



Programlama Öğretimi için Bir Model Önerisi: Yedi Adımda Programlama

Kürşat Ali Erümit ¹, Hasan Karal ², Güven Şahin ³, Dilara Arzugül Aksoy ⁴,
Ayşegül Aksoy Gencan ⁵, Ali İhsan Benzer ⁶

Öz

Alanyazında programlama öğretiminin pek çok bilişsel beceriye olumlu etkilerini ortaya koyan çalışmalar yer almaktadır. Buna karşın programlama öğretiminin sistematik bir şekilde gerçekleştirilebilmesi amacıyla pedagojiye yönelik çalışmalar oldukça sınırlı sayıdadır. Bu nedenle çalışmada, öğrencilerin bilişsel becerilerinin gelişimine yönelik bir programlama öğretiminin nasıl yapılması gerektiği üzerine odaklanılmıştır. Çalışmanın amacı; Programlama öğretimi sürecinde öğrencilerin çeşitli bilişsel becerilerinin gelişimini sağlamak amacıyla kullanılabilir bir öğretim modeli önerisinde bulunmaktır. Çalışma kapsamında, oluşturulan programlama öğretimi modeline Yedi Adımda Programlama (YAP) modeli adı verilmiştir. Araştırma deseni olarak nitel araştırma yöntemlerinden özel durum çalışması kullanılmıştır. Araştırmanın, YAP modelinin oluşturulması aşamasında araştırma grubunu tasarım ekibi (1 BÖTE öğretim üyesi ile 2 yüksek lisans, 2 doktora öğrencisi) ve tasarım değerlendirme ekibi (MEB'e bağlı ortaokullarda görevli 10 Bilişim Teknolojileri ve Yazılım Dersi öğretmeni) oluşturmaktadır. Araştırma süreci boyunca YAP modeli adımlarının belirlenmesi amacıyla doküman analizi ve tasarım değerlendirme ekibi ile yapılan görüşmelerin analizi gerçekleştirilmiştir. Bu çalışmada, modelin oluşturulması için izlenen süreç, oluşturulan modelin özellikleri ve tasarım değerlendirme ekibi ile yapılan görüşmelere yer verilmiştir. Bu çalışma ile ortaya konulan modelin, ortaokul düzeyinde programlama öğretimi yapmak isteyen eğitimciler için yol gösterici olması hedeflenmektedir.


Anahtar Kelimeler


Programlama öğretimi
Problem çözme
Algoritmik düşünme
YAP öğretim modeli
Ders planları
Ortaokul
Kodlama


Makale Hakkında


Gönderim Tarihi: 29.12.2017
Kabul Tarihi: 25.09.2018
Elektronik Yayın Tarihi: 18.12.2018


DOI: 10.15390/EB.2018.7678


¹  Trabzon Üniversitesi, Fatih Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi, Türkiye, kursaterumit@gmail.com

²  Trabzon Üniversitesi, Fatih Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi, Türkiye, karalhasan@gmail.com

³  Trabzon Üniversitesi, Fatih Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi, Türkiye, guvensahinn@gmail.com

⁴  Trabzon Üniversitesi, Fatih Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi, Türkiye, dilaraarzugulaksoy@gmail.com

⁵  Trabzon Üniversitesi, Fatih Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi, Türkiye, aysegul.aksoy.61@gmail.com

⁶  Trabzon Üniversitesi, Fatih Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi, Türkiye, aibenzer@gmail.com

Giriş

Bilim ve teknolojideki gelişmelerle birlikte bireylerden beklenen beceriler de farklılaşmaktadır. Trilling ve Fadel (2009), 21. yy. becerilerini sözel ve yazılı iletişim, eleştirel düşünme ve problem çözme, profesyonellik ve iş etiği, işbirliği ve takım çalışması, teknolojiyi uygulayabilme, liderlik ve proje yönetimi, çeşitli ekiplerde çalışma becerileri olarak tanımlamaktadırlar. ISTE (2016) ise öğrencilerin etkili öğrenmeler gerçekleştirebilmeleri için bazı standartlara sahip olmaları gerektiğini vurgulamıştır. Bu standartlar; yaratıcılık ve yenilik, iletişim ve işbirliği, araştırma ve bilgi akışı, eleştirel düşünme, problem çözme ve karar verme, dijital vatandaşlık ve teknoloji kullanımı şeklinde gruplandırılmıştır. Bu becerilerin yanında bilgi-işlemsel düşünme becerisi de gerekli görülmektedir (Wing, 2006).

Problem çözme, sistem tasarımı ve insan davranışlarını anlayabilme becerilerini kapsayan geniş bir kavram (Wing, 2006) olarak tanımlanan bilgi-işlemsel düşünme, bilişsel süreçleri ve problem çözme becerisini geliştirmek için bilgisayarların kullanımını da ifade etmektedir (Sanford ve Naidu, 2016).

Problemleri etkili bir şekilde çözmek için algoritmaları kullanmayı içeren bilgi-işlemsel düşünme becerisi (Atmatzidou ve Demetriadis, 2016; Choi, Lee ve Lee, 2016) bu özelliği ile bilgisayar biliminin merkezindedir. Bu nedenle öğrenenin algoritma tasarım ilkelerini anlamasının yanında bu ilkeleri kullanarak uygun algoritmalarla bir problemi çözmesi, öğrenenin bilgi-işlemsel düşünme becerisini geliştirebilir (Choi vd., 2016). Bir problemin çözümü için uygun algoritmalar tasarlayabilmek, algoritmik düşünme becerisi ile yakından ilişkilidir. Algoritmik düşünme becerisi bireyin, yaratıcı ve mantıksal düşünerek, eylemlerin sıralanması olarak ifade edilmektedir (Ziatdinov ve Musa, 2012) ve programlamayla doğrudan ilişkili olarak diğer bilişsel becerilerin gelişimini sağlayabilecek anahtar bir beceridir.

Futschek (2006)'e göre algoritmik düşünme, anlama ve yapılandırma ile ilgili çeşitli alt becerilerden oluşur. Bu alt beceriler; verilen problemleri analiz edebilme, bir problemi tam olarak ifade edebilme, verilen sorun için strateji üretebilme, stratejileri kullanarak verilen bir soruna doğru bir algoritma oluşturma yeteneği, olası tüm özel ve normal durumlarda düşünme becerisi ve bir algoritmanın verimliliğini artırma becerisidir. Futschek (2006)'e göre algoritmik düşünme; bilgisayar ortamından bağımsız olarak farklı stratejileri uygulayabilecekleri oyunlar (Maze) ve sıralama (Paralel Sorting) benzeri uygulamalarla da geliştirilebilmektedir. Bu nedenle öğrenciler bir problemin çözümüne yönelik geliştirdikleri algoritmayı bilgisayar ortamına geçmeden öncede test edebilirler.

Algoritmik düşünme becerisini geliştirme yollarından biri de programlama öğretimidir. İlgili alanyazında programlamanın öğrenenlerde çeşitli becerileri geliştirdiği ifade edilmektedir:

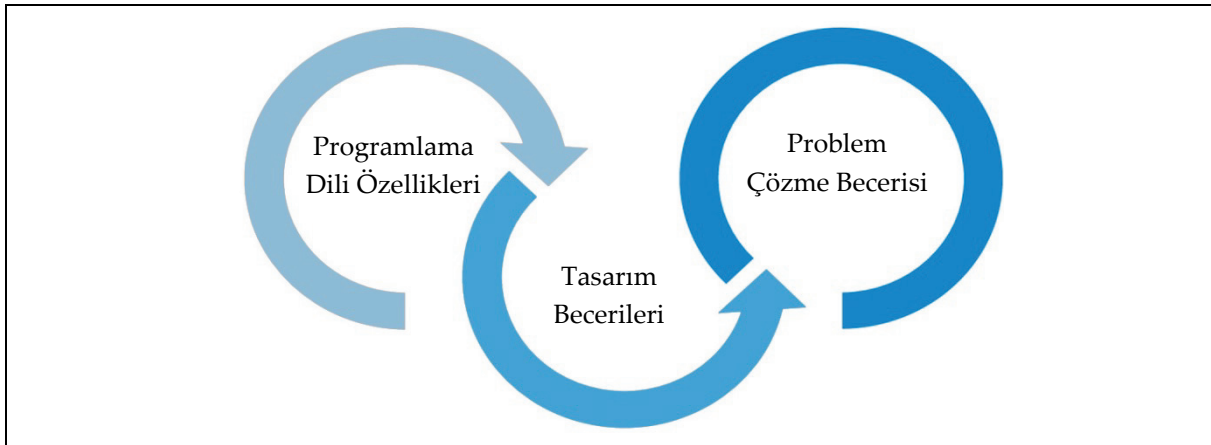
- Programlamanın öğrencilerin problem çözme becerilerini geliştirdiği, (Bergersen ve Gustafsson, 2011; Brown vd., 2013; Kalelioğlu ve Gülbahar, 2014; Lai ve Lai, 2012; Lai ve Yang, 2011),
- Bilişsel öğrenmeyi olumlu yönde etkilediği (Clements ve Sarama, 2003; Crescenzi, Malizia, Verri, Diaz ve Aedo, 2012; Grover ve Pea, 2013; Utting, Cooper, Kölling, Maloney ve Resnick, 2010),
- Öğrencilerde üst düzey düşünme becerilerini geliştirdiği (Kafai ve Burke, 2014; Shih, 2014),
- Motivasyonu artırdığı (Akpınar ve Altun, 2014; Sáez-López, Román-González ve Vázquez-Cano, 2016),
- Yaratıcı düşünme becerisini geliştirdiği (Fesakis ve Serafeim, 2009; Kobsiripat, 2015) bilinmektedir.

Programlama, öğrencilerin bilişsel becerilerini geliştirmekle birlikte öğreniminde de üst düzey düşünme becerileri gerektirmektedir (Law, Lee ve Yu, 2010). Bu sebeple programlama öğretim sürecinin oldukça zor olduğu (Helminen ve Malmi, 2010) ve öğrencilerin bu dersteki başarı oranlarının düşük olduğu (Robins, Rountree ve Rountree, 2003) ifade edilmektedir. Belirtilen bu zorluklar ise; program yapısı (Lahtinen, Ala-Mutka ve Järvinen, 2005), döngüler (Ginat, 2004), algoritma yapısı (Seppälä, Malmi ve Korhonen, 2006) gibi temel programlama kavramlarının öğrenme sürecinde

meydana gelmektedir. Aynı zamanda öğretim metodundan kaynaklı olarak da programlama dersinde öğrenciler zorluk yaşayabilmektedir. Birçok araştırmada belirtilen zorluklara karşı programlama öğretiminin nasıl olması gerektiği tartışılmaktadır (Coull ve Duncan, 2011; Lahtinen vd., 2005).

Programlama başarısını artırmak ve anlamayı kolaylaştırmak için öncelikli olarak algoritma mantığının öğrencilere kazandırılması gerekmektedir (Ala-Mutka, 2004). Bu nedenle, programlama eğitimine yeni başlayanlar için öğrenimi kolaylaştırmak amacıyla ilgi çekici ve eğlenceli görsel programlama dilleri geliştirilmiştir (Schwartz, Stagner ve Morrison, 2006). Metin tabanlı programlamada kod dizimi, öğrencilerin en çok problem yaşadığı konulardan biri olarak belirtilirken (Özmen ve Altun, 2014), görsel programlamada sunulan kod blokları bu problemi aşarak başlangıç düzeyinde ki programcılar için anlamayı ve uygulamayı kolaylaştırmaktadır (Wilson ve Moffat, 2010). Sáez-López ve diğerleri (2016); görsel programlarla projeler üzerinde çalışmanın öğrencilerde motivasyon ve isteği artırdığını ifade etmişlerdir. Pek çok çalışmada görsel programlama kullanılmasının kavramayı hızlandırdığı ifade edilmiştir (Naharro-Berrocal, Pareja-Flores, Urquiza-Fuentes ve Velazquez-Iturbide, 2002). Programlamaya yeni başlayan öğrenenler için Scratch (Malan ve Leitner, 2007; Wu, Chang ve He, 2010), kodu (Stolee ve Fristoe, 2011), StarLogo (Klopfer ve Yoon, 2005), ve Alice (Kelleher, Pausch ve Kiesler, 2007) programları önerilmektedir.

Robins ve diğerleri (2003)'e göre; programlama öğretimi sadece yeni dil özelliklerini öğrenmeye değil, aynı zamanda bu özelliklerin farklı durumlarda kullanımı, özellikle de temel program tasarımının altında yatan konuya odaklanmalıdır. Linn ve Dalbey (1989) programlama öğretimi için "bilişsel yeterlilikler zinciri" sunmaktadır. Bu zincir, dilin özellikleri ile başlar. İkinci halka; şemaları tasarlama, planlama, test etme ve yeniden düzenlemeyi içeren tasarım becerilerini kapsar. Üçüncü halka; bilgi ve stratejileri yeni bir programlama diline uyarlamayı içeren problem çözme becerisidir.



