edinimi ve kural otomatikleştirme problem çözme performansının temel bileşenleri olarak varsayılır (Ward ve Sweller, 1990). Şema, çeşitli problem çözme adımlarını gruplandıran ve uygun prosedürün uygulanmasına olanak tanıyan uzun süreli bellekte tutulan, alana özgü bilişsel yapılar olarak tanımlanır (Chi, Glaser ve Rees, 1981; Schoenfeld ve Herrmann, 1982). Kural otomatikleştirme ise problem çözümü için gereken kuralların, farkında olmadan ya da çok az bilinçle probleme uygulanmasıdır (Kotovsky, Hayes ve Simon, 1985). Buna, konuyla ilgili yeterli deneyimi olan bir öğrenenin geometrideki dik üçgen probleminde 5 ve 12 olan kenarları görüp hipotenüsü matematiksel işlem yapmadan 13 olarak düşünmesi ya da bir bilgisayar programı yazarken kullandığı programlama diline özgü sözdizimi kurallarını otomatik olarak uygulaması örnek gösterilebilir. Bu da öğrenenlerin problem çözümü için daha az bilişsel kaynak sarf etmesini sağlar. Çözümlü örnekler, yeterli deneyimi olmayan öğrenenlere yeni bir problemle karşılaştıklarında şema edinimi ve kural otomatikleştirme açısından rehberlik sağlamaktadır.

Çözümlü örnek yöntemi bilişsel yük kuramına (BYK) (Sweller, 1988; Sweller, van Merrienboer ve Paas, 1998; van Merrienboer ve Sweller, 2005) dayanır. BYK'ye göre şema edinimi ve kural otomatikleştirme öğrenmenin temel bileşenleridir ve öğrenen şema edinimi ve kural otomatikleştirme ile ilgili olmayan etkinliklere bilişsel kaynak tahsis ettiği zaman öğrenme süreci olumsuz etkilenir (Kalyuga vd., 2001). Bilişsel yük ya da diğer adıyla kısa süreli bellek yükü öğrenme görevinin doğasından (zorluk-kolaylık) kaynaklanan asıl bilişsel yük, öğrenme materyalinin sunuluş biçiminden kaynaklanan konu dışı bilişsel yük ve öğrenmenin

gerçekleşmesini sağlayan etkili bilişsel yük olarak üçe ayrılır. Programlama öğretiminde çözümlü örnek kullanımı yeterli deneyimi olmayan öğrenenlerin konu dışı bilişsel yüklenmelerini azaltarak bilişsel kaynaklarını daha verimli kullanmalarını sağlamaktadır (Abdul-rahman ve Boulay, 2014; Lee, 2013).

Özetle, algoritma geliştirme ve programlama öğretiminde uygun yöntem ve stratejilerin kullanımı önemli görülmektedir. Alanyazında programlama öğretiminde yapılan çalışmalar incelendiğinde genellikle geleneksel problem çözme yöntemlerinin ele alındığı görülmüştür. Ancak yapılan bazı çalışmalarda, yeterli deneyimi olmayan öğrenenlerde çözümlü örnek yönteminin, geleneksel problem çözme yöntemine göre daha etkili olduğu bulgusuna ulaşılmıştır (Abdul-rahman ve Boulay, 2014; Margulieux, Catrambone ve Guzdial, 2015; van Merrienboer, 1990). Bununla birlikte son yıllarda yapılan çalışmalarda çözümlü örnek ile problem çözme yöntemlerini karşılaştırmak yerine hangi çözümlü örnek biçiminin daha etkili olduğuna yönelik karşılaştırmalar yapılmaktadır. Algoritma veya programlama öğretiminde Türkiye'de çözümlü örneklerin etkililiğinin sınandığı bir çalışmaya ulaşılamamıştır. Benzer şekilde bu konuda yurt dışında yapılan yayınlar incelendiğinde de yeterince çalışma olmadığı görülmüştür. Ayrıca alanyazında programlama öğretiminin temellerini oluşturmada sıkça kullanılan algoritma öğretiminde çözümlü örnek yöntemlerinin kullanıldığı yurt içi ve yurt dışında yapılan bir çalışmaya ulaşılamamıştır.

1.2. Araştırmanın Amacı ve Önemi

Bu çalışmanın amacı, programlamanın temeli olan algoritma öğretiminde kullanılan geleneksel çözümlü örnek ve karartılmış çözümlü örnek yöntemlerinin bir web ortamı aracılığıyla üniversite öğrencilerinin başarılarına ve bilişsel yüklenmelerine etkisini incelemektir.

Bu araştırmanın programlama ya da sıkça kullanılan tabiriyle kodlama eğitiminin gerekliliği konusunda tartışmaların sürdüğü günümüzde programlamanın temeli olan algoritma öğretiminde uygulanacak öğretim yöntem ve stratejilerinin nasıl olması gerektiği konusunda alanyazına katkı sağlayacağı düşünülmektedir. Türkiye'de bu konuda yapılan çalışmaya rastlanılmaması ile birlikte dünya genelinde çözümlü örnek alanyazınındaki çalışmalar incelendiğinde bu konuda yeterince çalışma olmadığı görülmüştür.

Türkiye'de ilköğretimde "Bilişim Teknolojileri ve Yazılım" dersi için 2017 yılında geliştirilen müfredata göre algoritma öğretiminin 5. ve 6. sınıflarda zorunlu, 7. ve 8. sınıflarda seçmeli olarak gerçekleştirilmesi planlanmaktadır. Bu nedenle algoritma ve programlama eğitimi oldukça önemli görülmektedir. Algoritma geliştirme ve programlama öğretiminde yeni yöntemlerin etkililiğinin sınanmasının hem ilköğretim öğrencileri için hem de bu öğrencileri yetiştirecek Bilişim Teknolojileri öğretmen adayları için katkı sağlayıcı olabileceği öngörülmektedir.

Çalışmada ayrıca çözümlü örnek alanyazınında etkililiği sınanmış açıklamalı yardım, basitten karmaşığa sıralama, alt hedeflere ayırma ve basit ilişkilendirme ilkeleri kullanılmıştır. Bu ilkeler geliştirilen web ortamının tasarımını oluşturmuştur. Dolayısıyla hem araştırmacılar hem de geliştiriciler açısından bu ilkelerin çözümlü örnek tasarımında nasıl kullanılması gerektiği konusunda bu çalışmanın alanyazına önemli katkılar sağlayacağı öngörülmektedir.

Yapılan çalışmalarda programlama öğretiminde çözümlü örnek yönteminin problem çözme yöntemine göre daha verimli olduğu bulgusuna ulaşılmış, ancak hem çözümlü örnek yöntemlerinden hangisinin daha etkili olduğu konusunda yeterince çalışmaya ulaşılamamış hem de programlamanın temeli olan algoritma öğretiminde çözümlü örnek kullanımının etkililiği sınanmamıştır. Ayrıca programlama veya algoritma öğretiminde karartılmış çözümlü örneklerin etkililiğinin incelendiği bir çalışmaya ulaşılamamıştır. Bütün bu nedenlerden dolayı yapılacak olan bu çalışmanın alanyazına önemli katkılar getireceği öngörülmektedir.

1.3. Problem Cümlesi

Algoritma öğretiminde kullanılan iki farklı çözümlü örnek yönteminin üniversite öğrencilerinin başarılarına ve bilişsel yüklerine etkileri nelerdir?

1.3.1. Alt Problemler

1. Algoritma öğretiminde geleneksel çözümlü örnek ve karartılmış çözümlü örnek yöntemlerinin kullanıldığı grupların son test performanslarında anlamlı bir farklılık var mıdır?

2. Algoritma öğretiminde geleneksel çözümlü örnek ve karartılmış çözümlü örnek yöntemlerinin kullanıldığı grupların bilişsel yüklenme düzeylerinde anlamlı bir farklılık var mıdır?
3. Algoritma öğretiminde öğretim verimliliği açısından geleneksel çözümlü örnek ile karartılmış çözümlü örnek yöntemleri arasında anlamlı bir farklılık var mıdır?
4. Geleneksel çözümlü örnek ve karartılmış çözümlü örnek yöntemlerinin kullanıldığı grupların süreçteki problem çözme performanslarında anlamlı bir farklılık var mıdır?
5. Algoritma öğretiminde kullanılan öğretim yöntemleri Programlama Dilleri I dersi final puanlarını yordamakta mıdır?

1.4. Sayıltı ve Sınırlılıklar

- Başarı testinin güvenirliği için pilot uygulamanın yapıldığı grupta soruların samimi bir şekilde yanıtlandığı,
- Uygulama esnasında kullanılan web ortamına girişte katılımcıların yalnızca kendi öğrenci numaralarıyla giriş yaptıkları,
- Çalışmada hesaba katılmayan değişkenlerin deney ve kontrol grubunu benzer oranda etkilediği varsayılmıştır.

Çalışmanın sonuçları, uygulamanın gerçekleştirildiği üniversitenin Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü'nde öğrenim görmekte olan 33 öğrenciyle sınırlıdır.

1.6. Tanımlar

Çözümlü örnek: Çözümlü örnek, bir sorunun nasıl çözüldüğünü ya da verilen görevin nasıl gerçekleştirildiğini öğretebilmek amacıyla çözüm yolunun öğrenene adım adım verildiği bir öğretim yöntemidir (Clark, Nyugen, Sweller ve Baddeley, 2006).

Bilişsel yük kuramı: Kısa süreli bellek kaynaklarının sınırlı kapasiteye sahip olmasına dayanan kuram (Sweller, 1988).

Sözde kod: Programlama dillerinde kullanılan terimlerle algoritma oluşturma.

Akış şemaları: Algoritmanın şemalarla ifade edilerek görselleştirilmesini sağlayan araçlar.

2. İLGİLİ ARAŞTIRMALAR

2.1. Programlama Öğretimi ve Algoritma

Programlama kısaca bir problemin çözümüne yönelik oluşturulan işlem adımlarının programlama dilleri platformları aracılığıyla bilgisayarların anlayacağı dilde ifade edilmesi olarak tanımlanabilir (Durak, 2009). Programlama alanında öğrenenler formal eğitim sürecinde genellikle “Programlama Dilleri“, “Bilgisayar Bilimine Giriş”, “Programlamaya Giriş”, “Programlama Dilleri Temelleri” gibi dersler ile programlama ile tanışırlar. Bu derslerin verilmesinde genel yaklaşım bir platform üzerinde kod yazma sürecinden önce programlama mantığının öğretilmesi için algoritma öğretiminin gerçekleştirilmesidir.

Programlamanın temelinde programlama mantığı yatmaktadır. Programlama mantığı öğretimi için de algoritmalar sıklıkla kullanılan bir yöntemdir. Algoritma, bir problemin çözümünde kullanılan adımlar bütünü olarak ifade edilebilir (Eker, 2011). Algoritmaları aslında yaşamımızın her alanında kullandığımız problem çözme sürecine uyarlayabiliriz. Algoritmaların programlama temelleri öğretiminde bu kadar sık kullanılmasının önemli bir nedeni programlama sürecini gerçek yaşam durumlarıyla somutlaştırmaktır. Algoritma öğretiminde bilgisayar programı oluşturma sürecinde kullanılan değişkenler, karar yapıları, döngüler gibi kavramların somutlaştırılması amaçlanır.

Algoritma öğretiminde en sık kullanılan yöntemler işlem adımlarının metinsel olarak ifade edilmesi ve akış şemalarıyla gösterimidir. Algoritmanın metinsel olarak ifade edilmesi işlem adımlarının numaralandırılarak alt alta yazılmasıdır. Akış şeması ile gösterimde ise başla/bitir, karar yapıları, işlem, girdi/çıktı gibi alt süreçlerin her biri belirli bir sembolle gösterilir. Akış şeması ile gösterimde temel amaç problemin çözüm adımlarını görselleştirerek daha verimli bir program taslağı oluşturmaktır. Ancak algoritmanın metinsel olarak ifade edilmesi ve akış şemaları gerçek bir programlama platformuna geçişte yetersiz kalabilmekte ve bu nedenle yarı programlama dili yarı günlük dilin kullanıldığı sözde kod (pseudo-code) kullanılmaktadır. Sözde kod uygulamalarında algoritma yine metinsel olarak ifade edilir ancak farklı olarak “Sonucu ekrana yazdır” yerine “YAZ degiskenadi” gibi komutlar kullanılır. Böylece gerçek programlama platformlarına daha yumuşak bir geçiş yapılmış olunur.

Programlama dersleri genel anlamda zor olarak algılanmakta ve öğrencilerin en fazla başarısız olduğu derslerden biri olarak görülmektedir (Robins, Rountree ve Rountree, 2003). Dolayısıyla son yıllarda programlama başarısını ve başarısızlığını etkileyen faktörlerin keşfedilmesi ile ilgili bir eğilim söz konusudur (Ferrer-Mico, Fernandez ve Sanchez, 2012; Hawi, 2010; Jegede, 2009; Lau ve Yuen, 2009; Lau ve Yuen, 2011; Shaw, 2012; Sivasakthi ve Rajendran, 2011). Bu konuda yapılan çalışmalar programlama başarısının cinsiyet (Lau ve Yuen, 2011; Sullivan ve Bers, 2012; Yurdugül ve Aşkar, 2013), programlama deneyimi (Bergersen ve Gustafsson, 2011; Jegede, 2009; Lau ve Yuen, 2011), akademik başarı ve matematik performansı (Lau ve Yuen, 2009), öz yeterlik (Altun ve Mazman, 2012; Jegede, 2009) ve problem çözme becerilerinden (Fessakis, Gouli ve Mavroudi, 2013; Yurdugül ve Aşkar, 2013) etkilendiğini göstermektedir. Özmen ve Altun (2014) ise öğrencilerin programlama konusunda yaşadığı başlıca güçlüklerin programlama bilgisinin (sözdizimi, fonksiyonlar, kavramlar ve ilkeler, değişken atama, karar yapıları ve döngüler) eksikliğinden, programın mantığını kavrayamamaktan, hata ayıklama ve programlama becerilerinin yetersizliğinden kaynaklandığını belirtmişlerdir. Ayrıca programlamayı etkileyen faktörler arasında motivasyon, programlamaya karşı tutum ve öğretim yöntem ve teknikleri bulunmaktadır (Jenkins, 2002). Benzer şekilde Robins ve arkadaşları (2003) da programlama başarısını etkileyen en önemli faktörlerden birinin kullanılan öğretim yöntem ve stratejileri olduğunu belirtmiştir. Burada dikkat edilmesi gereken nokta programlamayı etkileyen birçok faktör olmasına karşın programlama eğitimcilerine yönelik çıkarım yapılabilecek en önemli faktör kullanılan öğretim yöntem ve stratejileridir. Her ne kadar bir bilgisayar programı oluşturmak için herhangi bir programlama dili bilgisinin gerekmediği platformlar (Scratch, App Inventor, vb.) son yıllarda yaygınlaşsa da programlamanın sadece söz diziminden ibaret olmadığı düşünüldüğünde programlama öğretiminde kullanılan yöntem ve stratejiler önemini korumaktadır.

Bilgisayar bilimi ya da yazılım mühendisliği gibi alanların programlamayla ilgili temel sorusu “Etkili bir yazılım projesi için neler yapılabilir?” olurken eğitim teknolojisi gibi alanlar için temel soru “Yeterli deneyimi olmayan bireylere etkili bir öğrenme-öğretme süreci tasarlamak için hangi yöntem ve araçları kullanmalıyım?” olmuştur. Programlama öğretiminde kullanılan yöntem ve stratejiler konusunda

alanyazın incelendiğinde yelpaze oldukça geniş olduğundan genel bir yorum yapmak doğru olmayabilir. Ancak Mayer (2013) ve Skinner (2016) programlama öğretimi ile ilgili yapılan çalışmaların programlama öğretiminde kullanılan ortamlara veya programlara odaklanılmasının doğru bir yaklaşım olmadığını ve "ne" sorusundan ziyade "nasıl" sorusuna cevap aranması gerektiğini belirtmişlerdir. Son yıllarda programlama öğretiminde öğrenene rehberlik sağlayacak yöntem ve stratejilerin etkililiğin sınandığı birçok çalışma bulunmaktadır (Abdul-rahman ve Boulay, 2014; Lee, 2013; Margulieux, Catrambone ve Guzdial, 2016; Margulieux ve Catrambone, 2016; Si, Kim ve Na, 2014). Mayer (2013) programlama öğretiminde keşife dayalı öğretim stratejilerinden ziyade öğrenenlere rehberlik sağlayacak stratejilerin uygulanması gerektiğini belirtmiştir. Bilişsel yük kuramına dayanan çözümlü örnek yöntemi yeterli deneyimi olmayan öğrenenler için bilişsel kaynaklarını daha verimli kullanmalarına olanak sağlamasından dolayı son yıllarda tercih edilen bir yöntem olmuştur.

2.2. Bilişsel Yük Kuramı

Bilişsel yük kuramı (Sweller, 1988), kısa süreli bellek kaynaklarının sınırlı kapasiteye sahip olmasına dayanmaktadır. Öğrenme sürecinde bilişsel kaynaklar kısa süreli bellek tarafından sınırlandırılmaktadır. BYK'nin savunduğu temel ilke, öğrenme sürecinde kısa süreli bellek kaynaklarını gereksiz yere etkileyebilecek etkenleri elimine ederek kısa süreli bellekteki kaynakların öğrenme için daha verimli kullanılmasını sağlamaktır (Renkl ve Atkinson, 2003). Yani bilişsel kaynakların şema edinimi ve kural otomatikleştirmeyi sağlayan öğrenme etkinliklerine yönlendirilmesi amaçlanır. Bilişsel yük ya da diğer adıyla kısa süreli bellek yükü öğrenme görevinin doğasından kaynaklanan asıl bilişsel yük, öğrenme materyalinin sunuluş biçiminden kaynaklanan konu dışı yük ve asıl öğrenmenin gerçekleştiği etkili yük olarak üçe ayrılır.

Asıl bilişsel yük öğrenilecek olan konunun doğasından kaynaklanan zorlukla ilgilidir. Eğer sunulan bilgi öğrenen için karmaşık ve zorsa asıl yük yüksek olmaktadır (Sweller, 1994; Sweller, 2010; Sweller ve Chandler, 1991; Tindall-Ford, Chandler ve Sweller, 1997). Asıl bilişsel yük, öğrenilmesi gereken materyallerin ya da görevlerin birim etkileşimliliğinin ne ölçüde olduğuna bağlı olarak kısa süreli

bellekte eş zamanlı işlenmesi gereken birimin sayısına bağlıdır (van Merrienboer ve Sweller, 2010). Paas ve Sweller'a (2014) göre asıl bilişsel yük öğrenme görevinin doğasının ya da öğrenenin ön bilgisinin değişmesiyle farklılık gösterebilir. Yani "çift" sözcüğü bu yazıyı okuyanlar için tek bir birim olarak ele alınırken, okumayı yeni öğrenen bir çocuk için oldukça karmaşık ve yüksek birim etkileşimliliği olan bir kelime olabilir. Konu dışı yük ise iyi tasarlanmamış öğretim tasarımı sonucunda oluşur (Kılıç Çakmak, 2007; van Merrienboer ve Sweller, 2010). Uygun olmayan öğretim tasarımı kısa süreli bellekte gereksiz kaynak kullanımına yol açar (Paas ve Sweller, 2014). Dolayısıyla öğretim tasarımcılarının amacı konu dışı yükü en alt seviyeye çekerek kısa süreli bellek kaynaklarını daha verimli hale getirmek olmalıdır. Etkili yük ise konu dışı yük yerine asıl yüke tahsis edilen kısa süreli bellek kaynaklarıdır (Paas ve Sweller, 2014). Çözümlü örnekler aracılığıyla öğrenmeyi arttırmayı amaçlayan neredeyse bütün öğretimsel müdahaleler ya etkili yükü arttırmayı ya da konu dışı yükü azaltmayı amaçlamaktadır (Kalyuga vd., 2001).

2.2.1. Bilişsel Yük'ün Ölçülmesi

Öğretim tasarımı alanındaki en önemli kuramlardan biri olarak görülen BYK, çoklu ortam ve diğer öğrenme materyallerinin tasarımı sürecinde etkili bir rehber olarak ele alınmaktadır. Bilişsel yükün ölçülmesinde iki temel yaklaşım uygulanmaktadır; fizyolojik ölçümler gibi bilişsel yükün nesnel ölçümü ve öğrenenlerin algılanan bilişsel yüklenme düzeylerini derecelendirdiği öznel ölçüm (Brünken, Seufert ve Paas, 2010). Bilişsel yükün nesnel ölçümünde göz izleme analizleri, kalp atış verileri, öğrenme çıktıları ve çalışma süreleri değerlendirilmektedir. Öznel ölçümde ise bir derecelendirme ölçümüne göre elde edilen veriler değerlendirilmektedir. Bu ölçümlerin hangisinin daha kullanışlı olduğu araştırma sorularına göre değişkenlik gösterebilir (Brünken vd., 2010). Buna ek olarak her iki ölçümün harmanlandığı verimlilik ölçümleri de kullanılmaktadır (Paas ve van Merrienboer, 1993; Paas, Tuovinen, Tabbers ve van Gerven, 2003). Verimlilik ölçümü hem öğrenenler tarafından doldurulan derecelendirme ölçeği puanları hem de performans puanları standartlaştırılarak oluşturulur. Brünken vd., (2010) verimlilik ölçümü kullanılmasının gerekçesini şöyle açıklamıştır:

> *Aynı içeriğe sahip materyallerin iki farklı öğretim yöntemiyle sunulduğu iki öğrenci grubu olduğunu varsayalım: İlk grup düşük düzeyde zihinsel çaba harcadığını belirtirken, diğer grup öğrenme çıktıları açısından daha başarılı olmuştur. Bu durumda hangi öğretim yöntemi önerilebilir? Böyle bir soruya cevap vermek bu duruma göre uygun olmayabilir. Hataların telafisi olmadığı durumların öğretiminde bilişsel çabadan ziyade somut çıktılar daha önemli görülebilir. Ancak yeterli deneyimi olmayan öğrenenlere yeni bilgi edindirme sürecinde aşırı yüklenmeden kaçınmak oldukça önemlidir. Bu duruma yönelik Paas ve van Merrienboer (1993) zihinsel çaba ve performansı harmanlayan bir yaklaşım önermiştir (s.191).*

Bilişsel yük ölçümüne ve verimlilik formülüne ilişkin ayrıntılı bilgi yöntem bölümünde veri toplama araçları başlığı altında yer almaktadır.

Çözümlü örneklerin etkililiği BYK'ye dayanmaktadır (Paas ve van Gog, 2006; Sweller, van Merrienboer ve Paas, 1998). Gerekli yönlendirmenin sağlanmadığı durumlarda problem çözme süreci, yeterli deneyimi olmayan öğrenenlerde konu dışı yüklenmeyi arttırmakta, hatta aşırı bilişsel yüklenmeye sebep olabilmektedir (Renkl, 2014). Çözümlü örnekler, problem çözme sürecinden kaynaklanan konu dışı bilişsel yüklenmeyi elimine etmek için geliştirilmiş bir yöntemdir (Sweller, 1989).

2.3. Öğrenme ve Öğretme Sürecinde Çözümlü Örnek Kullanımı

Çözümlü örnek, öğrenenlerin herhangi bir yönlendirmeye ihtiyaç duymadan problem çözme aşamasından önce çözüm adımları verilen problemlere çalışarak bir uzman gözüyle problem çözümündeki kural ve ilkeleri öğrenmesini sağlayan bir stratejidir (Renkl, 2005). Çözümlü örneklerde bir uzman gözüyle problemin çözümünün gösterimi, Bandura'nın (1986) sosyal-bilişsel öğrenme kuramı ile ilişkilendirilebilir. Sosyal-bilişsel öğrenme kuramına göre öğrenenler bir model üzerinden gözlemleyerek öğrenir. Burada modelden kasıt sadece bir beceriyi gösteren birey olmayabilir, benzer şekilde bir bireyin problem çözümüne dair kural ve ilkeleri de model olarak düşünülebilir (Renkl, 2014). Yani Bandura'ya göre öğrenme, öğrenenin sadece diğerlerini gözlem yoluyla taklit etmesi değil, çevredeki olayları veya durumları bilişsel olarak işlemesiyle gerçekleşir. Mayer (2004) son 50 yılda eğitim alanında yapılan çalışmalarda buluşsal öğrenme stratejisi olarak problem çözme yöntemi kullanımı ile karşılaştırıldığında öğrenene rehberlik sağlayacak öğretimin (guided instruction) lehine bulgular elde edildiğini belirtmiştir.

Çözümlü örnek ve problem çözme yöntemlerinin etkililiği öğrenenlerin ön bilgisine göre farklılık göstermektedir. Bu nedenle çözümlü örnek veya problem çözme yöntemlerinin hangisinin kullanılacağına öğrenenlerin ilgili alandaki ön bilgisi belirlendikten sonra karar verilmelidir (Kalyuga, Chandler, Tuovinen ve Sweller, 2001). Yapılan çalışmalar yeterli deneyimi olmayan öğrenenlerde çözümlü örneklerin geleneksel problem çözme yöntemine göre matematik (Cooper ve Sweller, 1987; Sweller, 1989; Sweller ve Cooper, 1985), kimya (Crippen ve Earl, 2004; Crippen ve Earl, 2007), fizik (Richey ve Nokes-Malach, 2015), mühendislik (Pollock, Chandler ve Sweller, 2002) ve bilgisayar programlama (Abdul-rahman ve Boulay, 2014; Lee, 2013; Margulieux, Catrambone ve Guzdial, 2016; Margulieux ve Catrambone, 2016; Si, Kim ve Na, 2014) gibi iyi yapılandırılmış alanlarda daha etkili olduğunu göstermiştir. Dolayısıyla rehberlik sağlandığı takdirde yeterli deneyimi olmayan öğrenen, bilişsel kaynaklarını verimli bir şekilde kullanabilmektedir. BYK alanyazınında buna “Çözümlü örnek etkisi” denilmektedir (Sweller, van Merrienboer ve Paas, 1998). Çözümlü örnek etkisi öğrenenin problem çözme uygulamalarına göre sınırlı bilişsel kaynaklarını daha etkili kullanmasının sonucudur (Moreno, 2006).

Öğrenenler ilgili alanda uzmanlaştıkça çözümlü örnek etkisi ilk başta kaybolmakta ardından tersi bir etki yapmaktadır (Kalyuga vd., 2001). BYK alanyazınında bu duruma “Uzmanlığın ters tepme etkisi” denilmektedir (Kalyuga, 2007). Öğrenenler konu hakkında uzmanlaştıkça çözümlü örnekler gereksiz hale gelmekte ve konu dışı bilişsel yüklenmeye sebep olmaktadır (van Merrienboer ve Sweller, 2010). Bu nedenle alanyazında çözümlü örneklerin yeterli deneyimi olmayan öğrenenler için kullanılması önemle vurgulanmaktadır.

Yapılan çalışmalar çözümlü örneklerin özellikle iyi yapılandırılmış alanlarda etkili olduğunu göstermektedir. İyi yapılandırılmış alanlarda sınırlı sayıda kavramlar, kurallar, ilkeler uygulanır, ayrıca açık bir hedef ve algoritmik bir çözüm yolu bulunur (Nievelstein, van Gog, van Dijck ve Boshuizen, 2013). Buna ek olarak iyi yapılandırılmış alanlardan biri olan programlama öğretiminde öğrencilerin çözümlü örnekler üzerinden öğrenmesini sağlamanın bilişsel becerilerin ve şemaların edinimi ve yönetimine yardım etmede önemli bir pedagojik strateji olduğu vurgulanmaktadır (Atkinson, Derry, Renkl ve Wortham, 2000; Chi, Bassok, Lewis,

Reimann, ve Glaser, 1989; Pirolli ve Anderson, 1985; van Merriënboer ve Paas, 1990).

Alanyazında çözümlü örneklere ilişkin farklı kullanım biçimleri ve tasarım ilkeleri ortaya konulmuştur. Çözümlü örneklerin kullanılıp kullanılmadığından ziyade öğrenenin bilişsel kaynaklarını daha verimli kullanmasını sağlayacak şekilde tasarlanması önemli görülmektedir (Ward ve Sweller, 1990). Benzer şekilde araştırmacılar çözümlü örneklerin etkililiğinin, örneklerin yapısına veya tasarımına bağlı olduğunu belirtmişlerdir (Catrambone, 1994; Catrambone ve Holyoak, 1990; Mwangi ve Sweller, 1998; Ward ve Sweller, 1990; Zhu ve Simon, 1987). Ward ve Sweller'a (1990) göre tasarımından veya yapısından dolayı bazı çözümlü örnekler, öğrenenin kısa süreli belleğini beklenilenin aksine olumsuz etkileyebilir. Bu nedenle çözümlü örnekleri kullanırken nasıl tasarlandığı da göz önünde bulundurulmalıdır. Bir sonraki bölümde bu çalışmada kullanılan geleneksel çözümlü örnek ve tamamlamalı çözümlü örnek yöntemlerine ve çözümlü örnek alanyazınında uygulanan diğer yöntem ve tasarım ilkelerine değinilecektir.

2.4. Çözümlü Örnek Uygulama Yöntemleri ve Tasarım İlkeleri

Bu bölümde çözümlü örnek alanyazınında etkililiği sınanmış uygulama yöntemleri ve tasarım ilkeleri ele alınmıştır. Bu çalışmada geleneksel çözümlü örnek ve bir tamamlamalı çözümlü örnek yöntemi olan karartılmış çözümlü örnek yöntemlerinin etkililiği karşılaştırılmıştır. Buna ek olarak bu bölümde yer alan alt hedeflere ayırma, basit ilişkilendirme, basitten karmaşığa doğru sıralama ve açıklamalı yardım ilkeleri çalışma için geliştirilen web ortamında kullanılırken doğru ve yanlış çözümlü örneklerin birlikte kullanılması, imgeleme ve öz açıklama ilkeleri bu çalışmada kullanılmamıştır. Bu ilkelerin kullanılıp kullanılmama durumları ve gerekçelerine ilişkin ayrıntılı bilgilere yöntem bölümündeki öğrenme ortamı başlığı altından ulaşılabilir.

2.4.1. Geleneksel Çözümlü Örnekler (Traditional Worked Examples)

Geleneksel çözümlü örneklerden kasıt çözümlü örneklerin herhangi bir manipülasyon olmadan öğrenenlere sunulmasıdır. Yani problem çözme aşamasından önce benzer problemlere ilişkin kural ve ilkelerin yer aldığı örnekler,

öğrenenlere tam çözümleriyle birlikte sunulur. Alanyazın incelendiğinde birçok farklı çözümlü örnek yöntemi sınanmasına karşı geleneksel çözümlü örneklerin işe koşulması ya da geliştirilen bir yöntemin geleneksel çözümlü örnek yöntemiyle karşılaştırılması bu yöntemin çözümlü örnek araştırmalarındaki etkililiğinin devam ettiğini göstermektedir. Geleneksel çözümlü örnek yönteminde genellikle örnek-problem çiftleri kullanılmaktadır (Sweller ve Cooper, 1985). Örnek-problem çiftlerinde her bir problemden önce bir örnek sunulur ve örnek-problem çiftleri birbirine özdeş problemlerden oluşur. Renkl ve Atkinson (2010) ve Atkinson vd. (2000) tamamlamalı örneklerin ya da alternatifi karartılmış örneklerin geleneksel çözümlü örnek yöntemine göre yeni beceri ediniminde öğrenenlerin bilişsel kapasitesini verimli bir şekilde kullanmada etkili olduklarını belirtmişlerdir.

2.4.2. Tamamlamalı Çözümlü Örnekler (Completion Strategy)

Tamamlamalı çözümlü örnekler, geleneksel çözümlü örneklere bir alternatif olarak ortaya atılmıştır. Burada amaç çözüm adımlarının bir bölümünün boş bırakılarak öğrenenin sürece daha aktif katılımını ve bilişsel kaynaklarını daha verimli kullanmasını sağlamaktır. Tamamlamalı örneklerde öğrenenler tarafından tamamlanması beklenen kısmi çözüm adımları bulunur. Bu yöntem geleneksel çözümlü örnek ile geleneksel problem çözme yöntemleri arasında bir köprü olarak düşünülebilir. Yani geleneksel çözümlü örnekler, tamamlamalı örneklerin tamamı çözümlü hali olarak düşünülebilir. Diğer bir açıdan geleneksel problem çözme yöntemi tamamlamalı örnek yönteminin tamamı çözümsüz hali olarak düşünülebilir. Yapılan çalışmalarda programlama öğretiminde tamamlamalı çözümlü örnek yönteminin geleneksel çözümlü örnek yöntemine göre daha etkili olduğu görülmüştür (van Merrienboer, 1990; van Merrienboer ve Krammar, 1987; Chang, Chiao, Chen ve Hsiao, 2000). Tamamlamalı çözümlü örnek yöntemi boşlukların rastgele bırakıldığı yöntemle uygulanabilirken, adımların sistematik bir şekilde arttırıldığı karartılmış çözümlü örnekler olarak da uygulanabilir.

2.4.2.1. Karartılmış Çözümlü Örnekler (Faded Worked Examples)

Karartılmış çözümlü örnekler de aslında bir tamamlamalı çözümlü örnek yöntemi olarak düşünülebilir. Karartılmış örneklerde ilk olarak tam çözümlü örnek sunulur. Ardından adım adım örnekteki çözümler boş bırakılır, bu işlem bütün adımlar

çıkarılana kadar devam eder. Böylece en son sadece problem kalır ve öğrenenden çözmesi beklenir. Bu şekilde çözümlü örneklerden problem çözmeye geçişte rehberlik azalarak son bulur. Karartılmış örnekler iki farklı yolla uygulanmaktadır. Bunlardan birisi çözüm adımlarındaki boşlukların en sondan geriye doğru arttırılması (backward fading) diğeri de en baştan sona doğru (forward fading) arttırılmasıdır. Yapılan çalışmalar özellikle geriye doğru karartma yönteminin kullanıldığı çözümlü örneklerin hem yakın transfere hem de uzak transfere olumlu etkiler yaptığını göstermiştir (Renkl, 2014). Karartılmış çözümlü örnekler geleneksel örnek-problem çiftinin sunulduğu yönteme göre daha etkili bir stratejidir (Kissane, Kalyuga, Chandler ve Sweller, 2008; Renkl ve Atkinson, 2003). Karartılmış çözümlü örnekler aynı zamanda giderek artan problem çözme becerisinin uygulanmasından dolayı uzmanlığın ters tepme etkisine karşı etkili bir yöntemdir (Kalyuga, 2007). Ayrıca karartılmış çözümlü örneklerin özellikle iyi yapılandırılmış alanlarda daha etkili olduğu belirtilmektedir (Renkl, 2014). Buna ek olarak birçok çalışmada karartılmış çözümlü örneklerin geleneksel çözümlü örneklere göre bilişsel yüklenme ve çalışma süreleri açısından da daha verimli olduğu ortaya konulmuştur (Atkinson, Renkl ve Merrill, 2003; Renkl, Atkinson ve Große, 2004). Ancak yapılan bazı çalışmalarda karartılmış çözümlü örneklerin geleneksel çözümlü örneklere göre daha etkili olmadığı bulgusuna ulaşılmıştır (Demiraslan Cevik ve Andre, 2014; Reisslein, Atkinson, Seeling ve Reisslein, 2006; Schwonke, Renkl, Krieg, Wittwer, Aleven ve Salden, 2009). Bu nedenle hangi yöntemin daha etkili olduğu konusu tartışmaya açıktır.

