

44. BÖLÜM

PROGRAMLAMA EĞİTİMİ ÜZERİNE BİR İNCELEME: YAŞANAN ZORLUKLAR, MEVCUT UYGULAMALAR VE GÜNCEL YAKLAŞIMLAR

Arş. Gör. Şenol SAYGINER

Hatay Mustafa Kemal Üniversitesi

Prof. Dr. Hakan TÜZÜN

Hacettepe Üniversitesi

Özet

Ülkelerin müfredat programlarında programlama eğitimine ağırlıklı olarak lisans düzeyinde yer verilmektedir. Eğitimin geç yaşlarda vermeye başlanılmasında, mevcut programlama ortamlarının karmaşık yapısı, süreçte üst düzey bilgi ve becerileri işe koşmanın gerekliliği ve kod yazmanın zorluğu gibi sebeplerden bahsedilebilir. Ancak son zamanlarda, okul öncesi dönemini de kapsayacak nitelikte görsel ortamların geliştirilmesi bu durumu tersine çevirmiştir. Günümüzde yazılımı sevdirmek, öğrencilerin programlama öğrenmelerini kolaylaştırmak ve öğrencileri daha erken yaşlarda programlama ile tanıştırmak bu eğitimleri daha geniş bir alana yaymak amacıyla bazı düzenlemeler yapılmaktadır. Bu süreçte hemen her yaşa uygun programlama ortamları geliştirilmekte, ulusal ve uluslararası niteliğe sahip platformlar oluşturulmakta ve birçok ülkede eğitim müfredatları güncellenmektedir. Bu doğrultuda yapılan bu çalışmada, öncelikle programlama eğitiminde ne tür zorlukların yaşandığı incelenmiş, ardından bu zorluklara ilişkin mevcut uygulamaların ve güncel yaklaşımların neler olduğu ortaya konulmuştur. Ayrıca ülkelerin ilköğretim müfredatlarında programlama eğitimine nasıl yer verdikleri ve programlama eğitimini yaygınlaştırmak amacıyla oluşturulan platformların hangi özelliklere sahip oldukları hakkında bilgiler sunulmuştur.

Anahtar kelimeler: Programlama, programlama eğitimi, görsel programlama ortamları, erken yaşta programlama.

Hazırlık Soruları

1. Programlama eğitiminde ne tür zorluklar yaşanmaktadır?
2. Görsel programlamanın programlama eğitimine katkıları neler olabilir?
3. Okul öncesi, ilkokul ve ortaokul düzeylerinde programlama eğitimi amacıyla hangi ortamlar kullanılabilir?
4. Yurt dışı ve yurt içinde programlama eğitimi üzerine ne tür çalışmalar yapılmaktadır?

Giriş

Bilgi ve iletişim teknolojilerine ilişkin uygulamalar, insanların günlük yaşamlarının neredeyse her alanında yer edinmeye başlamıştır. Örneğin, birbirleriyle İnternet üzerinden iletişim kurabilen nesnelere, üç boyutlu sanal ortamlar, giyilebilir teknolojiler veya yapay zekâya sahip robotik sistemler günümüzde karşılıklı olduğumuz uygulamalar arasındadır. Trafikte sürücüsüz araçlar, cerrahi operasyon yapan robotlar, kargo dağıtımını yapan dronelar ve üç boyutlu sanal ortam gezileri gibi günlük yaşama dair ihtiyaçlar bu teknolojilerle karşılanabilmektedir. Belirtilen bu teknolojiler farklı teknik yapılarla sahip olsa da hepsinin ortak noktası, sistemi kontrol eden veya harekete geçiren bir mekanizmaya yani “programlama” yapısına sahip olmalarıdır.

Programlama, donanım ile yazılım arasındaki iletişimi sağlayan bilgi ve beceriler bütünü olup günümüzde her bireyin öğrenmesi gereken bir yetkinlik olarak görülmektedir. Özellikle 21. yy becerileri olarak belirtilen yaratıcılık, problem çözme, mantıksal düşünme, farklı açılardan bakabilme ve üretim gibi becerilerin programlama eğitimleriyle geliştirilebileceği vurgulanmaktadır (EARGED, 2011; Pinto ve Escudeiro, 2014). Ancak yapılan çalışmalar, programlama eğitiminde çok sayıda zorluğun yaşandığını ortaya koymaktadır (Gomes ve Mendes, 2007; Kinnunen ve Malmi, 2008). Yaşanan bu zorluklara çözüm getirmek ve programlama eğitiminin okul öncesi dönemi itibarıyla verilmesini sağlamak düşüncesiyle günümüzde çeşitli yaklaşımlar kullanılmaktadır. Bu bağlamda yapılan bu çalışmanın amacı, öncelikle programlama eğitiminde yaşanan zorlukları tespit etmek, ardından bu zorluklara yönelik mevcut uygulamaları ve yeni yaklaşımları gözden geçirmektir. Buna ek olarak, ülkelerin eğitim müfredatlarında programlamaya nasıl yer verdikleri ve programlama eğitimini yaygınlaştırmak amacıyla oluşturulan platformların hangi özelliklere sahip oldukları konusunda bilgiler sunmaktır. İlk olarak programlamaya ilişkin temel kavramlar açıklanmış, sonrasında programlama eğitiminde yaşanan zorluklar incelenmiştir.