Şekil 1. Programlama Öğretimi İçin Gerekli Bilişsel Yeterlikler (Linn ve Dalbey, 1989)

Şekil 1'de gösterilen bu zincir programlamaya girişin temelini oluşturmaktadır. Öğretici, kontrol yapılarına, veri yapılarına, program tasarım ve problem alanını açık bir şekilde ifade etmelidir. Aksi halde sağlam bir alt yapı oluşturmak mümkün olmayacaktır (Winslow, 1996). Öğretim sürecinde görsel ve animasyon tabanlı programlama etkinlikleri verilerek öğrenci motivasyonu artırılabilir. Bu noktada öğreticinin görsel tabanlı bu etkinliği programlamanın temel kavramlarına dayandırarak açıklaması önemli görülmektedir (Kurland, Pea, Clement ve Mawby 1989). Spohrer ve Soloway (1989) de programlama öğretimi için bazı öneriler sunmuşlardır:

- Akış kontrolü için grafiksel diller kullanın.
- Basit bir makine modeli kullanın
- Değişken ve sabit isimleri vermeden önce kısa, basit ve tutarlı bir adlandırma kuralı belirleyin
- Uzamsal metaforlara dayalı tasarım örnekleri sunun
- Öğrenciye sunduğunuz destekleri yavaş yavaş azaltın.

Programlama öğretim sürecinde öğrencilere hızlı ve sürekli düzeltmeler verilmelidir. Bununla birlikte öğrencilerin kendi öğrenmelerine imkân verilmesi (Linn ve Dalbey, 1989) ve işbirlikli çalışmalar yapılması (Van Gorp ve Grissom, 2001; Williams, Wiebe, Yang, Ferzli ve Miller, 2002) önerilmektedir.

Programlama öğretiminin nasıl yapılması gerektiği ile ilgili pedagojik yaklaşım öneren çalışmaların yok denecek kadar az olması eğitim kurumlarında aynı yaş grubuna yönelik düzenlenen eğitimlerde de farklılıkların ortaya çıkmasına neden olmaktadır. Bu farklılıklar programlama öğretimi için hazırlanan etkinliklerin ve izlenen adımların farklılaşması şeklinde görülmektedir. Amacı öğrencilerin bilişsel becerilerinin gelişimini sağlamak olan bir öğretim programı için bu durum;

1. Dersin bilişsel becerileri geliştirmeye yönelik etkinliklerden çok animasyon oluşturma ve boş zaman aktivitesine dönüşmesi
2. Derste animasyon tarzı etkinliklerin ağırlıklı olması sonucu öğrencilerin programlamadan uzaklaşması
3. Bilişim teknolojileri ve yazılım dersinin diğer derslere olabilecek olumlu etkilerinin kaybolması
4. Dersin öğrenciler, veliler ve yetkililer nazarında öneminin yitirilmesi
5. Öğretim programının hedefine ulaşamaması şeklinde olumsuz sonuçların ortaya çıkmasına neden olabilir.

Bu durumun temel nedenlerinden biri de Web 2.0 teknolojilerinin erişilmesi ve öğrenilmesi kolay yapısından dolayı öğretmenlerin okullarda öğretmeye çalıştığı animasyona yönelik etkinliklerin öğrenciler tarafından kendiliğinden yapılabilmesidir (Karaman, Yıldırım ve Kaban, 2008; Kam ve Katerattanakul, 2014). Bilişsel becerilerin gelişimine yönelik etkilerinin göz ardı edilmesi, öğrencilerde programlama öğretiminin yalnızca animasyon hazırlama eğitimi olarak algılamasına (Lee, 2011; Scaffidi ve Chambers, 2011), ve bu durum öğrencilerin dersin önemini kavrayamamasına ve okullarda verilen eğitimlerin sıradanlaşarak sonunda sorgulanmasına neden olabilir (Kalelioğlu ve Gülbahar, 2014; Kukul ve Gökçearslan, 2014; Oluk ve Saltan, 2015; Yükseltürk, Altıok ve Üçgül, 2016).

21.yy'da bireylerinin üretken ve aktif olabilmeleri için gerekli becerilerin geliştirilmesinde programlama öğretimi önemli bir yere sahiptir. Bu nedenle dünyanın pek çok ülkesinin öğretim programlarında programlama öğretimi yer almaya başlamıştır (Balanskat ve Engelhardt, 2015). Ülkelerin programlama öğretimini, öğrencilerin yalnızca programlama becerilerini geliştirmek amacıyla değil aynı zamanda mantıksal düşünme ve problem çözme becerilerini geliştirmek amacıyla da öğretim programlarına entegre ettikleri görülmektedir (Balanskat ve Engelhardt, 2015). Ancak programlama öğretimi öğrenciler için algılanması zor bir süreç olarak görülmektedir (Porter ve Calder, 2004). Bunun nedeni ise programlama araçlarının ve programlama dillerinin anlaşılması zor bir yapıya sahip olmaları, programlama dilinin söz dizim(syntax) kurallarının zorluğu, programlama öğretiminde kullanılan geleneksel yöntemler ve öğrencilerin algoritmik düşünme becerilerinin yetersizliği gösterilmektedir (Byrne ve Lyons, 2001; Futschek, 2006). Bu çerçevede çalışmanın amacı; ortaokul seviyesinde öğrencilerin, algoritmik düşünme, problem çözme ve programlama becerilerinin gelişimine yönelik olarak yapılacak programlama öğretimi sürecinde kullanılacak bir öğretim modeli önerisinde bulunmaktadır.

Yöntem

Araştırma Modeli

Çalışmada; ortaokul seviyesinde problem çözme, algoritmik düşünme ve programlama becerileri geliştirmeye yönelik programlama öğretimi için bir model önerisinde bulunularak, öğretmenlerden oluşan Tasarım Değerlendirme Ekibi (TDE) görüşleri alınmıştır. Bu doğrultuda öncelikle programlama öğretimi için uygun olabilecek adımların neler olduğuna yanıt aranmış ve YAP modeli adımları doküman analiziyle belirlenmiştir. Daha sonra YAP modeli doğrultusunda hazırlanan ders planlarına yönelik olarak; TDE ile yapılan görüşmelerle, bilişsel seviye, süre ve kalabalık sınıflarda uygulanabilirlik ölçütlerine göre görüşleri yarı yapılandırılmış görüşme formu kullanılarak alınmıştır. Alınan görüşler doğrultusunda ilgili güncellemeler gerçekleştirilmiştir. Araştırma deseni olarak nitel araştırma yöntemlerinden özel durum çalışması kullanılmıştır. Özel durum çalışması; gözlem, görüşme, doküman analizi gibi nitel veri toplama yöntemlerinin kullanıldığı algıların ve olayların doğal ortamda gerçekçi ve bütüncül bir şekilde ortaya konmasına yönelik bir sürecin takip edildiği araştırma olarak tanımlanmaktadır (Yin, 2017). Ayrıca araştırmanın tasarlanması ve yürütülmesi süreçlerinde araştırmacıya esneklik sağlamaktadır (Silverman, 2013).

Bu çalışmada:

- YAP modeli adımlarının belirlenmesi amacıyla belirli anahtar kelimeler ve veri tabanlarında doküman analizi yapılması
- Oluşturulan ders planı, etkinlikler ve YAP modeli adımlarının belirli TDE öğretmenleri ile yapılan görüşmeler sonrasında derinlemesine analizi,
- Nitel yöntemlerle duygu, düşünce ve hislerin daha iyi anlaşılacak istenmesi,
- Bulguların betimleyici şekilde değerlendirilmesinden dolayı özel durum çalışması tercih edilmiştir.

Araştırma Grubu

Araştırmanın, YAP modelinin oluşturulması ve oluşturulan modele göre hazırlanan ders planlarının değerlendirilmesi aşamalarında çalışma grupları farklılık göstermektedir. Bu doğrultuda YAP modelinin oluşturulması aşamasında tasarım ekibi (TE) 5 kişiden (1 BÖTE öğretim üyesi ile 2 yüksek lisans, 2 doktora öğrencisi) ve tasarım değerlendirme ekibi(TDE) 10 kişiden (MEB'e bağlı ortaokullarda görevli 10 Bilişim Teknolojileri ve Yazılım Dersi öğretmeni) oluşmaktadır. TE bilgileri Tablo 1'de ve TDE bilgileri ise Tablo 2'de sunulmaktadır.

Tablo 1. Tasarım Ekibi

Ekip Üyesi	Ünvanı/Öğrenim Düzeyi	Deneyimi		
		Programlama	Programlama Öğretimi	Lisans Mezuniyeti
C1	Yrd. Doç. Dr.	21 yıl	14 yıl	Bilgisayar Öğretmenliği
SP1	Öğr. Gör. -Doktora Öğrencisi	17 yıl	14 yıl	Bilgisayar Öğretmenliği
SP2	Doktora Öğrencisi	8 yıl		Bilgisayar ve Öğretim Teknolojileri Eğitimi
SM1	Yüksek Lisans Öğrencisi	6 yıl		Bilgisayar ve Öğretim Teknolojileri Eğitimi
SM2	Yüksek Lisans Öğrencisi	6 yıl		Bilgisayar ve Öğretim Teknolojileri Eğitimi

YAP modelinin, modele uygun ders planlarının ve etkinliklerin tasarlanması aşamasında Tablo1' de belirtilen tasarım ekibi görev almıştır. Çalışma 2016-2017 güz yarıyılında 16 haftalık sürede haftalık 6 saatlik çalışmalarda gerçekleştirilmiştir. Oluşturulan taslak model, ders planları ve

etkinliklerin değerlendirilmesi; Tablo 2’de belirtilen ve aktif olarak okullarında ortaokul seviyesinde programlama öğretimi veren öğretmenler (tasarım değerlendirme ekibi) tarafından gerçekleştirilmiştir. Öğretmenlerin YAP modeli adımlarını somut olarak görebilmeleri ve böylelikle daha gerçekçi bir değerlendirme yapabilmeleri amacıyla TE tarafından her konu için etkinlikler ve ders planları hazırlanmıştır. TDE ile yapılan görüşmelerde; öncelikle YAP modelinin adımları açıklanarak ders planlarında karşılık gelen işlemler ve etkinlikler gösterilmiş daha sonrada öğretmenlerin görüşleri alınmıştır. Öğretmenlerin tamamı Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü lisans mezunudur.

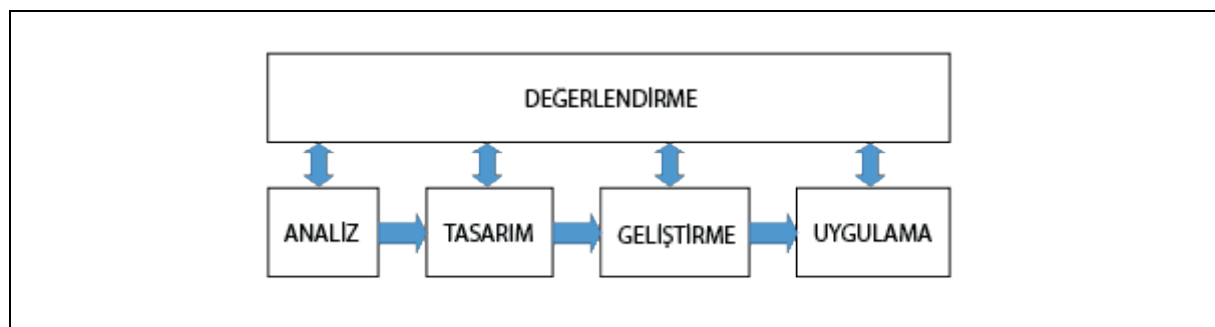
Tablo 2. Tasarım Değerlendirme Ekibi

Öğretmen	Deneyimi	
	Öğretmenlik	Programlama Öğretimi
Ö1	13 yıl	4 yıl
Ö2	10 yıl	4 yıl
Ö3	10 yıl	4 yıl
Ö4	6 yıl	4 yıl
Ö5	6 yıl	4 yıl
Ö6	14 yıl	3 yıl
Ö7	12 yıl	2 yıl
Ö8	10 yıl	2 yıl
Ö9	10 yıl	2 yıl
Ö10	10 yıl	1 yıl

TE tarafından hazırlanan etkinlikler ve ders planları, TDE ile yapılan görüşmelerle; bilişsel seviyeye uygunluk, kalabalık sınıflarda uygulanabilirlik ve süreye uygunluk ölçütleri açısından değerlendirilmiş ve görüşler doğrultusunda model, ders planları ve etkinlikler geliştirilmiştir.