2.4.3. Diğer Çözümlü Örnek Yöntemleri ve Tasarım İlkeleri

Alt Hedeflere Ayırma İlkesi (Subgoal Labeling)

Çözümlü örneklerin çözüm adımlarının alt hedeflerine ayrılarak sunulması, öğrenenlerin problemin altında yatan mantığı kavramasında yardımcı bir strateji olarak kullanılmaktadır (Catrambone ve Holyoak, 1990; Marguileux, Catrambone ve Guzdial, 2012; Stasko, Catrambone ve Guzdial, 2016). Alt hedefleri anlamak için basit bir program oluşturma adımlarını düşünelim. Örneğin, ilk olarak değişkenler tanımlanır, kullanıcıdan değer girmesi için ekrana uyarı oluşturulur, kullanıcıdan alınan sayı değişkenlere atanır, bu değişkenler üzerinden işlem gerçekleştirilir ve sonuç ekrana yazdırılır. Ön bilgisi yetersiz olan öğrenciler için bu

programı oluşturmak daha karmaşık bir kod yapısı gerektirse de temelde bu alt hedefler problemin çözümünü oluşturur. Yani alt hedefler öğrenen için çözüm adımlarını gruplayıp, vurgulayarak problemin yapısını anlamasında bir taslak sunar.

Çözümlü örneklerde alt hedef kullanımı matematik, fizik gibi alanlarda yaygın bir şekilde kullanılmakla birlikte son yıllarda programlama öğretiminde de kullanılmaktadır (Margulieux, Catrambone ve Guzdial, 2012; Margulieux, Catrambone ve Guzdial, 2013; Schaeffer, 2015; Morrison, Margulieux ve Guzdial, 2015; Margulieux, Catrambone ve Guzdial, 2016; Margulieux ve Catrambone, 2016). Programlama öğretiminde yapılan çalışmalarda çözümlü örneklerde alt hedef kullanıldığı durumlarda öğrenenlerin sarf ettiği bilişsel çaba ve çalışma süreleri açısından anlamlı bir farklılık görülmese de alt hedeflerin kullanıldığı gruplar, alt hedeflerin kullanılmadığı gruplara göre daha yüksek başarı göstermişlerdir. Ayrıca çözümlü örneklerde alt hedef kullanıldığında öğrenenlerin çözüm adımlarını daha nitelikli bir şekilde açıkladığı görülmüştür (Margulieux, ve Catrambone, 2016). Sonuç olarak, programlama öğretiminde çözümlü örnek adımlarının alt hedeflerine ayrılması ve alt hedeflerin vurgulanması önemli bir pedagojik strateji olarak görülmektedir.

Öz Açıklama İlkesi (Self-explanation Principle)

Öz açıklama ilkesi, öğrenenlerin çözümlü örneklere çalışırken bilginin pasif alıcısı olmak yerine, çözüm adımlarını açıklayarak örneklere çalışırken daha aktif bir süreç geçirmesinin sağlanması ile ilgilidir. Öz açıklama ilkesini ilk olarak Chi, Bassok, Lewis, Reimann ve Glaser (1989) kullanmışlardır. Chi ve arkadaşları, öğrenenlerin çözüm adımlarını açıklamalarının problemin yapısını daha iyi anlayarak daha derin öğrenmelerine yol açtığını ileri sürmüşlerdir. Sweller (2010) ise öz-açıklama sürecinde öğrenenin bilişsel kaynaklarını konuya yönelik daha etkili kullandığını ve dolayısıyla konu dışı yükün azalmasına yardımcı bir strateji olduğunu belirtmiştir.

Ancak öğrenenler bazen öz açıklama sürecinde üretken olmakta zorlanıp doğru ifadeleri kullanamayabilirler (Atkinson vd., 2000; Berthold ve Renkl, 2009; Renkl, 2002). Bunun için öz açıklamaların niteliğini arttırmaya yönelik birtakım stratejiler kullanılmaktadır. Bu stratejiler; öz açıklama süreci ile ilgili eğitim verilmesi, yapısal

manipülasyonlar ile öz açıklama sürecinin desteklenmesi ve sosyal pekiştireçlerle öz açıklama sürecinin teşvik edilmesidir (Atkinson vd., 2000; Renkl, 2014). Öz açıklama sürecinden önce bununla ilgili bir eğitim verilmesi daha etkili bulgulara yol açsa da yeterince etkili olmadığı görülmüştür (Nathan, Mertz ve Ryan, 1994). Aynı zamanda bu süreçle ilgili bir eğitim verilmesi verimlilik açısından düşünüldüğünde olumsuz sonuçlara sebep olabilir. Bir diğer yöntem olan yapısal manipülasyonlara, alt hedeflerin belirlenmesi (Catrambone, 1996), tamamlamalı örneklerin kullanımı (Stark, 1999) ve bölünmüş dikkat etkisinden kaçınmak için bütünleşik örneklerin kullanımı (Mwangi ve Sweller, 1998) örnek gösterilebilir. Yapısal manipülasyonlar öz açıklama sürecinin niteliğini arttırmada en etkili strateji olarak görülmektedir (Atkinson vd., 2000). Bir diğer yöntem olan sosyal pekiştireçlerin kullanımında öğrenenler çözüm adımlarını bir başka öğrenene açıklamaktadır. Bu yöntemin kullanımı özellikle öğrenenlerin stres yapmasından ve yeterli ön bilgilerinin olmamasından dolayı öğrenme açısından olumsuz sonuçlara yol açmıştır (Mwangi ve Sweller, 1998; Webb, 1991). Renkl (1997) bu yöntemin etkili olması için öğrenenlerin hem içerik konusunda ön bilgilerinin olması gerektiğini hem de öğretim deneyimi yaşamış olması gerektiğini belirtmiştir. Aksi takdirde öğrenenler için bu durum stres oluşturabilmekte ve aşırı bilişsel yüklenmeye sebep olabilmektedir.

Her ne kadar öz açıklama sürecinin kullanımının etkili olduğu görülse de bu yöntemin kullanımı oldukça risklidir. Yukarıda belirtilen yöntemler uygulansa bile konu ile ilgili yeterli deneyimi olmayan öğrenenler için öz açıklama sürecinde aşırı bilişsel yüklenme oluşabilir. Bu durumda öğrenenler genellikle çözüm adımlarını tekrar okuma ya da sözcüklerin yerlerini değiştirerek açıklama gibi istenmeyen sonuçlara yönelebilirler (Stark, 1999). Aynı zamanda karmaşık görevlerde öz açıklama kullanımı da aşırı bilişsel yüklenme ile sonuçlanarak öğrenmeyi olumsuz etkilemektedir (Renkl, 2014).

Açıklamalı Yardım İlkesi (Explanation-help Principle)

Açıklamalı yardım ilkesi, öğrenenlere çözümlü örneklere çalışırken odaklanmaları gereken kural ve ilkelerin metinsel olarak açıklanması ile ilgilidir. Açıklamalı yardım ilkesi belirli bir kuralı ve çözüm yolu olan iyi yapılandırılmış alanlarda etkilidir. Öz açıklamaların kullanıldığı örnekler, metinsel açıklamalarla desteklenen çözümlü

örneklere göre daha etkili bir yöntemdir (Schworm ve Renkl, 2006). Ancak yukarıda belirtildiği gibi öz açıklama kullanımı yeterli deneyimi olmayan öğrenenlerde olumlu sonuçlar doğurmayabilir. Bu durumda açıklamalı yardım ilkesinin kullanımı daha etkilidir (Schunk ve Zimmerman, 2007). Açıklamalı yardım ilkesi bir problemin çözümünün tamamının açıklanması şeklinde de kullanılabilirken, problemin çözümü ile ilgili yapısal ipuçları verilerek de kullanılabilir (Renkl, 2014). Buna ek olarak açıklamalı yardımın bağlamdan bağımsız, konu ile ilgili genel bilgileri veya ipuçlarını sağlaması önem arz etmektedir. Catrambone (1995), konu alanıyla ilgili daha genel açıklamaların sunulduğu öğrenenlerin, daha spesifik açıklamaların sunulduğu öğrenenlere göre öğrenmenin transferi konusunda daha yüksek performans gösterdikleri bulgusuna ulaşmıştır. Ancak öğrenenlerin açıklayıcı metinleri problemlere uygulamasının oldukça güç olduğu belirtilmektedir (Eiriksdottir ve Catrambone, 2011; VanLehn, Jones ve Chi, 1992). Bunun için, açıklayıcı metinlerin özellikle alt hedefler ve çözümlü örneklerle birlikte kullanılmasının en etkili yöntem olabileceği öngörülmektedir (Margulieux ve Catrambone, 2016).

Basitten Karmaşığa Sıralama İlkesi (Simple to Complex Strategy)

Yeterli deneyimi olmayan öğrenenler için konu dışı bilişsel yükü ne kadar azaltırsak azaltalım, asıl bilişsel yük karmaşık görevler için yüksek olabilir. Van Merrienboer ve Sweller (2010) bunu karmaşık görevlerde asıl bilişsel yükü oluşturan yüksek birim etkileşimliliğinden kaynaklanabileceğini belirtmişlerdir. Bu durumda içeriğin basitten karmaşığa doğru sıralanması önem arz etmektedir. Burada amaç asıl bilişsel yükün düşürülmesi değildir, çünkü sunulan karmaşık görevin daha basit hale getirilmesi öğrenmeyi olumlu yönde etkilemeyebilir. Asıl bilişsel yükün azalması, öğrenenlerin şema oluşumunu sağlayan etkili bilişsel yükü arttırmak için harcadıkları bilişsel çabanın da düşeceği anlamına gelir. Burada amaç sunulan içeriğin birim etkileşimliliğini kademeli bir şekilde arttırmaktır. Sunulan içeriğin ilk aşamada basit olması, yani düşük birim etkileşimliliğine sahip olması ardından gittikçe daha karmaşık, yani yüksek birim etkileşimliliğine sahip olması her aşamada aynı birim etkileşimliliğine sahip materyallerin sunumuna göre daha etkili bir yöntemdir (Pollock, Chandler ve Sweller, 2002).

Basit İlişkilendirme İlkesi (Easy-mapping Principle)

Basit ilişkilendirme ilkesi, problemin çözüm adımlarının gösteriminde farklı kaynakların ilişkilendirilerek öğrenenin konu dışı bilişsel yüklenmesinin azaltılmasıyla ilgilidir. Çözümlü örnek alanyazınında çözüm adımlarının ilişkilendirilmesi, Tarmizi ve Sweller (1988) tarafından bölünmüş dikkat etkisine karşı metinlerin ve şekillerin entegre biçiminde sunulması, biçem etkisi olarak bilinen görsel ve işitsel kaynakların birlikte kullanılması (Mayer ve Moreno, 2003), birbiriyle ilişkili bileşenlerin renk kodlamasıyla sunulması (Berthold ve Renkl, 2009) gibi yollarla uygulanmaktadır. Alanyazında birçok çalışmada farklı enformasyon kaynaklarının ilişkilendirildiği durumlarda öğrenenlerin daha az bilişsel çaba ile daha yüksek performans gösterdikleri görülmüştür (Berthold ve Renkl, 2009; Renkl, 2014; Renkl, 2017).

Bölünmüş dikkat etkisi aynı kanala (görsel veya işitsel) hitap eden farklı enformasyon kaynaklarının bütünleşik olarak verilmesi ile ilgilidir. Örneğin geometri probleminde şeklin ve metinlerin ayrı bir şekilde sunulması öğrenmeyi olumsuz etkileyebilir. Buna karşılık Tarmizi ve Sweller (1988) ile Ward ve Sweller (1990) yaptıkları çalışmalarda geometri öğretiminde iki farklı enformasyon kaynağını entegre ederek olumlu sonuçlar elde etmişlerdir. Chang, Hsu ve Yu (2011) bu yöntemi programlama öğretiminde uygulamış ve benzer sonuçları elde etmiştir. Tarmizi ve Sweller (1988) bölünmüş dikkat etkisinin hesaba katılmadığı durumlarda çözümlü örnek kullanımının tersine bir etki yaratacağını ve bu durumda problem çözme uygulamalarının daha etkili olabileceğini belirtmiştir.

Biçem etkisi farklı kanallara (görsel ve işitsel) hitap eden iki enformasyon kaynağının birlikte kullanımı ile ilgilidir. Mousavi, Low ve Sweller (1995) içeriğin metinsel olarak sunulmasına ek olarak sesli anlatımın da eklenmesinin daha etkili bir yöntem olduğunu belirtmişlerdir. Ancak Jeung, Chandler ve Sweller (1997) karmaşık görevlerde içeriğin hem görsel hem de işitsel olarak sunulmasının, sadece görsel olarak sunulmasına göre daha etkili olmadığını göstermiştir. Ayrıca içeriğin görsel olarak sunulmasına göre işitsel olarak sunulması bazı teknik detaylar (sınıf ortamında çalışılıyorsa kulaklık gibi) gerektirdiğinden tercih edilen bir yöntem olmayabilir (Renkl, 2005). Buna ek olarak öğrenenlerin sunulan materyale aşina olmadığı ve konunun karmaşık olduğu durumlarda sadece görsel kanala uygun sunumun gerçekleştirilmesi daha etkili olmaktadır (Renkl, 2005).

Renk kodları yardımıyla enformasyon kaynaklarının ilişkilendirilmesi özellikle karmaşık bilgilerin sunulduğu durumlarda etkili bir stratejidir (Berthold ve Renkl, 2009; Keller, Gerjetz, Scheiter ve Garsoffky, 2006; Ozcelik, Karakus, Kursun ve Cagiltay, 2009). Renk kodları kullanımında dikkat edilmesi gereken tutarlılıktır. Bir örnekte belirli bir ögeyi ifade eden renk, devam eden örnek veya problemde de aynı ögeyi ifade etmelidir. Renk kodları konu ile ilgili öğrenmeyi olumsuz etkileyecek arama stratejilerinden kaynaklanan konu dışı bilişsel yüklenmeyi azaltırken öğrenenin dikkatini gerekli enformasyon kaynaklarına odaklamasını sağladığından etkili yükü de arttırmaktadır (Mayer, 2001).

İmgeleme İlkesi (Imagery Principle)

İmgeleme ilkesi, öğrenenin çözüm adımlarına çalıştıktan sonra bu adımları zihinde canlandırması ile ilgilidir. Cooper, Tindall-Ford, Chandler ve Sweller (2001) çözümlü örneklere çalıştıktan sonra çözüm adımlarını hayal etmenin öğrenme üzerinde olumlu etki yaptığını belirtmişlerdir. Benzer şekilde Renkl (2014) çözüm adımlarını zihinde canlandırmanın problem çözme ile eş değer etki yaratabileceğini belirtmiştir. Ancak imgeleme yoluyla çözümlü örneklerden öğrenme stratejisi yeterli deneyime sahip olmayan öğrenenler üzerinde etkili olmamıştır (Cooper vd., 2001). Dolayısıyla bu yöntemin kullanımında öğrenenlerin konu hakkında ön bilgilerinin olması gerekmektedir.

Doğru ve Yanlış Çözümlü Örneklerin Birlikte Kullanımı (Studying Error Principle)

Örnek problemleri hem doğru hem de yanlış çözümleriyle birlikte sunmak çözümlü örnek alanyazınında kullanılan bir diğer stratejidir. Bu durumda öğrenenlerin, sadece doğru olan çözüm adımlarına değil yanlış çözüm adımlarına da çalışarak, problem çözümünde karşılaşabilecekleri olası yanlışlara karşı hazırbulunuşluğu artabilir. Siegler ve Chen (2008) öz açıklama sürecinde doğru ve yanlış çözümlerin birlikte kullanımının sadece doğru çözüm adımlarının kullanımına göre daha etkili bir strateji olduğu bulgusuna ulaşmıştır. Öğrenenlerin yanlış çözümleri açıklaması daha sonraki problem çözümünde karşılaşacakları olası hatalardan kaçınmasına yardımcı bir stratejidir (Durkin ve Rittle-Johnson, 2012). Ancak öğrenenlerin yanlış çözümlü örneklere çalışması öğrenmenin ilk aşamasında olumlu sonuçlar

doğurmayabilir (Große ve Renkl, 2007). Bu nedenle bu yöntemde de öğrenenlerin yeterli ön bilgiye sahip olması bir ön koşul olarak düşünülebilir. Ayrıca bu yöntemin etkili olması için yanlışların daha belirgin hale getirilmesi (Große ve Renkl, 2007), çözümlerin neden doğru veya yanlış olduğunun metinsel açıklamalarla belirtilmesi (Stark, Kopp ve Fischer, 2011) gibi stratejilerin uygulanması önerilmektedir.

2.5. Programlama Öğretiminde Çözümlü Örnek Kullanımı ile İlgili Araştırmalar

Bu bölümde, alan yazında programlama öğretiminde çözümlü örnek kullanımıyla ilgili yapılan çalışmalar kronolojik bir sırada verilmiştir.

Van Merrienboer (1990), programlamaya giriş dersi kapsamında 57 üniversite öğrencisi ile yürüttüğü çalışmada programlama öğretiminde tamamlamalı çözümlü örnek yönteminin öğrenenlerin programlama performansı üzerindeki etkilerini incelemeyi amaçlamıştır. Katılımcılar deney ve kontrol grubu olmak üzere iki gruba rastgele atanmıştır. Deney grubuna tamamlamalı çözümlü örnekler sunulurken, kontrol grubunda geleneksel problem çözme yöntemi uygulanmıştır. 10 haftalık uygulama sürecinde COMAL-80 programlama dili kullanılmıştır. Deney ve kontrol grubuna uygulama öncesi programlama ile ilgili temel kavramlar konusunda kitapçıklar dağıtılmıştır. Uygulama sonunda her iki gruba da 18 sorudan oluşan çoktan seçmeli program oluşturma testi uygulanmıştır. Çalışmada ayrıca öğrencilerin programı tamamlama süreleri, algılanan zorluk ve var olan programı yorumlama becerileri de ölçülmüştür. Sonuç olarak tamamlamalı çözümlü örnek yöntemi program oluşturma sürecinde daha etkili olmuştur. Algılanan zorluk açısından bir farklılık görülmemiştir. Çalışma süreleri değerlendirildiğinde geleneksel problem çözme yönteminin uygulandığı grubun daha kısa sürede programı oluşturduğu görülmüştür. Bunun nedeni olarak da materyalin tasarımı gösterildiği için beklenen bir sonuç olarak belirtilmiştir. Ayrıca öğrenenler için her iki yöntem de aynı düzeyde merak uyandırıcı olmuştur. Van Merrienboer, programlama öğretiminde tamamlamalı çözümlü örnek yönteminin geleneksel problem çözme yöntemi için iyi bir alternatif olabileceğini ileri sürmüştür.

Hohn ve Moraes (1998), 102 öğrenciyle programlamaya giriş dersi kapsamında bir uygulama gerçekleştirmiş ve açıklayıcı metinlerle desteklenmiş çözümlü örneklerin etkililiğini sınamıştır. Çalışmada öğrenciler 3 gruba ayrılmış, bir gruba geleneksel

problem çözme yöntemi, bir gruba geleneksel çözümlü örnek yöntemi bir gruba da açıklayıcı metinlerle desteklenmiş çözümlü örnek yöntemi uygulanmıştır. Çalışmada Pascal programlama dili tercih edilmiştir. Öğrenenlerin performansı değerlendirilirken verilen kodları gruplama ve açıklama problemleri ve Pascal yeterlilik testi kullanılmıştır. Araştırma sonucuna göre açıklayıcı metinlerle desteklenmiş çözümlü örnek yöntemi diğer iki yönteme göre kodların gruplandırılması, açıklanması ve son test performansı değerlendirildiğinde daha etkili olmuştur. Ayrıca açıklayıcı metin destekli çözümlü örneklere çalışan öğrenenler son test esnasında açıklayıcı metinleri kullandıklarını belirtirken, geleneksel problem çözme grubundakiler uygulama sırasında sunulan materyali son test esnasında kullanmadıklarını belirtmişlerdir. Araştırmacılar elde edilen sonuçların, kullanılan yöntemin konu dışı bilişsel yükü azaltmasından ve öğrenenlerin kod yapılarına odaklanmaktan ziyade problemin nasıl çözüldüğüne odaklanmasından kaynaklanmış olabileceğini belirtmişlerdir.

Chang, Chiao, Chen ve Hsiao (2000), 45 lise öğrencisiyle yaptıkları çalışmada programlama öğretiminde tamamlamalı çözümlü örnek yönteminin etkililiğini sınamışlardır. Katılımcılar 2 gruba rastgele dağıtılarak, bir gruba tamamlamalı çözümlü örnek yöntemine göre tasarlanan bir ortam sunulurken bir gruba problemler verilerek çözmeleri beklenmiştir. Çalışma için Basic programlama dili tercih edilmiş ve örneklere çalışılması için web tabanlı bir sistem geliştirilmiştir. Oliver (1993) tarafından geliştirilen “Algoritma Süreci Testi” katılımcılara uygulanarak başarı ölçütü olarak değerlendirilmiştir. Çalışmada ayrıca öğrenenlerden uygulanan yönteme ilişkin görüş alınmıştır. Araştırma sonucuna göre programlama öğretiminde tamamlamalı çözümlü örnek yöntemi, geleneksel problem çözme yöntemine göre daha etkili olmuştur. Ayrıca katılımcılar tamamlamalı çözümlü örneklerden öğrenmenin daha motive edici ve verimli olduğunu belirtmişlerdir.

Lee (2013), 147 lise öğrencisiyle yürüttüğü çalışmasında programlama öğretimi için öz açıklama ilkesine dayalı çözümlü örnek yöntemi ile geleneksel problem çözme yöntemini karşılaştırmıştır. Lee, ayrıca öğrenenlerin bu iki yöntemden hangisini tercih ettiklerini ve nedenlerini de analiz etmiştir. Çalışma için çözümlü örnekler ve çoklu ortam tasarım ilkelerine göre hazırlanan web tabanlı e-öğrenme ortamı kullanılmıştır. Grupların başarı ölçütü için son test performansı ve uygulama

sonunda katılımcıların yöntemlere ilişkin tutumları incelenmiştir. Çalışma sonucunda son test performansına göre bu iki yöntemin öğrenme çıktıları açısından bir farklılık göstermediği belirlenmiş, ancak öğrenenler öz açıklama ilkesine dayalı çözümlü örneklerden daha iyi öğrendiklerini ifade etmişlerdir.

Margulieux, Catrambone ve Guzdial (2013), programlama öğretiminde alt hedeflere ayrılmış çözümlü örneklerin K-12 öğretmenlerinin programlama performansına etkisini incelemeyi amaçlamışlardır. Çalışmaya 18 K-12 öğretmeni katılmış ve katılımcılar deney ve kontrol grubu olmak üzere 2 gruba rastgele atanmıştır. Çalışma için bir öğrenme yönetim sistemi kullanılmış ve App Inventor programı kullanılmıştır. Gruplardan birine alt hedeflere ayrılmış çözümlü örnek yöntemine göre tasarlanan e-içerikler sunulurken diğer gruba geleneksel çözümlü örnek yöntemine göre tasarlanan e-içerikler sunulmuştur. K-12 öğretmenlerinin programlama performansını ölçmek için hem süreç içinde hem de süreç sonunda problemler sunulmuştur. Aynı zamanda e-içeriklere çalışma süreleri analiz edilmiş ve her bir uygulama sonunda algılanan zorluk düzeyleri ölçülmüştür. Araştırma sonucuna göre alt hedeflere ayrılmış çözümlü örnekler geleneksel çözümlü örneklerden daha etkili olmuştur. Çalışma süreleri ve algılanan zorluk düzeyleri incelendiğinde aralarında anlamlı bir farklılık görülmemiştir. Araştırmacılar bu sonucun e-öğrenme ortamında öğretmenlerin olmadığı bir öğrenme sürecinde alt hedeflerin katılımcılar için rehber olmasından kaynaklanabileceğini öne sürmüşlerdir.

Si, Kim ve Na (2014), 96 üniversite öğrencisiyle yürüttükleri çalışmada C programlama dili öğretimi için öğrenenlerin karartılmış çözümlü örneklerden öğrenmesinde uyarlanabilir öğretimin ve çoklu ortam ilkesinin verimliliğini incelemişlerdir. Katılımcılar uygulama için 4 gruba rastgele atanmıştır. Uyarlanabilir öğrenme ortamında katılımcılar 9 adımdan oluşan problem çözme sürecinde belirlenen ölçütlere göre beklenen performansa ulaşamadıklarında aynı adımı tekrarlarken, uyarlanabilir olmayan grupta öğrenenin yanıtlarına bakılmaksızın bir sonraki aşamaya geçilmiştir. İki grup anlatımın sadece görsel materyal sunularak ve uyarlanabilir öğrenme ortamının kullanılıp kullanılmamasına göre, iki grup da hem görsel hem sesli anlatımın sunularak uyarlanabilir öğrenme ortamının kullanılıp kullanılmamasına göre ayrılmıştır. Uygulama ortamı olarak web tabanlı bir sistemin geliştirildiği çalışmada uygulamanın etkililiğini yani nasıl

uyarlandığını ölçmek için araştırmacılar tarafından geliştirilen başarı testi ve bilişsel yükü ölçmek için 9'lu derecelendirme ölçeği (Paas ve van Merrienboer, 1993) kullanılmıştır. Çalışmanın sonucunda uyarlanabilir öğrenme ortamının kullanıldığı gruplarda daha verimli öğrenme çıktıları elde edilmiştir. Ayrıca sadece görsel olarak tasarlanan örneklerle hem görsel hem işitsel olarak tasarlanan örnekler arasında anlamlı bir farklılık bulunmamıştır. Ancak görsel-işitsel gruptaki öğrenenler daha az bilişsel çabayla daha fazla bilgi edinmişlerdir. Sonuç olarak uyarlanabilir öğrenme ortamının kullanımı ve hem görsel hem de işitsel olarak tasarlanan örnekler bilişsel yükü azaltmış, şema edinimini ve yönetimini kolaylaştırmıştır.

Abdul-Rahman ve Boulay (2014), çözümlü örnekler aracılığıyla programlama öğreniminde öğrenme stillerini (aktif ve yansıtıcı öğrenen) hesaba katarak tamamlamalı örnekler, yapının vurgulandığı örnekler ve bu iki yöntemin sırayla sunulduğu eşlenik örnekler olmak üzere üç farklı çözümlü örnek yönteminin verimliliğini incelemeyi amaçlamışlardır. Çalışmaya ilk defa programlama dersi alan 110 üniversite öğrencisi katılmış ve katılımcılar uygulanan yöntemlere göre 3 gruba dağıtılmıştır. Gruplara deney öncesi öğrenme stili envanteri (Felder ve Soloman, 2011) uygulanmış ve grupların aktif ve yansıtıcı öğrenenler açısından denk olmaları sağlanmıştır. Uygulama için LECSES adında bir sistem geliştirilmiştir. Yapının vurgulandığı örneklerde öğrenenlerin LECSES sisteminde verilen örneklerin altında yatan mantığı açıklamaları beklenmiştir. Tamamlamalı örneklerde öğrenenlerin LECSES sisteminde kısmi çözümü verilen örnekleri tamamlamaları beklenmiştir. Eşlenik örneklerde öğrenenlerin uygulamanın ilk aşamasında kısmi olarak verilen örnekleri tamamlamaları sonrasında da örneklerin altında yatan mantığı açıklamaları beklenmiştir. Öğrenenlerin bilişsel yüklenme düzeylerini ölçmek amacıyla Kalyuga, Chandler ve Sweller (1998) tarafından geliştirilen 5'li derecelendirme ölçeği kullanılmıştır. Öğrenenlere deney öncesi ön test uygulanmış ve edindikleri bilişsel şemaların niteliği son test performansıyla değerlendirilmiştir. Sonuç olarak eşlenik örneklerin diğer iki yönteme göre daha verimli olabileceği öngörülse de tamamlamalı örneklerin daha verimli olduğu belirlenmiştir. Araştırmacı bunun nedeninin eşlenik örneklerde bir yöntemden diğerine geçişin öğrenenleri olumsuz etkilemesinden kaynaklanabileceğini belirtmiştir. Yapının vurgulandığı örneklerin diğer iki yönteme göre daha az verimli

olmasının nedeni olarak da yeterli deneyimi olmayan öğrenenlerde çözümlerin açıklanmasının aşırı bilişsel yüklenmeye sebep olmasından kaynaklandığı ileri sürülmüştür. Ayrıca her bir yöntem grubundaki öğrenme stillerine bakıldığında aktif ve yansıtıcı öğrenenler için hem bilişsel yüklenme düzeylerinde hem de başarılarında anlamlı bir farklılık görülmemiştir.

Schaeffer (2015), 132 üniversite öğrencisi ile yürüttüğü çalışmada açıklayıcı metinler ile çözümlü örneklerin sunuluş sırasının programlama öğretimindeki etkilerini incelemeyi amaçlamıştır. Çalışma için 2x2 (“materyal sunuluş sırası” x “alt hedeflere ayrılıp ayrılmama durumu”) faktöriyel desen kullanılmıştır. Katılımcılara programlama öğretimi App Inventor programı aracılığıyla gerçekleştirilmiştir. Materyal sunuluş sırasında iki gruba çözümlü örnekler açıklayıcı metinlerden önce verilirken, iki gruba açıklayıcı metinlerden sonra verilmiştir. Araştırma için geliştirilen performans testi iki puanlayıcı tarafından değerlendirilmiştir. Çalışmanın sonucuna göre hem materyalin sunuluş sırası hem de çözümlü örneklerin alt hedeflere ayrılıp ayrılmaması açısından gruplar arasında anlamlı bir farklılık görülmemiştir. Araştırmacı bu sonucun açıklayıcı metinlerin yazılı olarak değil video aracılığıyla verilmiş olmasından kaynaklanabileceğini ileri sürmüştür. Bunun nedeni olarak da videolardaki alt hedeflerin yazılı olarak verilen alt hedeflerle aynı ölçüde verimli olmaması ihtimalinden kaynaklanabileceğini ileri sürmüşlerdir.

Harms, Rowlett ve Kelleher (2015), yaşları 10 ile 15 arasında olan 27 katılımcı ile yaptıkları araştırmada programlama öğretiminde tamamlamalı çözümlü örnekler ile özel öğretici yazılımların programlama performansı üzerindeki etkilerini incelemişlerdir. Araştırma için her iki yönteme göre bir program geliştirme yazılımı hazırlanmıştır. Katılımcılar deney ve kontrol grubu olmak üzere 2 gruba rastgele atanmıştır. Çalışmada programlama performansı 4 sorudan oluşan transfer testi ile ölçülmüştür. Ayrıca katılımcıların içsel motivasyonu, bilişsel yükü ve uygulama süreleri de analiz edilmiştir. Araştırma sonucuna göre tamamlamalı çözümlü örneklerin sunulduğu grup diğer gruba göre %23 daha az zaman harcarken, %50 daha fazla bilişsel çaba sarf etmiştir. Tamamlamalı çözümlü örnekler ile özel öğretici yazılımlara göre tasarlanan öğrenme ortamları içsel motivasyon açısından değerlendirildiğinde gruplar arasında anlamlı bir farklılık görülmemiştir. Transfer testi sonuçları analiz edildiğinde tamamlamalı çözümlü örnek yönteminin daha etkili olduğu bulgusuna ulaşılmıştır.

Vieira, Yan ve Magana (2015), programlamaya giriş dersinde 35 üniversite öğrencisi ile programlama öğretiminde çözümlü örnek kullanımını incelemişlerdir. Araştırma 3 aşama olarak uygulanmış ve katılımcılar iki gruba ayrılmıştır. Deney grubuna dersin uygulama aşamasında çözümlü örnekler sunulurken kontrol grubuna sadece problem sunularak öğrenenlerden çözmeleri beklenmiştir. Çalışmada ayrıca deney ve kontrol grupları kendi içinde programlama deneyimi olan ve olmayanlar olarak ayrılarak analiz edilmiştir. Örnekler, Visual Studio programında çözümüyle birlikte akış şemaları da gösterilerek sunulmuştur. Sunulan kodlara çözümlü örnek grubu kod parçaları ile ilgili açıklamalar girmişlerdir. Araştırmada nicel veri olarak ön test - son test, çözümlü örnek kullanımına ilişkin tutum ölçeği ve uygulama esnasındaki puanlar, nitel veri olarak açık uçlu sorular ve örneklere girilen açıklamalar değerlendirilmiştir. Sonuç olarak araştırmanın ilk iki aşamasında gruplar arasında bir farklılık görülmezken son aşamasında çözümlü örnek yönteminin kullanıldığı grup daha iyi performans göstermiştir. Ayrıca yeterli deneyimi olmayan öğrenenlerin çözümlü örneklerden daha iyi öğrendiği görülmüştür. Çalışmada çözümlü örnek olarak sunulan kodların öğrenciler tarafından detaylı bir şekilde açıklanmasının etkili bir strateji olduğu önerilmiştir.

Margulieux, Catrambone ve Guzdial (2016), programlama alanında yeterli deneyimi olmayan öğrenenlerde alt hedeflere ayrılmış çözümlü örnek ile geleneksel çözümlü örnek yöntemlerinin etkililiğini karşılaştırmak için bir dizi deney uygulamışlardır. Kırk psikoloji öğrencisine uygulanan ilk deneyde alt hedeflere ayrılmış çözümlü örnek ile geleneksel çözümlü örnek yöntemleri App Inventor programı kullanılarak uygulanmıştır. İkinci deneyde öğrenenlerden farklı olarak çözümlü örnek adımlarını açıklamaları istenmiştir, verilerin analizinin güçlüğü dikkate alınarak bu çalışmaya 12 öğrenci katılmıştır. Üçüncü deneyde ise ilk deneyde olduğu gibi alt hedeflere ayrılmış örnekler ile geleneksel çözümlü örnek karşılaştırılmış, farklı olarak öğrenciler değil K-12 öğretmenlerine uygulanmış ve öğretim laboratuvar ortamında değil e-öğrenme ortamında gerçekleştirilmiştir. Bu deneyde ilk aşamada 27 öğretmen katılmasına rağmen uygulamanın tamamına katılan 10 öğretmenden elde edilen veriler analiz edilmiştir. Üç deneyde de katılımcılara bir dizi problem sunulmuş ve öğrenenlerden çözmeleri beklenmiştir. Probleme verilen cevaplar iki puanlayıcı tarafından değerlendirilerek, kullanılan

yöntemlerin etkililiğinde değerlendirme ölçütü olarak baz alınmıştır. Araştırmanın sonucuna göre alt hedeflere ayrılmış çözümlü örnek, geleneksel çözümlü örnek yöntemine göre hem öz açıklama sürecinin dahil edildiği deneyde, hem öz açıklama sürecinin dahil edilmediği deneyde daha etkili bir strateji olmuştur. Ayrıca bu bulgulara katılımcıların öğretmenlerden oluştuğu deneyde de ulaşılmıştır.

Margulieux ve Catrambone (2016), programlama öğretiminde 240 üniversite öğrencisi ile alt hedeflere ayrılmış açıklayıcı metin ve alt hedeflere ayrılmış çözümlü örnek yöntemlerinin etkililiğini araştırmıştır. İki deneyden oluşan çalışmada her iki deneyde de 4 grup oluşturulmuştur. Bu gruplar alt hedeflere ayrılmış çözümlü örnek ve alt hedeflere ayrılmamış çözümlü örnek yöntemleri ile alt hedeflere ayrılmış açıklayıcı metin ve alt hedeflere ayrılmamış açıklayıcı metin yöntemlerinin kullanılmasına göre oluşturulmuştur. İkinci deneyin ilk deneyden tek farkı katılımcıların değerlendirme esnasında çalışılan materyallere bakabilmesi olmuştur. Çalışmada başarı ölçütü olarak bir dizi problem ele alınmıştır. Katılımcıların yanıtları iki puanlayıcı tarafından değerlendirilmiştir. Çalışmada ayrıca verimliliği ölçmek için problem çözme süreleri analiz edilmiştir. Yapılan deneylerin sonucuna göre alt hedeflere ayrılmış açıklayıcı metinler, ancak alt hedeflere ayrılmış çözümlü örnek yönteminin kullanımıyla birlikte daha etkili olmuştur. Alt hedeflere ayrılmış açıklayıcı metinler öğrenenlerin programlama esnasında süreci doğru açıklayabilmeleri konusunda yardımcı bir strateji olurken, alt hedeflere ayrılmış çözümlü örnekler süreci uygulayabilme konusunda yardımcı bir strateji olmuştur. Ayrıca alt hedeflerin hem açıklayıcı metinler hem de çözümlü örnekler de kullanımı, yalnızca çözümlü örneklerde kullanımına göre daha etkili bir strateji olmuştur.