1. Programlama, Programlama Eğitimi ve Problem Çözme

Programlama, bir teknolojinin (bilgisayar, cep telefonu veya mikrodalga fırın) nasıl çalışması gerektiğini belirleyen ve bu doğrultuda yönlendirmelerde bulunan komutlar bütünüdür. Diğer bir ifadeyle programlama, karşılaşılan bir problemin çözümü için gerekli olan komutların bilgisayarın anlayabileceği şekilde komutlara çevrilmesi, derlenmesi ve çalıştırılmasıdır (Kesici ve Kocabaş, 2001). Burada bilgisayar dili olarak adlandırılan özel kelime ve sembollerin bir araya gelerek oluşturdukları yapı ise programlama dilidir. Günümüzde çok sayıda programlama dilli geliştirilmiş ve her geçen gün yenileri de geliştirilmeye devam etmektedir. Her bir dilin kendine özgü kalıp ve yazım kuralları ile amaca uygun komutlar oluşturulur. Dile özgü komutların yazılması süreci kodlama ya da programlama, ortaya çıkan son ürün ise program ya da uygulama olarak adlandırılır (Ersoy, Madran ve Gülbahar, 2011).

Programlar, bir işin nasıl yapılacağını bilmeyen ancak işin yapılması için gerekli tüm donanımına sahip olan bilgisayarlara, adım adım işin nasıl yapılacağını anlatan yapılardır (Çağiltay ve Fal, 2014). Bu nedenle bir program yazılmadan önce, programı yazan kişinin öncelikle problemi çok iyi anlaması ve adım adım süreci tasarlayabilmesi gerekmektedir. Problemin adım adım detaylı bir şekilde tanımlanması süreci ise algoritma olarak tanımlanabilir. Algoritma, bir görevi yerine getirmek veya bir problemi çözmek için sıralanan adımların kümesidir. Algoritma, özellikle programlama alanında kullanılan bir kavram olmakla birlikte aslında günlük hayatta bilinçli ya da bilinçsiz olarak yaptığımız eylemlerimizde de görülmektedir. Örneğin, evde yemek pişirirken, birine yol tarifi yaparken, trafik ışıklarında beklerken ve birine telefon ederken hep bir sıralamayı takip ederiz. Bu işlemsel sıralamalar birer algoritmadır. Benzer şekilde, bilgisayara işlerimizi yaptırtmak veya bir problemi çözdürtmek gerektiğinde, o işi hangi adımlarda yapması gerektiğini mantıklı bir sırada ve bilgisayarın anlayacağı bir dille (programlama dili) sunmak gerekir. Yani problemin çözüm algoritmasının oluşturulmasına ihtiyaç duyulur.

Buraya kadar ortaya konulan görüşlerde, eylemlerimizin bir algoritma temelinde geliştiği, algoritmaların bir araya gelerek bilgisayarın anlayabileceği bir dille (programlama dili) işe koşulması sürecinin ise programlama olarak nitelendirildiği belirtilmiştir. Yapılan bu tanımlamadan hareketle şu yargıda bulunulabilir: “Programlama bir problem çözme sürecidir”. Bu görüş beraberinde şu soruyu gündeme getirmiştir: “Programlama eğer bir problem çözme süreci ise bu süreç öğrenende ne tür beceriler geliştirir?”

Bu soruya ilişkin olarak öncelikle problem çözmenin ne olduğu tartışılarak bu becerinin programlama ile ilişkisi ortaya konulacaktır. Problem çözme, karşılaşılan bir sorunla başa çıkma sürecidir (Heppner ve Petersen, 1982). Kaya’ya (2011) göre problem çözme, bir problemle karşı karşıya kalındığında problemi çözmek için durumun analiz edilmesi, gerekli bilgilerin toplanması, bunlardan çözüme götürücü olanların seçilmesi ve seçilen bilgilerin uygun bir biçimde düzenlenerek kullanılmasıdır. Pea ve Kurland (1987), problem çözme sürecinin dört aşamadan oluştuğunu belirtmektedir. Bunlar, problemin anlaşılması, çözüm için plan yapma, planın uygulanması, problemin çözülmesi ve değerlendirilmesidir (Pea ve Kurland, 1987). Problem çözme üzerine yapılan tanımlamalarda bir süreçten bahsedildiği görülmektedir. Benzer süreçler programlama için de geçerlidir. Programlama, analiz, tasarım, geliştirme ve test

olmak üzere dört aşamadan oluşmakta olup bu aşamalar ile problem çözme süreçleri arasındaki ilişki devam eden bölümde özetlenmiştir.

Problem anlama: Problem çözmeye karşılaşılan önemli güçlüklerden biri, problemin gerektiği gibi okunup anlaşılabilmesidir. Programlamada da bu durum sıklıkla yapılan bir hatadır. Yeni bir program yazarken ya da var olan programda değişiklik yapmaya gidildiğinde öncelikle problemin ne olduğu net bir şekilde ortaya konulmalıdır.

Çözüm için plan yapma: Bu aşama, problemi çözmeye yardımcı olabilecek bütün bilgilerin toplandığı ve bu bilgiler ışığında değişkenler arasındaki ilişkilerin kurulduğu, hangi değişkenlerin ilişkili ya da ilişkisiz olduğunun belirlendiği ve olası bütün çözüm yollarının ortaya konulduğu aşamadır. Program yazarken de problemin anlaşılmasının ardından olası bütün çözüm yolları belirlenmeye ve çözüm için bir yol haritası oluşturulmaya çalışılır.