Araştırma Süreci

Çalışmanın amacı doğrultusunda araştırma süreci ADDIE tasarım modeline göre yapılmıştır. ADDIE adını Analyze, Design, Develop, Implement, Evaluate kelimelerinin baş harflerinden oluşan ve beş adımdan meydana gelen bir öğretim tasarım modelidir. ADDIE modeli tüm öğrenimler için uygun olabilen temel bir model olması ve diğer öğretim tasarımı modellerinin bileşenlerini içermesi sebebiyle tercih edilmiştir. Bu modele göre araştırma süreci şekil-2’de yer verilmiş olup süreçte izlenen adımlar detaylı olarak anlatılmıştır.



Şekil 2. ADDIE Tasarım Modeline Göre Araştırma Süreci

(1) Analiz

Analiz aşaması; problem analizi, görev analizi ve öğretim analizi olmak üzere 3 adımda gerçekleşmiştir. Problem analizinde, TE tarafından alanyazın taraması yapılmıştır. Bu tarama sonucunda ortaokul düzeyinde programlama öğretimi için pedagojik bir yaklaşım öneren çalışmaların

azlığı dikkat çekmiştir. Ayrıca programlama öğretiminin nasıl yapılabileceğiyle ilgili ortak bir yaklaşımın olmadığı görülmüştür. Bu nedenle çalışmaya 'Ortaokul seviyesinde programlama öğretimi nasıl yapılmalı?' sorusundan hareketle başlanmıştır. Görev analizinde görev, problem analizinden elde edilen sonuca göre 'Ortaokul seviyesinde algoritmik düşünme ve problem çözme becerisi yeterliklerini içeren programlama öğretimi için bir model önerisinde bulunulmak' olarak belirlenmiştir. Öğretim analizi kapsamında ise; "Problem Çözme Kavramları ve Yaklaşımları", "Programlama Ortamını Tanıyalım", "Değişkenler", "Koşul Yapıları" ve "Döngüler" konularının kazanımları Milli Eğitim Bakanlığı [MEB] (2017) Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı'ndan alınmıştır.

(2) Tasarım: Öğretim Adımlarının Belirlenmesi

Öğretim adımlarının belirlenmesi aşamasında yapılan doküman analizinde Yıldırım ve Şimşek'in (2011) belirttiği şekilde; dokümanlara ulaşma, orijinalliği teyit etme, dokümanların anlaşılması, verilerin analiz edilmesi ve verilerin kullanılması adımları izlenmiştir.

Dokümanlara Ulaşma

YAP modeli adımlarının belirlenmesi için programlama öğretiminde temel teşkil eden algoritmik düşünme becerisine yönelik çalışmalara; Algoritmik düşünme, Algorithmic Thinking, Curriculum, Primary and secondary education, Problem solving, Computational Thinking, Deeper Learning anahtar kelimeleriyle Google Scholar, Science Direct ve Eric veri tabanlarında arama yapılarak ulaşılmıştır. Elde edilen 15 çalışma içerisinden öğretim sürecine yönelik kullanılabilir seviye ve adımların yer aldığı 6 çalışma belirlenmiştir. Etkinliklerin uygulanmasına yönelik belirli pedagojik yaklaşımların (ikili kodlama, drama vb.) kullanıldığı ancak bütün olarak bir ders için hazırlanmamış çalışmalar ile sürece ilişkin yeterli bilgi sağlamayan çalışmalar elenmiştir. Araştırma kapsamında makalelerin kullanılmasının nedeni; araştırmanın yapılabilirliği düşünülerek araştırmanın sınırlarını net olarak belirleyebilmektir. Bu doğrultuda incelenen çalışmalar Tablo 3'de listelenmiştir.

Tablo 3. İncelenen Dokümanlar

Doküman Türleri	İncelenen Kaynaklar
Makaleler	Zsakó ve Szlávi (2012), ICT Competences: Algorithmic thinking Vasconcelos (2007), Basic Strategy for Algorithmic Problem Solving Futschek (2006), Algorithmic thinking: The key for understanding computer science Committee on Logic Education(2008) Szántó, (2002) (aktaran Zsakó ve Szlávi, 2012, s. 55) Garner (2003), Learning resources and tools to aid novices learn programming

Orijinalliği Teyit Etme

Yapılan çalışmanın geçerliğinin sağlanması açısından ulaşılan dokümanların orijinalliğinin teyit edilmesi önem taşımaktadır. Bu bağlamda Google Scholar, Science Direct ve Eric veri tabanlarında belirlenen anahtar kelimeler ile arama yapılmış ve ulaşılan makalelerin hangi dergilerde yayınlandığı incelenerek orijinalliği teyit edilmiştir. Bu kapsamda dergilerin dizinlenme bilgileri ve hakemli dergi olma durumları kontrol edilmiştir.

Dokümanların Anlaşılması

Bu aşamada belirlenen dokümanlar sistematik olarak birbirleriyle karşılaştırılmıştır. Bu noktada makalelerde öğretim sürecinde kullanılabilir seviye ve adımlar incelenmiş ve karşılaştırılmıştır. Makaleler TE tarafından içerik analizine tabi tutularak temalar belirlenmiştir.

Verilerin Analiz Edilmesi

Dokümanların anlaşılması aşamasında TE tarafından yapılan karşılaştırma işlemi sonucu ortaya çıkan temalar Tablo 4'de verilmektedir.

Tablo 4. Öğretim Adımları İçin Belirlenen Temaların Karşılaştırılması

Çalışmalar	Problemi Tanıma ve Anlama	Strateji Oluşturma	Strateji Karşılaştırma	Algoritmayı Oluşturma	Algoritmayı Kodlama	Farklı Bir Koddaki Hatayı Belirleme ve Düzeltme	Yeni Algoritmalar Hazırlama ve Kodlama
Zsakó ve Szlávi (2012)	✓	✓	✓	✓	✓	✓	✓
Vasconcelos (2007)	✓	✓		✓	✓		✓
Futschek (2006)	✓	✓		✓	✓		
Comittee on Logic Education (2008)	✓		✓	✓			
Szántó (2002)				✓	✓		
Garner (2003)	✓			✓	✓	✓	

Analiz süreci fikir birliği doğrultusunda TE'nin eş zamanlı çalışması ile tamamlanmıştır. TE tarafından bulunan temalar yapılan toplantılar ile incelenmiş ve bu inceleme sonucunda temaların; Problemi Tanıma ve Anlama, Strateji Oluşturma, Strateji Karşılaştırma, Algoritmayı Oluşturma, Algoritmayı Kodlama, Farklı bir Koddaki Hatayı Belirleme ve Düzeltme, Yeni Algoritmalar Hazırlama ve Kodlama faktörlerinde kesiştiği görülmüştür. Temaların belirlenmesi sürecine ilişkin bilgiler "Araştırmanın geçerliği ve güvenilirliği" başlığında verilmiştir.

Verilerin Kullanılması

Doküman analizi sonucunda elde edilen verilere ve sonuçlara çalışmanın bulgular ve sonuç bölümlerinde yer verilmiştir. Öğretim adımları olarak kullanılacak seviye ve adımlar incelendikten sonra çalışmada yer alan adımların detaylı açıklanmasına bulgular bölümünde, oluşturulan YAP modeli adımlarına ise sonuç bölümünde yer verilmiştir.

Ders Planı Değerlendirme Ölçütlerinin Belirlenmesi

Ek-1'de bulunan ders planı değerlendirme ölçütleri formu ile 150 öğretmene "Ders planı hazırlarken dikkat edilmesi gereken noktalar sizce nelerdir?" sorusu yöneltilmiştir. Soruya verilen cevapların TE tarafından değerlendirilmesi neticesinde 3 ölçüt belirlenmiştir. "Bilişsel seviyeye uygunluk", "Kalabalık sınıflarda uygulanabilirlik" ve "Süreye uygunluk" olarak belirlenen ölçütlerin dağılımı Tablo 5'te verilmiştir.

Tablo 5. Belirlenen Ölçütlerin Dağılımı (n=150)

	f	%
Bilişsel Seviyeye Uygunluk	96	64
Süreye Uygunluk	64	42,66
Kalabalık Sınıflarda Uygulanabilirlik	86	57,33

Tablo 5'e göre, öğretmenler ders planı hazırlanırken dikkat edilmesi gereken en önemli nokta olarak tasarlanacak etkinliklerin bilişsel seviyeye uygun olması gerektiğini belirtmişlerdir. İkinci husus olarak tasarlanan bu etkinliklerin kalabalık sınıflarda da uygulanabilecek şekilde olması gerektiğini vurgulamışlardır. Dikkat çektikleri bir diğer hususun ise etkinliklerin ortalama olarak ders süresine uygun olmasını ifade etmişlerdir.

(3) Geliştirme: Tasarım Ekibi Tarafından Etkinlikler ve Ders Planlarının Hazırlanması

MEB öğretim programı göz önüne alınarak, ders planlarında kullanılacak etkinlikler ve ders planları TE tarafından geliştirilmeye başlanmıştır. YAP modeli adımlarının uygulanabilir nitelikte

olması için geliştirilecek olan etkinlikler oldukça önemlidir Etkinlikler geliştirilirken öğrencilerin bilişsel seviyelerine uygunluğu, kalabalık sınıf ortamında uygulanabilirliği, ders süresine uygunluğu ve bilgisayar ortamına uyarlanabilir olmasına dikkat edilmiştir. Etkinlikler belirlenirken dikkat edilen bir diğer nokta geliştirilecek bu etkinliklerin analojik bir şekilde tasarlanabilir olmalarıdır. Analoji, bilinmeyen bir olay ya da durumu bilinen bir olay ya da durum ile karşılaştırma yaparak ve ilişkiler kurarak anlama sürecini içermektedir (Çıray ve Erişti, 2014). Analoji yöntemi, öğrencilerin derse aktif katılım sağlaması, sorgulayıcı ve yaratıcı özelliklerinin gelişmesinde önemli rol oynamaktadır. (Yuretich, Khan, Leckie ve Clement, 2001). Çalışmada etkinlikler, analojik yapıda olup, öğrencilerin etkinlik sırasında kendilerini rahat bir şekilde ifade etmelerine imkân tanıyacak şekilde geliştirilmiştir. Etkinliklerde, günlük hayattan örnekler yer verilmesine özen gösterilmiştir. Bu türden etkinlikler öğrencilerin, anlamakta zorlandıkları programlama konularını kendi düzeylerine göre indirgeyebildikleri için konuların daha rahat anlaşılmasını sağlamaktadır. Ayrıca geliştirilen etkinliklerin bir problem durumu ve problemi çözüme ulaştıracak birden fazla stratejinin yer almasına dikkat edilmiştir. TE üyeleri, yapmış oldukları çalışmaları yapılan grup toplantılarında sunmuş ve bu çalışmalar diğer ekip üyeleri tarafından değerlendirilmiştir. Yapılan toplantılarda çalışmalar için alınan kararlar not edilmiş ve çalışmalar güncellenerek tekrar değerlendirilmiştir. Bu şekilde nihai olarak elde edilen ders planları TE üyeleri tarafından onaylanmıştır.

(4) Uygulama: Planların Öğretmen Görüşüne Sunulması

TE görüşleri ile hazırlanan ders planları, TDE ile yapılan yarı yapılandırılmış görüşmelerde, belirlenen ders planı değerlendirme ölçütlerine göre değerlendirilmiştir. Öğretmenlerle yapılan görüşme sorularına Ek-2'de yer verilmiştir.

(5) Değerlendirme: Planların Güncellenmesi

TDE öğretmenlerinden görüşme yoluyla elde edilen veriler göz önünde bulundurularak ders planları ile ilgili güncellemeler gerçekleştirilmiştir. Ders planlarının oluşturulması için 10 haftalık sürecin ardından yaklaşık 6 haftalık bir süreçte tüm öğretmen görüşmeleri ve güncellemeler tamamlanarak planlar uygulama sürecine hazır hale getirilmiştir. Bu çalışmada YAP modeli ve modele ilişkin görüşlere odaklanıldığından modele uygun ders planlarının geliştirilmesi süreci anlatılmamıştır. Bu konuda detaylı bilgiye ve örnek ders planlarına Şahin (2018)'den ulaşılabilir.