Margulieux, Morrison, Guzdial ve Catrambone (2016), 120 üniversite öğrencisi ile yaptıkları çalışmada programlamada kullanılan döngüler konusunun öğretiminde öz açıklamalara dayalı alt hedeflere ayrılmış çözümlü örnek yönteminin etkililiğini sınamışlardır. Katılımcılar 6 gruba rastgele eşit olarak (n=20) atanmıştır. Gruplar 3x2 (“alt hedeflerin uygulanma durumu” x “yakın ve uzak transferi ölçen problemler”) faktöriyel desene uygun olarak oluşturulmuştur. Alt hedeflerin uygulanma durumunda grupların birine alt hedeflere ayrılmamış çözümlü örnekler, birine alt hedeflere tasarımcılar tarafından ayrılmış çözümlü örnekler sunulurken, bir gruba çözümlü örnekler sunularak alt hedefleri katılımcıların oluşturması

beklenmiştir. Bu şekilde öz açıklama sürecini tetiklemeyi amaçlamışlardır. Uygulama sürecinde katılımcılara 3 çözümlü örnek ve 3 problem sunulmuştur. Uygulama sonrasında 4 sorudan oluşan performans testi uygulanmıştır. Performans testindeki 2 soru yakın transferi ölçerken 2 soru uzak transferi ölçecek şekilde tasarlanmıştır. Çalışmada ayrıca uygulama sürecindeki problemlerin çözüm süresi ve alt hedeflerin açıklandığı grubun verdikleri yanıtlar da analiz edilmiştir. Yapılan analizler sonucunda yakın transferde yani sunulan örneklere özdeş problemlerin sunulmasında en etkili yöntem tasarımcı tarafından oluşturulan alt hedeflere ayrılmış çözümlü örnek yöntemi olurken en az etkili yöntem alt hedeflerin öğrenen tarafından oluşturulduğu çözümlü örnek yöntemi olmuştur. Uzak transferde yani sunulan örneklerden daha farklı olarak oluşturulan problemlerin sunulması durumunda en etkili yöntem alt hedeflerin öğrenen tarafından oluşturulduğu çözümlü örnek yöntemi olurken, en az etkili yöntem ise alt hedeflerin uygulanmadığı çözümlü örnek yöntemi olmuştur. Alt hedeflerin öğrenenler tarafından oluşturulduğu grupların açıklamalarının niteliği bağlama bağlı oluşturulan açıklamalar ve bağlamdan bağımsız oluşturulan açıklamalar olarak değerlendirilmiştir. Öz açıklamaların niteliği analiz edildiğinde uygulama esnasında uzak transferi ölçen problemlerin sunulduğu grup yakın transferi ölçen problemlerin sunulduğu gruba göre daha nitelikli açıklamalar oluşturmuştur. Ayrıca çalışma süreleri incelendiğinde uzak transferi ölçen problemlerin sunulduğu grup daha kısa sürede uygulamayı tamamlamıştır. Sonuç olarak uygulama esnasında çözümün alt hedeflere tasarımcı tarafından ayrıldığı çözümlü örnek yönteminin uygulanması ve çözümlü örnekten sonra uzak transferi ölçen problemlerin sunulmasının daha etkili olduğu bulgularına ulaşılmıştır.

2.5.1. İlgili Araştırmaların Değerlendirilmesi

Programlama öğretiminde çözümlü örnek kullanımı ile ilgili araştırmalar incelendiğinde çalışmaların %75'inden fazlasının son 5 yılda (2012'den sonra) yapıldığı görülmüştür. Bu da son yıllarda bu konuyla ilgili çalışmalara yönelik bir eğilim olduğunu göstermektedir.

Yapılan çalışmalarda genellikle uygulanan yöntemlerin, problem çözme başarısına, öğrenenlerin bilişsel yüklenmelerine ve çalışma sürelerine etkilerine bakılmıştır. Bu konuda yapılan ilk çalışmalarda çözümlü örnek yönteminin

geleneksel problem çözme yöntemiyle karşılaştırıldığı ve bu çalışmalarda çözümlü örnek yönteminin problem çözme yöntemine göre daha verimli olduğu bulgularına ulaşılmıştır (Hohn ve Moraes, 1998; Chang vd., 2000; van Merrienboer, 1990). Bu bulgular yeterli deneyimi olmayan bireylere programlama öğretiminde çözümlü örnek yöntemlerinin problem çözme yöntemine göre daha etkili olduğunu ve son yıllarda bu yöntemleri karşılaştırmak yerine hangi çözümlü örnek yönteminin daha etkili olduğu konusunda çalışmalar yapılması gerekliliğini göstermektedir. Yapılan bazı çalışmalarda (Chang vd., 2000; Lee, 2013) çözümlü örneklerden öğrenmenin motivasyonu arttırdığı ve öğrencilerin çözümlü örneklerden daha iyi öğrendiklerini ifade ettikleri ortaya konulmuştur.

Programlama öğretiminde çözümlü örnek kullanılırken alt hedeflerin vurgulanması sıkça kullanılan bir stratejidir (Margulieux, Catrambone ve Guzdial, 2013, 2016; Margulieux ve Catrambone, 2016; Margulieux, Morrison, Catrambone ve Guzdial, 2016; Scheaffer, 2015). Çözümlü örneklerin alt hedeflerinin vurgulandığı durumda öğrenenler, vurgulanmadığı durumdaki öğrenenlere göre daha üstün başarı göstermişlerdir. Ancak bu grupların çalışma süreleri ve bilişsel yüklenme düzeyleri arasında bir farklılık görülmemiştir. Ayrıca alt hedeflerle birlikte öz açıklamanın kullanımı öğrenme üzerinde olumsuz etki yapmış ve öğrenenlerin aşırı bilişsel yüklenmelerine sebep olmuştur (Margulieux vd., 2016). Bununla birlikte çözümlü örneklerde öz açıklama süreci yerine alt hedeflerle açıklayıcı metinlerin birlikte kullanılması öğrenmeye olumlu katkı sağlamıştır (Margulieux ve Catrambone, 2016).

Programlama öğretiminde yaygın olarak kullanılan bir diğer strateji de tamamlama stratejisidir (Abdul-rahman ve Boulay, 2014; Harms vd., 2015; van Merrienboer, 1990). Tamamlamalı çözümlü örnek yöntemi, geleneksel problem çözme (van Merrienboer, 1990) ve öz açıklamaların kullanıldığı çözümlü örnek (Abdul-rahman ve Boulay, 2014) yöntemlerine göre başarı açısından daha etkili olmuştur. Ancak tamamlamalı çözümlü örnek yönteminin bilişsel yüklenme ve çalışma süreleri üzerindeki etkileri çelişkilidir. Bu nedenle tamamlamalı çözümlü örnek yönteminin sadece başarıya etkisi değil, çalışma süreleri ve/veya bilişsel yüklenme düzeyleri de hesaba katılarak öğretim verimliliğinin belirlenmesi önemli görülmektedir. Ayrıca öğrenenlerin bilişsel yüklerinin daha verimli kullanmasını sağlayabileceği bir

tamamlamalı çözümlü örnek yöntemi olan karartma stratejisinin programlama öğretiminde kullanılması da önemli olabilir.

Yapılan çalışmalar incelendiğinde programlama öğretiminde çözümlü örnek kullanımı ile ilgili Türkiye'de yapılan bir çalışmaya rastlanmamıştır. Genel olarak alanyazın incelendiğinde de programlama öğretiminde matematik ve fizik gibi alanlarda kullanılan bazı stratejilerin (kolay işaretleme ilkesi, imgeleme ilkesi, doğru ve yanlış çözümlü örneklerin birlikte kullanımı, karartılmış çözümlü örnekler) etkililiğinin sınanmadığı görülmüştür. Ayrıca programlama öğretiminde döngüler, karar yapıları veya daha geniş perspektifte bir program (App Inventor vb.) anlatımıyla ilgili çözümlü örnekler kullanılırken algoritma öğretiminde çözümlü örnek yönteminin kullanıldığı çalışmalar bulunmamaktadır.

3. YÖNTEM

3.1. Araştırmanın Yöntemi

Bu çalışmada algoritma öğretiminde kullanılan geleneksel çözümlü örnek ve karartılmış çözümlü örnek yöntemlerinin öğrenenlerin başarı ve bilişsel yüklerine etkisini karşılaştırmak amacıyla gerçek deneysel araştırma yöntemlerinden ön test – son test kontrol gruplu model kullanılmıştır. Gerçek deneysel modelde bir katılımcı havuzundan katılımcılar deney ve kontrol gruplarına seçkisiz olarak atandıkları için deneysel çalışmalarda oluşabilecek iç geçerlik tehditlerine karşı güçlü bir savunma oluşturulur.

Araştırmanın bağımsız değişkeni olarak öğretim yöntemi (karatılmış çözümlü örnekler ve geleneksel çözümlü örnekler) ele alınmıştır. Bağımlı değişkenler ise ön test - son test performansı, bilişsel yük ve öğretim verimliliğidir.

3.2. Çalışma Grubu

Araştırmanın çalışma grubu, bir devlet üniversitesinin Bilgisayar ve Öğretim Teknolojileri Eğitimi (BÖTE) Bölümü 2. sınıf öğrencilerinden oluşmuştur. Katılımcılar 28 öğrenci deney, 23 öğrenci kontrol grubuna olmak üzere seçkisiz olarak atanmıştır. Uygulama sürecine dâhil edilmesi planlanan katılımcılardan 9 deney 7 kontrol grubu olmak üzere 16 öğrenci uygulamalara katılmamış, deney grubundan 2 öğrenci de uygulamalara katılmış ancak başarı testine katılmamıştır. Sonuç olarak çalışma, 17 deney grubu, 16 kontrol grubu olmak üzere 33 katılımcı ile yürütülmüştür. Çalışma grubunun cinsiyete göre dağılımı Tablo 3.1'de gösterilmiştir.

Tablo 3.1: Deney ve Kontrol Gruplarının Cinsiyete Göre Dağılımı

Grup	*Kadın*		*Erkek*		*Toplam*
	n	*%*	*n*	*%*	
Deney	14	%82	3	%18	17
Kontrol	10	%63	6	%37	16
Genel	24	%73	9	%27	33

Tablo 3.1'de görüldüğü üzere çalışma grubunun %70'inden fazlasının kadın öğrencilerden oluştuğu görülmektedir. Deney grubunda kadın öğrenciler grubun %82'sini oluştururken, kontrol grubunda katılımcıların %63'ü kadın öğrencilerden

oluşmuş ve biraz daha dengeli bir dağılım söz konusu olmuştur. Genel olarak incelendiğinde cinsiyete göre denk olmayan gruplar oluştuğu söylenebilir.

3.3. Veri Toplama Araçları

Çalışmada 4 adet veri toplama aracı kullanılmıştır. Çalışmada uygulanan yöntemin etkililiğini ölçmek amacıyla başarı testi, uygulama esnasındaki bilişsel yüklenme düzeylerini belirlemek için bilişsel yük derecelendirme ölçeği kullanılmıştır. Ayrıca uygulamanan yöntemin programlama final sınavı puanlarını yordama derecesini belirlemek için dersin final sınavı puanları ve uygulama esnasındaki süreç performanslarını ölçmek için veri tabanında tutulan öğrenci yanıtları analiz edilmiştir.

3.3.1. Başarı Testi

Bu çalışmada kullanılan başarı testi algoritma öğretiminde akış şemaları ve sözde kod konularını kapsayacak şekilde çoktan seçmeli 15 maddeden oluşmaktadır (Bkz. Ek 4). Her bir maddenin doğru yanıtı 1 puan, yanlış veya boş bırakılan yanıtlar ise 0 puan olarak değerlendirilmiş ve testten alınacak maksimum puan 15 olarak belirlenmiştir. Başarı testi deney ve kontrol gruplarının uygulama öncesi ve sonrasındaki bilgilerini ölçmek amacıyla uygulamadan bir hafta önce ön-test olarak, uygulamadan bir hafta sonra son-test olarak sunulmuştur.

İlk olarak, araştırmacı tarafından algoritma oluşturma sürecinde sözde kod ve akış şemaları konularını kapsayacak şekilde 28 maddeden oluşan bir madde havuzu oluşturulmuştur. Ardından programlama alanında çalışan iki öğretim üyesinden uzman görüşü alınarak dönütler sonrasında madde sayısı 20'ye indirilmiştir. Uzman görüşü doğrultusunda bu maddelerin 6'sı kolay, 8'i orta, 6'sı zor zorlukta olarak değerlendirilmiştir. Ayrıca Türkçe Eğitimi Bölümü'nde görev yapan bir öğretim üyesinden yazım yanlışlarına ilişkin görüş alınarak maddelerin açık ve anlaşılır olmaları sağlanmaya çalışılmıştır.

Başarı testinin geliştirilme sürecinde katılımcılar, bir devlet üniversitesinin BÖTE Bölümü'nde öğrenim gören 1., 3. ve 4. sınıftaki 75 öğrenciden oluşturulmuştur. Çalışma grubuna uygulanacak maddelerin seçimi için pilot uygulamadan elde edilen veriler üzerinde alt-üst %27 grup ortalamaları farkına dayalı madde analizi

yapılarak madde ayırt edicilik indeksi (rjx) ve madde güçlük indeksi (pj) hesaplanmıştır.

Rjx testte yer alan her bir madde için ayırt etme gücünü temsil eder. Rjx'i 0,40'tan büyük olan maddeler çok iyi, 0,30 ile 0,39 arasında olan maddeler iyi ayırt edici olarak değerlendirilebilir, 0,20 ile 0,29 arasında olan maddeler düzeltilerek kullanılabilir, 0,20'den düşük olan maddeler ise ölçekten çıkartılmalıdır (Crocker ve Algina, 1986; Tekin, 1996). Pj ise testteki maddelerin zorluk derecesini göstermektedir. Pj 0,40'tan küçük ise madde kolay, 0,40 ile 0,60 arasında ise orta, 0,60'tan büyük ise maddenin zor olduğu söylenebilir. Testin ortalama pj'sinin 0,50'e yakın olması istenen durumdur (Büyüköztürk, 2014; Sönmez ve Alacapınar, 2016). Yapılan analiz sonucunda geliştirilen testin pj'si 0,54 olarak bulunmuştur.

Madde analizi sonucunda madde ayırt edicilik indeksi 0,20'nin altında olan 3 madde çıkartılmış, 0,20 ile 0,30 arasında olan 2 madde de uzman görüşü alınarak kapsam geçerliğini bozmayacağı düşünülerek testten çıkartılmıştır. Böylece testin 15 maddelik son hali oluşturulmuştur. Madde analizine ilişkin sonuçlar Ek 8'de gösterilmiştir.

Madde analizi yapılarak 15 maddeye indirilen testin güvenirlik analizi için KR-20 iç tutarlılık katsayısı hesaplanmıştır. KR-20 yöntemi geliştirilen aracın pilot uygulama grubuna bir kez uygulanacağı durumda ve madde güçlüklerinin eşit olmadığı durumda kullanılan bir yöntemdir (Büyüköztürk, Kılıç Çakmak, Akgün, Karadeniz ve Demirel, 2016). KR-20 iç tutarlılık katsayısının en az 0,70 olması istenilen bir durumdur (Fraenkel, Wallen ve Hyun, 2012).

Tablo 3.2: Başarı Testinin Geliştirilmesine İlişkin Analiz Sonuçları

N	*Ortalama güçlük indeksi*	*Ortalama ayırt edicilik indeksi*	*KR-20 iç tutarlılık katsayısı*
75	0,54	0,53	0,871

Tablo 3.2 incelendiğinde geliştirilen başarı testinin KR-20 iç tutarlılık katsayısı yapılan analiz sonucunda 0,871 olarak hesaplanmıştır. Bu durumda hazırlanan başarı testinin güvenilir olduğu söylenebilir.

3.3.2. Bilişsel Yük Ölçeği

Bilişsel yük, öğrenenlerin hedeflenen görevi yerine getirirken harcadıkları bilişsel çabayı ifade etmektedir. Bu çalışmada Paas ve van Merrienboer (1993) tarafından

geliştirilen Kılıç ve Karadeniz (2004) tarafından Türkçe'ye uyarlaması yapılan tek maddelik 9'lu derecelendirme ölçeği, uygulama sürecindeki her iki haftada da problem çözümü tamamlandıktan sonra öğrencilere sunulmuştur. Türkçe'ye uyarlaması yapılan ölçekte tek madde olarak öğrencilere "Verilen görevi tamamlarken ne kadar çaba sarf ettiniz?" sorusu sorulmuş ve 1 "çok çok az", 9 "çok çok fazla" olmak üzere 1'den 9'a kadar sarf ettikleri çabayı işaretlemeleri istenilmiştir. Kılıç ve Karadeniz (2004) ölçeğin güvenilirliğine ilişkin yapılan analizler sonucunda Cronbach Alfa iç tutarlılık katsayısını 0,78 olarak hesaplamışlardır.

3.3.3. Öğretim Verimliliğinin Ölçülmesi

Öğretim verimliliğini test etmek için Paas ve van Merrienboer (1993), son test performansıyla birlikte bu testi çözerken oluşan bilişsel yükü hesaba katarak bir formül geliştirmişlerdir. Çalışmada test esnasındaki bilişsel yüklenme değil öğrenme sürecindeki bilişsel yüklenme temel alınacağından uyarlanmış öğretim verimliliği formülü (van Gog ve Paas, 2008) kullanılmıştır. Öğretim verimliliğinin nasıl hesaplanacağı Şekil 3.1'de gösterilmiştir.

$$\text{Öğretim Verimliliği} = \frac{Z_{\text{Başarı}} - Z_{\text{Bilişsel Yük}}}{\sqrt{2}}$$

Şekil 3.1: Öğretim Verimliliği Formülü

Bu formüle göre ilk olarak öğrencilerin öğrenme sürecindeki bilişsel yük puanlarının ortalamaları ve başarı puanları standartlaştırılmıştır. Öğretimin etkililiği analiz edilirken olduğu gibi öğretim verimliliği analiz edilirken de düzeltilmiş son test puanları kullanılmıştır. Her bir öğrencinin düzeltilmiş son test puanını elde etmek için değişim puanı formülü (Cronbach & Furby, 1970; Gardner & Neufeld, 1987) kullanılmıştır.

3.3.4. Öğrenme Sürecine İlişkin Performans Puanları

Uygulama esnasında katılımcıların web ortamında çözümlü örneklere çalıştıktan sonra probleme verdikleri yanıtlara ilişkin veriler bir veritabanına kaydedilmiştir. Bu

verileri değerlendirmek amacıyla araştırmacı tarafından 10 maddelik bir rubrik hazırlanmıştır. Uzman görüşü alınarak 4 madde çıkarılıp 1 madde eklenerek 7 maddelik rubrik, puanlayıcıların değerlendirmesine sunulmak üzere son haline getirilmiştir (Bkz. Ek 6). Her bir madde 2 puan üzerinden değerlendirilmiş, en yüksek başarı puanı 14 olarak belirlenmiştir.

Uygulama esnasında katılımcıların girmiş oldukları verilerin değerlendirilmesinde araştırmacı ve dersi veren öğretim üyesi olmak üzere 2 puanlayıcı, geliştirilen rubrik üzerinden her bir öğrenci için puan vermişlerdir. Bu puanların tutarlılığının incelenmesi için kodlayıcılar arası güvenirlik katsayıları kullanılmıştır. Kodlayıcılar arası güvenirlik için uzlaşma yüzdesi, Scott'ın pi, Cohen'in kappa, Krippendorff'un alpha, Spearman rho ve Pearson r kullanılmakta olup hangi katsayının kullanılacağı konusunda genel bir fikir birliği oluşmamıştır (De Wever, Schellens, Valcke ve Van Keer, 2006). Bu çalışmada hem rubriğin ölçütlerinin kategorik olması hem de 2 puanlayıcı tarafından değerlendirildiğinden dolayı Cohen'in Kappa katsayısı kullanılmıştır. Cohen'in Kappa katsayısı iki uygulama için de ayrı ayrı hesaplanmış, sözde kod uygulaması için güvenirlik katsayısı 0,638, akış şemaları uygulaması için de 0,658 olarak bulunmuştur. Kappa katsayısı için değer aralığı 0,75 (bazen 0,80) üzeri ise mükemmel uyum, 0,40 ın aşağısı zayıf uyum, 0,40 ile 0,75 arası ise iyi uyum şeklinde yorumlanabilir (Krippendorff, 1980; Neuendorf, 2002). Buna göre her iki uygulama için puanlamaların uyumlu ve güvenilir olduğu söylenebilir.

3.4. Verilerin Analizi

Verilerin betimsel analizinde ortalama ve standart sapma (ss) değerleri kullanılmıştır. Araştırmada yapılan tüm analizlerde istatistiksel anlamlılık seviyesi 0,05 olarak kabul edilmiştir. Araştırmada elde edilen bulguların etki büyüklüğünü belirlemek için eta kare etki büyüklüğü değeri hesaplanmıştır. Etki büyüklüğü değerlerini yorumlarken, .01-.05 arası küçük, .06-.13 arası orta, 0,14 ve üzeri değerleri geniş etki büyüklüğü olarak değerlendirilmiştir (Cohen, 1988).

Deney ve kontrol grubunun uygulanan yönteme göre son test performanslarını analiz etmek için kovaryans analizi (ANCOVA) kullanılmıştır. Deney ve kontrol grubunun uygulanan yönteme göre bilişsel yüklenme düzeylerini analiz etmek için Tekrarlı Ölçümler için Varyans Analizi (ANOVA) kullanılmıştır. Geleneksel çözümlü

örnek ve karartılmış çözümlü örnek yöntemlerinin verimliliğini ölçmek için t testi kullanılmıştır. Uygulama sırasındaki süreç performanslarını ölçmek için Tekrarlı Ölçümler için ANOVA kullanılmıştır. Uygulanan yöntemlerin Programlama Dilleri I dersini yordama derecesini analiz etmek için Basit Doğrusal Regresyon Analizi kullanılmıştır.

Başarı testlerinin geliştirilmesinde pj ve rjx, geliştirilen testin güvenirliğinin hesaplanmasında KR-20 iç tutarlılık katsayısı kullanılmıştır. Verilerin normallik sayıltısını sağlayıp sağlamadıklarını kontrol etmek için çarpıklık ve basıklık katsayıları ile birlikte Shapiro Wilk testi sonuçları kullanılmıştır. Verilerin homojenlik sayıltısını karşılayıp karşılamadığına bakmak için Levene testi sonuçlarına bakılmıştır. Elde edilen veriler SPSS 21 paket programı kullanılarak analiz edilmiştir.

3.5. Sayıltıların İncelenmesi

Araştırma problemlerine ilişkin bulguların elde edilmesi için gerçekleştirilen analizlerden önce kullanılacak analizlerin sayıltılarının karşılanması gerekmektedir. Bu sayıltılar karşılandığında parametrik testler, karşılanmadığı durumda da parametrik olmayan testler kullanılmaktadır. Bu bölümde çalışmada kullanılan analizlerin sayıltılarına ilişkin bulgulara yer verilmiştir.

Tablo 3.3: Çalışmada Kullanılan Analizlere İlişkin Basıklık-Çarpıklık Bulguları

Referans No	*Ölçümün Amacı*	*Normalliğine Bakılacak Ölçüm*	*N*	*Çarpıklık Değeri*	*Basıklık Değeri*
1	Akış şemaları uygulamasında oluşan bilişsel yüklenme düzeyini belirlemek	Tüm katılımcıların bilişsel yüklenme düzeyi	33	-0,661	-0,197
2	Sözde kod uygulamasında oluşan bilişsel yüklenme düzeyi	Tüm katılımcıların bilişsel yüklenme düzeyi	33	-0,960	0,160
3	Akış şemaları uygulamasında süreç performans puanlarını belirlemek	Tüm katılımcıların süreç performansı	33	0,040	-1,146
4	Sözde kod uygulamasında süreç performans puanlarını belirlemek	Tüm katılımcıların süreç performansı	33	-0,132	-0,754
5	Uygulanan yöntemleri öğretim verimliliği açısından karşılaştırmak	Karartılmış çözümlü örnek yöntemine ilişkin öğretim verimliliği puanı	33	0,350	-0,502
6		Geleneksel çözümlü örnek yöntemine ilişkin öğretim verimliliği puanı	33	0,107	-0,524
7	Deney ve kontrol grubunun ön test puanlarına göre düzeltilmiş son test performans puanlarını karşılaştırmak	Deney grubunun ön test performansı	33	0,217	-1,015
8		Kontrol grubunun ön test performansı	33	0,031	1,185
9		Deney grubunun son test performansı	33	-0,305	-1,035

10		Kontrol grubunun son test performansı	33	-0,431	-0,071
11	Uygulanan yöntemin final sınavı puanlarını yordama derecesini belirlemek	Hata terimleri	33	-0,487	-0,611

3.5.1. Bağımsız Örneklem T Testine İlişkin Sayıltılar

1. Her iki grup kendi içinde normal dağılım göstermelidir.

Bu varsayımı incelemek için basıklık ve çarpıklık değerlerine bakılmıştır. Bu değerlerin ±2 aralığında olması normal dağıldığını göstermektedir (Cooper-Cutting, 2010; George ve Mallery, 2003). Tablo 3.3'te 5 ve 6 referans no'lu değerler incelendiğinde grupların kendi içinde normal dağılım gösterdiği görülmüştür.

2. Her iki grubun dağılımlarına ait varyanslar birbirine eşit olmalıdır.

Diğer parametrik testlerde olduğu gibi t testinde de grupların varyanslarının homojen olması beklenmektedir. Bunun için Levene testi sonuçları incelenmiş ve varyansların homojen olduğu görülmüştür ($F(1,31)=1,056$, $p=0,312>0,05$).

3.5.2. ANCOVA'ya İlişkin Sayıltılar

1. Her iki grup için hem bağımlı değişken hem de kontrol değişkeni normal dağılım göstermelidir.

Tablo 3.3'te 7, 8, 9 ve 10 referans no'lu değerler incelendiğinde grupların kendi içinde normal dağılım gösterdiği görülmüştür.

2. Her iki grubun dağılımlarına ait varyanslar birbirine eşit olmalıdır.

Levene testi sonuçlarına göre varyansların homojen olduğu görülmüştür ($F(1,31)=0,00$, $p=0,995>0,05$).

3. Gruplar içi regresyon eğimleri eşit olmalıdır.

Yapılan analiz sonucunda gruplar içi regresyon eğimlerinin eşit olduğu görülmüştür ($F(1,31)=0,898$, $p=0,995>0,351$).

3.5.3. Basit Doğrusal Regresyon Analizine İlişkin Sayıltılar

1. Hata terimleri normal dağılım göstermelidir.

Tablo 3.3'te 11 referans no'lu değerler incelendiğinde hata terimlerinin normal dağıldığı görülmüştür.

2. Bağımlı ve bağımsız değişkenler arası doğrusal bir ilişki olmalıdır.

Tablo 3.4'e göre öğretim yöntemi ve final puanı arasında anlamlı bir ilişkinin olduğu söylenebilir.

Tablo 3.4: Basit Doğrusal Regresyon Sayıltılarına İlişkin Bulgular

Değişken	*n*	*r*	*p*	*Durbin-Watson*
Öğretim Yöntemi – Final Puanı	33	0,363	0,019	1,656

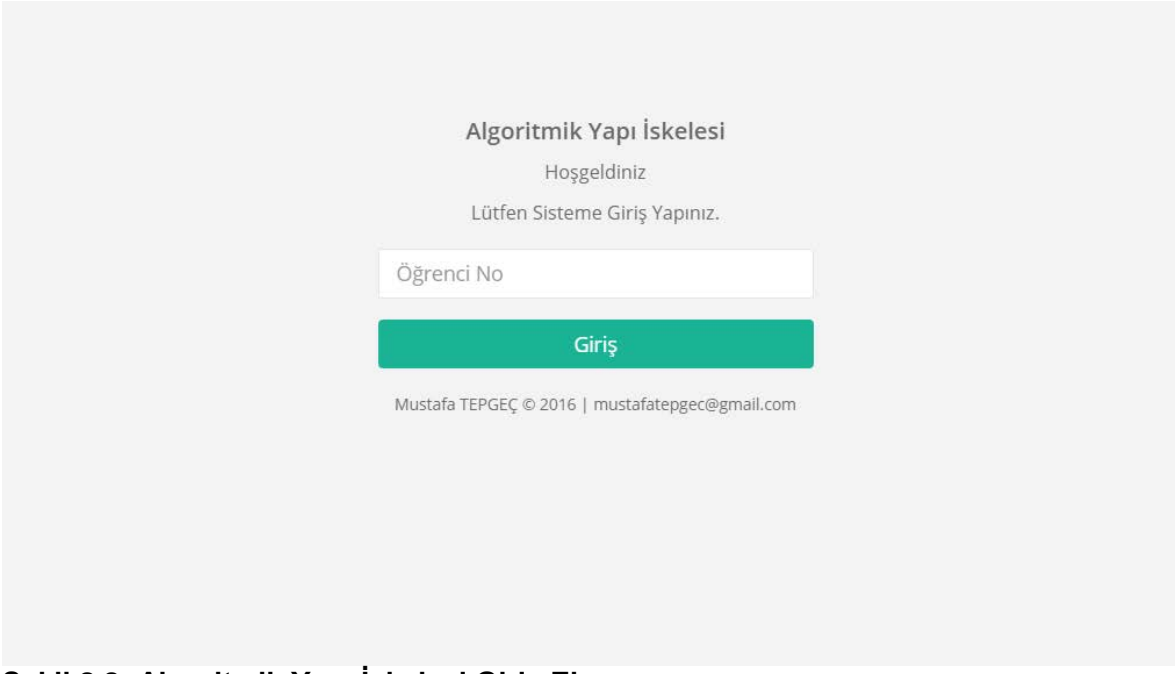
3. Hata terimleri arasında ilişki olmamalıdır (Otokorelasyon)

Tablo 3.4 incelendiğinde Durbin-Watson değerinin 1,5 ile 2,5 değerleri arasında yer almasından dolayı otokorelasyonun olmadığı, yani hata terimleri arasında ilişkinin olmadığı söylenebilir.

Araştırmada kullanılması planlanan parametrik test sayıltıları incelendiğinde bu sayıltıların karşılandığı görülmüştür. Dolayısıyla sayıltıları incelenen testler kullanılmış, parametrik olmayan test kullanımına gereksinim duyulmamıştır.

3.6. Çözümlü Örnek İlkelerine Dayalı Geliştirilen Çevrimiçi Öğrenme Ortamı (Algoritmik Yapı İskelesi)

Çalışma bağlamında deney ve kontrol grubuna yönelik iki farklı web tabanlı ortam tasarlanmıştır. Çözümlü örneklerde kullanılan tasarım ilkelerinin daha etkili bir şekilde sunulması için bilgisayar ortamı tercih edilmiştir. Katılımcıların girdikleri verilerin daha sağlıklı bir şekilde tutulması için de web tabanlı bir öğrenme ortamı tasarlanmıştr. Katılımcılar ortama algoritmikyapiiskelesi.com adresinden ulaşmışlar ve sisteme girişte öğrenci numaralarını kullanmışlardır (Bkz. Şekil 3.2).



Şekil 3.2: Algoritmik Yapı İskelesi Giriş Ekranı

Katılımcılar sisteme girmeden önce öğrenci numaraları veritabanına tanımlanmış ve deney grubuna atanan öğrenciler karartılmış çözümlü örnek yöntemine göre geliştirilen ortama, kontrol grubuna atanan öğrenciler ise geleneksel çözümlü örnek yöntemine göre geliştirilen ortama giriş yapmışlardır. Öğrenme ortamında kullanılan ve kullanılmayan çözümlü örnek ilkeleri, tanımları, kullanılma/kullanılmama gerekçeleri ve öğrenme ortamında nasıl kullanıldığı ile ilgili bilgiler Tablo 3.5'te yer almaktadır.