Planın uygulanması: Bu aşama, önceki bölümde bir problemi çözmek için belirlenen uygun yöntemin uygulamaya konulduğu aşamadır. İnsanlar bir problemle karşılaştıklarında bu problemi, en az zaman ve çaba ile çözmek isterler. Benzer durum, programlama için de geçerlidir. Kod yazımında en uygun komutların seçilmesi ve programın olabildiğince az sayıda kod satırından oluşması tercih edilen bir durumdur.

Problemin çözülmesi ve değerlendirilmesi: Problemin çözümü için seçilen yöntemin işe koşulmasının ardından başarıya ulaşıp ulaşılmadığının değerlendirildiği aşamadır. Bu durum programlamada programın yazıldıktan sonra doğru çalışıp çalışmadığının kontrol edilmesi aşamasına karşılık gelmektedir.

Buraya kadar yapılan açıklamalarda, ağırlıklı olarak problem çözme becerisi üzerinde durulmuştur. Çünkü bu becerinin pek çok bilişsel beceriyi de içeren şemsiye bir kavram olduğu düşünülmektedir. Bu durumu Soylu ve Soylu (2006) şu şekilde açıklamaktadır: Problem çözme, bilişsel, duygusal ve davranışsal etkinlikleri içeren karmaşık bir süreç olup problem çözenin bilişsel yönü eleştirel, yansıtıcı ve yaratıcı düşünmeyi, analiz yapabilme ve sentezleme becerilerinin kullanımını gerektirir. Aşkar'a (1989) göre ise mantıksal düşünme, problem çözenin alt aşamalarından birisidir. Bu ifade, mantıksal analiz veya muhakeme yapabilen bir kişinin karşılaştığı problemlere etkili çözümler üretebileceği şeklinde yorumlanabilir. Ayrıca bir problemin çözümü sürecinde sorgulama, analitik düşünme, soyut düşünme, karar verme, yaratıcı düşünme ve alternatif çözümler üretme gibi beceriler de işe koşulmaktadır (Monroy-Hernandez ve Resnick, 2008; Shin, Park ve Bae, 2013).

Daha önce de belirtildiği gibi, programlamanın bilişsel süreçlere etkisi yalnızca problem çözme becerisi ile sınırlı değildir. Örneğin, 21. yy becerileri olarak nitelendirilen sistematik düşünme, olaylar arasındaki ilişkileri görebilme ve mantıksal düşünme gibi yetiler programlama ile kazandırılabilir (Balım, 2013; Taylan, 2013; Küçüköğlü, 2014). Yapılan çalışmalarda ayrıca programlama eğitiminin soyut düşünme (Erdoğan, 2005), karşılaşılan problemleri küçük parçalara bölebilme ve analiz edebilme gibi becerilerin gelişimini de sağladığı (Çağiltay ve Fal, 2014) belirtilmektedir.

Programlama eğitiminin bilişsel becerilerin gelişimi üzerinde anlamlı bir etkisinin olduğu görülmektedir. Ancak geçmişten günümüze yapılan çalışmalarda programlama eğitimine

ilişkin çok sayıda zorluğun yaşandığı ortaya konulmakta olup bu zorlukların neler olduğu devam eden bölümde sunulmuştur.

2. Programlama Eğitiminde Yaşanan Zorluklar

Programlama, çok sayıda öğrenci tarafından öğrenimi sürecinde zorluklar yaşandığı belirtilen bir süreçtir. Örneğin Mccracken, Almstrum, Diaz, Guzdial, Hagan ve Kolikant (2001) tarafından gerçekleştirilen bir araştırmada, öğrencilerin sadece %21'inin programlama derslerinde başarılı oldukları belirtilmiştir. Benzer bir çalışma Kinnunen ve Malmi (2008) tarafından yapılmış olup bu çalışmada da dünya genelindeki üniversitelerde programlama temelleri ve programlamaya giriş dersine kayıt olan öğrencilerin %20-%40 oranında dersi veya bölümü daha ilk yılda bıraktıkları görülmüştür. Diğer bir araştırmada ise bilgisayar programlama dersinin eğitimin her kademesinde, özellikle de üniversite düzeyinde, öğrencilerin öğrenmekte en çok zorlandıkları derslerden biri olduğu belirtilmektedir (Jenkins, 2002).

Programlama öğreniminde yaşanan zorlukların sebepleri incelendiğinde, en temel sorun “programlama mantığının oluşturulamaması” olarak belirtilmektedir (Pea ve Kurland, 1983; Lahtinen, Ala-Mutka ve Jarvinen, 2005; Kinnunen ve Malmi, 2008; Özmen ve Altun, 2014). Buna ek olarak, programlama öğrenmeye yönelik öğrencilerin olumsuz bir tutuma sahip olmaları (Gomes ve Mendes, 2007; Hongwarittorn ve Krairit, 2010), programlama diline ait kavramların soyutluğu ve karmaşıklığı (Esteves ve Mendes, 2004; Ozoran, Çağultay ve Topalı, 2012) gibi durumlar da mevcuttur. Byrne ve Lyons (2001), programlama alanında yaşanan zorluğun alışlagelmiş öğretim yöntemlerinden kaynaklandığını belirtmektedir. Byrne ve Lyons'a göre, alışlagelmiş yöntemlerde kitaplardaki kurallar verilerek bu kuralların öğrenciler tarafından ezberlenmesi istenilmektedir.