Araştırmanın Geçerliliği ve Güvenirliği

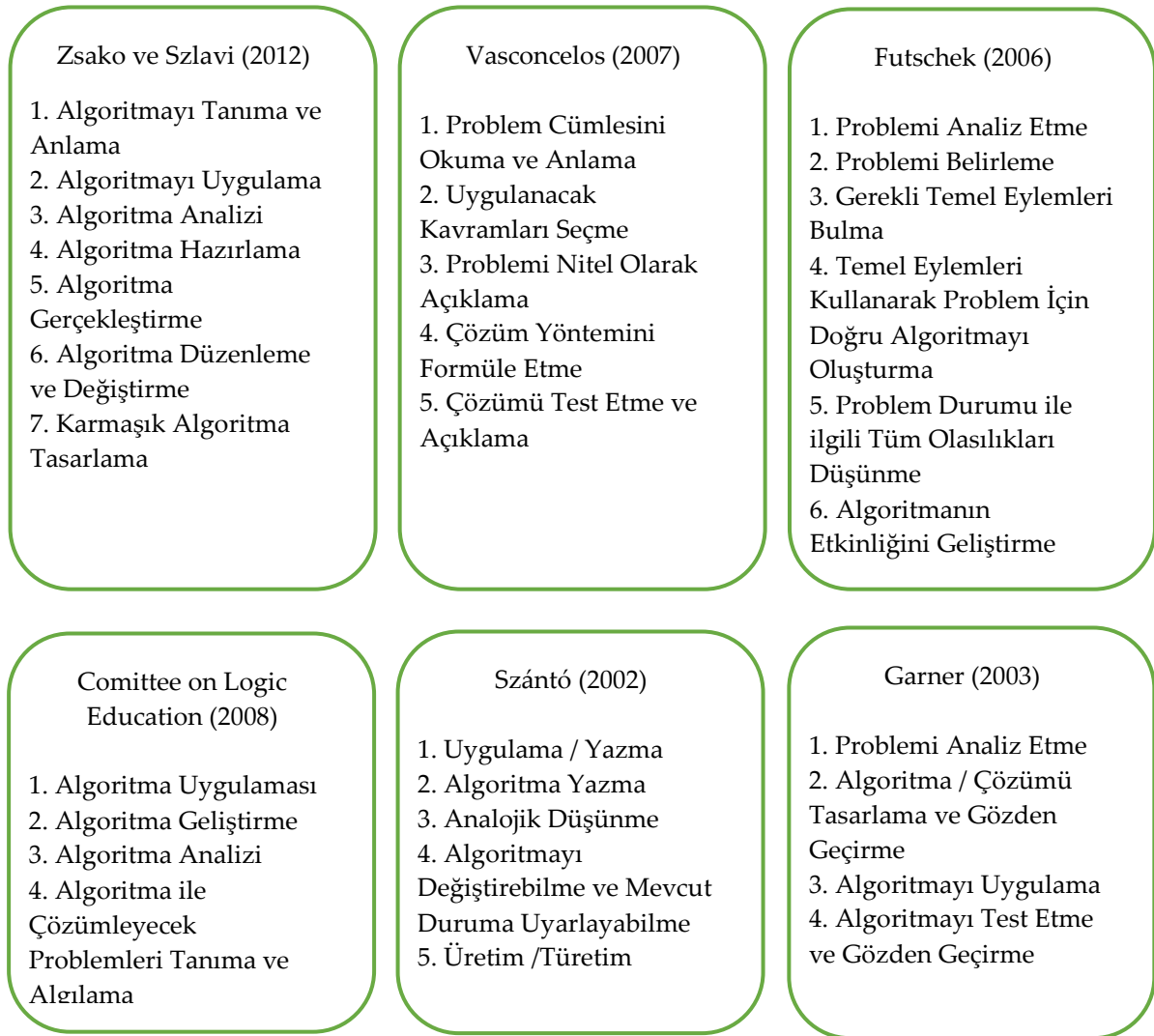
Katılımcıların ifade ettiklerini, araştırmacının gördüğünü ve okuduğunu birleştirme, indirgeme ve yorumlama sürecini içeren (Merriam ve Tisdell, 2015) nitel veriler, içerik analizi yöntemiyle çözümlenmiştir. Öğretmen ve öğrencilerle yapılan yarı yapılandırılmış görüşmeler kayıt altına alınmış ve daha sonra bu kayıtların çözümlenmesi yoluyla elde edilen veriler bilgisayar ortamına aktarılmıştır. Sonrasında Creswell (2013)'in de belirttiği gibi veriler düzenlenmiş, temalara göre kodlanmış ve bu temalar alıntılarla desteklenerek içerik analizi tamamlanmıştır. Araştırma süreci boyunca geçerlik ve güvenirliliği tehdit edebilecek faktörleri engellemek için birtakım tedbirler alınmıştır. Görüşmelere başlanmadan önce katılımcılar bilgilendirilmiş, katılım tamamen gönüllülük esasında gerçekleşmiş ve katılımcı bilgileri gizli tutulmuştur. Görüşmeler gerçekleştirilirken ses kaydı alınmış, araştırmacı notları tutulmuş her veri kayıt edilmeye çalışılmıştır. Elde edilen tüm veriler sonuçların doğrulanabilir olması amacıyla titizlikle saklanmıştır. Verilerin analizi aşamasında Bilgisayar ve Öğretim Teknolojileri Eğitimi alan uzmanı üç araştırmacı analizleri gerçekleştirmiş ve daha sonra analiz sonuçları arasında uyum sağlanmıştır. Her bir araştırmacı birbirlerinden ayrı olarak; verilerden temaları ve temaların ilişkili olduğu alanları belirlemiştir. Araştırmacıların ilk belirledikleri temaların birbirleriyle tutarlılık oranı %92'dir. Araştırmacılar daha sonra bir araya gelerek temalar ve ilişkili alanlar üzerinde görüş birliğine varmışlardır. Gerekli görülen yerlerde, elde edilen veriler katılımcılara tekrar ulaşılarak sorulmuş ve teyit edilmiştir. Son olarak araştırma raporunda tüm süreç ayrıntılı olarak betimlenmiş ve bu şekilde transfer edilebilir nitelikte bir araştırma ortaya konulmuştur. Araştırmacılar verilerin analiz edilmesi ve raporlaştırma aşamalarında aktif rol almışlardır.

Bulgular

Çalışmada öğretim adımlarının belirlenmesi amacıyla TE tarafından gerçekleştirilen alanyazın taraması sonucu elde edilen veriler TDE tarafından değerlendirilerek tüm ekibin uzlaştığı 7 adım kabul edilmiştir. Bu adımların belirlenmesinde öncelikle adımların bilişsel becerilerle olan ilişkisi daha sonra da öğrencilerin bilişsel seviyelerine uygunluğu ve kalabalık sınıflarda uygulanabilirliği göz önünde bulundurulmuştur. Bulgular bölümünde; YAP modeli adımlarını belirlemek amacıyla yapılan doküman analizi ve TDE görüşlerine ilişkin bulgular verilmektedir.

YAP Modeli Adımlarının Belirlenmesi

YAP modeli; ortaokul seviyesindeki öğrencilerin algoritmik düşünme, problem çözme ve programlama becerilerini geliştirmeye yönelik bir programlama öğretimi gerçekleştirebilmek amacıyla geliştirilmiştir. Bu amaçla yapılan doküman analizi sonucunda kriterlere uygun olan çalışmalarda yer alan bulgular şekil 3’de gösterilmektedir. Şekil 3’de gösterilen adımlar ilgili çalışmalarda, algoritmik düşünme aşamaları olarak tanımlanmaktadır.



Şekil 3. Algoritmik Düşünme Aşamaları

Şekil 3’de belirtilen çalışmalara yönelik TE tarafından yapılan değerlendirmeler şu şekildedir:

Zsako ve Szlávi (2012) tarafından tasarlanan algoritmik düşünme süreci gruplara ayrılarak yönetildiği görülmektedir. Örneğin, beşinci adımda kodlama işlemi gerçekleştirilirken diğer aşama

tiplerinde kodlamaya daha abuk geildiđi grlmektedir. alıřmada, her bir ařamanın bir nceki ařamanın zerine inřa edildiđi ifade edilmiř ve her bir ařamanın biliřsel beceriler ile iliřkisi nemle belirtilmiřtir. İleri ařamaların st dzey dřünme becerisi gerektirdiđi ifade edilmiřtir.

Vasconcelos'un (2007) beř ařamadan oluřan planında, problemi anlama ve aıklamaya daha fazla yer verildiđi grlmektedir. Ayrıca algoritma dzenleme, geliřtirme ve karmařık algoritma tasarlama adımlarına pek yer verilmediđi grlmüřtür. Ancak her ařamada ne ifade edildiđi ve nelerin yapılması gerektiđi ayrıntılı olarak aıklanması nemlidir.

Futschek'in (2006) altı ařamadan oluřan algoritmik dřünme srecinde, problemin anlaşılması ve aıklanmasına daha fazla yer verilirken karmařık algoritmalar tasarlama adımlarına dřük dzeyde yer verildiđi ve ařamaları oluřturan adımlara ayrıntılı olarak deđinilmediđi grlmüřtür.

Committee on Logic Education (2008)'ın algoritmik dřünme srecinin ařamalarında adımların birleřtirilme yoluna gidilerek gruplandırıldıđı ve zellikle de algoritmanın test iřlemi gerekleřtirildikten sonra hataların analiz edilmesi ve dzeltilmesi, karmařık algoritma tasarlama adımlarına daha az yer verildiđi grlmektedir. Bu đretimin ortaokul seviyesinde yapılması amalandıđı için ařamaların daha ayrıntılı bir řekilde ele alınması uygun olacaktır.

Szántó (2002), algoritmik dřünme srecinde birbirinden farklı adımları gruplandırmıř ve genel adım sayısında azaltmaya gittiđi grlmüřtür (aktaran Zsakó ve Szlávi, 2012).

Garner'in (2003), tasarlamıř olduđu adımları birleřtirerek ařama sayısını azalttıđı grlmektedir. Ařamaları oluřturan adımlara ayrıntılı olarak deđinilmemiřtir.

alıřmada algoritmik dřünme becerisinin seviyeleri olarak tanımlanan bu ařamalar; biliřsel becerileri kapsama ve kalabalık sınıflarda ders planı hazırlamaya uygunluk kriterlerine gre deđerlendirilerek đretim adımları belirlenmiřtir.

Tasarım Deđerlendirme Ekibinin YAP Modeline İliřkin Grřleri

TDE ile yapılan yarı yapılandırılmıř grřmelerin ardından YAP modelinin adımlarında iřlenen uygulamalara iliřkin bulgular Tablo 6'da modelin uygunluđuna ynelik deđerlendirme bařlıđı altında sunulmuřtur.

Tablo 6. Modelin Uygunluğunun Değerlendirilmesi

Adımlar	Uygundur (n=10)	Güncellenebilir (n=10)		Konular	Öneriler (n=10)
		Seviyeye Uygunluk	Uygulanabilirlik		
Problemi Tanıma ve Anlama	8	2	-	Koşul Yapıları	“Bence uygun bir adım zaten matematik dersinde de bu tarz etkinlikler hesaplamalar yaptıkları için zorlanmazlar seviyelerine uygun. Ama şunu tavsiye edeyim ben size düşündüğünüz şey güzel ama daha basitleştirilebilir mi? Zorlayabilir daha basit bir etkinlik çerçevesinde yapabilirseniz bence uygulaması daha kolay olur.”(Ö5)
				Döngüler	“Adım mutlaka gerekli ve mantıklı. Etkinlik öğrenci seviyesine göre bana biraz zor geldi. İlk başlarda çocuk zorlanırsa onda şöyle bir intiba olabiliyor ben yapamıyorum gibi olabiliyor dolayısıyla etkinlik daha basit olabilir.”(Ö7)
Strateji Oluşturma	8	2	-	Koşul Yapıları	“Öğrencilerin bireysel olarak altından kalkabilmeleri zor gibi görünüyor. Grup çalışması yapılırsa en azından kalkamayacak olanlar da oradan faydalanmış olur.” (Ö6) “Fikir çok güzel ama etkinlik belki biraz daha kısaltılarak basitleştirme yoluna gidilebilir. Bu haliyle biraz karmaşık görünüyor.”(Ö7)
Strateji Karşılaştırma	9	-	1	Döngüler	“Tahtaya yazma olayı değil de farklı bir şekilde daha pratik yazma veya aktarma olayı olursa 20dk fazla bile gelir ama tahtaya yazma işi geldiği için çok süre kaybı oluyor.”(Ö9)
Algoritmayı Oluşturma	5	-	5	Koşul Yapıları	“Kendi sınıflarımızı düşündüğümüz zaman sadece tahtanın önündeki alan mevcut buda yeterli olmayabilir. Böyle bir durumda sınıftan dışarı çıkma gibi bahçeyi veya koridoru kullanma gibi alternatif olursa daha rahat olur.”(Ö5)
				Döngüler	“Bilgisayarsız etkinliklerde sınıf ortamı tercih edilebilir”(Ö8)