Tablo 3.5: Öğrenme Ortamının Tasarlanmasında Değerlendirilen Çözümlü Örnek Tasarım İlkeleri

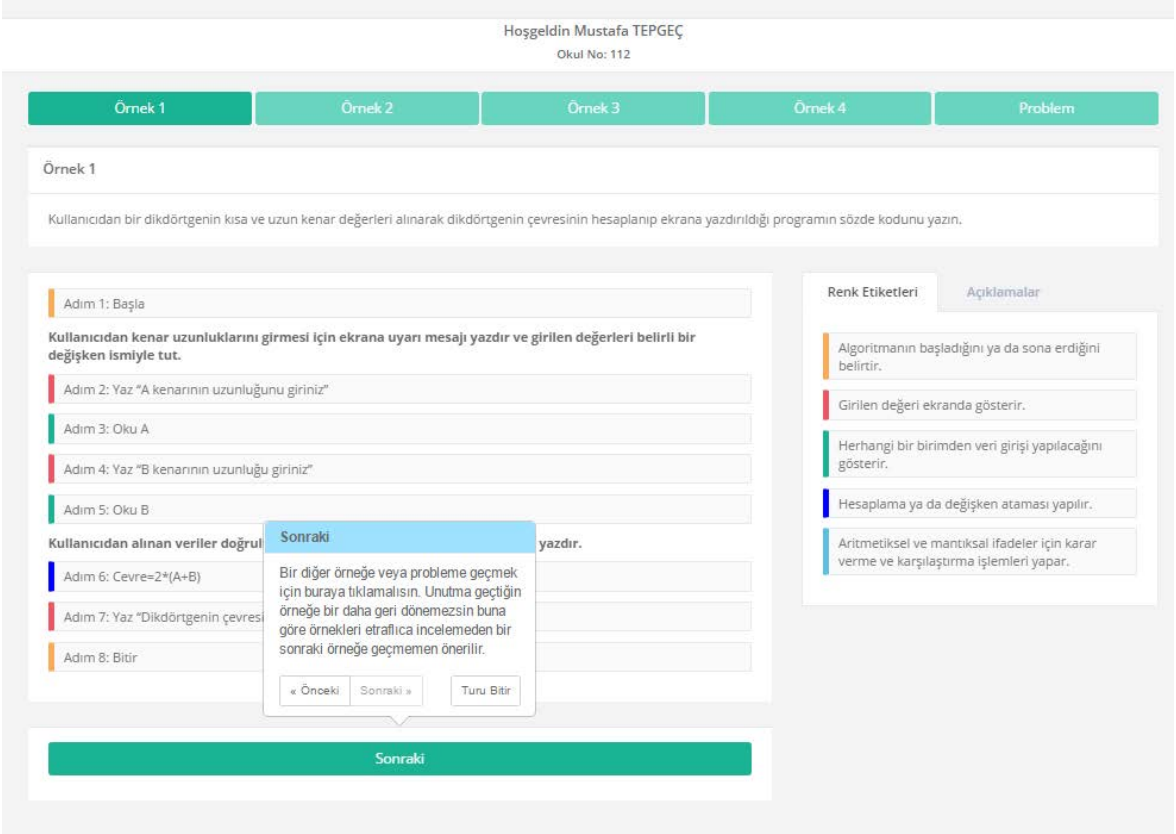
Tasarım İlkesi	*Açıklama*	*Öğrenme Ortamında Neden Kullanıldı/Kullanılmadı?*	*Öğrenme Ortamında Nasıl Kullanıldı?*
Karartılmış çözümlü örnek ilkesi	Örnekteki çözüm adımlarında sistematik bir şekilde boşlukları arttırarak rehberliğin azaltılması	Karartılmış çözümlü örnekler giderek artan problem çözme becerisinin uygulanmasından dolayı uzmanlığın ters tepme etkisine karşı etkili bir yöntemdir (Kalyuga, 2007). Ayrıca karartılmış çözümlü örneklerin özellikle iyi yapılandırılmış alanlarda daha etkili olduğu belirtilmektedir (Renkl, 2014). Bununla birlikte birçok çalışmada karartılmış çözümlü örneklerin geleneksel çözümlü örneklere göre bilişsel yüklenme ve çalışma süreleri açısından da daha verimli olduğu ortaya konulmuştur (Atkinson, Renkl ve Merrill, 2003; Renkl, Atkinson ve Große, 2004).	Deney grubu için geliştirilen öğrenme ortamında çözümlü örnekler geriye doğru karartma yöntemiyle kullanılmıştır. İlk olarak öğrenenlere örnek problemin tam çözümlü hali sunulmuş, ardından çözüm adımlarındaki boşluklar sondan geriye doğru azaltılarak uygulanmıştır. Son olarak öğrenenden sunulan çözümsüz problemi çözmeleri beklenmiştir.
Alt hedeflere ayırma ilkesi	Problemin çözümünü oluşturan adımların gruplanarak belirgin hale getirilmesi	Çözümlü örneklerin çözüm adımlarının alt hedeflerine ayrılarak sunulması, öğrenenlerin problemin altında yatan mantığı kavramasında yardımcı bir strateji olarak kullanılmaktadır (Catrambone ve Holyoak, 1990; Marguileux, Catrambone ve Guzdial, 2012; Stasko, Catrambone ve Guzdial, 2016). Programlama öğretiminde yapılan çalışmalarda çözümlü örneklerde alt hedef kullanıldığı durumlarda öğrenenlerin sarf ettiği bilişsel çaba ve çalışma süreleri açısından anlamlı bir farklılık görülmese de alt hedeflerin kullanıldığı gruplar, alt hedeflerin kullanılmadığı gruplara göre daha yüksek başarı göstermişlerdir.	Öğrenme ortamında alt hedefler hem deney hem kontrol grubunda örnek problemlerin çözüm adımlarının yer aldığı bölümde kullanılmıştır. Sözde kod uygulamasında çözüm adımları kalın yazı tipiyle açıklanarak alt hedeflerine ayrılırken, akış şemaları uygulamasında çözüm adımlarının sol tarafında çözüm adımları gruplandırılıp açıklanarak alt hedeflerine ayrılmıştır.
Öz açıklama ilkesi	Problemin çözüm adımlarının öğrenen tarafından açıklanarak daha aktif bir öğrenme süreci geçirilmesi	Yapılan çalışmalarda öz açıklama sürecinin karartılmış çözümlü örneklerle birlikte kullanılmasının öğrencilerde aşırı bilişsel yüklenmeye neden olabileceği belirtilmiştir (alıntılayan İltüzer, 2017). Ayrıca yeterli deneyimi olmayan öğrenenlerde öz açıklama sürecinin öğrenmeyi olumsuz etkilediği görülmüştür.	Bu ilke, çalışma yeterli deneyimi olmayan öğrenciler ile uygulandığı ve karartılmış örneklerle birlikte kullanıldığında aşırı bilişsel yüklenmeye sebep olabileceği için geliştirilen ortamda değerlendirilmemiştir.
Açıklamalı yardım ilkesi	Çözüm adımlarına ek olarak problem çözümüyle ilgili yapısal ipuçlarının verilmesi	Açıklamalı yardım ilkesi belirli bir kuralı ve çözüm yolu olan iyi yapılandırılmış alanlarda etkilidir. Öz açıklamaların kullanıldığı örnekler, metinsel açıklamalarla desteklenen çözümlü örneklere göre daha etkili bir yöntemdir (Schworm ve Renkl, 2006). Ancak yukarıda belirtildiği gibi öz açıklama kullanımı yeterli deneyimi olmayan öğrenenlerde olumlu sonuçlar doğurmayabilir. Bu durumda açıklamalı yardım ilkesinin kullanımı daha etkilidir (Schunk ve Zimmerman, 2007). Ayrıca açıklayıcı metinlerin özellikle alt hedefler ve çözümlü örneklerle birlikte kullanılmasının en etkili yöntem olabileceği öngörülmektedir (Margulieux ve Catrambone, 2016).	Öğrenme ortamında açıklayıcı metinler hem deney hem de kontrol grubunda örnek problemlerin çözüm adımlarının sağ tarafında yer alan açıklamalar bölümünde kullanılmıştır. Açıklayıcı metinlerin direkt problemin çözümüne yönelik olmasından ziyade öğrenen için problemin altında yatan mantığı kavramasında kullanabilecekleri ipuçları şeklinde sunulmuştur. Açıklayıcı metinler ayrıca öğrenenlerin öğrenme ortamına alışma sürecini hızlandırmak için ortamla ilgili bilgiler verilerek kullanılmıştır.
Basitten karmaşığa sıralama ilkesi	Öğrenenlere sunulan örnek problemlerin başlangıçta birim etkileşimliliği düşük olandan yükseğe doğru sıralanması	Sunulan içeriğin ilk aşamada basit olması, yani düşük birim etkileşimliliğine sahip olması ardından gittikçe daha karmaşık, yani yüksek birim etkileşimliliğine sahip olması her aşamada aynı birim etkileşimliliğine sahip materyallerin sunumuna göre daha etkili bir yöntemdir (Pollock, Chandler ve Sweller, 2002).	Öğrenme ortamında kullanılacak örnek problemler tasarlanırken bu ilke dikkate alınmıştır. İlk aşamada daha basit örnek problemler sunulurken, öğrenenler ilerledikçe problemler zorlaşmakta, yani birim etkileşimliliği artmaktadır.

Basit ilişkilendirme ilkesi 1 (Entegre çözümlü örnekler ile bölünmüş dikkat etkisinden kaçınma)	Problemin çözüm adımlarının gösteriminde farklı enformasyon kaynaklarının ilişkilendirilerek öğrenenin konu dışı bilişsel yüklenmesinin azaltılması, enformasyonun bütünleşik bir şekilde sunulması	Tarmizi ve Sweller (1988) ile Ward ve Sweller (1990) yaptıkları çalışmalarda geometri öğretiminde iki farklı enformasyon kaynağını (şekil ve metin) entegre ederek olumlu sonuçlar elde etmişlerdir. Chang, Hsu ve Yu (2011) bu yöntemi programlama öğretiminde uygulamış ve benzer sonuçları elde etmiştir. Tarmizi ve Sweller (1988) bölünmüş dikkat etkisinin hesaba katılmadığı durumlarda çözümlü örnek kullanımının tersine bir etki yaratacağını ve bu durumda problem çözme uygulamalarının daha etkili olabileceğini belirtmiştir.	Geliştirilen öğrenme ortamında hem deney grubu hem de kontrol grubu için alt hedefler akış şemalarıyla bütünleşik bir şekilde sunulmuştur. Yani problemin çözüm adımlarının gruplandırılarak açıklandığı alt hedefler akış şemalarının hemen yanında yer almıştır.
Basit ilişkilendirme ilkesi 2 (Biçem ilkesi-görsel ve işitsel kaynakların birlikte kullanılması)		Jeung, Chandler ve Sweller (1997) karmaşık görevlerde içeriğin hem görsel hem de işitsel olarak sunulmasının, sadece görsel olarak sunulmasına göre daha etkili olmadığını göstermiştir. Ayrıca içeriğin görsel olarak sunulmasına göre işitsel olarak sunulması bazı teknik detaylar (sınıf ortamında çalışılıyorsa kulaklık gibi) gerektirdiğinden tercih edilen bir yöntem olmayabilir (Renkl, 2005). Bununla birlikte öğrenenlerin sunulan materyale aşina olmadığı ve konunun karmaşık olduğu durumlarda sadece görsel kanala uygun sunumun gerçekleştirilmesi daha etkili olmaktadır (Renkl, 2005).	Bu ilke, kulaklık gibi teknik gereksinimler gerektirdiği için geliştirilen ortamda değerlendirilmemiştir.
Basit ilişkilendirme ilkesi 3 (Birbiriyle ilişkili bileşenlerin renk kodlamasıyla belirtilmesi)		Renk kodları yardımıyla enformasyon kaynaklarının ilişkilendirilmesi özellikle karmaşık bilgilerin sunulduğu durumlarda etkili bir stratejidir (Berthold ve Renkl, 2009; Keller, Gerjetz, Scheiter ve Garsoffky, 2006; Ozcelik, Karakus, Kursun ve Cagiltay, 2009). Renk kodları konu ile ilgili öğrenmeyi olumsuz etkileyecek arama stratejilerinden kaynaklanan konu dışı bilişsel yüklenmeyi azaltırken öğrenenin dikkatini gerekli enformasyon kaynaklarına odaklamasını sağladığından etkili yükü de arttırmaktadır (Mayer, 2001).	Bu ilke hem deney hem de kontrol grubunda, hem sözde kod hem de akış şeması uygulamalarında, hem çözümlü örneklerin yer aldığı hem de problemin yer aldığı ekranlarda kullanılmıştır. Renk kodları algoritma oluşturma sürecinde kullanılan, girdi/çıktı, ekrana yazdırma, işlem gibi adımlar için uygulanmıştır.
İmgeleme ilkesi	Çözümlü örneğe çalışırken çözüm adımlarının zihinde canlandırılmasını sağlama	İmgeleme yoluyla çözümlü örneklerden öğrenme stratejisi yeterli deneyime sahip olmayan öğrenenler üzerinde etkili olmamıştır (Cooper, Tindall-Ford, Chandler ve Sweller, 2001). Dolayısıyla bu yöntemin kullanımında öğrenenlerin konu hakkında ön bilgilerinin olması gerekmektedir.	Bu ilke, çalışma yeterli deneyimi olmayan öğrenciler üzerinde uygulandığı için geliştirilen ortamda değerlendirilmemiştir.
Hatalara çalışma ilkesi	Doğru ve yanlış çözümlü örneklerin birlikte kullanılarak problem çözümünde olası yanlışlara karşılık öğrenenlerin hazırbulunuşluklarını arttırması	Öğrenenlerin yanlış çözümlü örneklere çalışması öğrenmenin ilk aşamasında olumlu sonuçlar doğurmayabilir (Große ve Renkl, 2007). Bu nedenle bu yöntemde de öğrenenlerin yeterli ön bilgiye sahip olması bir ön koşul olarak düşünülebilir.	Bu ilke, çalışma yeterli deneyimi olmayan öğrenciler ile uygulandığı için geliştirilen ortamda değerlendirilmemiştir.

Araştırmada karartılmış çözümlü örnek ve geleneksel çözümlü örnek yöntemlerinin algoritma başarısına ve bilişsel yüke etkileri karşılaştırıldığı için öğrenme ortamında kullanılan çözümlü örnekler temelde bu iki yönteme göre tasarlanmıştır. Buna ek olarak Tablo 3.5'te belirtilen çözümlü örnek alanyazınında etkililiği kanıtlanmış ve çalışmanın kapsamına uygun ilkeler öğrenme ortamında her iki grup için de değerlendirilmiştir. Çözümlü örnek alanyazınında etkililiği kanıtlanan öz açıklama, hatalara çalışma, görsel-işitsel ögelerin birlikte kullanımı ve imgeleme ilkelerinin de neden bu çalışma bağlamında kullanılmadığı gerekçelendirilmiştir. Çalışma bağlamında geliştirilen öğrenme ortamında çözümlü örnek tasarım ilkeleri dışında herhangi bir manipülasyon yapılmamış ve gerekçelendirilemeyen hiçbir içerik veya tasarım ögesi öğrenme ortamına dahil edilmemiştir. Bir sonraki bölümlerde deney ve kontrol grubu için kullanılan öğrenme ortamına ilişkin ekran görüntüleri uygulama bazında ayrı ayrı olarak sunulmuştur (Öğrenme ortamına ilişkin tüm ekran görüntüleri için Bkz. Ek 5).

3.6.1. Sözde Kod Uygulaması

Deney ve kontrol grubuna uygulanan sözde kod uygulamasıyla ilgili ekran görüntüleri bu bölümde gösterilmiştir. Kullanıcılar öğrenci numaraları ile ortama giriş yaptıkları anda Şekil 3.3'te gösterilen ekran görüntüsüyle karşılaşmışlardır. Burada uygulamada sunulacak 4 çözümlü örnekten ilki görüntülenmektedir. İlk çözümlü örnek hem deney grubunda hem de kontrol grubunda tam çözümlü hali ile sunulmuştur. Dolayısıyla kullanıcı bu çözümlü örneğin üzerinde herhangi bir işlem gerçekleştirmemiştir.



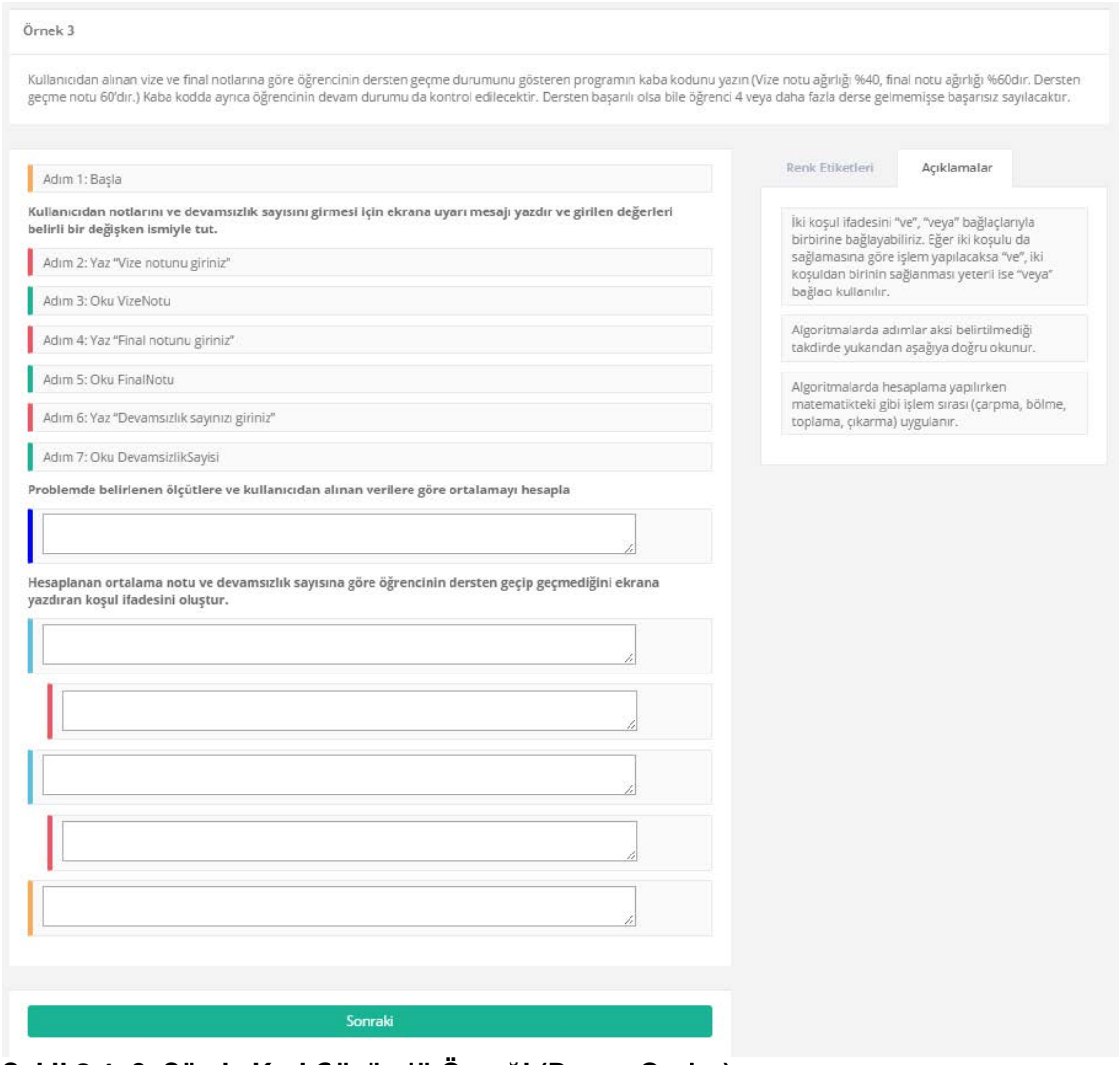
Şekil 3.3: 1. Sözde Kod Çözümlü Örneği (Deney ve Kontrol Grubu)

Deney ve kontrol grubunda öğrencilerin sisteme giriş yapmasıyla birlikte ortamın nasıl kullanılacağına dair kısa bilgiler verilmektedir. Kullanıcıların etkileşim içinde bulunacağı bütün ögeler bu yöntemle açıklanmaktadır. Şekil 3.3'te görüldüğü gibi ekranın en üstünde katılımcıların hangi problem üzerinde olduğu belirginleştirilirken, hemen alt tarafında problem ifadesi yer almaktadır. Ekranın sol tarafında örnek problemlerin çözüm adımları ve alt hedefler yer alırken, sağ tarafta renk kodları ve metinsel açıklamalar yer almaktadır. Renk kodları ve metinsel açıklamalar aynı alanda kullanıcı etkileşimine göre açılmakta olup, varsayılan olarak renk kodları görüntülenmektedir. Katılımcılar bir sonraki çözümlü örneğe geçmek için en altta yer alan "Sonraki" düğmesine tıklarken, bir önceki çözümlü örneğe erişime olanak tanınmamıştır.

Ortamda kullanılan algoritma adımları için (Başla/bitir, işlem, ekrana yazdırma, veri girişi, karar) bir renk verilmiş ve sağ tarafta bu renkler açıklanmıştır. Uygulama boyunca algoritma adımları için aynı renkler kullanılmıştır (Örneğin başla/bitir komutları için turuncu renk). Öğrenme ortamında renk kodları bu şekilde kullanılmıştır. Açıklamalar bölümünde algoritmanın sözde kod ile ifade edilmesinde

dikkat edilmesi gereken kurallar açıklanmıştır. Kullanıcının sisteme girmesiyle birlikte de çözümlü örneklerle birlikte ortamın kullanımıyla ilgili ipuçları verilmiştir. Öğrenme ortamında açıklayıcı yardım ilkesi bu şekilde kullanılmıştır.

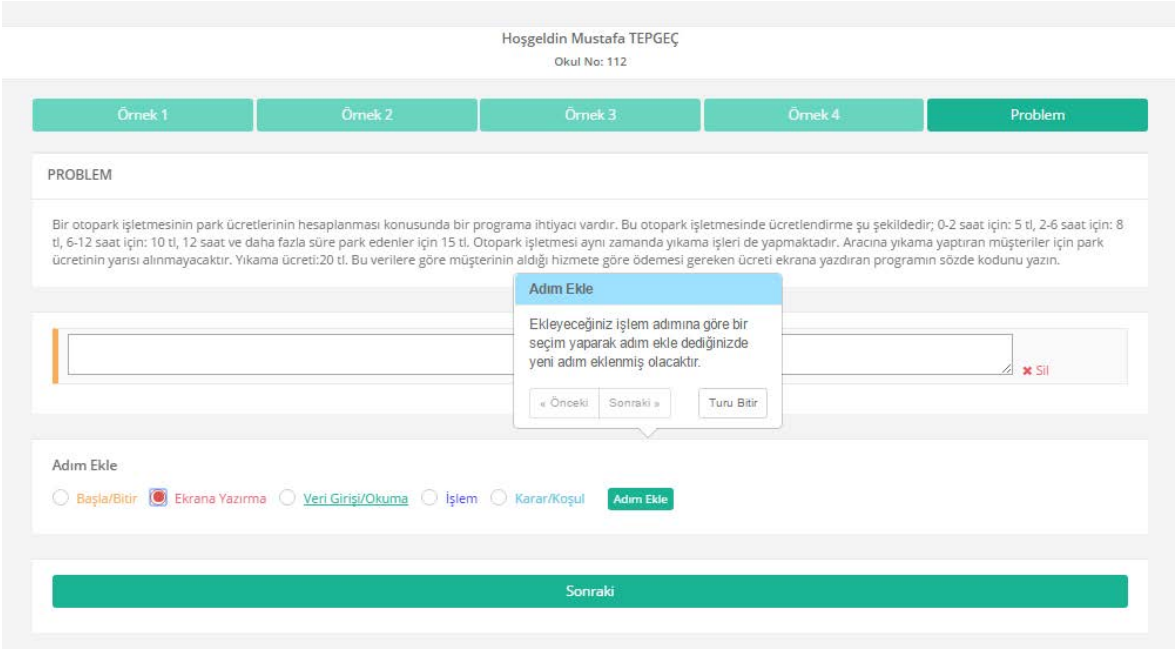
İlk örnekte (Şekil 3.3) hem deney grubu hem de kontrol grubu tam çözümlü örneğe çalışmışlardır. İkinci örnekten itibaren deney grubu için örneklerin çözüm adımları sondan başa doğru boşluklar arttırılarak sunulmuş ve en son problem durumu belirtilerek çözülmesi beklenmiştir. Kontrol grubu için bütün örnek problemler tam çözümleriyle birlikte sunulmuştur.



Şekil 3.4: 3. Sözde Kod Çözümlü Örneği (Deney Grubu)

Şekil 3.4'te görüldüğü gibi karartılmış çözümlü örneklere çalışan deney grubu için çözüm adımlarının bir bölümü boş bırakılarak öğrenenlerden çözmeleri beklenmiştir. Boş bırakılan adımların her biri renk etiketlerine uygun olarak

tasarlanmıştır. Ayrıca problemin çözüm adımlarının gruplandırılarak belirginleştirildiği alt hedefler çözümü boş bırakılan adımlarda da kullanılmıştır. Çözümlü örneklerde alt hedefler ilkesi bu şekilde uygulanmıştır. Aynı zamanda çözüm adımlarına ilişkin alt hedefler, problemin çözümünden bağımsız olarak değil çözüm adımlarının içinde sunulmuştur. Geliştirilen ortamda basit ilişkilendirme ilkesi bu şekilde uygulanmıştır. Örnek 3, kontrol grubuna Şekil 3.3'te olduğu gibi tam çözümlü haliyle sunulmuştur.



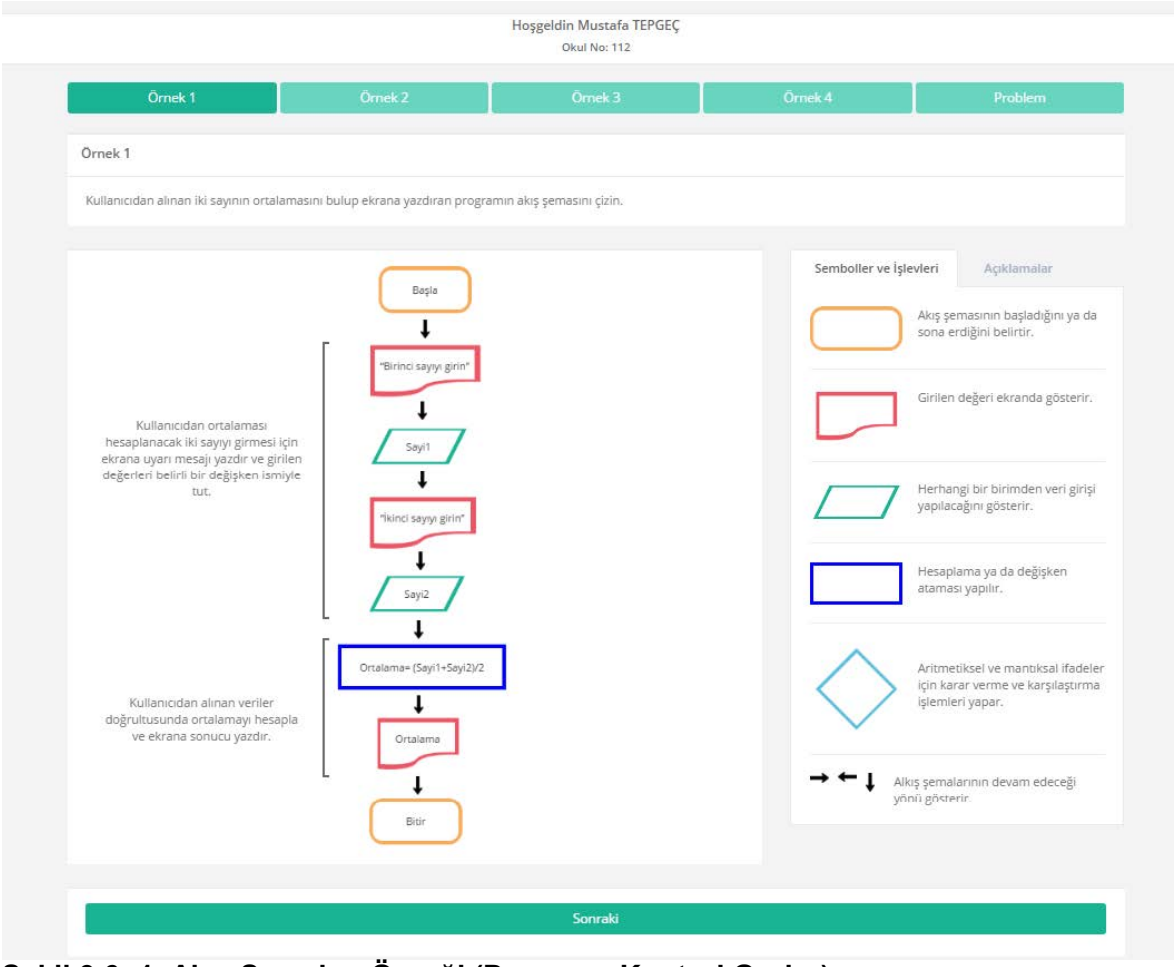
Şekil 3.5: Sözde Kod Problem Sorusu (Deney ve Kontrol Grubu)

Hem deney grubu hem de kontrol grubu 4 çözümlü örneğe çalıştıktan sonra Şekil 3.5'te gösterilen problem ifadesiyle karşılaşmışlardır. Problem bölümünde farklı yollar ile sonuca ulaşılabileceğinden hangi çözüm adımlarını kullanacağı belirtilmemiş, öğrenenlerden bu adımları kendilerinin eklemesi beklenmiştir.

3.6.2. Akış Şemaları Uygulaması

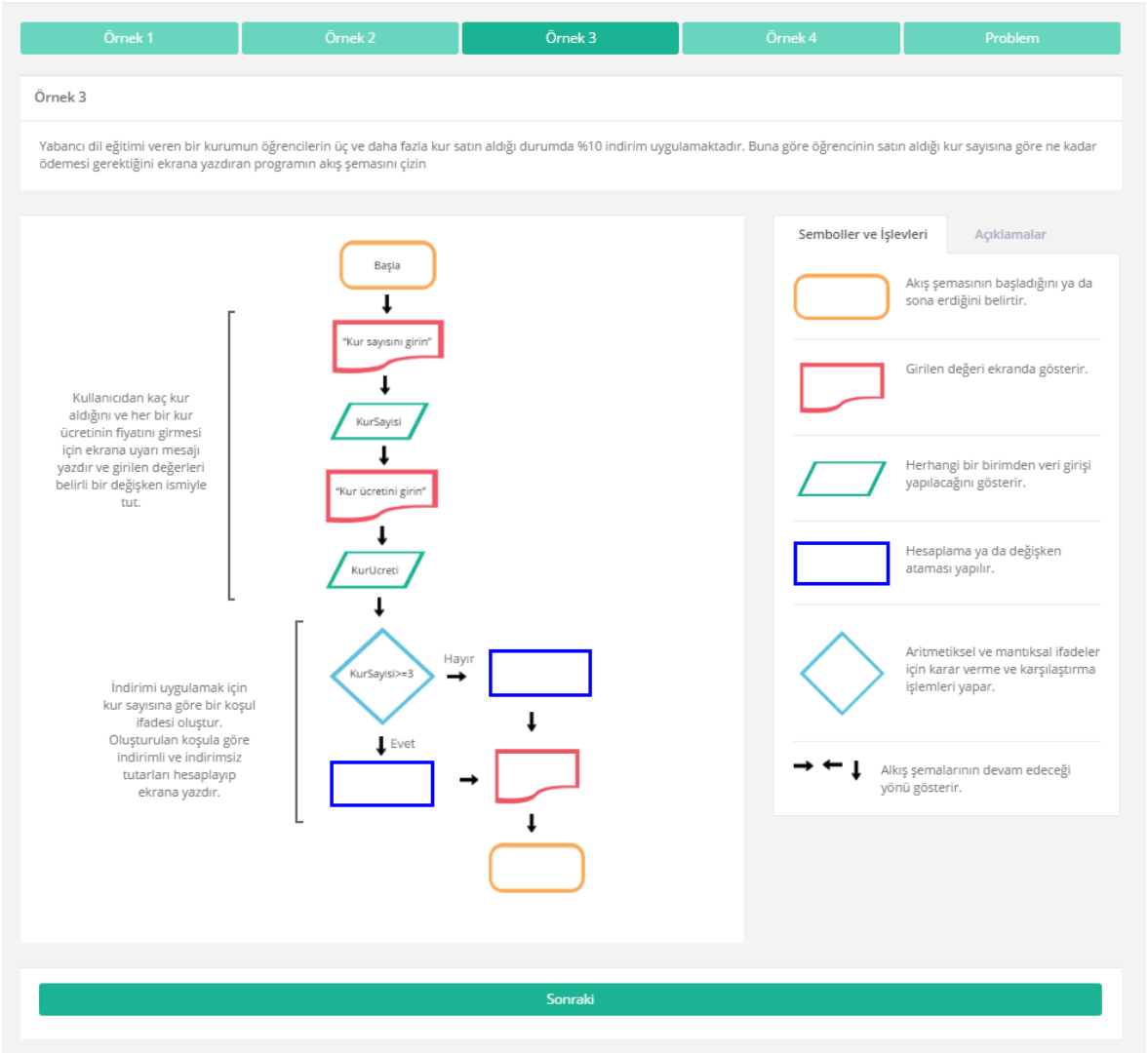
Akış şemaları uygulamasında kullanılan ilkeler ve arayüz sözde kodlar uygulamasına benzer şekilde tasarlanmıştır. Farklı olarak renk etiketleri yerine semboller ve işlevleri eklenmiştir. Bu sekme altında, akış şemalarında kullanılan sembollerin kullanımıyla ilgili bilgiler verilmiştir. Örneklerin çözüm adımlarının alt hedeflere ayrılarak gruplandırılması akış şemaları uygulamasında da gerçekleştirilmiş ve çözüm adımlarının sol tarafında belirtilmiştir. Deney grubunda

karartılan çözüm adımları için şekiller eklenmiş ve katılımcılardan problemin akışına göre doldurmaları beklenmiştir.



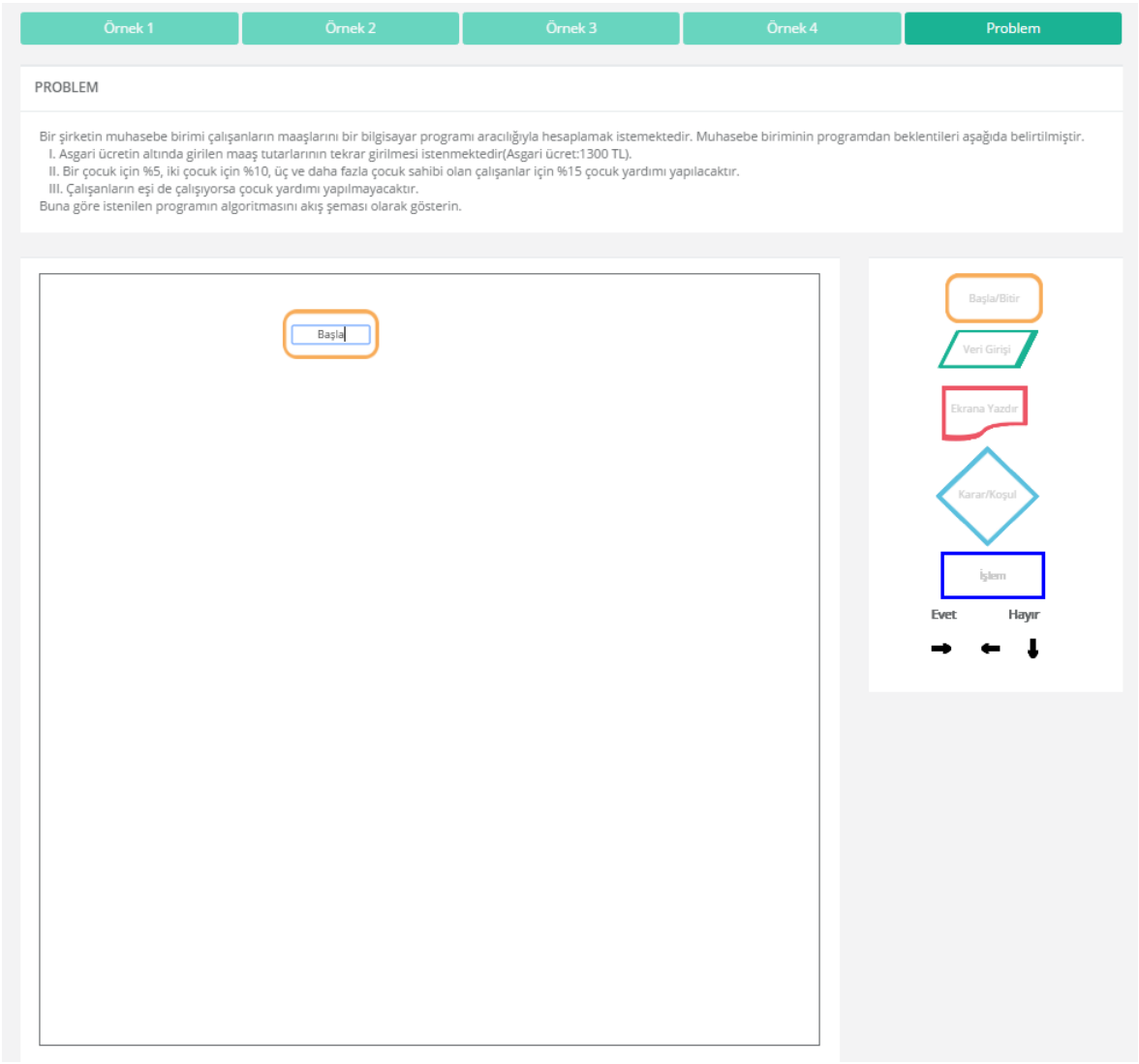
Şekil 3.6: 1. Akış Şemaları Örneği (Deney ve Kontrol Grubu)

İlk örnekte (Şekil 3.6) hem deney grubu hem de kontrol grubu tam çözümlü örneğe çalışmışlardır. İkinci örnekten itibaren deney grubu için örneklerin çözüm adımları sondan başa doğru boşluklar arttırılarak sunulmuş ve en son problem durumu belirtilerek çözülmesi beklenmiştir.



Şekil 3.7: 3. Akış Şemaları Örneği (Deney Grubu)

Şekil 3.7'de görüldüğü gibi karartılmış çözümlü örneklere çalışan deney grubunda çözüm adımlarının bir bölümü boş bırakılarak öğrenenlerden çözmeleri beklenmiştir. Boş bırakılan adımların her biri renk etiketlerine uygun olarak tasarlanmıştır. Örnek 3, kontrol grubuna Şekil 3.6'da olduğu gibi tam çözümlü olarak sunulmuştur.