Programlama, temelinde yoğun bir bilişsel süreci barındırdığı için öğrenimi sürecinde çeşitli becerileri de gerektirmektedir. Örneğin, Gomes ve Mendes (2007) tarafından yapılan çalışmada, bazı öğrencilerin problemin nasıl çözüleceğini bilmemelerinden, bazılarının ise yeterli matematiksel ve mantıksal bilgiye sahip olmamalarından dolayı programlamayı yapamadıklarını belirtmişlerdir. Linn ve Clancy (1992) ise öğrencilerin sadece programı yazmaya odaklandıklarını, mantıksal çıkarımlar yapmadıklarını belirtmişlerdir. Bu sonuç, mantıksal düşünme ve problem çözme becerilerinin programlama sürecindeki önemine de ışık tutmaktadır.

Yapılan açıklamalarda programlamanın öğrenimi sürecinde zorluklar yaşandığı belirtilerek bu zorluklara ve sebeplerine dair açıklamalara yer verilmiştir. Tablo 1'de geçmişten günümüze doğru programlama eğitimine dair yaşanan zorluklar özetlenerek sunulmuştur.

Tablo 1. Programlama eğitiminde yaşanan zorluklar

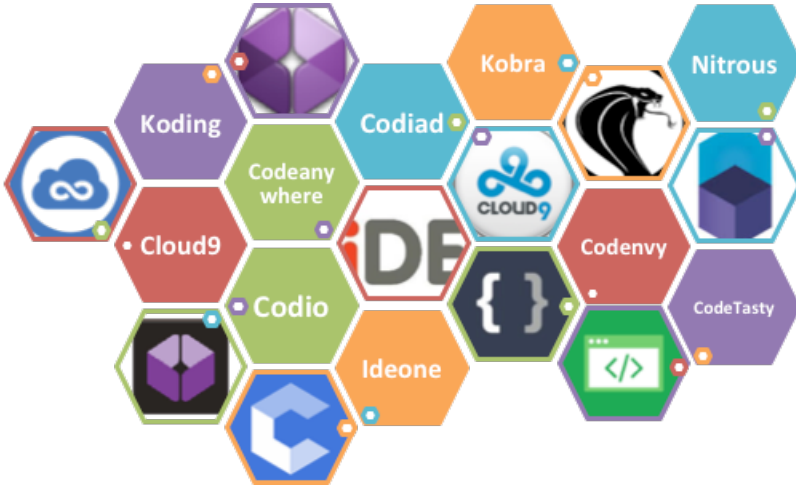
Çalışmayı Yapanlar	Yazılım/ Ders	Hedef Kitle	Algoritma oluşturma	Programlama dilinin yapısı	Öğrenme-öğretme yöntemleri	Hatayı bulma, ayıklama	Motivasyon	Sayısal bilgi-beceri eksikliği	Kod yazma, koda dökme	Programlamaya karşı tutum	Program geliştirme ortamı
Bayman ve Mayer (1983)	Basic	Üniversite D. ve Öğretmenler	*								
Du Boulay (1986)	-	Öğretmenler	*						*		
Linn ve Clancy (1992)	Pascal	Lise Düzeyi	*								
Proul (2000)	BP	-	*	*							
Byrne ve Lyons (2001)	Basic	Üniversite Düzeyi						*		*	
Duncan (2002)	BP	Üniversite Düzeyi	*	*	*				*		
Jenkins (2002)	-	-	*		*		*	*			
Lahtinen vd. (2005)	C, Java ve Pascal	Üniversite D. ve Öğretmenler	*	*		*					*
Gomes ve Mendes (2007)	-	-	*	*	*		*				
Kinnunen ve Malmi (2008)	Java	Üniversite Düzeyi	*			*	*	*			
Tang, Ting ve Ling (2009)	BP	Üniversite Düzeyi	*	*		*					
Hawi (2010)	BP	Üniversite Düzeyi			*						
Hongwarittorn vd. (2010)	Java	Üniversite Düzeyi								*	
Sivasakthi ve Rajendran (2011)	Java	Üniversite Düzeyi	*			*					*
Ozoran, Çağıltay vd. (2012)	C	Üniversite Düzeyi		*			*				
Derus ve Ali (2012)	BP	Üniversite Düzeyi	*	*	*						
Mhashi ve Alakeel (2013)	C#	Üniversite Düzeyi	*		*				*		
Piteira ve Costa (2013)	-	Üniversite D. ve Öğretmenler	*	*					*		
Gomes ve Mendes (2014)	BP	Üniversite D. ve Öğretmenler	*				*	*			
Özmen ve Altun (2014)	BP	Üniversite Düzeyi	*		*	*					
Cevahir ve Özdemir (2017)	BP	Öğretmenler		*	*			*			

“BP: Bilgisayar Programlamaya Giriş Dersi”

Programlama eğitiminin bilişsel becerilerin gelişimine olumlu bir etki yaptığı belirtilmiştir. Gelineen noktada ise programlama öğrenmenin zor bir süreç olduğu vurgulanmaktadır. O

halde soru şudur: Programlamanın beceri gelişimi üzerine anlamlı bir katkısı olduğu gerçeğinden hareketle bu eğitim öğrenciyi en az zorlanacağı şekilde nasıl bir yöntemle verilebilir? Sorunun cevabına ilişkin günümüzde iki farklı uygulamadan söz edilebilir. Bunlar, bulut tabanlı programlama ortamları ve görsel programlama ortamlarıdır.