				Değişkenler	<p>“Bu uygulama öncesinde çocuklara bir değişken oluşturup o değişkeni hesaplama içerisinde kullanma ile ilgili bir örnek yaptırabiliriz”(Ö1)</p> <p>“Böyle bir kodlamayı yapabilmesi için öncelikle çocuklara bu işin matematiği anlatılmalı. Değişkenler konusu ile ilgili bu şekilde bir uygulama ile başlamak yerine çok daha basit bir hesaplatma işlemi ile başlanabilir”(Ö3)</p> <p>“İlk verdiğimiz örnek çocuğun ne yaptığını kavrama noktasında sıkıntı olabilir. Çocuğun tek işlem yapacağı bir örnekle başlanabilir diye düşünüyorum”(Ö8)</p>
Algoritmayı Kodlama	2	8	-	Döngüler	<p>“Bilgisayar etkinliğine kadar olan süreçler çok güzel anlatılmış ancak bilgisayar ortamına geçişte ilk örnek olarak zor olmuş. Bilgisayar ortamındaki projelerde orta seviye olursa başlangıç için daha iyi olur diye düşünüyorum.”(Ö2)</p> <p>“Çocuğun keşfetmesini sağlayacak şekilde daha basit bir örnekle başlanabilir.”(Ö5)</p>
				Koşul Yapıları	<p>“Ben olsam daha basit bir şey gösteririm. Doğrudan, eğer böyleyse böyle değilse şöyle. Daha kısa bir örnekle başlayıp daha sonra zorlaştırmak daha sağlıklı olabilir. Bu ilk örnek için bana çok ağır geldi. Aynı örneğin daha basit hali de olabilir.”(Ö8)</p>
Farklı bir Koddaki Hatayı Belirleme ve Düzeltme	9	1	-	Koşul Yapıları	<p>“Kod düzenleme aşamasında öğrencilere verilecek uygulamaya geçmeden önce benzer kodların kullanıldığı bir yapı üzerinden anlatım olursa daha verimli olabilir.”(Ö4)</p>
Yeni Algoritmalar Hazırlama ve Kodlama	10	-	-		-

Tablo 6’da YAP modeli ile hazırlanan ders planlarının, tasarım değerlendirme ekibi tarafından bilişsel seviyeye uygunluk, süre ve kalabalık sınıflarda uygulanabilirlik ölçütlerine göre yapmış oldukları değerlendirmeler incelendiğinde, YAP modelinin adımları ağırlıklı olarak uygun bulunmuştur. Öğretmenlerin adımlara ilişkin değerlendirmeleri aşağıda başlıklar halinde verilmiştir.

“Problemi Tanıma ve Anlama”

Değişkenler- “Altıncı sınıf için uygun görüyorum yani gayet açıklayıcı cümleler var zaten problemde de, ifade olarak terim olarak anlamayacakları bir kelimedede yok gördüğüm kadarıyla bence anlaşılabilir seviyelerine uygun diye görüyorum.”(Ö9)

Koşul Yapıları- “Uygundur farklı bir konu dikkatlerini çeker diye düşünüyorum. Uygunluk açısından sorun yok biraz karmaşık gelebilir sadece. Resme bakıp anlamaları ya da konuyla ilgili yorum yapmaya çalışmaları hepsi karışabilir o yüzden adım adım düzenli gitmek gerekir.”(Ö4)

Döngüler- “Öğrenci seviyelerine göre uygun, eğlenceli dikkatlerini çekecek bir etkinliktir.”(Ö4)

“Strateji Oluşturma”

Değişkenler- “Uygundur çünkü zaten matematik dersinde bu tarz işlemler yapıyorlar sürekli. Altıncı sınıf seviyesindeki öğrenciler rahatlıkla algılayıp çözüme ulaştırabileceklerdir.”(Ö3)

Koşul Yapıları- “Öğrencilerin seviyelerine uygun olduğunu düşünüyorum. Sadece bir aktarım noktasından diğer aktarım noktasına giderken o aktarım noktalarını birazcık daha belirgin şekilde vurgulamak veya anlatırken resimde de iyi şekilde ortaya koymak gerektiğini düşünüyorum.”(Ö2)

Döngüler- “Araç geçişlerinin olması belki çocukları oyalayabilir onun dışında yapabilirler. En çabuk sürede hangi araçla gidecek bunu çocuklar seçer diye düşünüyorum. Altıncı sınıf öğrencilerimiz yapabilir.”(Ö9)

“Strateji Karşılaştırma”

Değişkenler- “15dk süre uygun bu madde için çocukların ortaya çıkardığı stratejilerin tartışılıp konuşulması uzun sürecek bir madde değil kendi stratejilerini birkaç dakika içinde ifade edebilirler.”(Ö7)

Koşul Yapıları- “Bu süre zaten bence izin versek iki derste konuşur. Çünkü o yaşta hep öyleler ama burada güzel olan diğerlerinin düşünemediği ya da grup halinde çalışıyorlarsa bir grubun düşünemediğini başka bir grup söyleyebiliyor.”(Ö4)

Döngüler- “Üçüncü madde için strateji incelemek, strateji belirledikten sonra onu incelemek için 20dk yeterli. Yeterli bir süre ayrılmış uygun.”(Ö10)

“Algoritmayı Oluşturma”

Tablo 7’ye göre 5 öğretmen “Algoritmayı Oluşturma” basamağını uygun olarak bulmuştur. 5 öğretmen ise “Algoritmayı Oluşturma” basamağında yer alan bazı drama oyunları için kalabalık sınıf ve fiziki alanın yetersiz olduğu sınıflarda uygulanabilirliğinin güç olduğu belirtilmiştir. Adıma ilişkin öğretmen görüşlerinden bazıları aşağıdaki gibidir:

Değişkenler- “Kalabalık sınıflarda pek sorun olacağını zannetmiyorum çünkü bu tarz etkinliklerde sınıf tamamen dikkatini oraya toplayabiliyor. Çok fazlada ciddiyetsizliğe yer vermeden denge sağlanırsa kalabalık sınıflarda uygulamada bir sorun olmaz.”(Ö7)

Koşul Yapıları- “Pratikte bu canlandırma bana çok zor gibi geliyor. Birincisi fiziki alan yaratmak ikincisi çocukların bu farklı kişilerle durak değiştirmesi ya da aktarma yapma mantığını anlaması zor gibi gözüküyor.”(Ö3)

Döngüler- “Kalabalık bir sınıfsa uygun ortam yaratmak sıkıntı oluyor. Kalabalık olduğunda arkadaki öğrenci etkinliği tam olarak takip edemeyecek. Bu tarz sorunlar olabilir.”(Ö4)

“Algoritmayı Kodlama”

Tablo 7’ye göre “Algoritmayı Kodlama” adımıyla uygulanacak bilgisayarlı etkinlikler 2 öğretmen tarafından uygun olarak görülürken, 8 öğretmen tarafından seviyeye uygunluk açısından güncellenebilir olduğu belirtilmiştir. Adıma ilişkin öğretmen görüşlerinden bazıları aşağıdaki gibidir:

Değişkenler- “Değişkenleri tanımladıktan sonra hani değişkenin ne olduğunu anladıktan sonra tanımlamayı yapabilirler ama değişkenlerle toplama, çarpma işte bunları başka bir değişkene aktarma işini yapamayabilirler.”(Ö1)

Koşul Yapıları- “Örnekler güzel ama ortaokul düzeyine göre örnekler mi daha üst düzey örnekler mi o biraz tartışılır. Daha basit olması daha iyi olacaktır. Örneklerin günlük hayatla ilişkilendirilmiş olması güzel.”(Ö7)

Döngüler- “Bu baya sene sonu projesi gibi bir şey. Genel olarak döndürme hareketini bile yapmak zor.”(Ö6)

Farklı bir Koddaki Hatayı Belirleme ve Düzeltme

Tablo 7’ye göre “Farklı bir Koddaki Hatayı Belirleme ve Düzeltme” adımıyla yapılacak bilgisayarlı etkinliği 8 öğretmen uygun olarak değerlendirirken, 2 öğretmen seviyeye uygunluk açısından güncellenebileceğini belirtmiştir. Adıma ilişkin öğretmen görüşlerinden bazıları aşağıdaki gibidir:

Değişkenler- “Hatayı düzeltmesi öğrenmesi için 10dk az gelebilir. Güzel bir etkinlik hatayı bulup düzeltmesi algoritmada fakat o hatayı tespit etmesi biraz uzun sürebilir. Düzeltmesi hızlı olabilir ama hatayı tespit edene kadar süre az kalabilir.”(Ö7)

Koşul Yapıları- “Öğrenciler çözümlenebilir basit bir uygulama. Bozuk kod yapısını düzenleyebileceklerini düşünüyorum. Öğrenci seviyesine uygun.”(Ö6)

Döngüler- “Yapılabilir gayet güzel. Böyle iyidir hepsinin çünkü bir hatası var hem kedinin hem köpeğin kontrol edecek onları zaten adım adım gittiği zaman sıkıntı kalmayacak adım adım okuduğu zaman hatayı da görebilecek.”(Ö9)

TDE ağırlıklı olarak YAP modelinin dördüncü ve beşinci adımı üzerinde önerilerde bulunmuştur. Ayrıca YAP modeli adımlarını uygun ve gerekli görmekle beraber, bu adımlardaki etkinliklerin uygulanmasının daha verimli olması açısından yapılabilecek güncellemeleri belirtmişlerdir.

YAP modeline göre hazırlanacak ders planlarında TDE görüşlerine göre dikkat edilmesi gerekenler Tablo 7’de gösterilmektedir.

Tablo 7. Planlarda Dikkat Edilmesi Gerekenler

YAP modeli adımları	TDE Önerileri
Problemi Tanıma ve Anlama	Ana etkinliğe geçmeden önce konu ile ilgili daha basit bir etkinlik ya da ana etkinliğin daha basit bir bölümüyle başlanmalı
Strateji Oluşturma	-
Strateji Karşılaştırma	Bu aşamada öğrencilerin oluşturdukları farklı stratejileri, adımları algoritma şeklinde ifade etmede yardımcı olması amacıyla öğretmen tahtaya yazmalı. Bu şekilde öğrencilere kendi stratejilerini geliştirebilmeleri ve diğer öğrencilerin stratejileri ile karşılaştırmaları için daha fazla süre verilirken yazma işlemi için daha az süre harcanmış olur.
Algoritmayı Oluşturma	-
Algoritmayı Kodlama	Oluşturulan algoritmanın tamamı kodlanmadan parçalara bölünerek basitten karmaşığa doğru adım adım ilerlenmeli. Dersin konusu olmayan ve öğrencinin kafasını karıştırıcı komutlar (örneğin karakteri hareket ettirme ya da kostüm değiştirme vb.) öğrencilere hazır olarak verilmeli. Böylelikle öğrencilerin dikkatini dağıtmadan, yalnızca o dersin konusuna odaklanması sağlanmalı.
Farklı bir Koddaki Hatayı Belirleme ve Düzeltme	Bu basamakta öğrencilerin yapmaları gereken işlem gereği, kodları analiz edebilmeleri için olabildiğince fazla süre tanınmalı.
Yeni Algoritmalar Hazırlama ve Kodlama	-

TDE öğretmenleri YAP modeli adımlarını ders planı hazırlama ve uygulamaya uygun olarak değerlendirmiştir. Tablo 7’de belirtilen öneriler YAP modeline göre hazırlanacak ders planlarına yöneliktir. Bu önerilerde özellikle; gerçekleştirilecek etkinliklerdeki problem durumunun daha küçük parçalara ayrılarak basitten karmaşığa doğru aşama aşama ilerlemesi gerektiği ve öğrencilerin kendi öğrenmelerini gerçekleştirebilmeleri amacıyla olabildiğince fazla zaman verilmesi gerektiği vurgulanmaktadır

Tartışma ve Sonuç

Çalışmada öncelikle TE tarafından, programlama öğretiminde kullanılacak adımlar doküman analiziyle belirlenmiştir. Daha sonra bu adımların değerlendirilmesi amacıyla hazırlanan ders planlarını, TDE ekibi incelemiş ve görüşlerini belirtmiştir. Çalışmada TE’nin doküman analizi ve TDE’nin görüşleri doğrultusunda ortaokul seviyesinde öğrencilerin algoritmik düşünme, problem çözme ve programlama becerilerinin gelişimine yönelik olarak YAP modeli adımları oluşturulmuştur. Modelin adımları hem bireysel hem de grup çalışmalarına uygun olacak şekilde hazırlanmıştır.

Bu adımlar:

1. Problemi tanıma ve anlama
2. Strateji oluşturma
3. Strateji karşılaştırma
4. Algoritmayı oluşturma
5. Algoritmayı kodlama
6. Farklı bir koddaki hatayı belirleme ve düzeltme
7. Yeni algoritmalar hazırlama ve kodlama şeklindedir.