Şekil 3.8: Akış Şemaları Problem Sorusu (Deney ve Kontrol Grubu)

Hem deney grubu hem de kontrol grubu 4 çözümlü örneğe çalıştıktan sonra Şekil 3.8'de gösterilen problem ifadesiyle karşılaşmışlardır. Problem bölümünde farklı yollar ile sonuca ulaşılabileceğinden hangi çözüm adımlarını kullanacağı belirtilmemiş, öğrenenlerden bu adımları kendilerinin eklemesi beklenmiştir. Çözüm adımlarını eklerken sürükle bırak yöntemi kullanılmıştır.

3.6.3. Öğrenme Ortamı'nın Teknik Özellikleri

Öğrenme ortamı sunucu taraflı açık kaynak kodlu PHP web programlama dili ile geliştirilmiştir. Arayüz tasarımında HTML ve Javascript dilleri kullanılmıştır. Veritabanı yönetim sistemi olarak da açık kaynak olan MySQL kullanılmıştır. Arayüz tasarımının sade, kullanışlı olması ve tarayıcı farkına düşmemesi için

"Bootstrap" CSS framework kullanılmıştır. Kayıtların rahat işlenebilmesi, tüm katılımcılara aynı anda ulaşılabilmesi için web yazılımı tercih edilmiştir.

3.7. Uygulama Süreci

Araştırma için ilk olarak Hacettepe Üniversitesi Etik Komisyonu'ndan gerekli izinler alınmıştır (Bkz. Ek 1). Etik onayının ardından uygulamanın yapılacağı üniversiteden de uygulama izni alınmıştır (Bkz. Ek 2).

Gerekli izinler alındıktan sonra geliştirilen ön test deney ve kontrol grubuna sınıf ortamında uygulanmıştır. Çalışmada kullanılan ön test ve son test katılımcılara kağıt kalem sınavı olarak uygulanmıştır. Ön test uygulandıktan bir hafta sonra öğrenme ortamı üzerinden uygulama süreci başlamıştır.

Uygulamada araştırmacı tarafından geliştirilen web tabanlı öğrenme ortamı kullanılmıştır. Uygulama aşamasında öğrenenler hem sözde kod hem de akış şemaları uygulamalarına algoritmikyapiiskelesi.com adresinden erişim sağlamıştır. Web tabanlı öğrenme ortamında uygulama süreci 2 hafta sürmüştür. İlk hafta sözde kod uygulaması gerçekleştirilirken, ikinci hafta akış şemaları uygulaması gerçekleştirilmiştir.

Katılımcılar deney ve kontrol grubuna göre web tabanlı öğrenme ortamına farklı arayüzler ile erişim sağlamıştır. Sözde kod uygulamasında deney ve kontrol grubu 4 adet çözümlü örneğe çalışıp, çözümü boş bırakılan problemi çözmüşlerdir. Problem çözümünün ardından bilişsel yük derecelendirme ölçeği sunulmuş ve katılımcılardan yanıtlamaları beklenmiştir. Öğrenme ortamında deney grubuna karartılmış çözümlü örnekler sunulmuştur. Yani ilk örnek tam çözümü ile birlikte, ikinci örnek sondan iki adımı boş bırakılmış haliyle sunulmuş ve çözümlü örneklerin adımlarındaki boşluklar giderek arttırılarak devam etmiştir. Kontrol grubunda ise 4 örneğin tamamı çözümlü olarak sunulmuştur. Ancak hem çözümlü örneklere çalıştıktan sonra sunulan problem, hem bilişsel yük ölçeği hem de ön-son test deney ve kontrol grubunda farklılık göstermemiştir.

Akış şemaları uygulamasında da sözde kod uygulamasına benzer bir süreç uygulanmıştır. Deney ve kontrol grubu yine 4 çözümlü örneğe çalışıp 1 problemi çözmüş, ardından da bilişsel yük ölçeğini doldurmuştur. Akış şemaları uygulamasında deney grubuna karartılmış çözümlü örnekler sunulurken, kontrol

grubuna geleneksel çözümlü örnekler sunulmuştur. Uygulama süreci tamamlandıktan bir hafta sonra uygulanan iki yöntemin etkililiğini belirlemek için son test uygulanmıştır.

Sözde kod ve akış şemaları uygulamaları, hem deney hem de kontrol grubuna birer ders saati süre verilerek dersin uygulama aşamasında bilgisayar laboratuvarlarında gerçekleştirilmiştir. Dersin kuramsal aşamasında öğretim üyesi ile birlikte konu anlatımı gerçekleştirilmiştir. Konu anlatımı hem deney grubuna hem de kontrol grubuna aynı zamanda ve aynı ortamda gerçekleştirilmiştir. Konu anlatımında, programlama ile ilgili temel kavramlara, algoritma kullanımının önemine ve algoritma oluşturma türlerine değinilmiştir. Öğrenenler algoritma oluşturma ile ilgili ilk deneyimi uygulama aşamasında yaşamışlardır.

Olası teknik problemlere karşı laboratuvarlar uygulamadan hemen önce test edilmiş, bağlantı sorunları giderilmiş ve bağlantı sorunu olan bilgisayarlar kullanıma kapatılmıştır. Aynı zamanda katılımcıların tamamı sisteme giriş yapana kadar diğer katılımcıların uygulamaya başlamaması gerektiği duyurulmuştur.

3.8. Araştırmanın İç ve Dış Geçerliği

3.8.1. Araştırmanın İç Geçerliği

İç geçerlik bir araştırmada bağımlı değişken üzerindeki etkinin gerçekten bağımsız değişkenden kaynaklı olup olmadığı ile ilgilidir. Araştırmalarda bağımsız değişken dışında kaynaklanabilecek bir dış etkinin kontrol altına alınması, yani iç geçerliği tehdit eden faktörleri ortadan kaldırmak veya en aza indirgemek amaçlanır.

Katılımcıların seçimi, katılımcı kaybı, uygulama ortamı, ön-test uygulanması, katılımcıların olgunlaşması, istatistiksel regresyon, veri toplama aracı, katılımcıların geçmişi, etkileşim ve beklentiler başlıca iç geçerlik tehditlerindendir (Büyüköztürk, 2014; Fraenkel vd., 2012). Araştırmada deneysel araştırma yöntemlerinden ön test – son test kontrol gruplu desen kullanılmıştır. Fraenkel ve arkadaşlarına (2012) göre bu desenin kullanılmasıyla hangi iç geçerlik tehditlerinin ne derecede kontrol altına alındığına ilişkin gösterimi ve araştırmacı olarak bu tehditlere karşı alınan önlemler Tablo 3.6'da gösterilmiştir.

Tablo 3.6: İç Geçerlik Tehditleri ve Gerçekleştirilen Deneysel Müdahaleler

İç Geçerlik Tehditleri	*Ön test – Son test Kontrol Gruplu Desen*	*Deneysel Müdahale*
Katılımcıların Seçimi	++	Katılımcılar gruplara yansız olarak atanmadan önce benzer geçmişlere sahip bir katılımcı havuzu oluşturuldu, dersi alttan alan öğrenciler katılımcı havuzundan çıkartıldı.
Uygulama Ortamı	-	Uygulama birbirine bilgisayar kapasitesi, konum, ışıklandırma açısından özdeş laboratuvarlarda gerçekleştirildi. Bununla birlikte ilk hafta 1. laboratuvarda uygulamaya katılan öğrenciler ikinci hafta 2. laboratuvarda, ilk hafta 2. laboratuvarda uygulamaya katılan öğrenciler ise ikinci hafta 1. laboratuvarda uygulamaya katılmışlardır.
Veri Toplama Aracının Bozulması	+	Bu etki veri toplama aracının gruplara farklı şekilde uygulanmasından ve notlandırılmasından kaynaklanabilir. Başarı testleri çoktan seçmeli olduğu için optik okuyucudan geçirilerek, uygulama esnasındaki veriler de araştırmacı tarafından uzman görüşü alınarak hazırlanan bir değerlendirme tablosuna göre notlandırılmıştır.
Veri Toplayıcı Özellikleri	-	Bu etki veri toplayıcının yaşından, cinsiyetinden, kültüründen ve benzeri özelliklerinden kaynaklanabilir. Sadece iki gruptan veri toplandığı için bir gruba araştırmacı bir gruba dersi veren öğretim üyesi gözetmenlik yapmıştır. İlgili dersi veren öğretim üyesiyle uygulama öncesinde öğrencilerin teknik problemler dışında herhangi bir sorusuna yanıt verilmemesi vb. konularda görüşülerek veri toplayıcı etkisini azaltmak amaçlanmıştır.
Veri Toplayıcı Yanlılığı	-	Veri toplayıcı yanlılığını önlemek için uygulamanın iki haftasında her veri toplayıcısı bir gruba birer kez gözetmenlik yapmıştır.
Test Etkisi	+	Deneysel müdahaleden önce grupların denk olup olmadığını görmek ve bağımlı değişkende gözlenmesi beklenen değişimin gerçekten bu müdahaleden kaynaklanıp kaynaklanmadığını görmek için gruplara ön test yapılmıştır. Ön test etkileşim tehditini azaltmak için son test öğrenenlere uygulamadan bir hafta sonra verilmiştir. Yani ön test ile son test arasında 3 hafta süre bırakılmıştır.
Katılımcıların Tutumu	-	Katılımcıların tutumunun olumsuz etkilenmemesi için yapılan uygulamanın bilimsel bir çalışmaya dayandığından bahsedilmiş, verilen başarı testlerinin ise herhangi bir şekilde değerlendirilmeyeceği açıklanmıştır.
İstatistiksel Regresyon	++	Katılımcılar gruplara yansız atanarak bu etkiye karşı güçlü bir kontrol oluşturulmuştur.
Uygulama Etkisi	-	Uygulama etkisini ortadan kaldırmak için uygulama haftalarının kuramsal derslerinde sadece dersi veren öğretim üyesi anlatımı gerçekleştirmiştir.

Not: (++) = güçlü kontrol, tehditin oluşmaması muhtemel; (+) = orta derecede kontrol, tehdit oluşabilir; (-) zayıf kontrol, tehditin oluşması muhtemel.

3.8.2. Araştırmanın Dış Geçerliği

Dış geçerlik çalışmadan elde edilen sonuçların evrene genellenebirlik derecesidir (Fraenkel vd., 2012). Bu araştırmanın katılımcılarını, BÖTE Bölümü'nde, Programlama Dilleri I dersini ilk kez alan öğrenciler oluşturmuştur. Dolayısıyla çalışmanın evrene genellenebilirliği sınırlı olup sadece benzer özellikteki gruplara genellenebilir.

4. BULGULAR

Bu bölümde her bir araştırma problemine yanıt aramak üzere gerçekleştirilen istatistiksel analiz bulgularına yer verilmiştir.

4.1. Uygulanan Öğretim Yöntemlerinin Başarıya Etkisine İlişkin Bulgular

Araştırmanın birinci alt problemi için "Algoritma öğretiminde geleneksel çözümlü örnek ve karartılmış çözümlü örnek yöntemlerinin kullanıldığı grupların son test performanslarında anlamlı bir farklılık var mıdır?" sorusunun yanıtlanmasında kovaryans analizi (ANCOVA) kullanılmıştır.

Katılımcıların deney ve kontrol grubuna göre ön test ve son testten aldıkları puanlara ilişkin betimsel istatistikler Tablo 4.1'de verilmiştir.

Tablo 4.1: Katılımcıların Ön Test - Son Test Puanlarına İlişkin Betimsel İstatistikler

	Grup	*n*	*Ortalama*	*ss*
Başarı Ön Test	Deney	17	7,59	2,12
	Kontrol	16	7,94	1,84
	Genel	33	7,76	1,97
Başarı Son Test	Deney	17	11,89	1,90
	Kontrol	16	10,56	2,13
	Genel	33	11,24	2,09

Tablo 4.1 incelendiğinde ortalama değeri temel alınarak kontrol grubunun deney grubuna göre daha yüksek ön bilgiye sahip olduğu söylenebilir. Son teste bakıldığında ise deney grubunun kontrol grubuna göre daha iyi performans gösterdiği görülmüştür. Bu sonuçların istatistiksel olarak anlamlı olup olmadığını görmek amacıyla tek faktörlü kovaryans analizi kullanılmıştır. ANCOVA'ya ilişkin bulgular Tablo 4.2'de gösterilmiştir.

Tablo 4.2: Başarı Ön Test Puanlarına Göre Düzeltilmiş Son Test Puanlarına İlişkin ANCOVA Sonuçları

Varyansın Kaynağı	*Kareler Toplamı*	*sd*	*Kareler Ortalaması*	*F*	*p*	*η2*
OnTest	18,52	1	18,52	5,18	0,030	0,147
Grup	17,32	1	17,32	4,85	0,036	0,139
Hata	107,18	30	3,57			
Düzeltilmiş Toplam	140,061	32				

Tablo 4.2'de görüldüğü üzere grupların ön test puanlarına göre düzeltilmiş son test puanları arasında, deney grubunun lehine istatistiksel olarak anlamlı bir farklılık

vardır ($F(1,30)=4,85$, $p<0,05$). Yani algoritma öğretiminde, karartılmış çözümlü örnek yönteminin geleneksel çözümlü örnek yöntemine göre daha etkili olduğu söylenebilir. Buna ek olarak Tablo 4.2'de etki büyüklüğü (η^2) değerine bakıldığında, bağımlı değişken üzerindeki değişimin %14'ünün uygulanan yöntemden kaynaklandığı söylenebilir. Cohen'e (1998) göre elde edilen bu sonuçtan çalışmanın geniş etki büyüklüğüne sahip olduğu çıkarılabilir.

4.2. Uygulanan Öğretim Yöntemlerinin Bilişsel Yüke Etkisine İlişkin Bulgular

Araştırmanın ikinci alt problemi için "Algoritma öğretiminde geleneksel çözümlü örnek ve karartılmış çözümlü örnek yöntemlerinin kullanıldığı grupların bilişsel yüklenme düzeylerinde anlamlı bir farklılık var mıdır?" sorusunun yanıtlanmasında tekrarlı ölçümler için ANOVA kullanılmıştır.

Katılımcıların deney ve kontrol grubuna göre sözde kod ve akış şemaları uygulamalarından hemen sonra sunulan bilişsel yük anketine verdikleri cevaplara ilişkin betimsel istatistikler Tablo 4.3'te gösterilmiştir.

Tablo 4.3: Katılımcıların Bilişsel Yük Puanlarına İlişkin Betimsel İstatistikler

Grup	*Sözde Kod Uygulaması Sürecinde Oluşan Bilişsel Yüklenme*			*Akış Şemaları Uygulaması Sürecinde Oluşan Bilişsel Yüklenme*		
	n	*Ortalama*	*ss*	*n*	*Ortalama*	*ss*
Deney	17	6,35	2,060	17	5,94	2,703
Kontrol	16	6,06	2,568	16	7,25	1,844
Genel	33	6,21	2,288	33	6,58	2,385

Tablo 4.4 incelendiğinde sözde kod uygulama sürecinde deney grubunun kontrol grubuna göre daha yüksek bilişsel yüklenmeye sahip olduğu görülürken, akış şemaları uygulama sürecinde deney grubunun kontrol grubuna göre daha düşük bilişsel yüklenmeye sahip olduğu görülmüştür. Ortalama değerleri temel alınarak uygulamanın genelinde kontrol grubunun deney grubuna göre daha yüksek bilişsel yüklenmeye sahip olduğu görülmektedir. Bu sonuçların istatistiksel olarak anlamlı olup olmadığını görmek amacıyla tekrarlı ölçümler için iki faktörlü ANOVA kullanılmıştır. Bu analize ilişkin bulgular Tablo 4.4'te gösterilmiştir.

Tablo 4.4: Gruplar Arası Bilişsel Yüklenme Düzeylerine İlişkin Tekrarlı Ölçümler İçin İki Faktörlü ANOVA Analizi Sonuçları

Varyansın Kaynağı	*Kareler Toplamı*	*sd*	*Kareler Ortalaması*	*F*	*p*	η^2
Gruplar Arası						
Grup (Deney/Kontrol)	4,274	1	4,274	0,570	0,456	0,018
Hata	232,483	31	7,499			
Gruplar İçi						
Ölçüm (1. BY Puanı/ 2. BY Puanı)	2,480	1	2,480	0,752	0,393	0,024
Grup*Ölçüm	10,541	1	10,541	3,195	0,084	0,093
Hata	102,278	31	3,299			

Tablo 4.5'te görüldüğü üzere deney ve kontrol grubunun her iki uygulama sonrasında ölçülen bilişsel yük puanları arasında anlamlı bir farklılık yoktur ($F(1,31)=0,570$, $p>0,05$). Yani farklı ölçümleri hesaba katmadan deney ve kontrol grubundaki öğrenenlerin çözümlü örneklere çalışırken algıladıkları zihinsel çabalarında anlamlı bir farklılık bulunmamıştır.

Ölçüm etkisi incelendiğinde farklı grupları hesaba katmadan öğrencilerin sözde kod ve akış şeması uygulamaları sonrası bilişsel yük puanları arasında anlamlı bir farklılık bulunmamıştır ($F(1,31)=0,752$, $p>0,05$). Yani katılımcıların 2 uygulamadaki bilişsel yüklenme düzeylerinde anlamlı bir farklılık görülmemiştir.

Etkileşim etkisi incelendiğinde ise deney ve kontrol gruplarının iki bilişsel yük ölçümünden elde edilen puanlarda anlamlı bir farklılık göstermediği görülmüştür ($F(1,31)=3,195$, $p>0,05$). Yani uygulanan yönteme göre deney ve kontrol gruplarının bilişsel yüklenme düzeylerinde anlamlı bir farklılık olmadığı belirlenmiştir.

4.3. Uygulanan Yöntemlerin Öğretim Verimliliği Açısından Karşılaştırılmasına İlişkin Bulgular

Araştırmanın üçüncü alt problemi için "Algoritma öğretiminde öğretim verimliliği açısından geleneksel çözümlü örnek ile karartılmış çözümlü örnek yöntemleri arasında anlamlı bir farklılık var mıdır?" sorusunun yanıtlanmasında bağımsız örneklem t-testi kullanılmıştır.

Uygulanan yönteme göre öğretim verimliliği puanlarına ilişkin betimsel istatistikler Tablo 4.5'te gösterilmiştir.

Tablo 4.5: Öğretim Verimliliği Puanlarına İlişkin Betimsel İstatistikler

Grup	*n*	*Ortalama*	*ss*
Karartılmış Çözümlü Örnekler	17	0,39	2,03
Geleneksel Çözümlü Örnekler	16	-0,41	1,18
Genel	33	0,00	1,11

Tablo 4.5 incelendiğinde ortalama değerleri temel alındığında karartılmış çözümlü örnek yönteminin daha verimli olduğu söylenebilir. Bu verilerin istatistiksel olarak anlamlı olup olmadığını görmek için t testi kullanılması planlanmıştır.

t testine ilişkin bulgular Tablo 4.6'de gösterilmiştir.

Tablo 4.6: Öğretim Verimliliğine İlişkin t Testi Sonuçları

Grup	*Ortalama*	*ss*	*sd*	*t*	*p*
Karartılmış Çözümlü Örnekler	0,39	2,03	31	-2,219	0,034
Geleneksel Çözümlü Örnekler	-0,41	1,18			

Tablo 4.6'da görüldüğü üzere karartılmış çözümlü örnek yöntemi ile geleneksel çözümlü örnek yönteminin öğretim verimliliği açısından aralarında anlamlı farklılık bulunmuştur ($t(31)=-2,219$, $p<0,05$). Yani karartılmış çözümlü örnek yöntemi geleneksel çözümlü örnek yöntemine göre daha verimlidir.

4.4. Uygulanan Yöntemlerin Süreç Performanslarına Etkisine İlişkin Bulgular

Araştırmanın dördüncü alt problemi için "Geleneksel çözümlü örnek ve karartılmış çözümlü örnek yöntemlerinin kullanıldığı grupların süreçteki problem çözme performanslarında anlamlı bir farklılık var mıdır?" sorusunun yanıtlanmasında tekrarlı ölçümler için ANOVA kullanılmıştır. Problem çözme performansları uygulama sürecinde web ortamında çözümlü örneklerin ardından sorulan tek soruluk probleme göre değerlendirilmiştir.

Katılımcıların deney ve kontrol grubuna göre sözde kod ve akış şemaları uygulamasında çözümlü örneklere çalıştıktan hemen sonra sunulan problemlere verdikleri cevaplara ilişkin betimsel istatistikler Tablo 4.7'de gösterilmiştir.

Tablo 4.7: Katılımcıların Çözümlü Örnekler Sonrasında Sunulan Problemi Çözme Performanslarına İlişkin Betimsel İstatistikler

Grup	*Sözde Kod Uygulaması Sürecinde Problem Çözme Performansı*			*Akış Şemaları Uygulaması Sürecinde Problem Çözme Performansı*		
	n	*Ortalama*	*ss*	*n*	*Ortalama*	*ss*
Deney	17	7,82	3,060	17	9,71	3,000
Kontrol	16	6,22	4,090	16	8,81	2,670

Genel	33	7,05	3,872	33	9,27	2,837

Tablo 4.7 incelendiğinde ortalama değerler baz alınarak hem sözde kod hem de akış şemaları uygulamaları sürecinde deney grubunun kontrol grubuna göre daha yüksek problem çözme performansı gösterdiği görülmüştür. Bu sonuçların istatistiksel olarak anlamlı olup olmadığını görmek amacıyla tekrarlı ölçümler için iki faktörlü ANOVA kullanılmıştır. Bu analize ilişkin bulgular Tablo 4.8'de gösterilmiştir.

Tablo 4.8: Gruplar Arası Çözümlü Örnekler Sonrasında Sunulan Problemi Çözme Performanslarına İlişkin Tekrarlı Ölçümler İçin İki Faktörlü ANOVA Analizi Sonuçları

Varyansın Kaynağı	*Kareler Toplamı*	*sd*	*Kareler Ortalaması*	*F*	*p*	η^2
Gruplar Arası						
Grup (Deney/Kontrol)	25,720	1	25,720	1,633	0,211	0,050
Hata	488,235	31	15,750			
Gruplar İçi						
Ölçüm (1. BY Puanı/ 2. BY Puanı)	82,571	1	82,571	11,572	0,002	0,272
Grup*Ölçüm	2,086	1	2,086	2,086	0,593	0,009
Hata	221,187	31	7,135			

Tablo 4.8'de görüldüğü üzere deney ve kontrol grubunun her iki uygulama sırasında ölçülen problem çözme performans puanları arasında anlamlı bir farklılık yoktur ($F(1,31)=1,633$, $p>0,05$). Yani farklı ölçümleri hesaba katmadan deney ve kontrol grubundaki öğrenenlerin çözümlü örneklere çalıştıktan sonraki problem çözme performanslarında anlamlı bir farklılık bulunmamıştır.

Ölçüm etkisi incelendiğinde farklı grupları hesaba katmadan öğrencilerin sözde kod ve akış şeması uygulamalarındaki problem çözme performans puanları arasında anlamlı bir farklılık bulunmuştur ($F(1,31)=11,572$, $p<0,05$). Yani problem çözme performansları açısından değerlendirildiğinde sözde kod ve akış şemaları uygulamalarında anlamlı bir farklılık görülmüştür.

Etkileşim etkisi incelendiğinde ise deney ve kontrol gruplarının sözde kod ve akış şeması problemlerinden elde edilen puanlarda anlamlı bir farklılık göstermediği görülmüştür ($F(1,31)=2,086$, $p>0,05$). Yani uygulanan yönteme göre deney ve kontrol gruplarının çözümlü örneklerin ardından sunulan problemi çözme performanslarında anlamlı bir farklılık göstermediği görülmüştür.

4.5. Uygulanan Yöntemlerin Final Puanlarını Yordama Derecesine İlişkin Bulgular

Araştırmanın beşinci alt problemi için "Algoritma öğretiminde kullanılan öğretim yöntemleri Programlama Dilleri I dersi final puanlarını yordamakta mıdır?" sorusunun yanıtlanmasında basit doğrusal regresyon analizi kullanılmıştır.

Katılımcıların deney ve kontrol grubuna göre Programlama Dilleri 1 dersi final sınavından aldıkları puanlara ilişkin betimsel istatistikler Tablo 4.9'da verilmiştir.

Tablo 4.9: Katılımcıların Final Puanlarına İlişkin Betimsel İstatistikler

	Grup	*n*	*Ortalama*	*ss*
Final Puanı	Deney	17	59,17	19,94
	Kontrol	16	43,56	21,28
	Genel	33	51,60	21,81

Tablo 4.9 incelendiğinde ortalama değerler baz alınarak deney grubunun kontrol grubuna göre Programlama Dilleri I dersi final puanında daha başarılı olduğu görülmüştür. Kullanılan öğretim yönteminin dersin final puanlarını yordama derecesini görmek için basit doğrusal regresyon analizi uygulanmıştır. Bu analize ilişkin bulgular Tablo 4.10'da yer almaktadır.

Tablo 4.10: Uygulanan Yöntemlerin Final Sınavını Yordama Derecesini Ölçen Regresyon Analizi Sonuçları

Model	*B*	*Standart Hata*	*β*	*t*	*p*
Sabit	27,949	11,473	2,086	2,436	0,021
Öğretim Yöntemi	15,614	7,191	7,135	2,171	0,038

Regresyon denklemi: Final Puanı = 27,949 + 15,614 * Öğretim Yöntemi

Tablo 4.10'de görüldüğü üzere kullanılan öğretim yöntemlerinin final sınavı puanlarının anlamlı bir yordayıcısı olduğu görülmektedir ($R=0,363$, $R^2=0,132$, $t=2,171$, $p<0,05$). Ayrıca final puanına ilişkin varyansın %13,2'sinin kullanılan öğretim yöntemiyle açıklandığı görülmüştür.

5. SONUÇ ve TARTIŞMA

Bu bölümde araştırmanın bulgularına dayalı olarak ulaşılan sonuçların özetine yer verilmiştir. Ayrıca bu sonuçlar alanyazında benzer çalışmaların sonuçlarıyla karşılaştırılarak bir değerlendirme yapılmıştır. Yapılan çalışmada genel olarak üniversite öğrencilerine algoritma öğretmede çözümlü örnek alanyazınında etkililiği kanıtlanmış geleneksel çözümlü örnek ve karartılmış çözümlü örnek yöntemlerinin bilişsel yüke ve başarıya etkileri incelenmiştir.

Karartılmış çözümlü örnek yönteminin uygulandığı durumda geleneksel çözümlü örnek yöntemine göre algoritma oluşturma konusunda üniversite öğrencilerinin başarısı daha yüksektir. Yani yeterli deneyimi olmayan öğrenenlere algoritma öğretmede karartılmış çözümlü örnek yöntemi başarı açısından daha etkili bir stratejidir. Bu sonuç, alanyazındaki bulguları destekler niteliktedir (Atkinson vd., 2003; Kissane vd., 2008; Renkl ve Atkinson, 2003; Renkl vd., 2004). Bu sonucun karartılmış çözümlü örneklerin giderek artan problem çözme becerisinin uygulanmasından dolayı uzmanlığın ters tepme etkisine karşı etkili bir yöntem (Kalyuga, 2007) olduğundan kaynaklanabileceği söylenebilir. Buna ek olarak programlamanın iyi yapılandırılmış bir alan olduğu düşünülerek karartılmış çözümlü örnek yönteminin iyi yapılandırılmış alanlarda daha etkili olduğu bulgusu da (Renkl, 2014) desteklenmiştir. Ancak bu sonuç karartılmış çözümlü örneklerin geleneksel çözümlü örneklere göre öğrenci başarısını arttırmadığı bulgusuna ulaşan çalışmalarla (Demiraslan Cevik ve Andre, 2014; Reisslein vd., 2006; Schwonke vd., 2009) çelişmektedir. Bunun için yalnızca öğrenci başarısına bakarak bir değerlendirme yapmak doğru olmayabilir. Uygulanan yöntemlerin sadece başarı açısından etkililiğinden ziyade öğretim verimliliğini incelemek daha anlamlı olacaktır (Brünken vd., 2010). Verimlilikten kasıt başarının yanı sıra bilişsel yüklenme, çalışma süreleri gibi değişkenleri de göz önünde bulundurmaktır. Bu çalışmada verimlilik ölçütü olarak başarı ve bilişsel yüklenme düzeyleri ele alınmıştır.

Uygulanan öğretim yöntemine göre deney ve kontrol grubunun bilişsel yüklenme düzeyleri arasında anlamlı bir farklılık yoktur. Bununla birlikte sözde kod ve akış şemaları uygulamaları için ayrı ayrı bilişsel yüklenme düzeyleri incelendiğinde

deney ve kontrol grupları farklılık göstermiştir. Sözde kod uygulamasında deney grubundaki öğrenenlerde daha fazla bilişsel yüklenme olduğu görülürken, akış şemaları uygulamasında kontrol grubundaki öğrenenlerde daha fazla bilişsel yüklenme olduğu görülmüştür. Ancak iki uygulama sonrası ölçülen bilişsel yüklenme düzeylerinin ortalamaları karşılaştırıldığında kontrol grubundaki öğrenenlerde daha fazla bilişsel yüklenme oluştuğu görülmüştür. Bu bulgular alanyazında ulaşılan sonuçlarla çelişkilidir (Demiraslan Cevik ve Andre, 2013; Schwonke vd., 2009). Alanyazında karartılmış çözümlü örneklerin geleneksel çözümlü örneklere göre daha fazla problem çözme gereksiniminden dolayı daha fazla bilişsel yüklenmeye neden olduğu belirtilmiştir. Bu sonuç, etkililiği karşılaştırılan iki yöntem dışında çözümlü örnek alanyazınında bilişsel kapasitenin daha verimli kullanılmasını sağladığı ortaya konulan tasarım ilkelerinin geliştirilen öğrenme ortamında kullanılması ile açıklanabilir. Yani alt hedeflere ayırma, açıklayıcı metinler ve basit ilişkilendirme ilkelerinin kullanımı gibi stratejiler konu dışı bilişsel yük oluşumunu engelleyerek/en aza indirgeyerek karartılmış çözümlü örnek yönteminde daha fazla oluşması ön görülen asıl bilişsel yükün öğrenmeye katkı sağlayacak etkili bilişsel yük olarak işlenmesine olanak sağlamış olabilir. Ayrıca Margulieux ve Catrambone (2016) çözümlü örneklerde hem alt hedeflerin vurgulanması hem de açıklayıcı metinlerin kullanılmasının en verimli strateji olabileceğini belirtmiştir.

Üniversite öğrencilerine algoritma öğretiminde karartılmış çözümlü örnek yönteminin geleneksel çözümlü örnek yöntemine göre daha verimli olmuştur. Yani yeterli deneyimi olmayan öğrenenlere algoritma öğretmede hem başarı hem de bilişsel yüklenme düzeyleri hesaba katıldığında karartılmış çözümlü örnek yöntemi daha verimli bir stratejidir. Bu sonuç, alanyazındaki bulguları destekler niteliktedir (Abdul-rahman ve Boulay, 2014; Atkinson vd., 2003; Renkl vd., 2004; van Merrienboer, 1990). Yapılan analiz sonucunda bu sonuç istatistiksel olarak anlamlı olsa da anlamlılık düzeyi verimlilik açısından bu iki yöntem arasında çok ciddi bir farklılık bulunmadığını göstermiştir. Bunun nedeni her iki grup için de çalışma sürelerinin bir ders saati olarak belirlenmesi olabilir. Çünkü deney grubundaki katılımcılar çözümlü örneklere çalışırken aynı zamanda kısmi çözüm adımları verilen problemleri tamamlarken, kontrol grubundaki katılımcılar yalnızca çözümlü örneklere çalışmış ve sadece örneklerden sonra sunulan problemi çözmüşlerdir.

Bu nedenle çalışma süreleri de hesaba katılarak bir değerlendirme yapıldığında daha anlamlı sonuçlar ortaya konulabilir.

Deney ve kontrol grubunun uygulama esnasındaki süreç performansları arasında anlamlı bir farklılık yoktur. Süreç performansları her iki uygulama için de öğrenenlerin çözümlü örneklere çalıştıktan sonra sunulan probleme verdikleri yanıtlar dikkate alınarak değerlendirilmiştir. Deney grubu ile kontrol grubunun süreç performansları arasında anlamlı bir farklılık görülmemesi alanyazında ulaşılan sonuçlarla çelişkilidir (Abdul-rahman ve Boulay, 2014; van Merrienboer, 1990; Vieira vd., 2015). Çalışma sürelerinin her iki grup için de sabit tutulması bu sonucun nedeni olabilir. Van Merrienboer (1990) ile Si vd. (2013) tarafından tamamlamalı çözümlü örneklerin daha zaman alıcı olduğu ortaya konulmuştur. Dolayısıyla bir tamamlamalı çözümlü örnek türü olan karartılmış çözümlü örnek yönteminin de daha fazla zaman alıcı olması beklenen bir durumdur. Bununla birlikte hem deney hem de kontrol grubundaki öğrenenler akış şemaları uygulamasında sözde kod uygulamasına göre daha başarılı olmuştur. Bunun sebebi olarak da deneyim etkisi gösterilebilir. Yani katılımcılar ilk uygulamada hem öğrenme ortamı hem de algoritma oluşturma konusunda deneyim kazandığı için, akış şemaları uygulamasında daha yüksek başarı göstermiş olabilir.

Algoritma öğretiminin karartılmış çözümlü örnek yöntemiyle gerçekleştirildiği deney grubunun, geleneksel çözümlü örnek yöntemiyle gerçekleştirildiği kontrol grubuna göre ders sonunda uygulanan final sınavı puanları açısından daha başarılıdır. Yani algoritma öğretimi sırasında karartılmış örnek yöntemi öğrenenlerin programlama performansını arttırmaktadır. Buna ek olarak kullanılan öğretim yöntemleri dersin final puanlarının anlamlı bir yordayıcısı olmuştur. Öğretim yönteminin, final puanlarına ilişkin varyansın %13,2'sini açıkladığı görülmüştür.

Özetle, üniversite öğrencilerine algoritma öğretmede kullanılan karartılmış çözümlü örnek yöntemi geleneksel çözümlü örnek yöntemine göre başarı açısından daha etkili bir stratejidir. Deney ve kontrol grubundaki öğrenenlerin bilişsel yüklenme düzeyleri arasında anlamlı bir farklılık yoktur. Bununla birlikte hem başarının hem de bilişsel yükün hesaba katılarak belirlendiği öğretim verimliliği açısından bakıldığında karartılmış çözümlü örnekler daha verimli bir stratejidir. Buna ek olarak algoritma öğretimi sonrası programlama becerileri incelendiğinde karartılmış çözümlü örnek yöntemi aracılığıyla öğrenen grubun

dersin final sınavı puanları dikkate alınarak programlama performansının daha yüksek olduğu söylenebilir. Ayrıca kullanılan öğretim yönteminin programlama final sınavı başarısını etkileyen önemli bir faktördür. Sonuç olarak, alanyazında etkililiği kanıtlanmış çözümlü örnek tasarım ilkelerine göre geliştirilen web tabanlı öğrenme ortamında karartılmış çözümlü örnek yönteminin öğrenenlerin algoritma oluşturma ve programlama performansını arttırdığı görülmüştür.

6. ÖNERİLER

Bu bölümde araştırma sonuçları doğrultusunda birtakım önerilere yer verilmiştir. Bu öneriler, gelecekte yapılması önerilen araştırmalar ve uygulamaya dönük araştırmalar olmak üzere iki başlıkta ele alınmıştır.

6.1. Araştırmaya Dönük Öneriler

Bu çalışmada üniversite öğrencilerine algoritma öğretiminde iki farklı çözümlü örnek yönteminin başarıya ve bilişsel yüke etkileri incelenmiştir. Algoritma öğretiminin yanı sıra, karar yapıları, döngüler, diziler ve fonksiyonlar gibi programlama bileşenlerinin öğretimine yönelik çözümlü örnek yöntemlerinin etkililiği incelenebilir. Ya da daha geniş kapsamda programlama ile ilgili tüm bileşenlerin öğretiminde çözümlü örnek kullanımının etkileri incelenebilir.