Bulut tabanlı programlama, bilgisayara herhangi bir yazılım kurmaya gerek kalmaksızın İnternet erişimi olan cihazlar aracılığıyla evde, işte veya yolda kod yazımına devam edebilme gibi olanaklar sunmaktadır. Bu ortamlar ile ayrıca farklı mekânlardaki kullanıcılar eş zamanlı olarak aynı kod üzerinde çalışabilmektedir. Bu durumun öğrenciler arasında işbirlikli projeleri yaygınlaştıracığı ve öğrenmeye yönelik motivasyonu arttıracığı söylenebilir. Şekil 1’de bulut tabanlı programlama ortamlarına ilişkin örnekler sunulmuştur.



Şekil 1. Bulut tabanlı entegre geliştirme ortamları (IDE)

Programlama eğitiminde yaşanan zorlukları gidermek üzere diğer bir çözümsel yaklaşım ise görsel programlama ortamlarıdır. Bu ortamlarda ses, resim ve müzik gibi çoklu ortam destekli projeler oluşturulabilmektedir. Birçok ortam kod yazmayı gerektirmeyen ara yüzlere sahip olup kullanıcılar sürükleyip bırak etkinlikleriyle projeler oluşturabilmektedir. Belirtilen bu özelliklerden hareketle programlama eğitimi sürecine olası katkılarına ilişkin bilgiler ilerleyen bölümde sunulmuştur.

3. Görsel Programlama Ortamları (GPO)

Önceki bölümde programlama eğitiminde “programlama mantığının oluşturulamaması, programlama dilinin yapısı, hata bulma, kod yazma” gibi sorunların yaşandığı belirtilmişti. Görsel ortamların bu sorunlara çözüm getirebileceği düşünülmektedir. Çünkü alışlagelmiş öğretim yöntemlerinde kodlar metin tabanlı ve soyut bir yapıda iken görsel ortamlarda adım adım somutlaştırarak sunulmakta ve anlık dönütler alınabilmektedir. Ayrıca GPO’ların büyük bir kısmında kodların bloklar halinde olması sayesinde program yazarken kodun unutulmasının (Lahtinen, Ala-Mutka ve Jarvinen, 2005; Kinnunen ve Malmi, 2008), kodların

veya kuralların mantığının öğrenilmesi yerine ezberlenmesinin (Byrne ve Lyons, 2001; İmal ve Eser, 2009) ya da programın algoritmasının oluşturulmasında güçlük çekilmesinin önüne geçilebilmektedir. Ek olarak, bu ortamların öğrencilere kod yazmayı sevdiğini ve program yazmaya istek oluşturduğunu belirtmek de mümkündür. Özellikle programlama öğrenmeye yeni başlayanların programlamaya olan merakını arttırarak onları araştırmaya sevk etmekte, sistematik ve mantıksal düşünme becerilerini geliştirmekte ve özgüvenlerini arttırmaktadır (Genç ve Karakuş, 2011).

Ayrıca yapılan çalışmalarda görsel programlama ortamlarının programlama öğrenimini kolaylaştırdığı (Wang, Huang ve Hwang, 2014), öğrencilerin problem çözme becerilerini geliştirdiği (Lai ve Yang, 2011; Lai ve Lai, 2012; Fessakis, Gouli ve Mavroudi, 2013), programlama öğrenmeye yönelik motivasyonu ve tutumu arttırdığı (Kereki, 2008; Nikiforos, Kontomaris ve Chorianopoulos, 2013; Nikou ve Economides, 2014; Ruf, Mühling ve Hubwieser, 2014; Wyffles, Martens ve Lemmens, 2014) belirtilmektedir.

Görsel programlama ortamlarının sahip olduğu potansiyelin programlama eğitimine olumlu yansımaları olduğu görülmektedir. Bu doğrultuda günümüzde okul öncesi dönemden üniversite düzeyine kadar kullanılabilir çeşitlilikte ve sayıda GPO geliştirildiği ve programlama eğitiminin erken yaşlardan itibaren verilmesine yönelik eğilimlerin arttığı gözlenmektedir. Ancak bu durum “hangi yaş ve eğitim düzeyinde hangi programlama ortamının kullanılabilirliği” konusunda bazı karışıklıkları da beraberinde getirmiştir. Bu noktadan hareketle bu çalışmada en fazla kullanılan 12 adet görsel programlama ortamı çeşitli değişkenler açısından incelenmiş ve Tablo 2’de karşılaştırmalı olarak sunulmuştur.