Problemi Tanıma ve Anlama: Bu aşama; öğrencilerin problemde verilenleri ve isteneni, çözüm sürecindeki yerini belirleyerek anlamasını gerektirir. Bu nedenle öğrencilerin soyutlama ve analogi yeteneklerini geliştirmeye ihtiyaç vardır. Soyutlama, öğrencinin problem içindeki gereksiz ve önemsiz bilgileri ayırt edebilmesidir. Analogi (Benzetim) ise bir kavramın öğretiminde öğrencilerin bildiği bir başka kavramla eşleştirmektir. Soyutlamayı öğrencilerin, analogiyi ise öğretmenlerin gerçekleştirmesi gerekmektedir. Bu aşama sınıf ortamında ve bilgisayarsız olarak gerçekleştirilir.

Strateji Oluşturma: Problemin çözümü için gerekli adımların belirlenmesi aşamasıdır. Öğrenciler bu aşamada farklı stratejiler düşünerek ve deneyerek problemin çözümü için uygun yolu belirlemeye çalışırlar. Önceki adımın devamı olarak sınıf ortamında ve bilgisayarsız olarak gerçekleştirilir.

Strateji Karşılaştırma: Öğrenciler bir önceki adımda belirledikleri stratejilerini sınıftaki diğer öğrencilerin çözümleriyle ve uygunluğunu değerlendirilir. Strateji karşılaştırma aşaması, farklı çözüm stratejilerinin karşılaştırılarak adımların; neden kullanıldığının, çözüm sürecindeki yerinin ve birbirleriyle ilişkisinin anlaşılmasını içerir. Öğrencilerin yalnızca kendi stratejilerini değil diğer öğrencilerin stratejilerini de anlayabilmesi gerekir. Öğrencilerin çözüm için atılan adımların gerekliliğini, sonraki adımlarla olan ilişkisini ve sonuçlarını anlaması bu adımda gerçekleşir. Bu çözüm için oluşturulan strateji adımlarının sırası belirlenir. Önceki adımın devamı olarak sınıf ortamında ve bilgisayarsız olarak gerçekleştirilir. Bu konuda detaylı bilgiye Şahin (2018)'den ulaşılabilir.

Algoritmayı Oluşturma: Bu aşama 2 alt adımdan oluşmaktadır.

i. Algoritmayı yazma: Öğrencilerin bir önceki adımda karşılaştırarak belirledikleri çözüm stratejisini adım adım yazmaları sağlanmalıdır. Öğrencilerin çözüm için kendi ifadelerini kullanmaları çözümü kavrayabilmelerini destekleyebilir.

ii. Algoritmayı oynama (Drama Etkinliği): Analogik yapıda bir problem, öğrencilerin günlük yaşantısıyla ilgili olacağından drama ile canlandırılması sağlanabilir. Bu durum farklı zeka tiplerine sahip öğrencilerin sürece dahil edilmesini ve öğrencilerin motivasyonlarının arttırılmasını sağlayabilir. Bu nedenle öğrencilerin kendilerini ifade etmeleri açısından drama etkinliği önemli görülmektedir.

Bu aşamalar önceki adımın devamı olarak sınıf ortamında ve bilgisayarsız olarak gerçekleştirilir. Bunan sonraki aşamalar ise bilgisayarlı süreçleri kapsamaktadır.

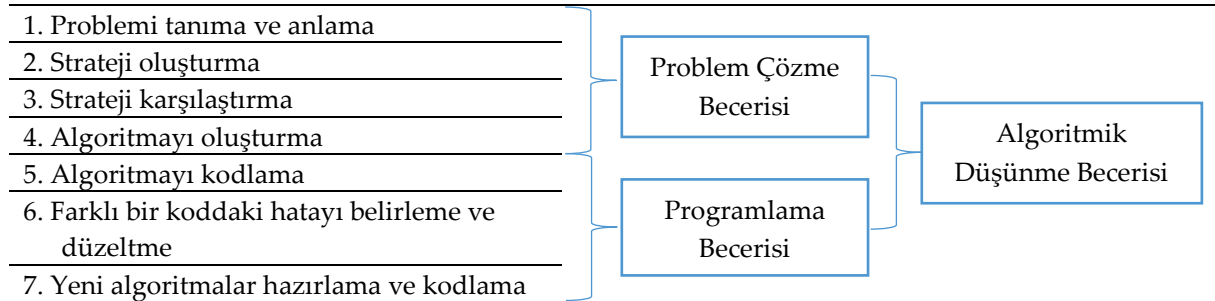
Algoritmayı Kodlama: Bu aşamada öğrencilerin oluşturdukları algoritmayı bilgisayar ortamında kodlaması gerekmektedir.

Kodlama öğrenciler tarafından zor görülen ve üst düzey düşünme becerileri gerektiren bir görevdir (Jenkins, 2002; Kinnunen ve Malmi, 2008). Bir algoritmanın kodlanması sırasında kullanılan her bir kod satırının kendinden önceki ve sonraki satırdaki kodla ilişkisinin ve tüm program içerisindeki görevinin anlaşılması gerekir. Yani kullanılan her bir kod satırı, kullanım nedeni ve programın tamamı için oluşturacağı sonuç açısından değerlendirilmelidir (Bayman ve Mayer, 1988).

Farklı Bir Koddaki Hatayı Belirleme ve Düzeltme: Farklı bir kodlamadaki sorunu anlamak ve düzeltmek, kendi algoritmamızı kodlamaktan daha üst beceri gerektirir. Bunun için o kodları yazan kişinin düşünme şeklini anlamak gerekmektedir. Farklı bir etkinlik olarak bilgisayar başında gerçekleştirilir.

Yeni Algoritmalar Hazırlama ve Kodlama: Bu aşamada öğrencilere program yazmaları için kullanmaları gereken asgari komutları belli ancak kendilerinin belirleyecekleri bir problem durumunu kodlamaları istenir. Bu şekilde öğrencilerin önceki süreçleri kendilerinin yeniden tekrar ederek öğrenmelerinin ve gelişimlerinin farkına varabilmeleri amaçlanmaktadır. Asgari komutların belirlenmesi öğrencilerin gereğinden kolay etkinlikler tasarlamalarını engellemek amacıyla. Bu aşamada öğretmen öğrencilerinin seviyelerine göre uygun görmesi durumunda sınırları belirlenmemiş ve genel ifadelerle ortaya konulmuş problemlerin çözümünü de öğrencilerinden isteyebilir. Öğretmen bu aşamadaki etkinliği ders dışı etkinlik olarak verebilir.

Belirtilen adımlarda en önemli noktalardan biri öğrencilere analogilere uygun problemlerin verilmesidir. Ayrıca öğrencilerin strateji oluşturup bu stratejileri yazarak, sınıf ortamında tartışarak ve drama etkinliğiyle oynayarak, sosyal becerilerinin geliştirilebileceği düşünülmektedir. Böylece öğrencilerde farklı bilişsel süreçlerin gelişimi de tetiklenebilir. Bununla beraber bu çalışmada temel olarak hedeflenen beceriler ve YAP adımlarının ilişkisi şekil 4’de gösterildiği gibidir.



Şekil 4. Yap Yöntemi Adımları ve Bilişsel Beceriler

Şekil 4’de belirtildiği gibi 7 adımdan oluşan YAP modeli adımlarının ilk dört adımı problem çözme becerisini geliştirmeye yönelik adımları kapsarken diğer üç adım ise programlama becerilerini geliştirmeye yönelik adımları kapsadığından dolayı her üç becerinin de gelişimi hedeflenmektedir. YAP modeli adımlarının belirtilen üç bilişsel beceri ile karşılaştırılması Tablo 8’de gösterildiği şekildedir.

Tablo 8. YAP Yönteminin Algoritmik Düşünme, Problem Çözme ve Programlama Seviyeleri İle İlişkisi

YAP Yöntemi	Algoritmik Düşünme Becerisi Alt Boyutları (Futschek, 2006)	Problem Çözme becerisi adımları (George Polya, 1957)	Programlama Seviyeleri
1. Problemi tanıma ve anlama	Verilen problemi analiz edebilme becerisi Problemi tam olarak anlama(soyutlama) ve ifade edebilme becerisi	1- Problemin anlaşılması	
2. Strateji oluşturma	Verilen probleme uygun temel stratejileri üretme becerisi	2- Çözüm ile ilgili stratejinin belirlenmesi 3- Stratejinin uygulanması	
3. Strateji karşılaştırma	Bir problemin olası tüm özel ve normal durumlarını düşünme becerisi	4- Çözümün değerlendirilmesi	
4. Algoritmayı oluşturma	Belirli bir probleme uygun stratejileri kullanarak doğru bir algoritma oluşturabilme becerisi		
5. Algoritmayı kodlama	Bir algoritmanın verimliliğini artırma becerisi		Temel seviye
6. Farklı bir koddaki hatayı belirleme ve düzeltme			Orta seviye
7. Yeni algoritmalar hazırlama ve kodlama			İleri seviye

Tablo 8'e bakıldığında, YAP modelinin ilk adımı, algoritmik düşünme becerilerinin alt boyutlarında yer alan "Verilen problemi analiz edebilme becerisi" ve "Problemi tam olarak anlama(soyutlama) ve ifade edebilme becerisi" adımları ile ilişkilidir. Aynı zamanda Polya'nın problem çözme adımlarının ilki olan "Problemin anlaşılması" adımı ile ilişkilidir. Belirtilen adımlarda yapılan işlemlerde; verilen herhangi bir problemi çözüme kavuşturmak için problem içeriği analiz edilerek verilen ve istenen veriler tespit edilir, çözüm için gerekli olmayan veriler ayrıştırılır, bu şekilde elde edilen verileri ifade etmeyi kapsamaktadır.

YAP modelinin ikinci adımı algoritmik düşünme becerilerinin alt boyutlarında yer alan "Verilen probleme uygun temel stratejileri üretme becerisi" ve Polya'nın problem çözme adımlarından "Çözüm ile ilgili stratejinin belirlenmesi" ve "Stratejinin uygulanması" adımları ile ilişkilidir. Belirtilen adımlar bir problemi çözmek için strateji geliştirmeyi ve geliştirilen bu stratejiyi uygulama sürecini kapsamaktadır.

YAP modelinin üçüncü adımı algoritmik düşünme becerilerinin alt boyutlarından biri olan "Bir problemin olası tüm özel ve normal durumlarını düşünme becerisi" ve Polya'nın problem çözme adımlarından "Çözümün değerlendirilmesi" adımı ile ilişkilidir. Belirtilen adımlarda, geliştirilen stratejiler ve bu stratejilerin uygulanması ile ulaşılan sonuçlar tartışılarak değerlendirilir. Problemin çözümü için elde edilen stratejilerden hangisinin en iyi olduğu belirlenir.

YAP modelinin dördüncü adımı algoritmik düşünme becerilerinin alt boyutlarında yer alan "Belirli bir probleme uygun stratejileri kullanarak doğru bir algoritma oluşturabilme becerisi" ile ilişkilidir. Belirtilen adımlarda, problemin çözümü için ortaya konulan en iyi strateji doğrultusunda problem çözümünün algoritmasının yazılması işlemi kapsamaktadır.

YAP modelinin beşinci ve altıncı adımları algoritmik düşünme becerilerinin alt boyutunda yer alan "Bir algoritmanın verimliliğini artırma becerisi" ile ilişkilidir. Belirtilen adımlarda algoritmanın uygulanmasıyla performansını arttırmak için geliştirmeler yapılabilmektedir. YAP modelinin beşinci adımı programlama seviyesi temel düzeyde olan öğrencileri kapsamaktadır. Programlama seviyesi bu düzeyde olan öğrenciler problemin çözümü için geliştirilen algoritmaya bağlı kalarak kodlama yaparlar (Weiser ve Shertz, 1983). Benzer şekilde YAP modelinin beşinci adımında kodlama yapılırken dördüncü adımda oluşturulan algoritmaya bağlı kalınmaktadır.

YAP modelinin altıncı adımı programlama seviyesi orta düzeyde olan bireyleri kapsamaktadır. Alanyazında programlama sürecinde gerçekleştirilen 5 işlem;

- 1- Problemin çözümü için gereksinimleri belirleme, probleme en uygun olacak şekilde çözümü geliştirme, bu işlemlerin tasarımını zihinsel olarak oluşturabilme
- 2- Daha önceden farklı kişiler tarafından yazılan bir algoritmayı ya da programı anlayabilme
- 3- Problemi çözüme ulaştırmak için bilgisayar ortamında kodlama işlemi yapabilme
- 4- Bir programdaki işlevsel bozukluğu tespit edebilme (Hata belirleme)
- 5- Belirli bir programda ihtiyaca göre gerekli değişikliği yapabilme (Geliştirme) (Shneiderman, 1976; Koubek, Salvendy, Dunsmore ve Lebold, 1989) şeklinde belirtilmektedir.