Bu çalışmada bilişsel yüklenme düzeylerini belirlemek için öznel ölçümler kullanılmıştır. Bununla birlikte bu ölçümlerin, bilişsel yüklenme düzeyini öğrenenlerin kalp atış hızına göre belirleme, göz izleme cihazlarıyla belirleme gibi nesnel ölçümlerle desteklenmesi daha güvenilir sonuçlar elde edilmesi açısından önemli görülmektedir.

Bu çalışmada ve alanyazındaki diğer çalışmalarda programlama öğretiminde çözümlü örnek kullanımı genellikle üniversite düzeyinde incelenmiş ve ilköğretim düzeyinde yapılan bir çalışmaya rastlanılamamıştır. Öğrenenlere çözümlü örnekler aracılığıyla yönlendirme sağlamanın daha önemli olabileceği yaş aralığında olan ilköğretim düzeyindeki öğrencilere programlama öğretiminde çözümlü örnek yöntemlerinin etkileri incelenebilir. Ayrıca kodlama eğitiminin ilköğretim sürecinin her kademesinde gündemde olması sebebiyle bu konuda yapılacak çalışmaların önemli olabileceği düşünülmektedir.

Bu çalışmada öğretim verimliliği ölçümünde başarı ve bilişsel yüklenme düzeyleri hesaba katılmıştır. Alanyazında yapılan birçok çalışmada çözümlü örnek yönteminin verimliliğinin belirlenmesinde çalışma süreleri de hesaba katılmıştır. Özellikle karartılmış çözümlü örneklerin geleneksel çözümlü örneklere göre daha fazla problem çözme gereksiniminden dolayı çalışma sürelerinin esnek tutulması

önerilmektedir. Dolayısıyla öğrenenlerin çözümlü örneklere çalışma ve problem çözme sürelerinin de değerlendirilmesi önemli görülmektedir.

Çalışmada karartılmış çözümlü örnek ve geleneksel çözümlü örnek yöntemleri karşılaştırılmış ve karartılmış çözümlü örneklerin daha verimli olduğu sonucuna ulaşılmıştır. Bundan sonraki yapılacak çalışmalarda programlama öğretiminde karartılmış çözümlü örnek yöntemi ile iyi yapılandırılmış alanlarda etkililiği kanıtlanmış öz açıklama sürecinin de entegre edildiği çözümlü örnek yöntemlerinin etkililiği karşılaştırılabilir.

Bu çalışmada sadece nicel veriler ele alınmıştır. Uygulanan yönteme ilişkin öğrenci görüşlerinin alınması çalışma sonuçlarını daha anlamlı hale getirebilir. Yapılacak çalışmalarda öğrenci görüşlerinin de hesaba katılması önemli görülmektedir.

6.2. Uygulamaya Dönük Öneriler

Karartılmış çözümlü örnek yöntemine göre geliştirilen web tabanlı öğrenme ortamı üniversite öğrencilerinin algoritma ve programlama performanslarına anlamlı katkılar sağlamıştır. Dolayısıyla sadece BÖTE bölümü değil, yeterli deneyimi olmayan öğrenenlere programlama dersi verilen bütün alanlarda karartılmış çözümlü örnek yönteminin kullanılması önerilmektedir.

Çözümlü örnek yöntemleri, alanyazında etkililiği kanıtlanan tasarım ilkeleriyle bir potada eritildiği takdirde başarılı olabilmektedir. Dolayısıyla öğrenme-öğretme sürecinde yalnızca uzman gözünden problemin çözümünü vermek yeterli olmayacaktır. Buna ek olarak tasarım ilkelerinin hesaba katılması önerilmektedir.

Programlama öğretiminde çözümlü örnek kullanılırken uzmanlığın ters tepme etkisi hesaba katılmalıdır. Öğrenen konu hakkında uzmanlaştıkça çözümlü örnekler gereksiz hale gelebilecektir. Bunun için öğrenme-öğretme sürecinin giderek artan problem çözme becerisi gerektiren uygulamalarla desteklenmesi önerilmektedir.

KAYNAKÇA

Abdul-Rahman, S. S., & Du Boulay, B. (2014). Learning programming via worked-examples: Relation of learning styles to cognitive load. *Computers in Human Behavior*, *30*, 286-298.

Altun, A., & Mazman, S. G. (2012). Programlamaya ilişkin öz yeterlilik algısı ölçeğinin Türkçe formumun geçerlilik ve güvenirlik çalışması. *Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi*, *3*(2), 297-308.

Atkinson, R. K., Renkl, A., & Merrill, M. M. (2003). Transitioning from studying examples to solving problems: Effects of self-explanation prompts and fading worked-out steps. *Journal of Educational Psychology*, *95*(4), 774.

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of educational research*, *70*(2), 181-214.

Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Englewood Cliffs, NJ: Prentice Hall.

Begosso, L. C., Begosso, L. R., Gonçalves, E. M., & Gonçalves, J. R. (2012). *An approach for teaching algorithms and computer programming using Greenfoot and Python.* In Frontiers in Education Conference (FIE), 2012 (pp. 1-6). IEEE.

Bergersen, G. R., & Gustafsson, J. E. (2011). Programming skill, knowledge, and working memory among professional software developers from an investment theory perspective. *Journal of Individual Differences.*

Berthold, K., & Renkl, A. (2009). Instructional aids to support a conceptual understanding of multiple representations. *Journal of Educational Psychology*, *101*(1), 70.

Bucks, G. W. (2010). *A phenomenographic study of the ways of understanding conditional and repetition structures in computer programming languages* (Unpublished Doctoral Dissertation). Purdue University.

Brünken, R., Seufert, T., & Paas, F. (2010). *Measuring cognitive load*. Cambridge University Press.

Büyüköztürk, Ş. (2014). *Deneysel desenler: Öntest sontest kontrol gruplu desen ve veri analizi.* (4. Baskı). *Pegem Yayınları, Ankara*.

Büyüköztürk, Ş., Çakmak, E. K., Akgün, Ö. E., Karadeniz, Ş., & Demirel, F. (2016). *Bilimsel araştırma yöntemleri.* 22. Baskı. Ankara: Pegem Akademi.

Catrambone, R., & Holyoak, K. J. (1990). Learning subgoals and methods for solving probability problems. *Memory & Cognition*, *18*(6), 593-603.

Catrambone, R. (1994). Improving examples to improve transfer to novel problems. *Memory & Cognition*, *22*(5), 606-615.

Catrambone, R. (1995). Aiding subgoal learning: Effects on transfer. *Journal of educational psychology*, *87*(1), 5.

Catrambone, R. (1996). Generalizing solution procedures learned from examples. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*(4), 1020.

Chang, K. E., Chiao, B. C., Chen, S. W., & Hsiao, R. S. (2000). A programming learning system for beginners-A completion strategy approach. *IEEE Transactions on Education*, *43*(2), 211-220.

Chang, T. W., Hsu, J. M., & Yu, P. T. (2011). A Comparison of Single-and Dual-Screen Environment in Programming Language: Cognitive Loads and Learning Effects. *Educational Technology & Society*, *14*(2), 188-200.

Chi, M. T., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive science*, *13*(2), 145-182.

Clark, R. C., Nguyen, F., Sweller, J., & Baddeley, M. (2006). Efficiency in learning: Evidence-based guidelines to manage cognitive load. *Performance Improvement*, *45*(9), 46-47.

Cooper, G., & Sweller, J. (1987). Effects of schema acquisition and rule automation on mathematical problem-solving transfer. *Journal of educational psychology*, *79*(4), 347.

Cooper, G., Tindall-Ford, S., Chandler, P., & Sweller, J. (2001). Learning by imagining. *Journal of Experimental Psychology: Applied*, *7*(1), 68.

Crippen, K. J., & Earl, B. L. (2004). Considering the efficacy of web-based worked examples in introductory chemistry. *Journal of Computers in Mathematics and Science Teaching*, *23*(2), 151-168.

Crippen, K. J., & Earl, B. L. (2007). The impact of web-based worked examples and self-explanation on performance, problem solving, and self-efficacy. *Computers & Education*, *49*(3), 809-821.

Crocker, L., & Algina, J. (1986). *Introduction to classical and modern test theory*. Holt, Rinehart and Winston, 6277 Sea Harbor Drive, Orlando.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* Lawrence Earlbaum Associates. Hillsdale, NJ, 20-26.

Cronbach, L. J., & Furby, L. (1970). How we should measure" change": Or should we?. *Psychological bulletin*, *74*(1), 68.

Çakmak, E. K. (2007). Çoklu ortamlarda dar boğaz: Aşırı bilişsel yüklenme.*Gazi Üniversitesi Gazi Eğitim Fakültesi Dergisi*, *27*(2).

Demiraslan-Çevik, Y., & Andre, T. (2013). Examining Preservice Teachers' Classroom Management Decisions in Three Case-Based Teaching Approaches. *International Journal of Education in Mathematics, Science and Technology, 1(1)*, 25-42

Demiraslan Cevik, Y., & Andre, T. (2014). Studying the impact of three different instructional methods on preservice teachers' decision-making. *Research Papers in Education*, *29*(1), 44-68.

De Wever, B., Schellens, T., Valcke, M., & Van Keer, H. (2006). Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review. *Computers & Education, 46(1)*, 6-28.

Dunican, E. (2002). Making the analogy: Alternative delivery techniques for first year programming courses. *Proceedings from the 14th workshop of the psychology of programming interest group, Brunel University* (89-99).

Durak, G. (2009). *Algoritma konusunda geliştirilen" programlama mantığı öğretici-PM Ö." yazılımının öğrenci başarısına etkisi* (Yayımlanmamış Yüksek Lisans Tezi). Anadolu Üniversitesi, Sosyal Bilimler Enstitüsü, Eskişehir.

Durkin, K., & Rittle-Johnson, B. (2012). The effectiveness of using incorrect examples to support learning about decimal magnitude. *Learning and Instruction*, *22*(3), 206-214.

Eiriksdottir, E., & Catrambone, R. (2011). Procedural instructions, principles, and examples: how to structure instructions for procedural tasks to enhance performance, learning, and transfer. *Human Factors*, *53(6),* 749-770.

Eker, M. (2011). *Algoritmayı anlamak.* (5. Baskı). *Nirvana Yayınları, Ankara*.

Ferrer-Mico, T., Prats-Fernàndez, M. À., & Redo-Sanchez, A. (2012). Impact of Scratch programming on students' understanding of their own learning process. *Procedia-Social and Behavioral Sciences*, *46*, 1219-1223.

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education, 63,* 87-97.

Fraenkel, J. R., Wallen, N. E., & Hyun, H. H. (2012). *How to Design and Evaluate Research in Education* (8th edt.). New York: McGram-Hill Companies.

Gardner, R. C., & Neufeld, R. W. (1987). Use of the simple change score in correlational analyses'. *Educational and Psychological Measurement*, *47*(4), 849-864.

George, D., & Mallery, M. (2003). *Using SPSS for Windows step by step: a simple guide and reference*. Boston, Allynn & Bacon.

Gerjets, P., Scheiter, K., & Catrambone, R. (2004). Designing instructional examples to reduce intrinsic cognitive load: Molar versus modular presentation of solution procedures. *Instructional Science*, *32*(1), 33-58.

Goldenson, D. (1996). Why Teach Computer Programming? Some Evidence about Generalization and Transfer. *National Educational Computing Conference.*

Große, C. S., & Renkl, A. (2007). Finding and fixing errors in worked examples: Can this foster learning outcomes?. *Learning and instruction, 17(6),* 612-634.

Harms, K. J., Rowlett, N., & Kelleher, C. (2015, October). Enabling independent learning of programming concepts through programming completion puzzles. In *Visual*

Languages and Human-Centric Computing (VL/HCC), 2015 IEEE Symposium on (pp. 271-279). IEEE.

Hawi, N. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. *Computers & Education*, *54*(4), 1127-1136.

Hohn, R. L., & Moraes, I. (1998). Use of rule-based elaboration of worked examples to promote the acquisition of programming plans. *Journal of Computer Information Systems*, *38*(2), 35-40.

Hubálovský, Š. (2012, April). *Research of Methods of a Multidisciplinary Approach in the Teaching of Algorithm Development and Programming.* In DIVAI 2012.

Jegede, P. O. (2009). Predictors Of Java Programming Self Efficacy Among Engineering Students In A Nigerian University. *International Journal of Computer Science and Information Security, 4(1 & 2)*.

Jenkins, T. (2002). *On the difficulty of learning to program.* In Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences (Vol. 4, No. 2002, pp. 53-58).

Jeung, H. J., Chandler, P., & Sweller, J. (1997). The role of visual indicators in dual sensory mode instruction. *Educational Psychology, 17(3),* 329-345.

Kalyuga, S., Chandler, P., & Sweller, J. (1998). Levels of expertise and instructional design. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *40*(1), 1-17.

Kalyuga, S., Chandler, P., Tuovinen, J., & Sweller, J. (2001). When problem solving is superior to studying worked examples. *Journal of educational psychology*, *93*(3), 579.

Kalyuga, S. (2007). Expertise reversal effect and its implications for learner-tailored instruction. *Educational Psychology Review*, *19*(4), 509-539.

Keller, T., Gerjets, P., Scheiter, K., & Garsoffky, B. (2006). Information visualizations for knowledge acquisition: The impact of dimensionality and color coding. *Computers in Human Behavior*, *22*(1), 43-65.

Kert, S. B., & Kurt, A. A. (2012). The effect of electronic performance support systems on self-regulated learning skills. *Interactive Learning Environments*, *20*(6), 485-500.

Kissane, M., Kalyuga, S., Chandler, P., & Sweller, J. (2008). The consequences of fading instructional guidance on delayed performance: The case of financial services training. *Educational Psychology*, *28*(7), 809-822.

Kılıç, E., & Karadeniz, Ş. (2004). Hiper ortamlarda öğrencilerin bilişsel yüklenme ve kaybolma düzeylerinin belirlenmesi. *Kuram ve Uygulamada Egitim Yönetimi Dergisi*, *10*(4), 562-579.

Kotovsky, K., Hayes, J. R., & Simon, H. A. (1985). Why are some problems hard? Evidence from Tower of Hanoi. *Cognitive psychology*, *17*(2), 248-294.

Krippendorff, K. (1980). *Content analysis.* Beverly Hills. California: Sage Publications.

Lau, W. W. F., & Yuen, A. H. K. (2009). Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy. *British Journal of Educational Technology, 40(4),* 696-712.

Lau, W. W. F., & Yuen, A. H. K. (2011). Modeling programming performance: Beyond the influence of learner characteristics. *Computers and Education, 57(1),* 1202-1213.

Lee, N. (2013). *The Effects of Self-Explanation and Reading Questions and Answers on Learning Computer Programming Language (*Unpublished Doctoral Dissertation). University of Nevada, Las Vegas.

Lister, R. (2011). Computing Education Research: Programming, syntax and cognitive load. *ACM Inroads, 2(2),* 21-22.

Margulieux, L. E., Catrambone, R., & Guzdial, M. (2016). Employing subgoals in computer programming education. *Computer Science Education*, *26*(1), 44-67.

Margulieux, L. E., & Catrambone, R. (2016). Improving problem solving with subgoal labels in expository text and worked examples. *Learning and Instruction*, *42*, 58-71.

May, J., & Dhillon, G. (2009). Interpreting beyond syntactics: A semiotic learning model for computer programming languages. *Journal of Information Systems Education*, *20*(4), 431.

Mayer, R. E. (2001). What good is educational psychology? The case of cognition and instruction. *Educational Psychologist*, *36*(2), 83-88.

Mayer, R. E., & Moreno, R. (2003). Nine ways to reduce cognitive load in multimedia learning. *Educational psychologist*, *38*(1), 43-52.

Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning?. *American psychologist*, *59*(1), 14.

Mayer, R. E. (2013). *Teaching and learning computer programming: Multiple research perspectives.* Routledge.

Margulieux, L. E., Guzdial, M., & Catrambone, R. (2012). *Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications.* In Proceedings of the ninth annual international conference on International computing education research (pp. 71-78). ACM.

Margulieux, L. E., Morrison, B. B., Catrambone, R., & Guzdial, M. (2016). *Training Learners to Self-Explain: Designing Instructions and Examples to Improve Problem Solving.* In Transforming Learning, Empowering Learners: The International Conference of the Learning Sciences (ICLS) (Vol. 1, pp. 98-105).

Moreno, G. (2006). *Building a fuzzy transformation system.* In International Conference on Current Trends in Theory and Practice of Computer Science (pp. 409-418). Springer Berlin Heidelberg.

Morrison, B. B. (2013). *Using cognitive load theory to improve the efficiency of learning to program.* In Proceedings of the ninth annual international ACM conference on International computing education research. ACM.

Morrison, B. B., Margulieux, L. E., & Guzdial, M. (2015). *Subgoals, context, and worked examples in learning computing problem solving.* In Proceedings of the Eleventh Annual International Conference on International Computing Education Research. ACM.

Mousavi, S. Y., Low, R., & Sweller, J. (1995). Reducing cognitive load by mixing auditory and visual presentation modes. *Journal of educational psychology,87(2),* 319.

Mwangi, W., & Sweller, J. (1998). Learning to solve compare word problems: The effect of example format and generating self-explanations. *Cognition and instruction, 16(2),* 173-199.

Nathan, M. J., Mertz, K., & Ryan, R. (1994). *Learning through self-explanation of mathematics examples: Effects of cognitive load*. In Presentation to the American Educational Research Association (AERA) annual meeting.

Nievelstein, F., Van Gog, T., Van Dijck, G., & Boshuizen, H. P. (2013). The worked example and expertise reversal effect in less structured tasks: Learning to reason about legal cases. *Contemporary Educational Psychology*, *38*(2), 118-125.

Oliver, R. L. (1993). Cognitive, affective, and attribute bases of the satisfaction response. *Journal of consumer research*, *20*(3), 418-430.

Ozcelik, E., Karakus, T., Kursun, E., & Cagiltay, K. (2009). An eye-tracking study of how color coding affects multimedia learning. *Computers & Education*, *53*(2), 445-453.

Özmen, B., & Altun, A. (2014). Undergraduate Students' Experiences in Programming: Difficulties and Obstacles. *Turkish Online Journal of Qualitative Inquiry*, *5*(3), 1-27.

Paas, F. G., & Van Merriënboer, J. J. (1993). The efficiency of instructional conditions: An approach to combine mental effort and performance measures. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *35*(4), 737-743.

Paas, F., Tuovinen, J. E., Tabbers, H., & Van Gerven, P. W. (2003). Cognitive load measurement as a means to advance cognitive load theory. *Educational psychologist, 38(1)*, 63-71.

Paas, F., & van Gog, T. (2006). Optimising worked example instruction: Different ways to increase germane cognitive load. *Learning and Instruction, 16(2)*.

Paas, F., & Sweller, J. (2014). Implications of cognitive load theory for multimedia learning. The Cambridge handbook of multimedia learning, 27, 27-42.

Pirolli, P. L., & Anderson, J. R. (1985). The role of learning from examples in the acquisition of recursive programming skills. *Canadian Journal of Psychology/Revue canadienne de psychologie*, *39*(2), 240.

Pollock, E., Chandler, P., & Sweller, J. (2002). Assimilating complex information. *Learning and instruction*, *12*(1), 61-86.

Reisslein, J., Atkinson, R. K., Seeling, P., & Reisslein, M. (2006). Encountering the expertise reversal effect with a computer-based environment on electrical circuit analysis. *Learning and instruction*, *16*(2), 92-103.

Renkl, A. (1997). Learning from worked-out examples: A study on individual differences. *Cognitive science, 21(1),* 1-29.

Renkl, A., & Atkinson, R. K. (2003). Structuring the transition from example study to problem solving in cognitive skill acquisition: A cognitive load perspective. *Educational psychologist*, *38*(1), 15-22.

Renkl, A., Atkinson, R. K., & Große, C. S. (2004). How fading worked solution steps works–a cognitive load perspective. *Instructional Science*, *32*(1), 59-82.

Renkl, A. (2005). *The worked-out-example principle in multimedia learning.* The Cambridge handbook of multimedia learning, 229-245.

Renkl, A., & Atkinson, R. K. (2010). *Learning from worked-out examples and problem solving.* In Cognitive load theory. Cambridge University Press.

Renkl, A. (2014). Toward an instructionally oriented theory of example-based learning. *Cognitive science*, *38*(1), 1-37.

Renkl, A. (2017). Learning from worked-examples in mathematics: students relate procedures to principles. *ZDM*, 1-14.

Renumol, V. G., Janakiram, D., & Jayaprakash, S. (2010). Identification of cognitive processes of effective and ineffective students during computer programming. *ACM Transactions on Computing Education (TOCE)*, *10*(3), 10.

Richey, J. E., & Nokes-Malach, T. J. (2015). Comparing four instructional techniques for promoting robust knowledge. *Educational Psychology Review,27(1),* 181-218.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, *13*(2), 137-172.

Schaeffer, L. M. (2015). *Interaction of instructional material order and subgoal labels on learning in programming* (Unpublished Doctoral dissertation). Georgia Institute of Technology.

Schoenfeld, A. H., & Herrmann, D. J. (1982). Problem perception and knowledge structure in expert and novice mathematical problem solvers. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *8*(5), 484.

Schwonke, R., Renkl, A., Krieg, C., Wittwer, J., Aleven, V., & Salden, R. (2009). The worked-example effect: Not an artefact of lousy control conditions. *Computers in Human Behavior*, *25*(2), 258-266.

Schworm, S., & Renkl, A. (2006). Learning argumentation skills through the use of prompts for self-explaining examples. *Journal of Educational Psychology,99(2)*, 285.

Schunk, D. H., & Zimmerman, B. J. (2007). Influencing children's self-efficacy and self-regulation of reading and writing through modeling. *Reading & Writing Quarterly*, *23*(1), 7-25.

Shaw, R. S. (2012). A study of the relationships among learning styles, participation types, and performance in programming language learning supported by online forums. *Computers and Education 58 (2012*), 111–120.

Si, J., Kim, D., & Na, C. (2014). Adaptive Instruction to Learner Expertise with Bimodal Process-oriented Worked-out Examples. *Educational Technology & Society*, *17*(1), 259-27

Siegler, R. S., & Chen, Z. (2008). Differentiation and integration: Guiding principles for analyzing cognitive change. *Developmental Science*, *11*(4), 433-448.

Sivasakthi, M., & Rajendran, R. (2011). Learning difficulties of 'object-oriented programming paradigm using Java': students' perspective. *Indian Journal of Science and Technology, 8(4)*, 983-985.

Skinner, B. F. (2016). *The technology of teaching.* BF Skinner Foundation.

Sönmez, V., & Alacapınar, F. G. (2016). *Örneklendirilmiş bilimsel araştırma yöntemleri*. Anı Yayıncılık.

Stark, R., Kopp, V., & Fischer, M. R. (2011). Case-based learning with worked examples in complex domains: Two experimental studies in undergraduate medical education. *Learning and Instruction*, *21*(1), 22-33.

Sullivan, A., & Bers, M. U. (2012). Gender differences in kindergarteners' robotics and programming achievement. *International Journal of Technology and Design Education, 23(3)*, 691-702

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and instruction*, *2*(1), 59-89.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive science*, *12*(2), 257-285.

Sweller, J. (1989). Cognitive technology: Some procedures for facilitating learning and problem solving in mathematics and science. *Journal of educational psychology*, *81*(4), 457.

Sweller, J., & Chandler, P. (1991). Evidence for cognitive load theory. *Cognition and instruction*, *8*(4), 351-362.

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and instruction*, *4*(4), 295-312.

Sweller, J., Van Merrienboer, J. J., & Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational psychology review*, *10*(3), 251-296.

Sweller, J. (2010). Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational psychology review*, *22*(2), 123-138.

Tarmizi, R. A., & Sweller, J. (1988). Guidance during mathematical problem solving. *Journal of educational psychology, 80(4),* 424.

Tekin, H. (1996). *Egitimde olcme ve degerlendirme.* Educational Measurement and Evaluation.

Tindall-Ford, S., Chandler, P., & Sweller, J. (1997). When two sensory modes are better than one. *Journal of experimental psychology: Applied*, *3*(4), 257.

VanLehn, K., Jones, R. M., & Chi, M. T. (1992). A model of the self-explanation effect. *The journal of the learning sciences*, *2*(1), 1-59.

Van Gog, T., & Paas, F. (2008). Instructional efficiency: Revisiting the original construct in educational research. *Educational Psychologist*, *43*(1), 16-26.

Van Merriënboer, J. J. (1990). Strategies for programming instruction in high school: Program completion vs. program generation. *Journal of educational computing research*, *6*(3), 265-285.

Van Merriënboer, J. J., & Paas, F. G. (1990). Automation and schema acquisition in learning elementary computer programming: Implications for the design of practice. *Computers in Human Behavior*, *6*(3), 273-289.

Van Merrienboer, J. J., & Sweller, J. (2005). Cognitive load theory and complex learning: Recent developments and future directions. *Educational psychology review*, *17*(2), 147-177.

Van Merriënboer, J. J., & Sweller, J. (2010). Cognitive load theory in health professional education: design principles and strategies. *Medical education*, *44*(1), 85-93.

Vieira, C., Yan, J., & Magana, A. J. (2015). Exploring design characteristics of worked examples to support programming and algorithm design. *J Comput Sci Educ*, *6*, 2-15.

Ward, M., & Sweller, J. (1990). Structuring effective worked examples. *Cognition and instruction*, *7*(1), 1-39.

Webb, N. M. (1991). Task-related verbal interaction and mathematics learning in small groups. *Journal for research in mathematics education*, 366-389.

Yurdugül, H., & Aşkar, P. (2013). Learning programming, problem solving and gender: A longitudinal study. *Procedia - Social and Behavioral Sciences, 83,* 605-610.

Zhu, X., & Simon, H. A. (1987). Learning mathematics from examples and by doing. *Cognition and instruction*, *4*(3), 137-166.

EKLER DİZİNİ

EK 1. ETİK KOMİSYON ONAY BİLDİRİMİ

T.C.
HACETTEPE ÜNİVERSİTESİ
Rektörlük

Sayı : 35853172/ 433-3738

12 Aralık 2016

EĞİTİM BİLİMLERİ ENSTİTÜ MÜDÜRLÜĞÜNE

İlgi: 24.11.2016 tarih ve 2712 sayılı yazınız.

Enstitünüz Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı tezli yüksek lisans programı öğrencilerinden **Arş. Gör. Mustafa TEPGEÇ**'in **Yrd. Doç. Dr. Yasemin DEMİRASLAN ÇEVİK** danışmanlığında yürüttüğü **"Algoritma Öğretiminde Çözümlü Örnek Kullanımının Öğrenci Başarısına ve Bilişsel Yüke Etkisi"** başlıklı tez çalışması, Üniversitemiz Senatosu Etik Komisyonunun **02 Aralık 2016** tarihinde yapmış olduğu toplantıda incelenmiş olup, etik açıdan uygun bulunmuştur.

Bilgilerinizi ve gereğini rica ederim.

Prof. Dr. Rahime M. NOHUTCU
Rektör a.
Rektör Yardımcısı

12/12/2016

Hacettepe Üniversitesi Rektörlük 06100 Sıhhiye-Ankara
Telefon: 0 (312) 305 3001 - 3002 • Faks: 0 (312) 311 9992
E-posta: yazimd@hacettepe.edu.tr • www.hacettepe.edu.tr

Ayrıntılı Bilgi için:
Yazı İşleri Müdürlüğü
0 (312) 305 1008

EK 2. UYGULAMA İZNİ ONAY BİLDİRİMİ

Evrak Tarih ve Sayısı: 18/01/2017-E.793

T.C.
MUSTAFA KEMAL ÜNİVERSİTESİ REKTÖRLÜĞÜ
Öğrenci İşleri Daire Başkanlığı

İVEDİ

Sayı : 19125448-730.08.03/
Konu : Tez Çalışması (Mustafa TEPGEÇ)

EĞİTİM FAKÜLTESİ DEKANLIĞINA

İlgi : Hacettepe Üniversitesi Rektörlüğü 26/12/2016 tarihli, 240-3897 sayılı ve Mustafa TEPGEÇ'in Tez Çalışması konulu yazı

Hacettepe Üniversitesi Eğitim Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı tezli yüksek lisans öğrencisi Mustafa TEPGEÇ'in, Yrd. Doç. Dr. Yasemin DEMİRASLAN ÇEVİK danışmanlığında yürüttüğü "Algoritma Öğretiminde Çözümlü Örnek Kullanımının Öğrenci Başarısına ve Bilişsel Yüke Etkisi" başlıklı tez çalışmasını Üniversitemiz Eğitim Fakültesi öğrencilerine uygulaması Rektörlüğümüzce uygun görülmüştür.
Bilgi ve gereğini arz ve rica ederim.

Prof.Dr. Veysel EREN
Rektör a.
Rektör Yardımcısı

DAĞITIM
Eğitim Fakültesi Dekanlığına
Hacettepe Üniversitesi Rektörlüğüne

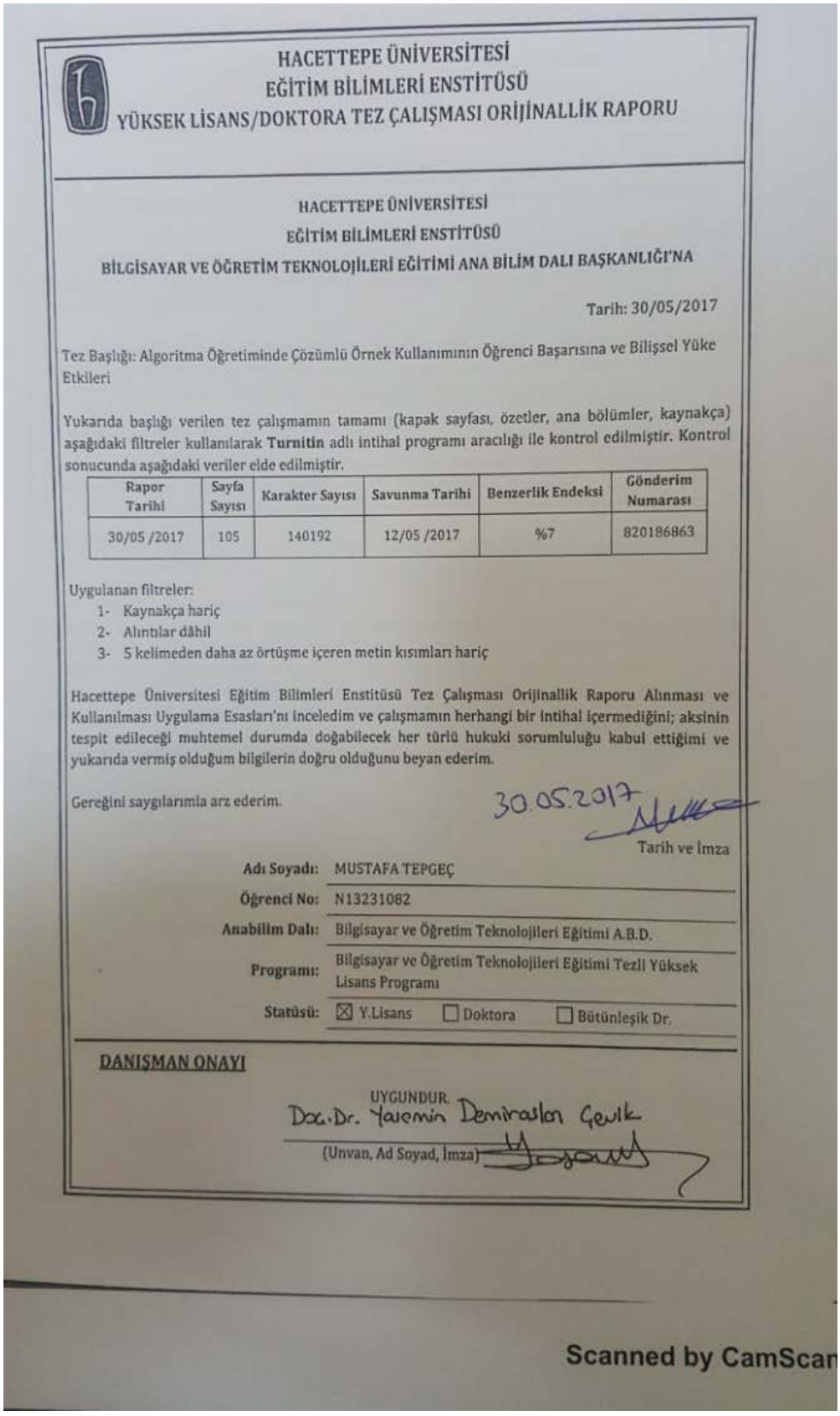
Mevcut Elektronik İmzalar
VEYSEL EREN (Rektör Yardımcılığı (Prof.Dr.Veysel EREN) - Rektör Yardımcısı) 18/01/2017 11:59

Evrakı Doğrulamak İçin : http://dogrula.mku.edu.tr/enVision-Sorgula/BelgeDogrulama.aspx?V=BEL94BP8N

Öğrenci İşleri Daire Başkanlığı
Tel: : 03262455915
E-Posta: : ogrenci@mku.edu.tr
Faks: 03262455916
Elektronik ağ:oidb.mku.edu.tr
Ayrıntılı bilgi için irtibat: Ayşegül Bilkay

Bu belge 5070 sayılı Elektronik İmza Kanununun 5. Maddesi gereğince güvenli elektronik imza ile imzalanmıştır.

EK 3. ORJİNALLİK RAPORU



HACETTEPE ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS/DOKTORA TEZ ÇALIŞMASI ORİJİNALLİK RAPORU

HACETTEPE ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANA BİLİM DALI BAŞKANLIĞI'NA

Tarih: 30/05/2017

Tez Başlığı: Algoritma Öğretiminde Çözümlü Örnek Kullanımının Öğrenci Başarısına ve Bilişsel Yüke Etkileri

Yukarıda başlığı verilen tez çalışmamın tamamı (kapak sayfası, özetler, ana bölümler, kaynakça) aşağıdaki filtreler kullanılarak **Turnitin** adlı intihal programı aracılığı ile kontrol edilmiştir. Kontrol sonucunda aşağıdaki veriler elde edilmiştir.

Rapor Tarihi	Sayfa Sayısı	Karakter Sayısı	Savunma Tarihi	Benzerlik Endeksi	Gönderim Numarası
30/05 /2017	105	140192	12/05 /2017	%7	820186863

Uygulanan filtreler:
1- Kaynakça hariç
2- Alıntılar dâhil
3- 5 kelimeden daha az örtüşme içeren metin kısımları hariç

Hacettepe Üniversitesi Eğitim Bilimleri Enstitüsü Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

30.05.2017

Tarih ve İmza

Adı Soyadı: MUSTAFA TEPGEÇ
Öğrenci No: N13231082
Anabilim Dalı: Bilgisayar ve Öğretim Teknolojileri Eğitimi A.B.D.
Programı: Bilgisayar ve Öğretim Teknolojileri Eğitimi Tezli Yüksek Lisans Programı
Statüsü: ☒ Y.Lisans ☐ Doktora ☐ Bütünleşik Dr.

DANIŞMAN ONAYI

UYGUNDUR.
Doç.Dr. Yasemin Demiraslan Çevik
(Unvan, Ad Soyad, İmza)

Scanned by CamScanner

HACETTEPE UNIVERSITY
GRADUATE SCHOOL OF EDUCATIONAL SCIENCES
THESIS/DISSERTATION ORIGINALITY REPORT

HACETTEPE UNIVERSITY
GRADUATE SCHOOL OF EDUCATIONAL SCIENCES
TO THE DEPARTMENT OF COMPUTER EDUCATION AND INSTRUCTIONAL TECHNOLOGIES

Date: 30/05/2017

Thesis Title : The Effects Of Using Worked Example Method In Teaching Algorithm On Students' Achievement And Cognitive Load

The whole thesis that includes the *title page, introduction, main chapters, conclusions and bibliography section* is checked by using **Turnitin** plagiarism detection software take into the consideration requested filtering options. According to the originality report obtained data are as below.