Tablo 2. Görsel programlama ortamları

Programlama Ortamı	Yazılımın İlk Geliştiricisi	Geliştirildiği Yıl	Kodlama Yapısı	Düzeğe Uygunluk	Türkçe Dil D.	Ücret Durumu
Lightbot lightbot.com	Daniel Yaroslavski (Waterloo Üniversitesi)	2008	Blok tabanlı	4-8 yaş (LightbotJr) ve 9 yaş ve üzeri (Lightbot)	Yok	LightbotJr: Ücretsiz, Lightbot: Ücretli
Scratch Junior scratchjr.org	MIT, Tufts Üniversitesi ve Playful şirketi	2014	Blok tabanlı	5-7 yaş arası	Yok	Ücretsiz
Kodable kodable.com	Grechen H. ve Jon Mattingly (Louisville Üniversitesi, ABD)	2012	Blok tabanlı	5-10 yaş arası	Yok	Belirli bir seviyeye kadar ücretsiz
Code Combat codecombat.com	Code Combat şirketi, ABD	2013	Python, JavaScript ve Lua	6 yaş ve üzeri	Var	Ücretsiz

Tynker tynker.com	Tynker şirketi, ABD	2013	Blok ve metin tabanlı: HTML, Python, JavaScript ve Robotik	7 yaş ve üzeri	Yok	20 seviye ücretsiz
Code Monkey playcode-monkey.com	Jonathan Schor, Brooklyn, NY	2014	Coffee Script	K-12 düzeyi	Var	Belirli bir seviyeye kadar ücretsiz
M. Small Basic small-basic.com	Microsoft	2008	Metin tabanlı-Basic	7 yaş ve üzeri	Var	Ücretsiz
Scratch scratch.mit.edu	MIT Medya Lab-Lifelong Kindergarten projesi	2003	Blok tabanlı	8 yaş ve üzeri	Var	Ücretsiz
Alice alice.org	Carnegie Mellon Üniversitesi	1998	Blok tabanlı	8 yaş ve üzeri	Var	Ücretsiz
MIT APP Inventor appinventor.mit.edu	Google ve MIT işbirliğiyle	2012	Blok tabanlı	K-12 ve üzeri	Yok	Ücretsiz
Lego Mindstorms mindstorms.lego.com	Colorado Üniversitesi	1994	Blok ve Metin tabanlı	K-12 ve üzeri	Yok	Bileşenler ücretli
Arduino arduino.cc	Ivrea Tasarım Enstitüsü projesi	2000	Metin tabanlı-C	K-12 ve üzeri	Var	Bileşenler ücretli

Tablo 2'den hareketle, günümüzde hemen her yaşa uygun yazılım bulmanın mümkün olduğu, dolayısıyla da bu durumun programlama eğitiminin erken yaşlardan itibaren verilmesi konusuna önemli bir dayanak oluşturduğu savunulabilir. Nitekim günümüzde çok sayıda ülke eğitim müfredatlarını güncelleyerek programlama eğitimine daha fazla yer vermeye başlamıştır. Ayrıca çok sayıda ulusal veya uluslararası organizasyon oluşturulmuş olup bu oluşumlar “çocuklara yönelik programlama eğitimleri” konularında girişimlerde bulunmaktadır. Bu bağlamda sonraki bölümde erken yaşlarda programlama eğitimine ilişkin uygulamalardan örnekler sunulacaktır.

4. Erken Yaşta Programlama Eğitimi

Günümüzde bireyler “21. yy bireyleri” ya da “dijital nesil” gibi ifadelerle anılmakta olup bu bireylerin problem çözme, mantıksal çözüm yolları üretebilme, üretim ve yaratıcılık gibi becerilere sahip olmaları beklenmektedir (EARGED, 2011; Pinto ve Escudeiro, 2014). Yapılan

çalışmalar, bu becerilerin kazandırılmasının programlama eğitimiyle mümkün olabileceğini ortaya koymaktadır (Monroy-Hernandez ve Resnick, 2008; Nam, Kim ve Lee, 2010; Lai ve Lai, 2012; Kobsiripat, 2015; Giordano ve Mairona, 2015). Bu doğrultuda gerek yurt dışında gerek yurt içinde programlama eğitiminin önemini farkına varılmış olup birçok ülke ilköğretim müfredat programında düzenlemeler yapmaktadır. Bu süreçte bazı ülkeler müfredatlarında yer alan bilişim ders içeriklerini güncellerken bazıları ise ilköğretim müfredat programına çeşitli kademelerde programlama derslerini ilk kez dâhil etmektedir. Ayrıca ülkeler arasında programlama eğitiminin verildiği sınıf düzeyi ve bu eğitimle geliştirilmek istenen beceriler de farklılık göstermektedir.

2015 yılında Avrupa Okul Ağı tarafından yapılan bir araştırmaya göre, ilköğretim müfredat programına programlama eğitimini dâhil eden 18 Avrupa ülkesinin bulunduğu ve bu ülkelerin müfredat programlarında programlama eğitimlerine farklı sebeplerden dolayı yer verdikleri belirtilmiştir. Bu ülkeler ve programlama eğitimini müfredatlarına dâhil etme sebepleri Tablo 3'te sunulmuştur.

Tablo 3. Programlama eğitimini müfredatına dâhil eden 18 Avrupa ülkesi ve bu eğitimi dâhil etme sebepleri

	Mantıksal Düşünme Becerisini Destekleme	Problem Çöz- me Becerisini Destekleme	Kodlama Becerilerini Destekleme	Öğrencileri BİT alanına dâhil etme	Diğer Anah- tar Yeterlikle- ri Destekleme	BİT İstihda- mını Destek- leme
İngiltere	*	*	*	*		*
Slovakya	*	*				
İspanya	*	*	*		*	
Portekiz	*	*			*	*
Polonya	*	*	*	*	*	*
Malta			*	*		
Litvanya	*		*			
Macaristan	*	*				
İsrail	*	*	*	*	*	*
İrlanda	*	*	*	*	*	
Fransa				*	*	*
Finlandiya	*	*	*			
Estonya	*	*		*	*	
Danimarka	*	*			*	
Çek Cumhuriyeti	*	*	*	*	*	*
Bulgaristan	*	*	*	*		
Belçika				*	*	*
Avusturya	*	*	*	*	*	*
Toplam	15	14	11	11	11	8