Programlamada öncelikle problem için en uygun çözümü geliştirebilmek, daha sonra yazılan bir programı kavramaktır. Bir program üzerinde meydana gelen işlev bozukluğunu tespit edip geliştirebilmek için programı anlamak gerekmektedir. Üçüncü madde problemi çözüme ulaştıracak kodlamayı yapma işlemidir. Dördüncü ve beşinci adımda program üzerinde meydana gelen hataları tespit ederek gerekli güncellemeyi yapabilme işlemidir. Feddon ve Charness (1999), programlama seviyesi farklı düzeyde olan kişiler ile programlama sürecinde gerçekleştirilen işlemler arasındaki ilişkiyi belirlemek için yapmış olduğu çalışmada, programlama seviyesi orta düzeyde olan kişiler ile bir programdaki hatayı tespit edebilme (dördüncü işlem) ve geliştirebilme (beşinci işlem) arasında anlamlı bir ilişki bulmuştur. Bu nedenle çalışmada öğrencilerin programlama becerilerinin gelişmesi için hata belirleme ve düzeltme çalışmalarının yapılması gerektiği belirtilmiştir. Benzer şekilde Masuck, Alves-

Foss ve Oman (2008) ise kodlama becerisine sahip olup programdaki hatayı belirleme ve düzeltme becerisine sahip olmayan öđrencilerin programlama sürecinin önemli bir aşamasından eksik kaldığını belirtmişlerdir.

YAP modelinin yedinci adımı programlama becerisi ileri düzeyde olan öđrencileri kapsamaktadır. Çünkü bu düzeydeki öđrenciler herhangi bir problem durumu için gerekli olan algoritmayı geliştirebilir ve bunu bilgisayar ortamında kodlayabilirler. Weiser ve Shertz (1983) programlama seviyesi ileri düzeyde olan bireylerin bir problemin çözümü için gerekli olan algoritmayı yazıp daha sonra bunu kodlayabildiklerini belirtmiştir.

YAP modeline göre geliştirilen ders planlarında öđretmen ve öđrencilerden beklenen görevler belirtilmiş ve ilgili yönergelerle desteklenmiştir. Aynı zamanda öđretmenlerin derslerinde kullanabileceđi formlar ve örnekler verilmiştir (Şahin, 2018). Ders planlarında yer alan etkinlikler öđretmenler için rehber niteliğinde ve zorluk seviyesi düzenlenebilecek şekilde geliştirilmiştir. Öđretmenler öđrencilerin bilişsel seviyeleri, ders süresi ve sınıfın fiziki imkânlarına göre etkinliklerde gerekli güncellemeleri yapabilirler. Bu çalışma kapsamında TDE öđretmenlerine sunulan ders planlarına genel bir bakış ve kullanımına ilişkin bilgiler Ek-3'de verilmiştir.

Sonuç olarak bu çalışmada eğitim-öđretim kurumlarında; öđrencilerin algoritmik düşünme, problem çözme ve programlama becerilerini geliştirmek amacıyla hazırlanan YAP (Yedi Adımda Programlama) modelinin oluşturulma süreci, adımları ve öđretmen görüşleri verilmiştir. Hazırlanan modelin sınıf ortamındaki uygulamalarda olumlu sonuçlar verebileceđi düşünülmektedir.

Öneriler

- YAP modeline göre oluşturulan ders planları farklı başarı seviyesindeki öđrencilere uygulanarak modelin etkililiđi değerlendirilebilir.
- YAP modeline göre oluşturulan ders planları farklı eğitim kademelerinde uygulanarak modelin etkililiđi değerlendirilebilir.
- YAP modeline göre oluşturulan ders planlarının uygulanmasının öđrenciler üzerindeki algoritmik düşünme, problem çözme ve programlama becerilerine etkisi incelenebilir.

Kaynakça

- Akpınar, Y. ve Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *İlköğretim Online*, 13(1), 1-4.
- Ala-Mutka, K. (2004). *Problems in learning and teaching programming*. https://www.cs.tut.fi/~edge/literature_study.pdf adresinden erişildi.
- Atmatzidou, S. ve Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661-670.
- Balanskat, A. ve Engelhardt, K. (2015). *Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across europe*. http://fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1-e1a6-4bf1-8105-ea27c0d2bbe0 adresinden erişildi.
- Bayman, P. ve Mayer, R. E. (1988). Using conceptual models to teach BASIC computer programming. *Journal of Educational Psychology*, 80(3), 291-298.
- Bergersen, G. R. ve Gustafsson, J. E. (2011). Programming skill, knowledge, and working memory among professional software developers from an investment theory perspective. *Journal of Individual Differences*, 32(4), 201-209.
- Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E. ve Fontecchio, A. (2013). *Computer aided instruction as a vehicle for problem solving: Scratch programming environment in the middle years classroom*. http://www.pages.drexel.edu/~dmk25/ASEE_08.pdf adresinden erişildi.
- Byrne, P. ve Lyons, G. (2001). The effect of student attributes on success in programming. *ACM SIGCSE Bulletin*, 33(3), 49-52.
- Choi, J. Lee, Y. ve Lee, E. (2016). Puzzle based algorithm learning for cultivating computational thinking. *Wireless Personal Communications*, 93(1), 131-145. doi: 10.1007/s11277-016-3679-9
- Clements, D. ve Sarama, J. (2003). Strip mining for gold: research and policy in educational technology a response to "fool's gold". *AACE Journal*, 11(1), 7-69. https://www.learntechlib.org/index.cfm/files/paper_17793.pdf?fuseaction=Reader.DownloadFullText&paper_id=17793%C2%A0 adresinden erişildi.
- Committee on Logic Education. (2008). *Algorithmic thinking*. <http://www.ucalgary.ca/aslcle/nctm/Q2A.html> adresinden erişildi.
- Crescenzi, P., Malizia, A., Verri, M. C., Diaz, P. ve Aedo, I. (2012). Integrating algorithm visualization video into a first-year algorithm and data structure course. *Educational Technology & Society*, 15(2), 115-124.
- Creswell, J. W. (2013). Steps in conducting a scholarly mixed methods study. *DBER Speaker Series*, 48. <http://digitalcommons.unl.edu/dberspeakers/48> adresinden erişildi.
- Coull, N. J. ve Duncan, I. M. (2011). Emergent requirements for supporting introductory programming. *Innovation in Teaching and Learning in Information and Computer Sciences*, 10(1), 78-85.
- Çıray, F. ve Erişti, B. (2014). Disiplinlerarası analogi tabanlı öğretimin farklı düzeylerde akademik başarılı ilköğretim öğrencilerinin fen ve teknoloji dersi öğrenme düzeyleri üzerindeki etkisi. *İlköğretim Online*, 13(3), 1049-1064.
- Fesakis, G. ve Serafeim, K. (2009). Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education. *ACM SIGCSE Bulletin*, 41(3), 258-262.

- Feddon, J. S. ve Charness, N. (1999). Component relationships depend on skill in programming. *11th Annual PPIG Workshop, University of Leeds, UK*, 1-11. <https://pdfs.semanticscholar.org/dccf/2f3cd095192abd0b5c624c5dda7948f5826b.pdf> adresinden erişildi.
- Futschek, G. (2006). Algorithmic thinking: the key for understanding computer science. R. T. Mittermeir (Ed.), *Informatics Education – The Bridge between Using and Understanding Computers* içinde (C. 4226, s. 159-168). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/11915355_15
- Garner, S. (2003). Learning resources and tools to aid novices learn programming. *Informing science & information technology education joint conference (INSITE)* içinde (s. 213-222). <https://pdfs.semanticscholar.org/21a6/68fb94878b040e4bffb0858d15896cbbdb8.pdf> adresinden erişildi.
- Ginat, D. (2004). On novice loop boundaries and range conceptions. *Computer Science Education*, 14(3), 165-181.
- Grover, S. ve Pea, R. D. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43.
- Helminen, J. ve Malmi, L. (2010). Jype - a program visualization and programming exercise tool for Python. *Proceedings of the 5th international symposium on Software visualization - SOFTVIS '10* içinde (s. 153). Salt Lake City, Utah, USA: ACM Press. <https://doi.org/10.1145/1879211.1879234>
- ISTE. (2016). *The ISTE National Educational Technology Standards (NETS) and Performance Indicators for Students*. <http://www.iste.org/standards/nets-for-students> adresinden erişildi.
- Jenkins, T. (2002, September). On the difficulty of learning to program. *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences* içinde. Leeds.
- Kam, H. J. ve Katerattanakul, P. (2014). Structural model of team-based learning using Web 2.0 collaborative software. *Computers & Education*, 76, 1-12.
- Karaman, S., Yıldırım, S. ve Kaban, A. (2008). Öğrenme 2.0 yaygınlaşıyor: Web 2.0 uygulamalarının eğitimde kullanımına ilişkin araştırmalar ve sonuçları. *XIII. Türkiye’de İnternet Konferansı Bildirileri* içinde (s. 35-40). Orta Doğu Teknik Üniversitesi, Ankara.
- Kalelioglu, F. ve Gülbahar, Y. (2014). The effects of teaching programming via Scratch on problem solving skills: a discussion from learners' perspective. *Informatics in Education*, 13(1), 33-50
- Kafai, Y. B. ve Q. Burke. (2014). *Connected code: Why children need to learn programming*. MIT Press.
- Kinnunen, P. ve Malmi, L. (2008, September). *CS minors in a CS1 course*. Paper presented at the Fourth international Workshop on Computing Education Research, Sydney.
- Kelleher, C., Pausch, R. ve Kiesler, S. (2007). Storytelling alice motivates middle school girls to learn computer programming. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM*, 1455-1464.
- Klopfer, E. ve Yoon, S. (2005). Developing games and simulations for today and tomorrow’s tech savvy youth. *TechTrends*, 49(3), 33-41.
- Kobsiripat, W. (2015). Effects of the media to promote the scratch programming capabilities creativity of elementary school students. *Procedia-Social and Behavioral Sciences*, 174, 227- 232.
- Koubek, R. J., Salvendy, G., Dunsmore, H. E. ve LeBold, W. K. (1989). Cognitive issues in the process of software development: review and reappraisal. *International Journal of Man--Machine Studies*, 30, 171-191.

- Kurland, D. M., Pea, R. D., Clement, C. ve Mawby, R. (1989). A study of the development of programming ability and thinking skills in high school students. *Journal of Educational Computing Research*, 2(4), 429-458.
- Kukul, V. ve Gökçearslan, Ş. (2014). *Scratch ile programlama eğitimi alan öğrencilerin problem çözme becerilerinin incelenmesi*. 8. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu'nda sunulan bildiri, Trakya Üniversitesi Bilgisayar Öğretmenliği Bölümü, Edirne.
- Law, K. M., Lee, V. C. ve Yu, Y. T. (2010). Learning motivation in e-learning facilitated computer programming courses. *Computers & Education*, 55(1), 218-228.
- Lahtinen, E., Ala-Mutka, K. ve Jarvinen, H. (2005) A Study of Difficulties of Novice Programmers. *In Acm Sigcse Bulletin, ACM*, 37(3), 14-18.
- Lai, A. F. ve Yang, S. M. (2011). The learning effect of visualized programming learning on 6th graders' problem solving and logical reasoning abilities. *2011 International Conference on Electrical and Control Engineering* içinde (s. 6940-6944). Yichang, China: IEEE. <https://doi.org/10.1109/ICECENG.2011.6056908>
- Lai, C. S. ve Lai, M. H. (2012). Using computer programming to enhance science learning for 5th graders in taipei. *2012 International Symposium on Computer, Consumer and Control* içinde (s. 146-148). Taichung, Taiwan: IEEE. <https://doi.org/10.1109/IS3C.2012.45>
- Lee, Y. J. (2011). Empowering teachers to create educational software: A constructivist approach utilizing Etoys, pair programming and cognitive apprenticeship. *Computers & Education*, 56(2), 527-538.
- Linn, M. C. ve Dalbey, J. (1989). Cognitive consequences of programming instruction. E. Soloway & J.C. Spohrer (Ed.), *Studying the novice programmer* içinde (s. 57-81). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Malan, D. J. ve Leitner, H. H. (2007). Scratch for budding computer scientists. *ACM SIGCSE Bulletin*, 39(1), 223-227.
- Masuck, C., Alves-Foss, J. ve Oman, P. (2008). Analysis of fault models for student use. *ACM SIGCSE Bulletin*, 40(2), 79-83.
- Merriam, S. B. ve Tisdell, E. J. (2015). *Qualitative research: A guide to design and implementation*. John Wiley & Sons.
- Milli Eğitim Bakanlığı. (2017). *Bilişim teknolojileri ve yazılım dergisi öğretim programı, 2016-2017*. Ankara: Milli Eğitim Basımevi.
- Naharro-Berrocal, F., Pareja-Flores, C., Urquiza-Fuentes, J. ve Velázquez-Iturbide, J. á. (2002). Approaches to comprehension-preserving graphical reduction of program visualizations. *Proceedings of the 2002 ACM symposium on Applied computing - SAC '02* içinde (s. 771). Madrid, Spain: ACM Press. <https://doi.org/10.1145/508791.508941>
- Oluk, A. ve Saltan, F. (2015). Effects of using the scratch program in 6th grade information technologies courses on algorithm development and problem solving skills [Special issue]. *Participatory Educational Research*, 10-20.
- Özmen, B. ve Altun, A. (2014). "Undergraduate students' experiences in programming: difficulties and obstacles." *Turkish Online Journal of Qualitative Inquiry*, 5(3), 1-27.
- Porter, R. ve Calder, P. (2004). Patterns in learning to program: an experiment? *Proceedings of the Sixth Australasian Conference on Computing Education - Volume 30* içinde (s. 241-246). Darlinghurst, Australia, Australia: Australian Computer Society, Inc.
- Polya, G. (1957). *How to solve It?* (2. bs.). Princeton, N.J.: Princeton University Press.