Time Submitted	Page Count	Character Count	Date of Thesis Defence	Similarity Index	Submission ID
30/05 /2017	105	140192	12/05 /2017	%7	820186863

Filtering options applied:
1. Bibliography excluded
2. Quotes excluded
3. Match size up to 5 words excluded

I declare that I have carefully read Hacettepe University Graduate School of Educational Sciences Guidelines for Obtaining and Using Thesis Originality Reports; that according to the maximum similarity index values specified in the Guidelines, my thesis does not include any form of plagiarism; that in any future detection of possible infringement of the regulations I accept all legal responsibility; and that all the information I have provided is correct to the best of my knowledge.

I respectfully submit this for approval. 30.05.2017

Date and Signature

Name Surname: MUSTAFA TEPGEÇ
Student No: N13231082
Department: Computer Education and Instructional Technologies
Program: Computer Education and Instructional Technologies Master Program
Status: ☒ Masters ☐ Ph.D. ☐ Integrated Ph.D.

ADVISOR APPROVAL

APPROVED.
Assoc. Prof. Dr. Yasemin Demiraslan Çevik

(Title, Name Surname, Signature)

EK 4. ALGORİTMA PERFORMANSINI ÖLÇMEK İÇİN KULLANILAN ÖN-SON TEST

Ad Soyad:
Öğrenci No:

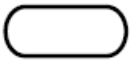
PROGRAMLAMA TEMELLERİ TESTİ

1-) Programlama sürecinde yapılması gereken işlem adımlarının sırası nasıl olmalıdır?

I-) Kodlama II-) Akış şeması oluşturma III-) Problem tanımlama
IV-) Programı sınama V-) Algoritma oluşturma

A-) I-II-III-IV-V
B-) III-V-I-II-IV
C-) III-I-II-V-IV
D-) III-IV-V-II-I
E-) III-V-II-I-IV

2-) Akış şemalarında aşağıdaki ifadelerden hangisi aritmetiksel ve mantıksal ifadeler için karar verme ve karşılaştırma işlemleri yapar?

A-) 
B-) 
C-) 
D-) 
E-) 

3-)
Adım 1: Başla
Adım 2: Yaz "İki sayı giriniz"
Adım 3: Oku sayi1, sayi2
Adım 4: Eğer sayi1 > sayi2
Yaz "büyüktür"
Değilse Yaz "küçüktür"
Adım 6: Bitir
(sayi1=10, sayi2=20)

Yukarıdaki algoritmanın ekran çıktısı nasıl olur?

A-) Ekrana bir şey yazmaz
B-) "büyüktür"
C-) "küçüktür"
D-) "Yaz küçüktür"
E-) Büyük olan sayıyı yazar.

4-) Aşağıdakilerden hangisi algoritmanın programlama diline çevrilmesidir?

A-) Hata ve böcek ayıklama
B-) Akış şeması oluşturma
C-) Kodlama
D-) Analiz
E-) Tasarım

5-) Akış şemalarında  sembolü ne anlama gelmektedir?

A-) Herhangi bir birimden veri girişi yapılacağını gösterir.
B-) Algoritmanın başladığı ya da sona erdiğini belirtmek için kullanılır.
C-) Ekran çıktısı oluşturmak için kullanılır.
D-) Şemanın akış yönünü gösterir.
E-) Hesaplama ya da değişken ataması yapmak için kullanılır.

6-)
```
Adım 1: Başla
Adım 2: Yaz "Yaşınız?"
Adım 3: Oku Yas
Adım 4: Eğer  Yas=>18 ise
            Yaz "Reşitsiniz"
            Değilse
            Yaz "Reşit Değilsiniz"
Adım 5: Bitir
```

Yukarıdaki ifade hangi problemin cevabı olabilir?

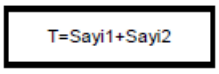
A-) Bir veritabanından kullanıcının yaşını çekip reşit olup olmadığını sorgulayan programın algoritması
B-) Kullanıcıya yaşı sorulup bilgisayarda kayıtlı bir metin dosyasına adını yazdıran programın algoritması
C-) Kullanıcının adına göre reşit olup olmadığını ekrana yazdıran programın algoritması
D-) Kullanıcıya yaşı sorulup reşit olup olmadığını ekrana yazdıran programın algoritması
E-) Kullanıcıya yaşı sorulup ekrana adını ve reşit olup olmadığını yazdıran programın algoritması

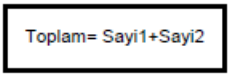
7-) "Bir algoritmanın yarı programlama dili kuralı, yarı konuşma diline dönük olarak ortaya konulması/ tanımlanmasıdır." ifadesi aşağıdaki kavramlardan hangisini açıklamaktadır?

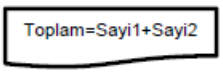
A-) Akış şeması
B-) Sözde kod
C-) Olay günlüğü
D-) Programlama
E-) Karar yapıları

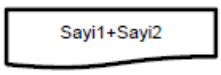
8-)
```
Adım 1: Başla
Adım 2: Yaz "Birinci sayı?"
Adım 3: Oku Sayi1
Adım 4: Yaz "İkinci sayı?"
Adım 5: Oku Sayi2
Adım 6: .....
Adım 7: Yaz Toplam
Adım 8: Bitir
```

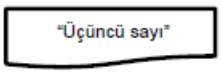
Yukarıdaki algoritmada kullanıcıdan alınan iki değerin toplamını ekrana yazdıran programın algoritması ifade edilmektedir. Buna göre boş bırakılan yerin akış şemasındaki gösterimi nasıl olmalıdır?

A-) 

B-) 

C-) 

D-) 

E-) 

9-) Aşağıda kullanıcıdan iki sayı alınıp büyük olan sayıdan küçük olan sayıyı çıkartıp ekrana **pozitif bir** sayı yazdıran programın hatalı veya eksik bir akış şeması bulunmaktadır. Şemadaki hatanın veya eksiğin nedeni aşağıdakilerden hangisidir?

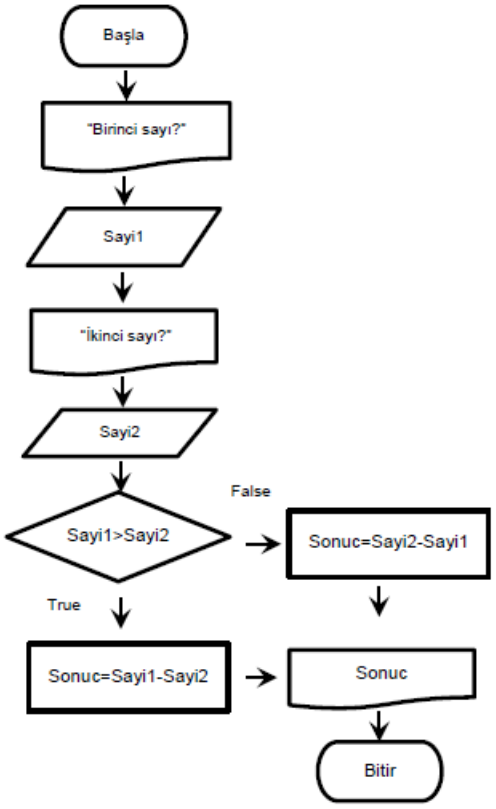
A-) Akış şemalarında başlangıç ve bitiş ifadeleri yer almaz.

B-) Akış her zaman dikey olmalıdır.

C-) Karar/koşul ifadesi daire sembolü ile gösterilir.

D-) Karar ifadelerinde her ihtimal düşünülmelidir.

E-) Değişkenler başla ifadesinden hemen sonra tanımlanır.

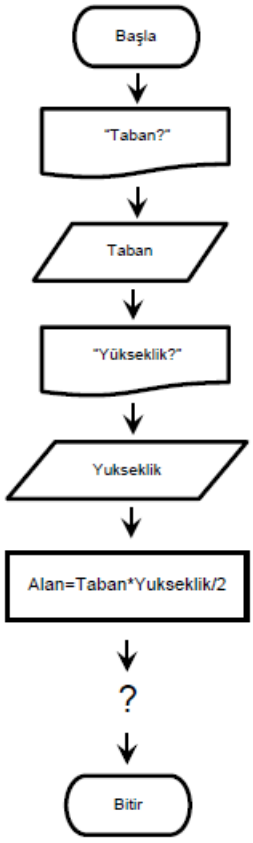


10-) Basit bir bankamatik algoritması yazılmak istendiğinde aşağıdaki boşluğa gelmesi gereken ifade seçeneklerden hangisinde yer almaktadır?

Adım 1: Başla
Adım 2: Yaz "lütfen şifrenizi giriniz"
Adım 3: Oku sifre
Adım 4: Eğer bilinen_sifre != sifre ise Git Adım 2
Adım 5: Yaz "hoş geldiniz"
Adım 6: Yaz "lütfen çekmek istediğiniz para miktarını giriniz"
Adım 7: Oku miktar
Adım 8: .....
Adım 9: Yaz "paranızı para haznesinden alınız-iyi günler"
Adım 10: Bitir

A) Eğer bankada_kalan_miktar<miktar ise Git 10 Değilse Git 9
B) Eğer bankada_kalan_miktar<miktar ise Git 9 Değilse Git 10
C) Eğer bankada_kalan_miktar=miktar ise Git 10 Değilse Git 9
D) Eğer miktar=5000 ise Git 10 Değilse Git 9
E) Eğer miktar>500 ise Git 10 Değilse Git 9

11-) Yandaki akış şemasında boş bırakılan alana aşağıdakilerden hangisi gelmelidir?



A-) Yukseklik

B-) Alan

C-) Alan

D-) Alan

E-) Yukseklik

12-)
```
Adım 1: Başla
Adım 2: Yaz "Maaşınızı girin"
Adım 3: Oku Maas
Adım 4: Yaz "Çocuk sayısını girin"
Adım 5: Oku Cocuksayisi
Adım 6: Eğer Cocuksayisi=>1 ise
            Maas = Maas + Maas*0,10
            Yaz "Alacağınız ücret = Maas"
            Değilse
            Yaz "Alacağınız ücret = Maas"
Adım 7: Bitir
```

Yukarıda bir çalışanın çocuk yardımı alıp almadığını ve maaş tutarını öğrenerek alacağı ücreti ekrana yazdıran programın hatalı veya eksik bir algoritması bulunmaktadır. Aşağıdakilerden hangisi hatanın nedenini ifade etmektedir?

A-) Ekrana yazdırılan mesajın tırnak içinde yazılmış olması
B-) Başla ve bitir ifadelerinin kullanılmış olması
C-) Değişkenin tırnak içinde yazılmış olması
D-) Döngü kullanılmamış olması
E-) Değişken isimlendirmesinin kurallarına uygun yapılmaması

13-) Adım 1: Başla
Adım 2: Toplam=0
Adım 3: Yaz "Bir sayı girin"
Adım 4: Oku Sayi
Adım 5: Toplam= Toplam + Sayi
Adım 6: Girilen_sayi= Girilen_sayi + 1
Adım 6: Eğer Girilen_sayi<5 ise Git Adım 3
Adım 7: Yaz Toplam
Adım 8: Bitir

Yukarıda kullanıcıdan alınan 5 sayının toplamının ekrana yazdırıldığı programın hatalı veya eksik bir algoritması bulunmaktadır. Aşağıdakilerden hangisi hatanın nedenini ifade etmektedir?

A-) Toplam değişkeninin ilk başta "0" olarak tanımlanması
B-) Toplamın ekrana koşul ifadesinden sonra yazdırılması
C-) Girilen_sayi değişkeninin tanımlanmamış olması
D-) Ekrana yazdırılan değerin tırnak içinde yazılmamış olması
E-) Koşul ifadesi sağlandığında gidilen adımın yanlış olması

14-) Adım 1: Başla
Adım 2: A=1
Adım 3: Yaz "Bir sayı girin"
Adım 4: Oku B
Adım 5: A=A*B
Adım 6: B= B-1
Adım 7: Eğer B>1 ise Git Adım 5
Adım 8: Yaz A
Adım 9: Bitir

Aşağıdaki seçeneklerden hangisi yukarıdaki algoritmayı tanımlamaktadır.

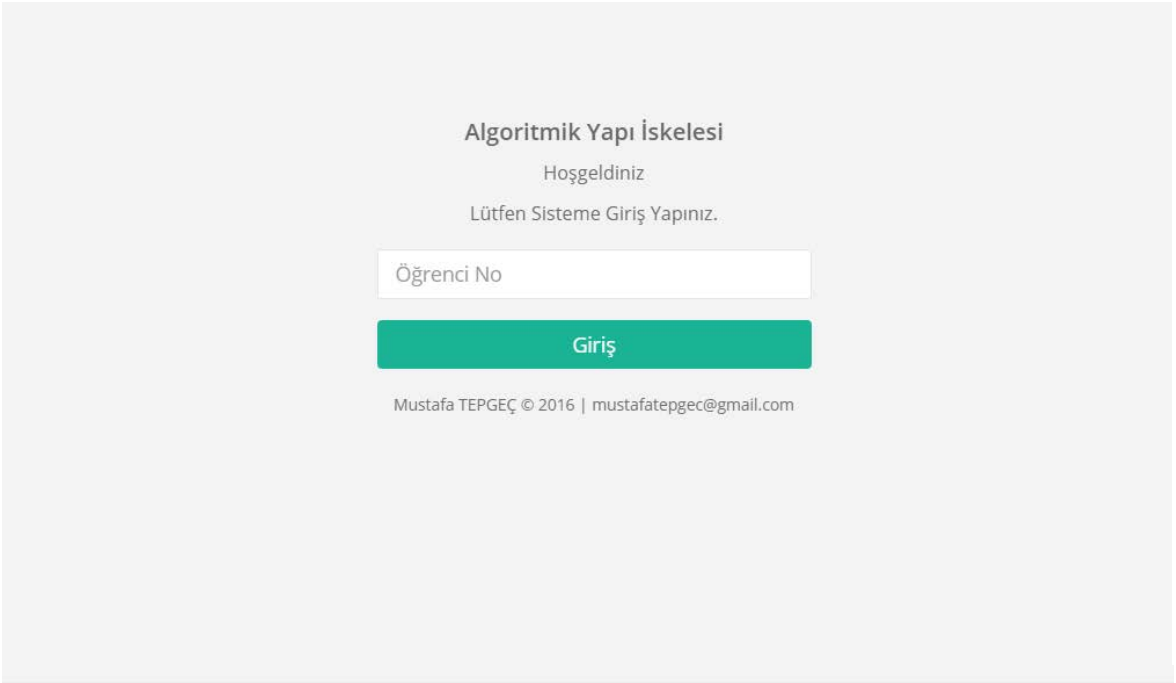
A-) Kullanıcının ekrana girdiği sayının logaritmasını ekrana yazdıran programın algoritması
B-) Girilen sayı ile 1 arasındaki sayıları toplayıp ekrana yazdıran programın algoritması
C-) Girilen sayı ile 1 arasındaki sayıların çarpımını (faktöriyel) ekrana yazdırır.
D-) Bir firmanın elinde kalan ürünlere göre toplam satış fiyatını yazdıran programın algoritması
E-) Girilen sayıdan başlayarak 1'e kadar olan sayıları ekrana yazdırır.

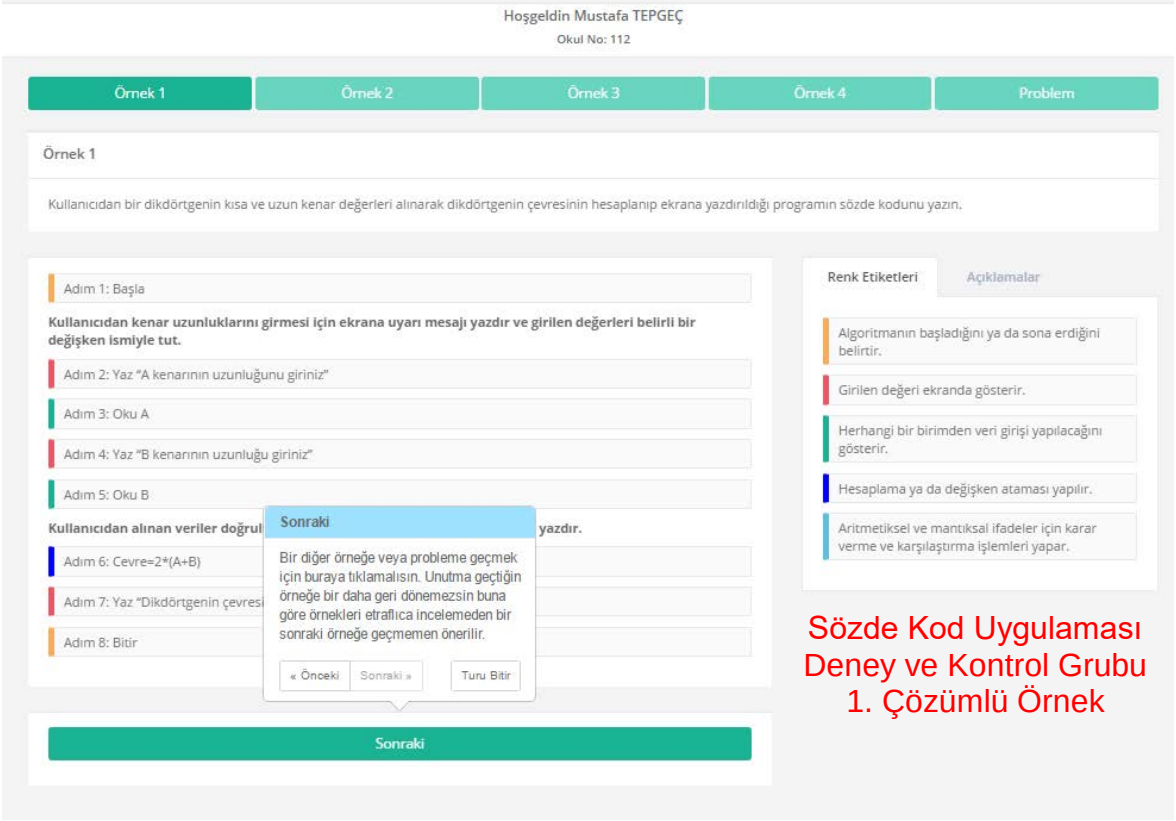
15-) I. Sadece bilgisayar konularında algoritmalar yazılabilir.
II. Algoritmalar matematiksel bir işlem adımı olmayan problemler için de yazılabilir.
III. Algoritma içerisinde değişken tanımlanmamaktadır.
IV. Algoritmalar aksi belirtilmediği takdirde yukarıdan aşağıya doğru okunur.

Yukarıdaki ifadelerden hangileri doğrudur?

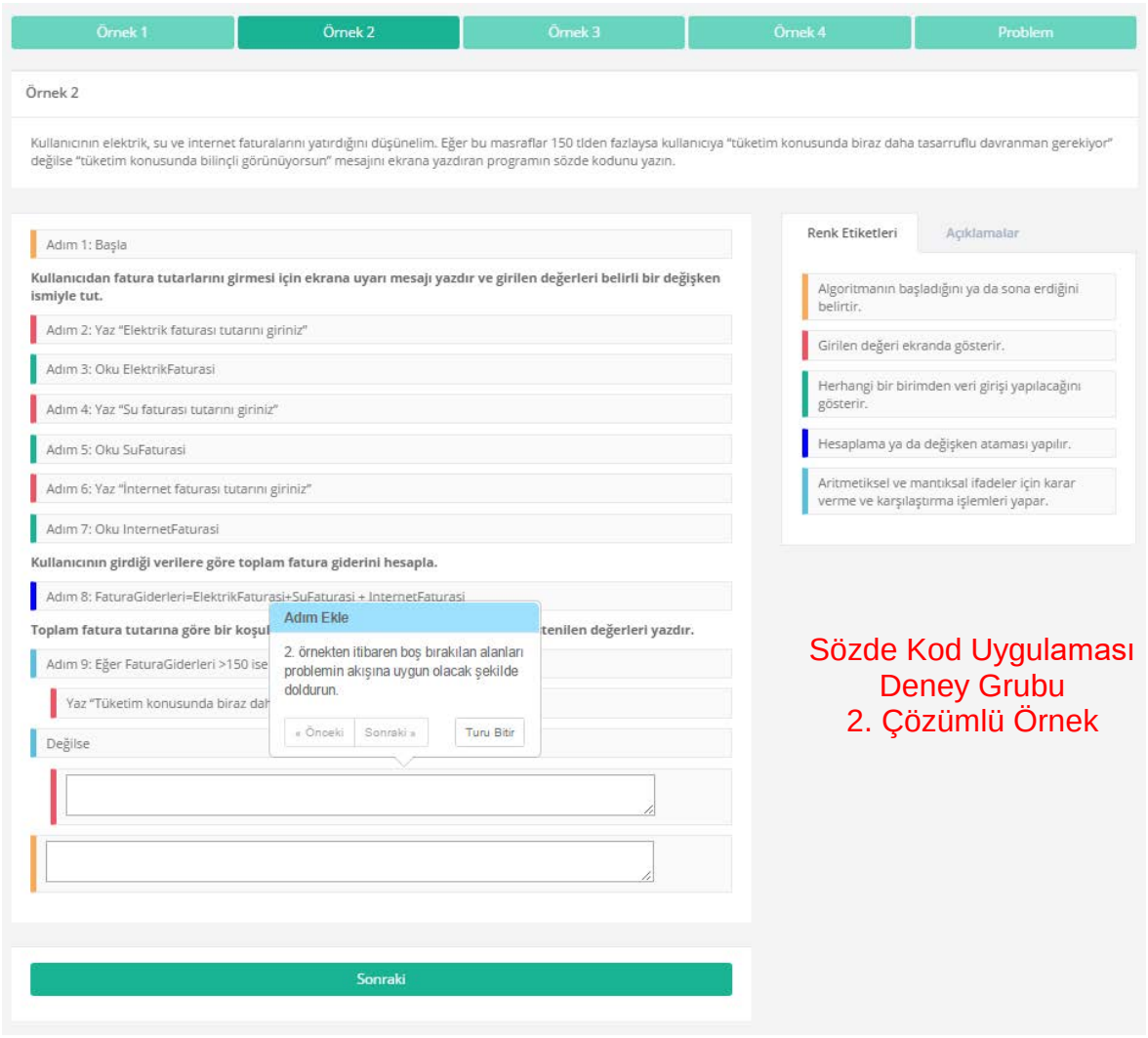
A-) I ve II B-) II ve III C-) I ve IV D-) II ve IV E-) II, III ve IV

EK 5. GELİŞTİRİLEN ÇEVRİMİÇİ ORTAMA İLİŞKİN EKRAN GÖRÜNTÜLERİ

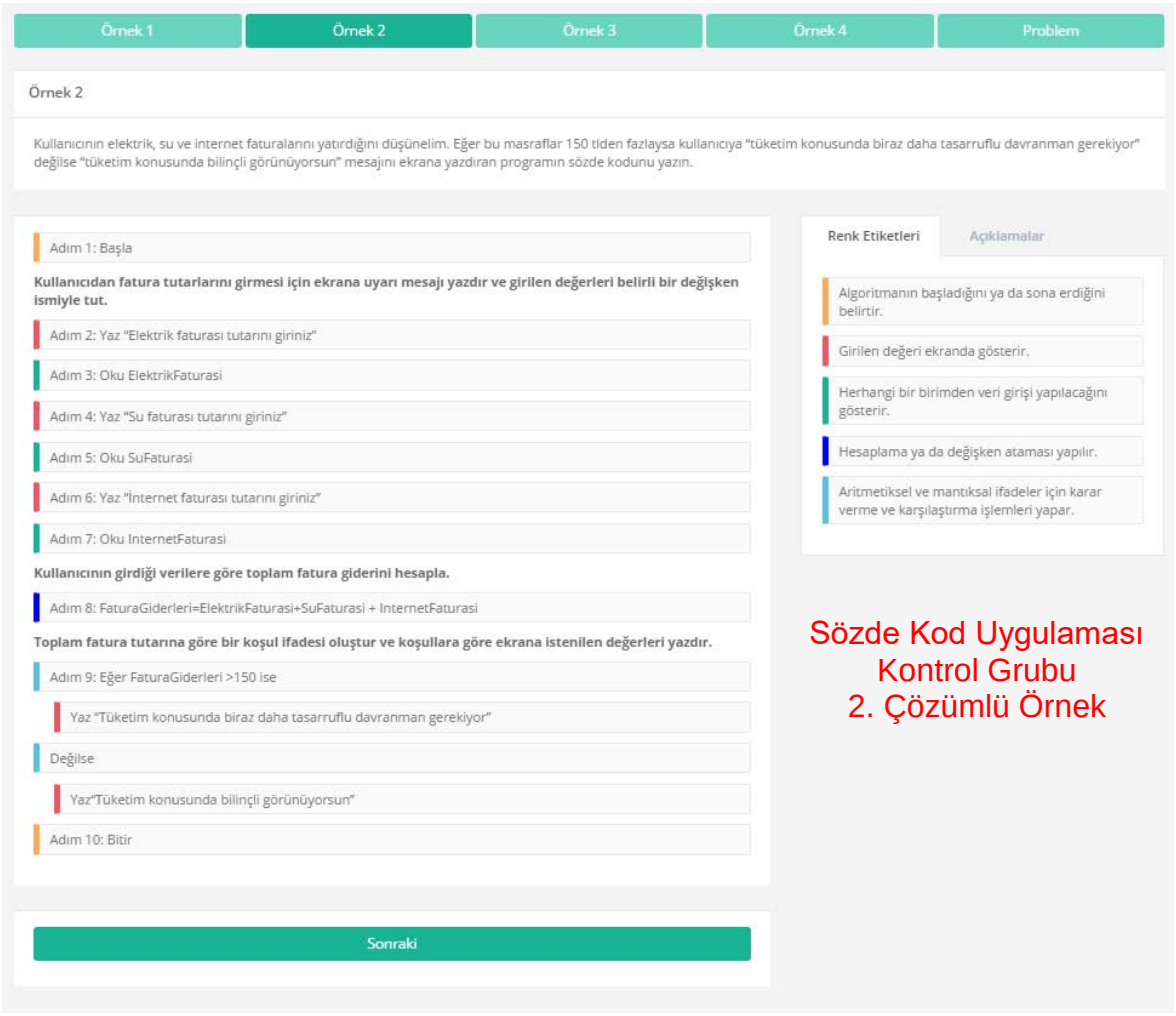




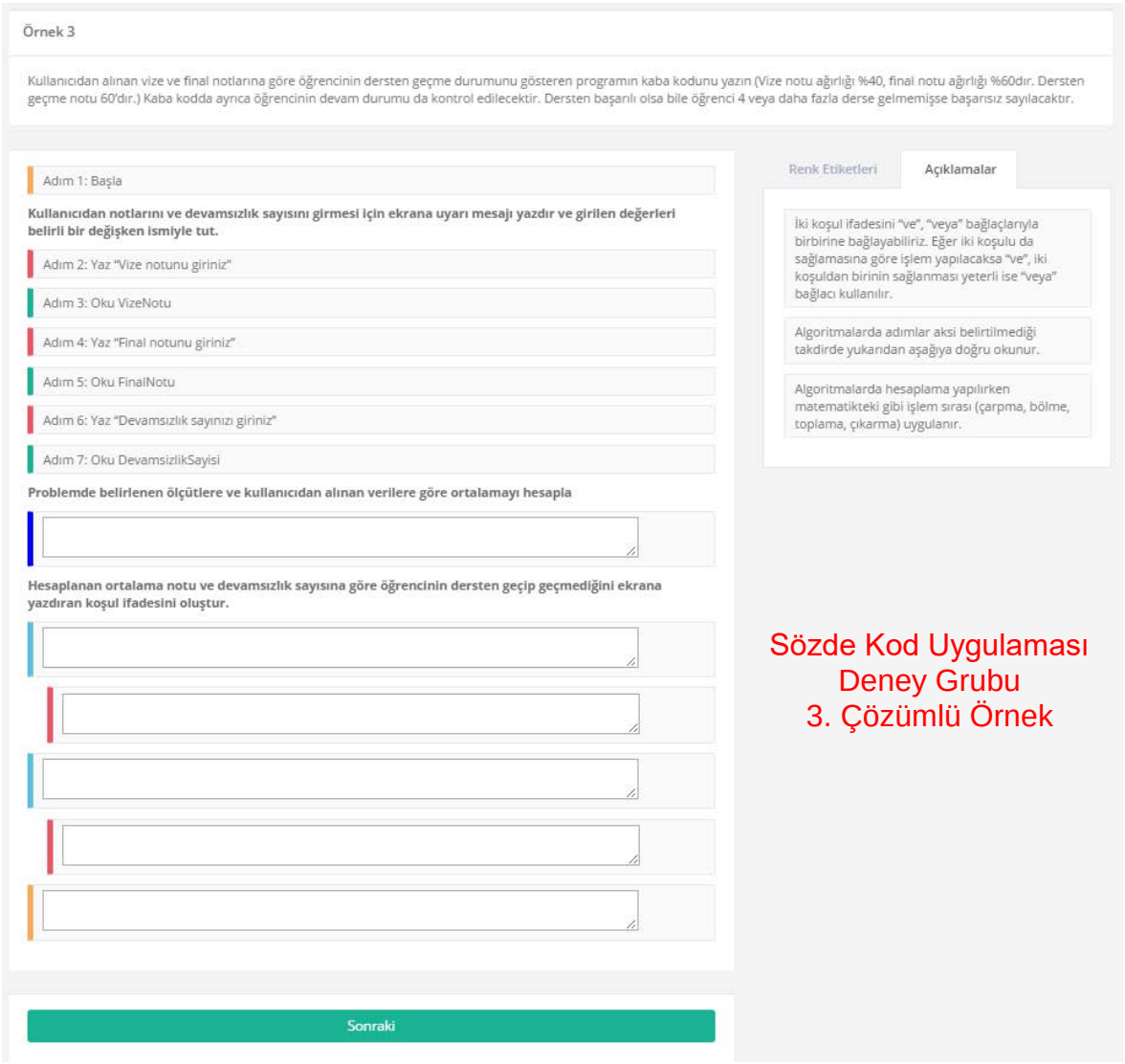
Sözde Kod Uygulaması
Deney ve Kontrol Grubu
1. Çözümlü Örnek



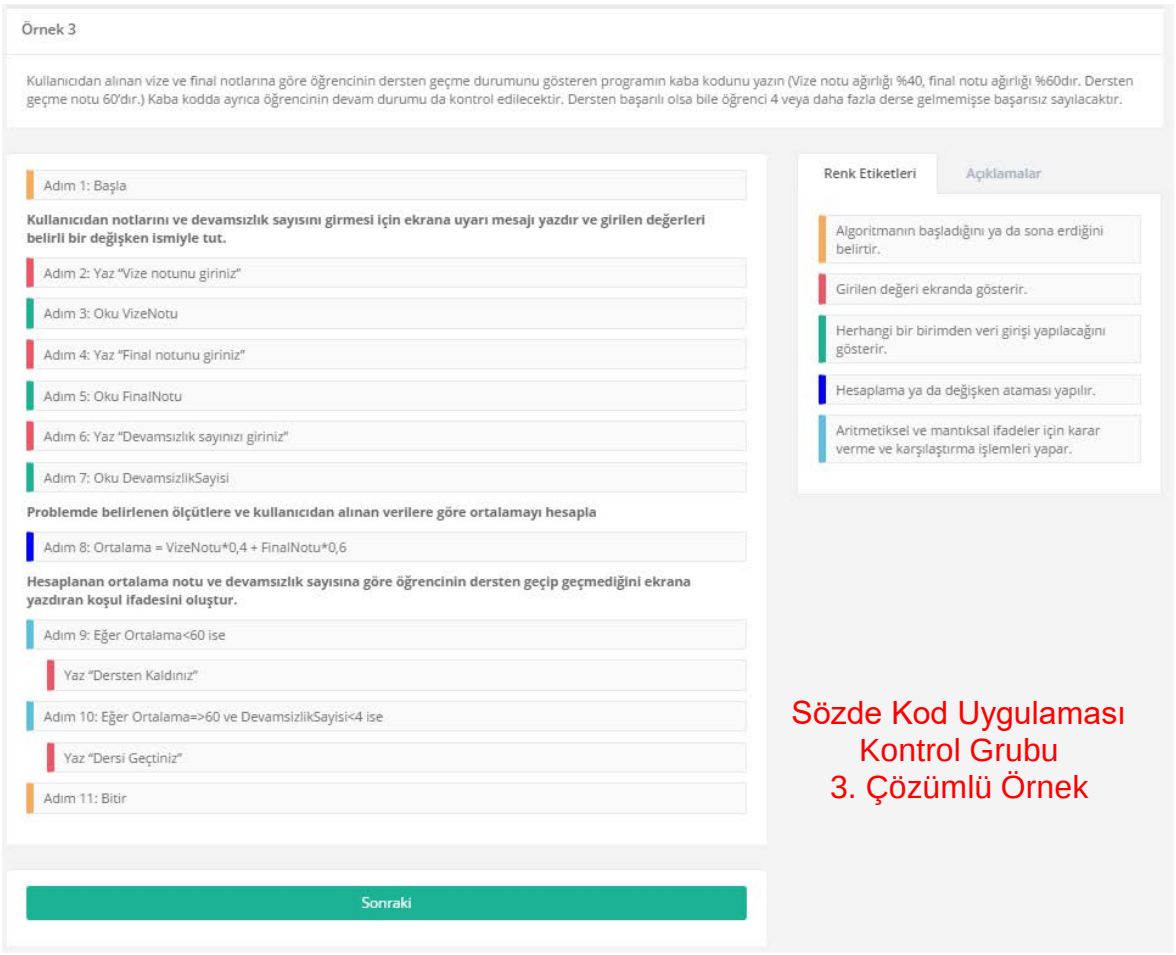
Sözde Kod Uygulaması
Deney Grubu
2. Çözümlü Örnek



Sözde Kod Uygulaması
Kontrol Grubu
2. Çözümlü Örnek



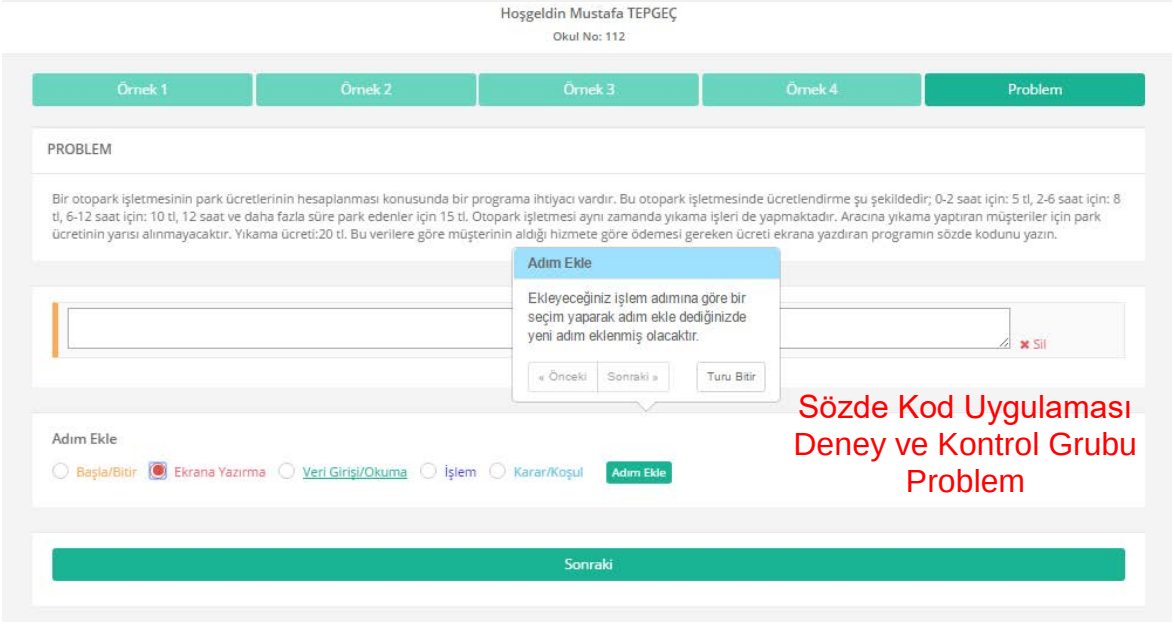
Sözde Kod Uygulaması
Deney Grubu
3. Çözümlü Örnek