18 Avrupa ülkesinin ilköğretim müfredatında programlama eğitimine en fazla sırasıyla “mantıksal düşünme becerisini desteklemek” ve “problem çözme becerisini desteklemek” amacıyla yer verdikleri görülmektedir. Ayrıca ülkelerin programlama eğitimine müfredatlarında yer vermelerindeki bir diğer önemli sebebin bilişim alanındaki istihdamı arttırmak olduğu görülmektedir. Çünkü diğer alanlara kıyasla bilişim sektöründe işgücü gereksinimi %12 oranında daha fazladır (Glass, 2017). Buna ek olarak, Avrupa’da 2020 yılına kadar 800.000 yazılım uzmanına gereksinim duyulacağı belirtilmektedir (European Schoolnet, 2015).

Programlama eğitimini müfredatına dâhil eden ülkeler yalnızca Avrupa ülkeleriyle sınırlı olmayıp dünyanın pek çok noktasında benzer girişimlerde bulunmaktadır. Örneğin Amerika Birleşik Devletleri; hükümet, sivil toplum kuruluşları, Microsoft ve Google gibi teknoloji ve yazılım şirketlerinin desteğiyle okullarda kodlama eğitimini yaygınlaştırmak ve programlama becerilerini geliştirmek amacıyla projeler yürütmektedir. Bu amaç doğrultusunda 2013 yılında code.org platformu kurulmuş ve hemen her yaşta öğrencinin hizmetine ücretsiz olarak sunulmuştur (Code.org, 2017). Öğrenciler Alice, Scratch, App Inventor ve Robomind gibi görsel programlama ortamları üzerinden kodlamalar yapabilmektedir.

Programlama eğitimi üzerine önemli yatırımlar yapan bir diğer ülke İngiltere’dir. Ülkedeki okullarda 2013 yılı itibarıyla programlama eğitime başlanmış ve 2014 yılı tüm ülkede “Kodlama Yılı” (Year of Code) olarak ilan edilmiştir (Özdemir, 2015). Ayrıca “çağa uygun olmadığı” gerekçesiyle bilgi ve iletişim teknolojileri dersinin içeriği değiştirilerek programlama içerikli konulara yer verilmiştir. Okullarda programlama eğitimi 3 kademeli olarak sürdürülmekte olup birinci kademede (5-7 yaş) algoritmanın ne olduğu, ikinci kademede (7-11 yaş) karmaşık problemlerin çözümü, üçüncü kademede (12-14) ise iki ya da daha fazla programlama diline hâkim olunması amaçlanmıştır (Öndeş, 2016).

Yazılım alanında önemli başarılarla sahip olan Hindistan’ın müfredat programında da yoğun şekilde programlama eğitime yer verilmektedir. Ülkede bir öğrenci henüz ilkokuldayken (1 ve 4. sınıflar arası) algoritma eğitimi almaktadır. Bu eğitime ortaokul düzeyinde BASIC ile devam edilmektedir. Liseye geçildiğinde ise ileri düzeyde programların yazılabilmesi beklenmektedir (SSRVM, 2007).

Programlama eğitimine müfredatında yer veren bir diğer ülke Estonya’dır. Ülkede 2012 yılında pilot bir program uygulamaya konulmuş ve ardından ilkokul 1. sınıf itibarıyla kodlama eğitiminin verilmesi kararlaştırılmıştır (Olson, 2012). Bu doğrultuda ülke genelinde 7-19 yaş arasındaki öğrencilere ProgeTiger adıyla zorunlu eğitim kapsamında bir proje yürütülmektedir (Menge, 2012; Kodlamaogreniyorum, 2015). Proje kapsamında ilkokul düzeyinde LEGO ve Kodu Game Lab, ortaokul düzeyinde Scratch ve LEGO Mindstorms, lise düzeyinde ise Python, JavaScript ve Robotik gibi ileri düzey eğitimler verilmektedir (HITSA, 2015).

Avustralya’da öğrenciler 5 yaşından itibaren 2 yıl boyunca temel programlama dilleri ve kod eğitimi almakta olup bu eğitimler ilerleyen senelerde ileri düzey programlama derslerine dönüşecek şekilde planlanmaktadır. Ortalama 7 yaşında bir öğrencinin programlama mantığına hâkim olması amaçlanmaktadır (Kahraman, 2015; Öçalan, 2015).