- Robins, A., Rountree, J. ve Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137-172.
- Sanford, J. F. ve Naidu, J. T. (2016). Computational thinking concepts for grade school. *Contemporary Issues in Education Research (Online)*, 9(1), 23.
- Sález-López, J. M., Román-González, M. ve Vázquez-Cano, E. (2016). "Visual programming languages integrated across the curriculum in elementary school: A two year case study using "scratch" in five schools.", *Computer & Education*, 97, 129-141.
- Scaffidi, C. ve Chambers, C. (2012). Skill progression demonstrated by users in the Scratch animation environment. *International Journal of Human-Computer Interaction*, 28(6), 383-398.
- Schwartz, J., Stagner, J. ve Morrison, W. (2006). Kid's programming language (Kpl). *ACM SIGGRAPH 2006 Educators program on - SIGGRAPH '06* içinde (s. 52). Boston, Massachusetts: ACM Press. <https://doi.org/10.1145/1179295.1179348>.
- Seppälä, O., Malmi, L. ve Korhonen, A. (2006). Observations on student misconceptions—A case study of the Build-Heap Algorithm. *Computer Science Education*, 16(3), 241-255.
- Shneiderman, B. (1976). Exploratory experiments in programmer behavior. *International Journal of Computer and Information Sciences*, 5, 123-143.
- Shih, I. J. (2014). *The effect of scratch programming on the seventh graders' mathematics abilities and problem solving attitudes* (Yayımlanmamış yüksek lisans tezi). Taipei University, Taiwan.
- Silverman, D. (2013). *Doing qualitative research: A practical handbook*. London: SAGE Publications.
- Spohrer, J. C. ve Soloway, E. (1989). Simulating student programmers. *Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 1* içinde (s. 543-549). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Stolee, K. T. ve Fristoe, T. (2011). Expressing computer science concepts through Kodu game lab. *Proceedings of the 42nd ACM technical symposium on Computer science education - SIGCSE '11* içinde (s. 99). Dallas, TX, USA: ACM Press. <https://doi.org/10.1145/1953163.1953197>
- Şahin, G. (2018). *Ortaokul seviyesinde programlama öğretimi için bir yöntem önerisi* (Yayımlanmamış yüksek lisans tezi). Karadeniz Teknik Üniversitesi, Eğitim Bilimleri Enstitüsü, Trabzon.
- Trilling, B. ve Fadel, C. (2012). *21st century skills: learning for life in our times* (1 bs.). San Francisco, Calif: Jossey-Bass.
- Utting, I., Cooper, S., Kölling, M., Maloney, J. ve Resnick, M. (2010). Alice, greenfoot, and scratch - a discussion. *ACM Transactions on Computing Education*, 10(4), 1-11.
- Vasconcelos, J. (2007). *Basic Strategy for Algorithmic Problem Solving*. <http://www.cs.jhu.edu/~jorgev/cs106/ProblemSolving.htm> adresinden erişildi.
- Van Gorp, M. J. ve Grissom, S. (2001). An empirical evaluation of using constructive classroom activities to teach introductory programming. *Computer Science Education*, 11(3), 247-260.
- Weiser, M. ve Shertz, J. (1983). Programming problem representation in novice and expert programmers. *International Journal of Man-Machine Studies*, 19(4), 391-398.
- Winslow, L. E. (1996). Programming pedagogy-a psychological overview. *ACM Sigcse Bulletin*, 28(3), 17-22.
- Williams, L., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In support of pair programming in the introductory computer science course. *Computer Science Education*, 12(3), 197-212.

- Wilson, A. ve Moffat, D.C. (2010). *Evaluating Scratch to introduce younger school children to programming*. <http://scratched.gse.harvard.edu/sites/default/files/wilson-moffat-ppig2010-final.pdf> adresinden erişildi.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wu, W. Y., Chang, C. K. ve He, Y. Y. (2010). Using Scratch as game-based learning tool to reduce learning anxiety in programming course. Global Learn sunulan sözlü bildiri, Association for the Advancement of Computing in Education (AACE).
- Yuretich, R. F., Khan, S. A., Leckie, R. M. ve Clement, J. J. (2001). Active-learning methods to improve student performance and scientific interest in a large introductory oceanography course. *Journal of Geoscience Education*, 49(2), 111-119.
- Yükseltürk, E, Altok, S. ve Üçgöl, M. (2016). *Oyun programlamanın ilköğretim öğrencilerinin problem çözme becerilerine etkileri: Bir yaz kampı deneyimleri*. 4. Uluslararası Öğretim Teknolojileri ve Öğretmen Eğitimi Sempozyumu'nda sunulan sözlü bildiri, Fırat Üniversitesi Bilgisayar ve Öğretim Teknolojileri Bölümü, Elazığ.
- Yıldırım, A. ve Şimşek, H. (2011). *Sosyal bilimlerde nitel araştırma yöntemleri*. İstanbul: Seçkin Yayınları.
- Yin, R. K. (2017). *Case study research and applications: Design and methods*. Los Angeles: SAGE.
- Zsakó, L. ve Szlávi, P. (2012). ICT competences: algorithmic thinking. *Acta Didactica Napocensia*, 5(2), 49-58.
- Ziatdinov, R. ve Musa, S. (2012). Rapid mental computation system as a tool for algorithmic thinking of elementary school students development. *European researcher, Series A*, (7), 1105-1110.

Ek 1. Ders Planı Deđerlendirme Ölüütleri Formu

Görev yaptığınız okulun adı:
Branşınız:
Öđretmenlik deneyiminiz:

1. Ders planı hazırlarken dikkat edilmesi gereken noktalar nelerdir?

Ek 2. Bilişim Teknolojileri ve Yazılım Dersi Öđretmenleri Görüşme Soruları

1. Ortaokul 6. Sınıf öđrencileri için programlama öđretimine yönelik hazırlanan ders planında yer alan etkinliđin, öđrencilerin seviyelerine uygunluk durumu nedir?
2. Ortaokul 6. Sınıf öđrencileri için programlama öđretimine yönelik hazırlanan ders planında yer alan bilgisayar uygulamalarının, öđrencilerin seviyelerine uygunluk durumu nedir?
3. Ortaokul 6. Sınıf öđrencileri için programlama öđretimine yönelik hazırlanan ders planında yer alan etkinliklere ayrılan süre yeterli midir? Neden?
4. Ortaokul 6. Sınıf öđrencileri için programlama öđretimine yönelik hazırlanan ders planında bilgisayar uygulamalarına ayrılan süre yeterli midir? Neden?
5. Ortaokul 6. Sınıf öđrencileri için programlama öđretimine yönelik hazırlanan ders planında yer alan etkinliklerin kalabalık sınıflarda uygulanabilirlik durumu nedir?
6. Ortaokul 6. Sınıf öđrencileri için programlama öđretimine yönelik hazırlanan ders planında yer alan bilgisayar uygulamalarının kalabalık sınıflarda uygulanabilirlik durumu nedir?

Ek 3. Ders Planlarına Genel Bakış ve Kullanımı

Tablo 9’da ders planları; konular, kazanımlar ve etkinlikler boyutunda özetlenerek, ders planlarının 7 adıma göre uygulanmasına yönelik genel bilgiler verilmiştir.

Tablo 9. Ders Planlarına İlişkin Özet Tablo

Konu	Kazanımlar	Etkinlik Adı	Adım	Türü	Bireysel/ Grup	Süre	Hafta
Algoritma Kavramı	<ul style="list-style-type: none"> • Problem çözme sürecindeki temel kavramları açıklar. • Bir problemin çözüm adımlarının doğru bir şekilde belirlenmesinin ve sıralanmasının önemini ifade eder. • Günlük hayatta karşılaştığı problemler için çözüm yolları önerir. • Verilen problemi çözmek üzere farklı algoritmalar tasarlar. • Algoritmayı analiz ederek sonucunu yordar. • Algoritmanın hatalarını giderir. • Algoritmayı daha verimli olması için düzenler. • Problem analiz ve çözüm aşamasında akranlarıyla ortak fikirler geliştirir. 	Çobana yardım	1-7	Bilgisayarsız	Bireysel veya Grup	40dk	1
		Dokumacı kuşlar	1-7	Bilgisayarsız	Bireysel veya Grup	40dk	
Programlama Ortamını Tanıyalım	<ul style="list-style-type: none"> • Sahneye karakter ekler. • Sahne arka planını değiştirir. • Karaktere hareket kazandırır. • Karakter kostümü değiştirerek hareket kazandırır. 	Serbest zaman	1-7	Bilgisayarlı	Bireysel	160dk	2-3
Değişkenler	<ul style="list-style-type: none"> • Verilen problemin çözümünde sabitleri ve değişkenleri belirler. • Verilen problemin çözümünde sabitleri ve değişkenleri kullanır. • Veri türlerini açıklar. • Veri türleri arasındaki farkı açıklar. • Veri türlerini verilen problemin çözümünde kullanır. 	“Manavdan meyve alma”	1	Bilgisayarsız	Bireysel veya Grup	5dk	4-5
		“Meyve seçiyorum”	2-4	Bilgisayarsız	Grup	35dk	
		“Meyve sepetim”	5-7	Bilgisayarlı	Bireysel	120dk	


Şart Yapıları	<ul style="list-style-type: none"> • Karar mantık yapısını açıklar. • Çoklu eğer yapısını açıklar. • Düz mantık yürütmeyi açıklar ve örnek verir. • Olumlu mantık yürütmeyi açıklar ve örnek verir. • Olumsuz mantık yürütmeyi açıklar ve örnek verir. • Mantıksal yapıları birbirine dönüştürür. • Karar tabloları oluşturur. • Problem çözüme süreçlerinde karar yapılarını kullanarak algoritma tasarlar. • Problem çözüme süreçlerinde karar yapılarını kullanarak akış şeması oluşturur. 	“Harita” Etkinliği	1	Bilgisayarsız	Bireysel veya Grup	10dk	6-7
		“Renkli adımlar”	2-4	Bilgisayarsız	Grup	30dk	
		“Labirent harita oyunu”	5-7	Bilgisayarlı	Bireysel	120dk	
Döngüler	<ul style="list-style-type: none"> • Döngü mantık yapısını açıklar. • Döngü mantık yapısı içerisindeki artış değerlerinin işlevini açıklar. • Döngü mantık yapısı içerisinde koşulları açıklar. • Döngü mantık yapısına uygun algoritma tasarlar. • Problem çözüme süreçlerinde döngü yapılarını kullanarak algoritma tasarlar. 	“Ali oyun oynuyor”	1	Bilgisayarsız	Bireysel veya Grup	5dk	8-9
		“Araba yarışı” Sınıf içi canlandırma	2-4	Bilgisayarsız	Grup	35dk	
		“Yakıt miktarım”	5-6	Bilgisayarlı	Bireysel	115dk	
		“Araba yarışı” Uygulaması	7	Bilgisayarlı	Bireysel	5dk	

Ek 4. Döngüler Konusuyla İlgili PAT ve SOT Formları

- **Problemi Anlama ve Tanıma Formu**

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

- **Strateji Oluşturma Formu**

Yol	Araçın Rengi	Yolda Geçen Süre	Araç Deđiřimi	Araç Deđiřimi Süresi
Toprak				
Asfalt				
Kum				
Çakıl				
				
Gidiř Süresi:			Deđiřim Süresi:	
Tur Süresi:			Toplam Süre:	