Sözde Kod Uygulaması
Kontrol Grubu
3. Çözümlü Örnek

Sözde Kod Uygulaması
Deney Grubu
4. Çözümlü Örnek

Sözde Kod Uygulaması
Kontrol Grubu
4. Çözümlü Örnek



Sözde Kod Uygulaması
Deney ve Kontrol Grubu
Problem

Hoşgeldin Mustafa TEPGEÇ

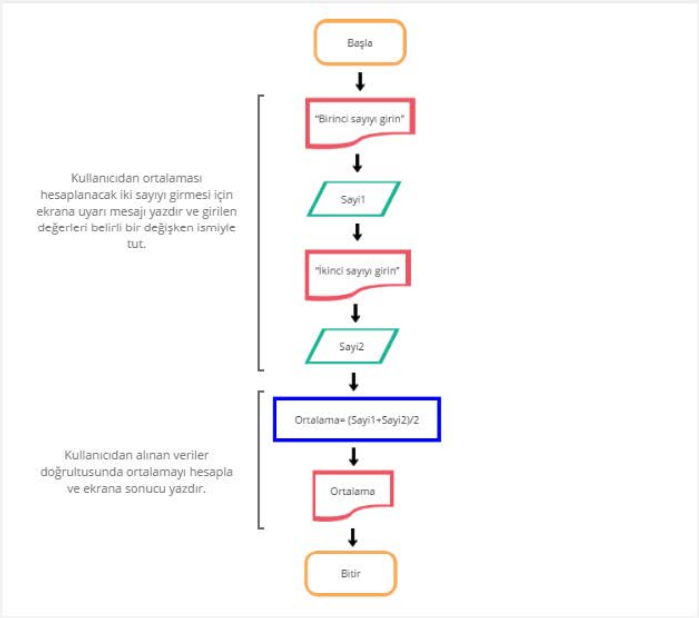
Okul No: 112

Örnek 1 | Örnek 2 | Örnek 3 | Örnek 4 | Problem

Akış Şemaları Uygulaması
Deney ve Kontrol Grubu
1. Çözümlü Örnek

Örnek 1

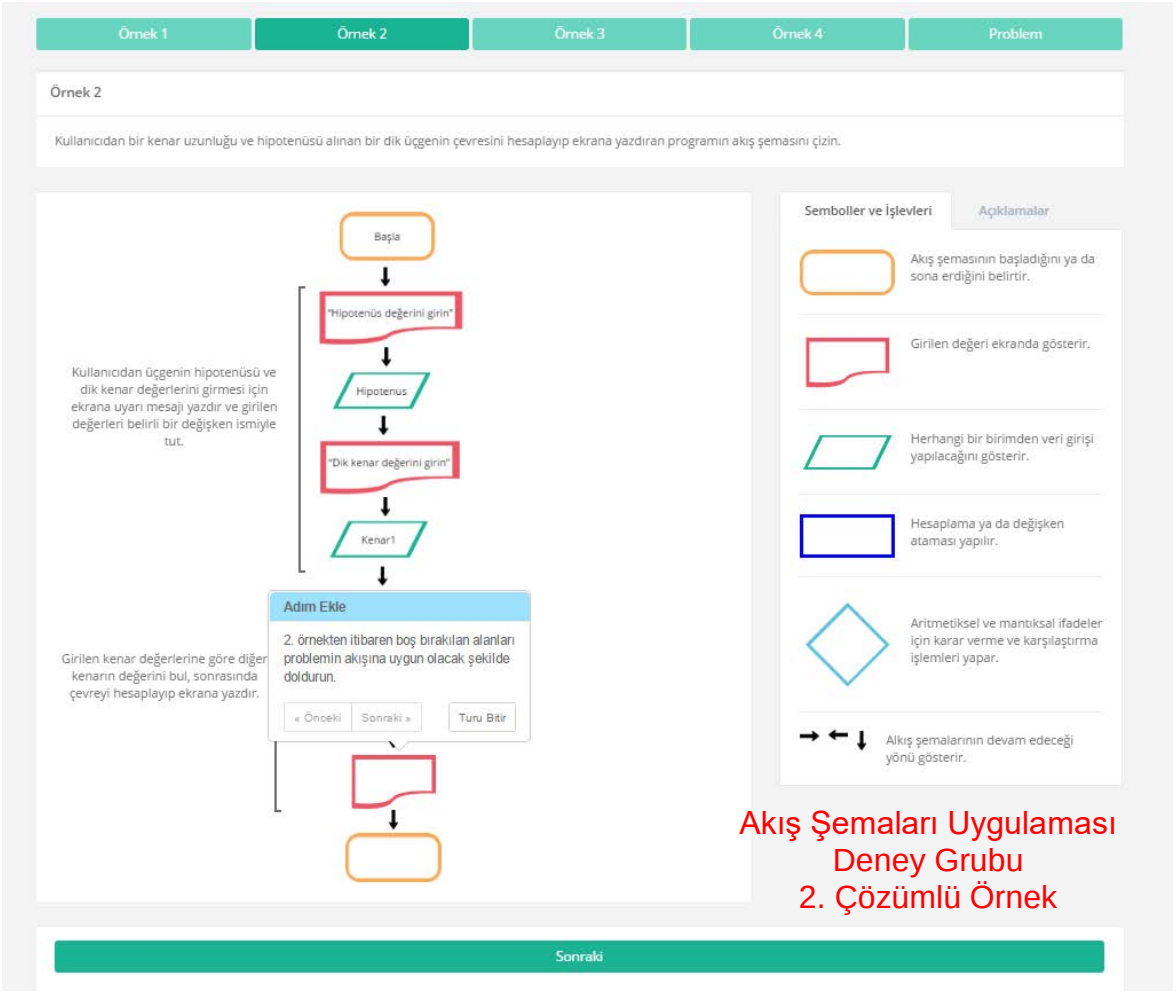
Kullanıcıdan alınan iki sayının ortalamasını bulup ekrana yazdıran programın akış şemasını çizin.



Semboller ve İşlevleri | Açıklamalar

- Akış şemasının başladığını ya da sona erdiğini belirtir.
- Girilen değeri ekranda gösterir.
- Herhangi bir birimden veri girişi yapılacağını gösterir.
- Hesaplama ya da değişken ataması yapılır.
- Aritmetiksel ve mantıksal ifadeler için karar verme ve karşılaştırma işlemleri yapar.
- Akış şemalarının devam edeceği yönü gösterir.

Sonraki

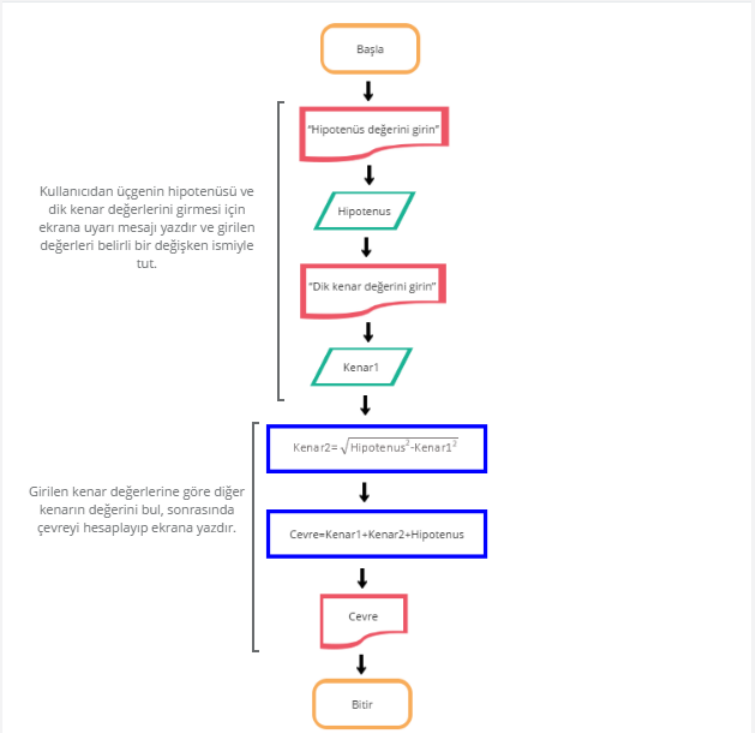


Akış Şemaları Uygulaması
Deney Grubu
2. Çözümlü Örnek

Örnek 1 | Örnek 2 | Örnek 3 | Örnek 4 | Problem

Örnek 2

Kullanıcıdan bir kenar uzunluğu ve hipotenüsü alınan bir dik üçgenin çevresini hesaplayıp ekrana yazdıran programın akış şemasını çizin.



Semboller ve İşlevleri | Açıklamalar

- Akış şemasının başladığını ya da sona erdiğini belirtir.
- Girilen değeri ekranda gösterir.
- Herhangi bir birimden veri girişi yapılacağını gösterir.
- Hesaplama ya da değişken ataması yapılır.
- Aritmetiksel ve mantıksal ifadeler için karar verme ve karşılaştırma işlemleri yapar.
- Akış şemalarının devam edeceği yönü gösterir.

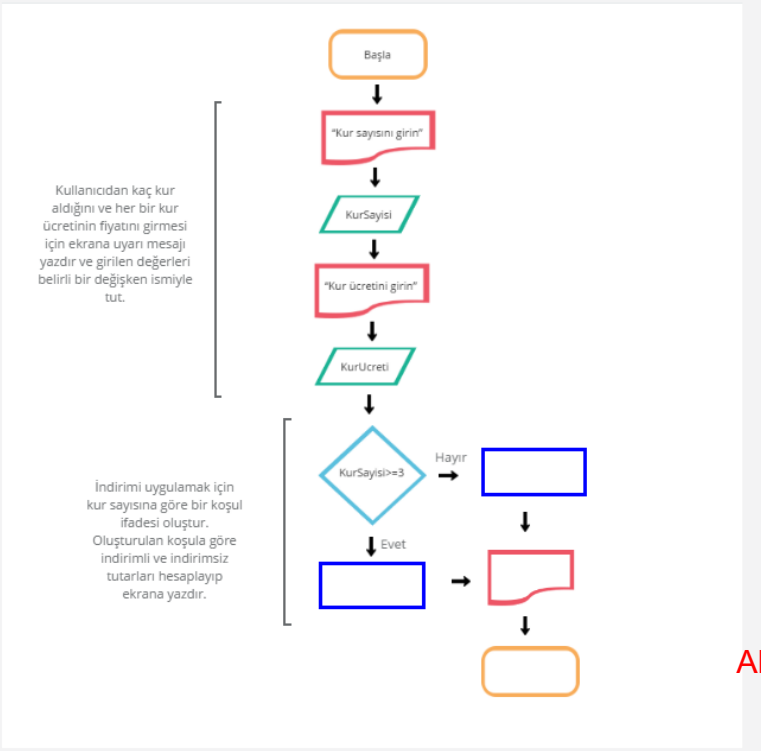
Akış Şemaları Uygulaması
Kontrol Grubu
2. Çözümlü Örnek

Sonraki

Örnek 1 | Örnek 2 | Örnek 3 | Örnek 4 | Problem

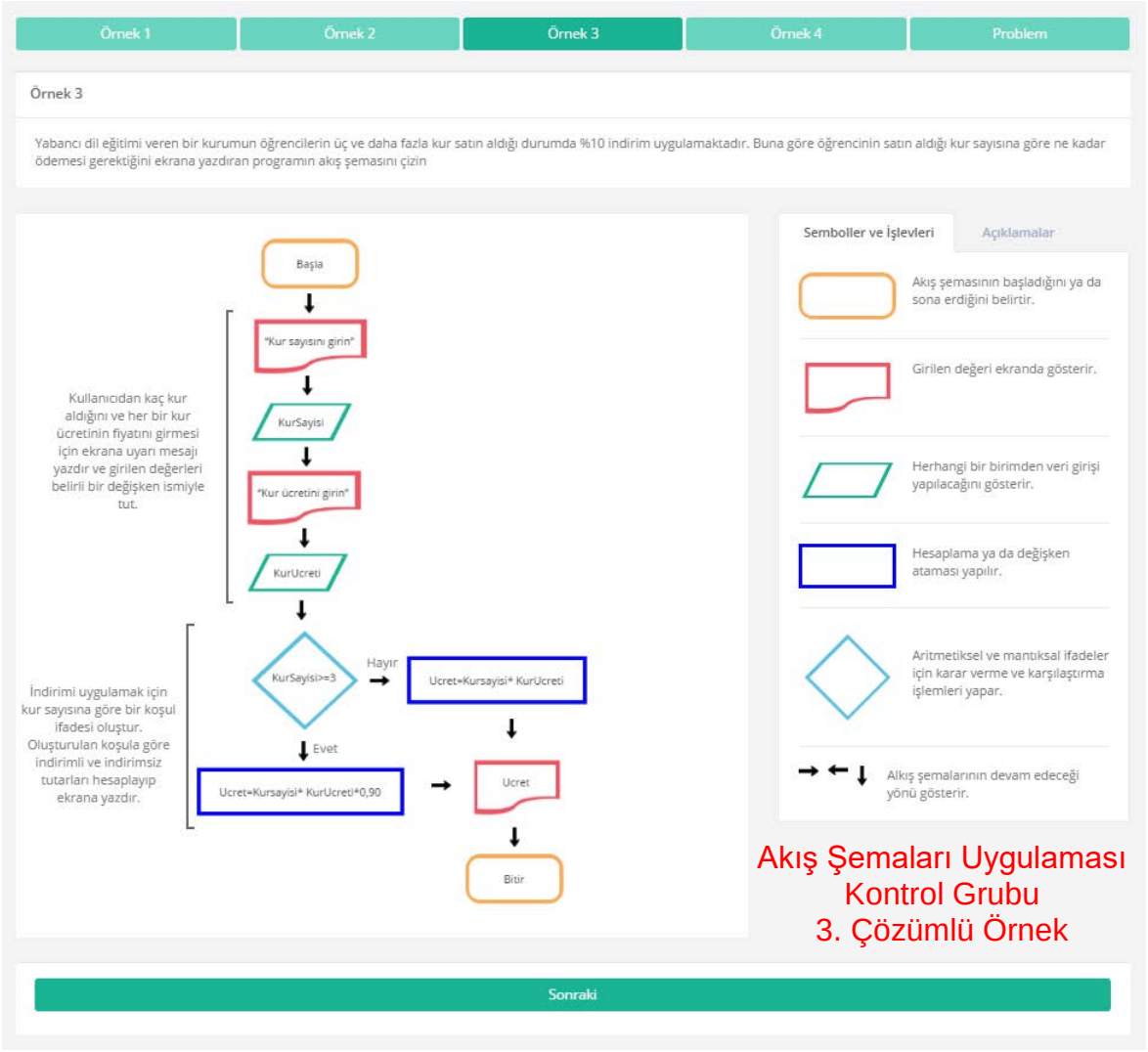
Örnek 3

Yabancı dil eğitimi veren bir kurumun öğrencilerin üç ve daha fazla kur satın aldığı durumda %10 indirim uygulamaktadır. Buna göre öğrencinin satın aldığı kur sayısına göre ne kadar ödemesi gerektiğini ekrana yazdıran programın akış şemasını çizin

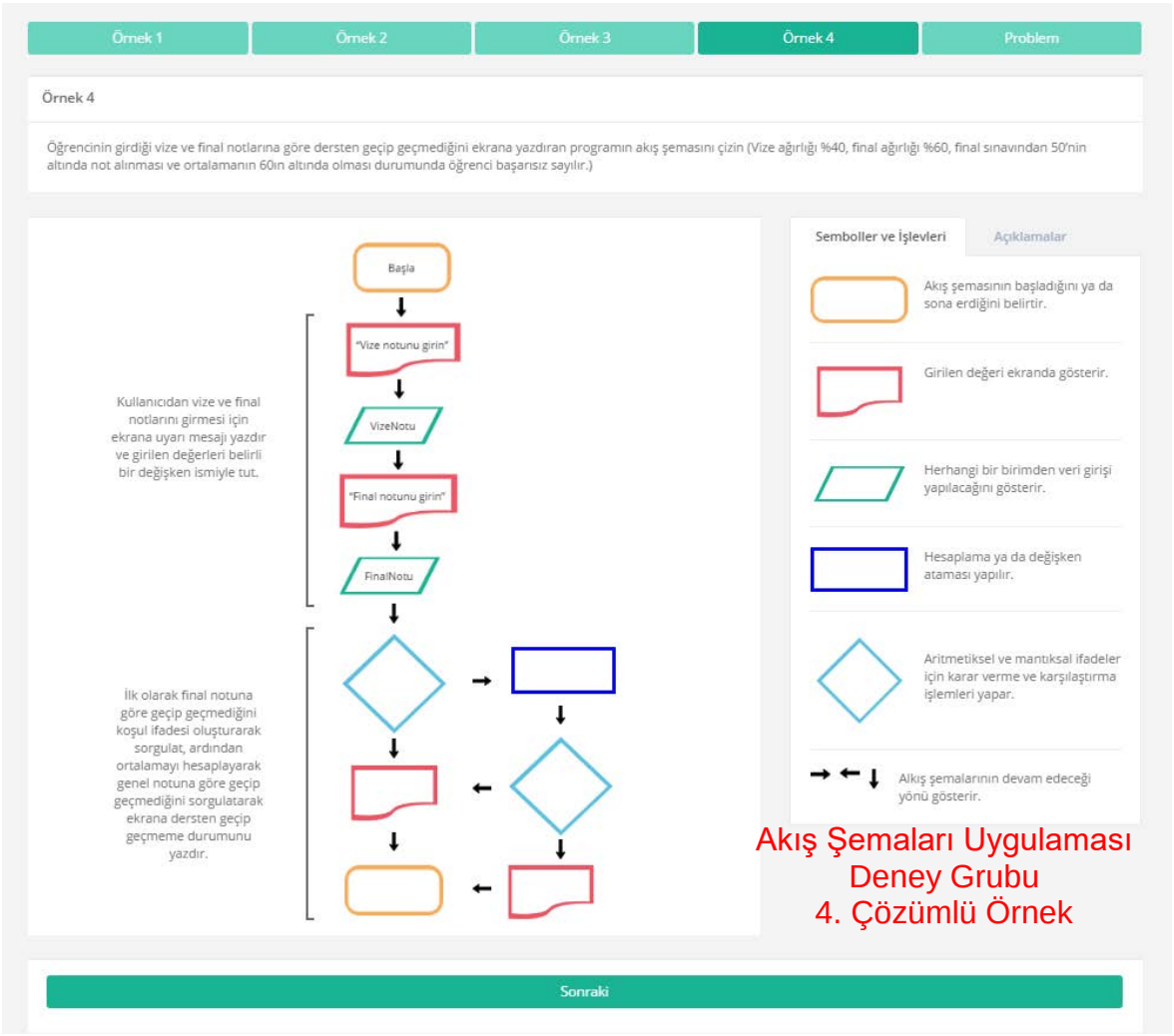


Akış Şemaları Uygulaması
Deney Grubu
3. Çözümlü Örnek

Sonraki



Akış Şemaları Uygulaması
Kontrol Grubu
3. Çözümlü Örnek

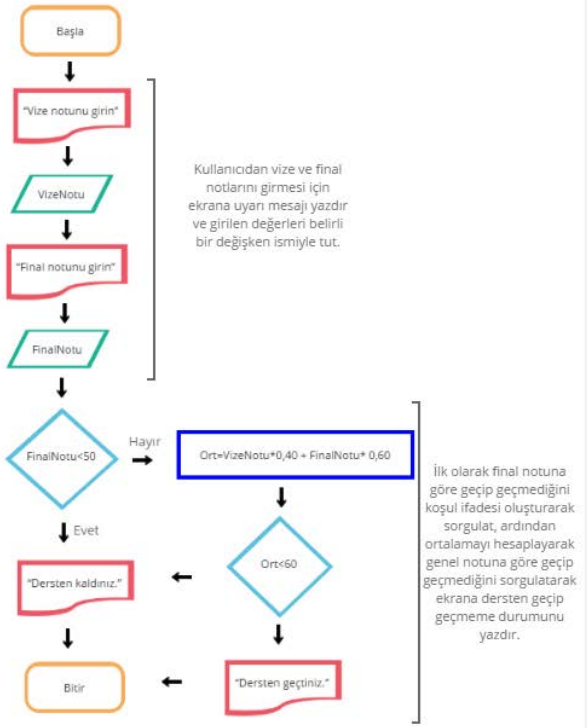


Akış Şemaları Uygulaması
Deney Grubu
4. Çözümlü Örnek

Örnek 1 | Örnek 2 | Örnek 3 | Örnek 4 | Problem

Örnek 4

Öğrencinin girdiği vize ve final notlarına göre dersten geçip geçmediğini ekrana yazdıran programın akış şemasını çizin (Vize ağırlığı %40, final ağırlığı %60, final sınavından 50'nin altında not alınması ve ortalamanın 60ın altında olması durumunda öğrenci başarısız sayılır.)



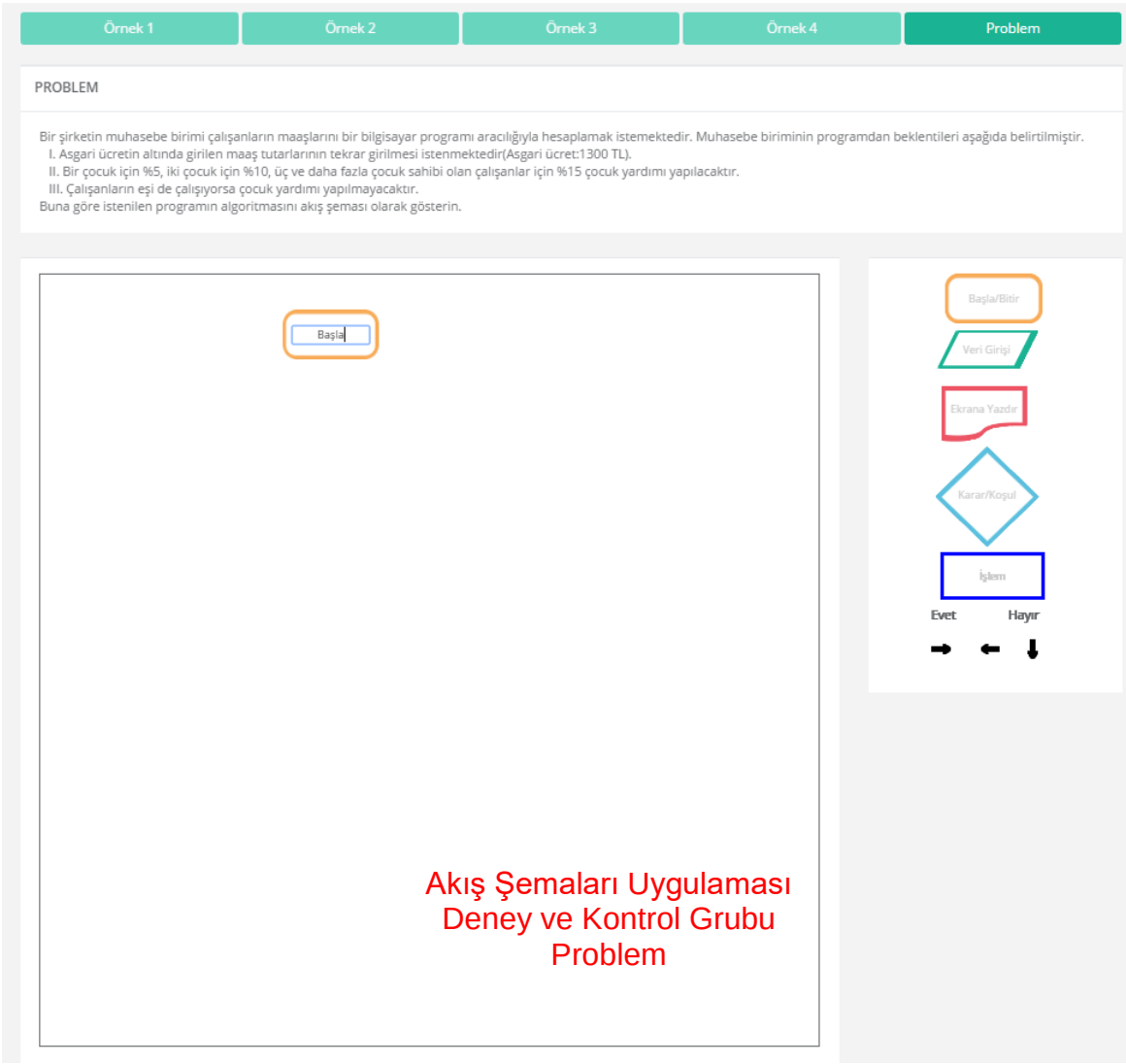
Semboller ve İşlevleri | Açıklamalar

Akış şemalarında hesaplama yapılırken matematikteki gibi işlem sırası (çarpma, bölme, toplama, çıkarma) uygulanır.

Algoritmalarda "=" ifadesi eşitliği değil atamayı ifade eder. Yani sayi=sayi+1 ifadesi görüldüğünde 0=1 gibi bir sonuca varılabilir. Ancak algoritma ve programlamada bu ifade "sayi değişkeninin değerini 1 artır" anlamına gelmektedir.

Sonraki

Akış Şemaları Uygulaması
Kontrol Grubu
4. Çözümlü Örnek


Örnek 1
Örnek 2
Örnek 3
Örnek 4
Problem
PROBLEM
Bir şirketin muhasebe birimi çalışanların maaşlarını bir bilgisayar programı aracılığıyla hesaplamak istemektedir. Muhasebe biriminin programdan beklentileri aşağıda belirtilmiştir.
I. Asgari ücretin altında girilen maaş tutarlarının tekrar girilmesi istenmektedir(Asgari ücret:1300 TL).
II. Bir çocuk için %5, iki çocuk için %10, üç ve daha fazla çocuk sahibi olan çalışanlar için %15 çocuk yardımı yapılacaktır.
III. Çalışanların eşi de çalışıyorsa çocuk yardımı yapılmayacaktır.
Buna göre istenilen programın algoritmasını akış şeması olarak gösterin.
Başla
Başla/Bitir
Veri Girişi
Ekrana Yazdır
Karar/Koşul
İşlem
Evet
Hayır
Akış Şemaları Uygulaması
Deney ve Kontrol Grubu
Problem


Hoşgeldin Mustafa TEPGEÇ
Okul No: 112
Ölçek Formu
Verilen görevi tamamlarken ne kadar çaba sarf ettiniz?
(1) Çok çok az
(2) Çok az
(3) Az
(4) Kısmen az
(5) Ne az ne fazla
(6) Kısmen fazla
(7) Fazla
(8) Çok fazla
(9) Çok çok fazla
Kaydet
Her iki uygulamanın
sonunda sunulan bilişsel
yük derecelendirme ölçeği

EK 6. DENEY VE KONTROL GRUBUNUN SÜREÇ PERFORMANSLARINI DEĞERLENDİRME RUBRİĞİ

Problem	Ölçütler	0	1	2
Sözde Kod Problemi	Sözde kodlarda başla ve bitir ifadelerini doğru yerde ve yazım kurallarına uygun şekilde kullanmıştır.			
	Sözde kodlarda veri girişi/okuma ifadelerini doğru yerde ve yazım kurallarına uygun şekilde kullanmıştır.			
	Sözde kodlarda ekrana yazdırma ifadelerini doğru yerde ve yazım kurallarına uygun şekilde kullanmıştır.			
	Sözde kodlarda işlem ifadelerini doğru yerde ve yazım kurallarına uygun şekilde kullanmıştır.			
	Sözde kodlarda karar yapılarını doğru yerde ve yazım kurallarına uygun şekilde kullanmıştır.			
	Problemi eksiksiz ve sonuca ulaştıracak şekilde çözümlemiştir.			
	Algoritmayı sonuca en kısa yoldan ulaştıracak şekilde tasarlamıştır.			
Akış Şeması Problemi	Akış şemalarında başla ve bitir ifadelerini doğru yerde, uygun sembollerle ve yazım kurallarına uygun şekilde kullanmıştır.			
	Akış şemalarında veri girişi/okuma ifadelerini doğru yerde, uygun sembollerle ve yazım kurallarına uygun şekilde kullanmıştır.			
	Akış şemalarında ekrana yazdırma ifadelerini doğru yerde, uygun sembollerle ve yazım kurallarına uygun şekilde kullanmıştır.			
	Akış şemalarında işlem ifadelerini doğru yerde, uygun sembollerle ve yazım kurallarına uygun şekilde kullanmıştır.			
	Akış şemalarında karar yapılarını doğru yerde, uygun sembollerle ve yazım kurallarına uygun şekilde kullanmıştır.			
	Problemi eksiksiz ve sonuca ulaştıracak şekilde çözümlemiştir.			
	Algoritmayı sonuca en kısa yoldan ulaştıracak şekilde tasarlamıştır.			

EK 7. PROGRAMLAMA DİLLERİ I DERSİ FİNAL SINAVI SORULARI

Uygulama Sorusu

1. Program çalıştığında aşağıdaki gibi bir menü ile karşılaşılır. Kullanıcı menüden istediği seçeneği seçip işlemini gerçekleştirebilmelidir. Program ancak kullanıcı çıkışı seçtiğinde sonlanmalıdır. Aksi takdirde bir işlemi yaptıktan sonra tekrar menüyü görmeli ve işlemlerine devam edebilmelidir. (20 Puan)

```
----Menü-------
1. Girilen sayının kaç basamaklı olduğunu bulma      (20 puan)
2. Basamak değerlerini toplama                       (20 Puan)
3. Rasgele 10 sayının çift olanlarının ortalamasını hesaplama(20 Puan)
4. Sayının mükemmel olup olmadığını bulma     (20 Puan)
5. Çıkış
İşleminizi seçiniz1-5:
```

Seçenekler çalışması hakkında aşağıdaki örnekleri inceleyiniz.

1 numaralı seçenek seçilmişse;
Kullanıcıdan en fazla 4 basamaklı pozitif tam sayı (0-9999) girmesi istenir (sayı büyüklüğü kontrolüne gerek yoktur) Program kullanıcının girdiği sayının kaç basamaklı bir sayı olduğunu ekrana yazar

Ornek Ekran çıktısı:
Sayı giriniz: *856*
Girilen sayı üç basamaklıdır.

2 numaralı seçenek seçilmişse;
Kullanıcıdan 4 basamaklı bir sayı girmesini ister (Kullanıcının yaptığı giriş eğer 1000'den küçük veya 9999'dan büyük ise program girişi tekrar ettirmelidir).
Program girilen sayının basamak değerlerinin toplamını bulup ekrana yazar

Örnek ekran çıktısı:
Bir sayı giriniz 3871
Basamaklar toplamı:19

3 numaralı seçenek seçilmişse;
Program **1-20** arasında 10 rasgele sayı üretir ve ekrana yazar. Bunların içindeki çift sayıların ortalamasını hesaplar.
Örnek Ekran çıktısı: Rasgele sayılar
10 12 8 5 16 3 4 17 19 9
Çift olanların toplamı: 50
Ortalaması: 10

Not: rasgele sayı üretmek için;
Srand(time(NULL)) kodunu main bloğunun başında bir kez yazılır
Örneğin bir değişkene 1 ile 10 arasında bir değer atamak için.
A= rand()%10+1; *yazılır*

4 numaralı seçenek seçilmişse; Kendisi hariç, pozitif tam bölenlerinin toplamı yine kendisine eşit olan sayıya mükemmel sayı denir.
Program bir sayı girdirir girilen sayının mükemmel sayı olup olmadığını ekrana yazar.

Örnek Ekran Çıktısı:
Bir sayı giriniz: 6
Sayının kendisi haric tam bölenleri: 1 2 3
Savı mükemmel savıdır.

5 numaralı seçenek seçilmişse;
Programı sonlandırmak için bir tuşa basınız.

EK 8. BAŞARI TESTİ MADDE ANALİZİ SONUÇLARI

Madde No	*Rjx*	*Ayırt Edicilik*	*Pj*	*Madde Güçlüğü*	*Durum*
1	0,55	Çok iyi	0,73	Kolay	Teste dâhil edildi
2	0,70	Çok iyi	0,45	Orta	Teste dâhil edildi
3	0,45	Çok iyi	0,78	Kolay	Teste dâhil edildi
4	0,60	Çok iyi	0,45	Orta	Teste dâhil edildi
5	0,60	Çok iyi	0,60	Orta	Teste dâhil edildi
6	0,30	İyi	0,75	Kolay	Teste dâhil edildi
7	0,15	Testten atılmalı	0,93	Kolay	Testten çıkarıldı
8	0,25	Düzenlenemez ise testten atılmalı	0,38	Zor	Testten çıkarıldı
9	0,35	İyi	0,28	Zor	Teste dâhil edildi
10	0,70	Çok iyi	0,65	Kolay	Teste dâhil edildi
11	0,40	Çok iyi	0,40	Orta	Teste dahil edildi
12	0,50	Çok iyi	0,55	Orta	Teste dahil edildi
13	0,45	Çok iyi	0,58	Orta	Teste dahil edildi
14	0,25	Düzenlenemez ise testten atılmalı	0,83	Kolay	Testten çıkarıldı
15	0,65	Çok iyi	0,43	Orta	Teste dahil edildi
16	0,60	Çok iyi	0,30	Zor	Teste dahil edildi
17	0,15	Testten atılmalı	0,28	Zor	Testten çıkarıldı
18	0,80	Çok iyi	0,55	Orta	Teste dahil edildi
19	0,35	İyi	0,78	Kolay	Teste dahil edildi
20	0,15	Testten atılmalı	0,13	Zor	Testten çıkarıldı

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı	Mustafa TEPGEÇ
Doğum Yeri	MERSİN
Doğum Tarihi	06.12.1991

Eğitim Durumu

Lise	Mersin Yahya Günsür Anadolu Bilgisayar ve Teknik Meslek Lisesi	2010
Lisans	Hacettepe Üniversitesi	2014
Yüksek Lisans	Hacettepe Üniversitesi	-
Yabancı Dil	İngilizce: Okuma (Çok iyi), Dinleme (Orta), Yazma (Orta), Konuşma (Zayıf). YDS: 81,25 – YÖKDİL: 91,25	

İş Deneyimi

Çalıştığı Kurumlar	YediSekizDokuz Eğitim Bilişim Danışmanlık Ltd. Şti.	2013-2014
	Radon Bilişim Ltd. Şti.	2014-2014
	Fonet Bilgi Teknolojileri A.Ş	2014-2015
	Mustafa Kemal Üniversitesi	2015-Devam Ediyor

Akademik Çalışmalar

Yayınlar (Ulusal, uluslararası makale, bildiri, poster vb gibi.)

Demiraslan Çevik & Tepgeç (2016). *Bilişim Teknolojileri Öğretmen Adaylarının Teknoloji Mentorluk Deneyimleri.* 3rd. International Conference on New Trends in Education, İzmir.

Tepgeç & Demiraslan Çevik (2016). *Programlama Öğretiminde Çözümlü Örnek Kullanımı: Bir Betimsel Tarama Çalışması.* 10th International Computer & Instructional Technologies Symposium, Rize.

İletişim

e-Posta Adresi	mustafatepgec@gmail.com
	mustafatepgec@mku.edu.tr

Jüri Tarihi	12 Mayıs 2017