Yurtdışında programlama eğitimini yaygınlaştırma konusunda yürütülen projeler incelendiğinde, her geçen gün ülkelerin bu eğitime verdiği önemin arttığı ve bu eğitimlerin erken yaşlarda verilmesine yönelik bir eğilimin olduğu görülmektedir. Ülkemizde de sivil toplum kuruluşları, üniversiteler ve Milli Eğitim Bakanlığı'nın da desteklediği çok sayıda proje gerçekleştirilmektedir. Bu bağlamda atılan adımlar incelendiğinde ilk olarak müfredat programındaki değişiklikler göze çarpmaktadır. Ülkemizde 2012 yılına kadar “Bilgisayar” ve “Bilişim Teknolojileri” olarak adlandırılan bilgisayar dersleri, 2012 yılında “Bilişim Teknolojileri ve Yazılım” olarak değiştirilmiştir (BTE Derneği, 2013). Bu değişimi önemli kılan özellik ise dersin isminde ilk defa yazılım ifadesine yer verilmesi ve dolayısıyla müfredata algoritma ve programlamayla ilgili konuların eklenmiş olmasıdır (BTE Derneği, 2013). Buna ek olarak, Milli Eğitim Bakanlığı bünyesinde geliştirilmiş olan EBA Portalı üzerinden öğrenci ve öğretmenler kendi programlarını yazabilmekte veya başkası tarafından yazılan bir programın kod satırlarına erişerek geliştirmeler yapabilmektedirler. Ayrıca 2014 yılında Türkiye Bilişim Derneği ve çeşitli üniversitelerin desteğiyle “Bilgisayar Programlama Çocuk Oyunağı” adlı bir etkinlik organize edilmiştir. Etkinliğin amacı, ilk, orta ve lise düzeylerindeki öğrencilerin her gün keyifle zaman geçirdikleri bilgisayar ve İnternet teknolojileriyle kendi programlarını da yazabileceklerini ve program yazmanın aslında basit olduğunu fark etmelerini sağlamaktır (TBD, 2014).

Ülkemizde faaliyet gösteren mobil operatör şirketleri de programlama eğitimini yaygınlaştırmak amacıyla projeler gerçekleştirmektedir. Örneğin, 2016 yılında hayata geçirilen ve “Yarım kodlayanlar” olarak adlandırılan bir proje ile 7-14 yaş aralığındaki çocuklara Scratch ve Arduino ile programlama eğitimleri verilmektedir (Hatisaru, 2016; Yüzak, 2016). Proje, Milli Eğitim Bakanlığı tarafından da desteklenmekte olup 2017-2018 yılları içerisinde 30 ilde 10 bin çocuğa programlama eğitimi verilmesi hedeflenmiştir (Türkiyevodafonevakfı, 2016). Ülkemizde bir operatör şirketi tarafından gerçekleştirilen diğer proje ise “Zekâ Küpü” projesidir. Proje, Bilim Sanat Merkezleri'nde (BİLSEM) öğrenim gören “özel yetenekli” öğrencileri destekleme amacını taşımaktadır (Zekâküpü, 2016). 2016 yılı içinde başlatılan projeye ilk yıl 7 ilde 2000 öğrenci dâhil edilmiş olup ilerleyen 3 yıl içerisinde bu sayının 10.000'e ulaşması hedeflenmektedir (Zekâküpü, 2016).

Buraya kadar yapılan açıklamalarda gerek yurt dışında gerek yurt içinde programlama eğitimi üzerine yapılan projelerde, ülkelerin eğitim bakanlıklarının, üniversitelerin ve özel kuruluşların desteklerinden bahsedilmiştir. Ancak programlamayı yaygınlaştırmak üzere yapılan projeler yalnızca bunlarla sınırlı değildir. Ulusal ve uluslararası alanda birçoğu gönüllülük esasına dayalı platformlar da hizmet vermektedir. “Coder Dojo, Code Clup, Code ve Bilişim Garaj Akademisi” bu platformlardan olup bunlara ilişkin karşılaştırmalı bilgiler ise Tablo 4'te sunulmuştur.

Tablo 4. Programlama eğitimini yaygınlaştırmak üzere oluşturulan platformlar

Platformun Adı	Geliştirildiği Yıl, Ülke	Eğitici Türü	Eğitim Türü	Yaş Düzeyi	Programlama Eğitimleri	Uygulanan Ülke S.	Desteklenen Dil S.
Coder Dojo coderdojo.com	2011, İrlanda	Gönüllü eğiticiler	Atölye çalışmaları	6-17 yaş arası	HTML, CSS, JavaScript, Ruby, Arduino, PHP, Raspberry Pi, Scratch ve Python	60+ ülke	-
Code Club codeclub.org.uk	2012, İngiltere	Gönüllü eğiticiler	Atölye çalışmaları	9-13 yaş arası	HTML, CSS, Scratch ve Python	100+ ülke	28 farklı dil desteği
Code Code.org	2013, ABD	Gönüllü eğiticiler	Çevrimiçi kurslar	K-12	Alice, Scratch, Kodu Game Lab, LighBot, MIT App Inventor ve Robomind	180+ ülke	54 farklı dil desteği
Bilişim Garaj Akademisi bilisimgarajakademisi.com	2012, Türkiye	Uzmanlar	Atölye çalışmaları	5-16 yaş arası	Scratch, APP Inventor, Kodu, Small Basic, Python, Alice ve Arduino	-	-

5. Sonuç

Bu çalışmada programlamaya ilişkin temel kavramlar üstünde durulmuş, programlama eğitiminde yaşanan zorluklar ve bu zorluklara yönelik çözüm odaklı platformlar belirtilmiş, ülkelerin eğitim müfredatlarında programlama eğitimine nasıl yer verdikleri ve programlama eğitimini yaygınlaştırmak üzere hangi organizasyonların bulunduğu dair açıklamalarda bulunulmuştur. Öncelikle programlamaya ilişkin kavramlar üstünde durularak geçmişten günümüze programlama eğitimi sürecinde ne tür zorlukların yaşandığı incelenmiştir.

Yapılan incelemelerde, programlama eğitiminde yaşanan en temel zorluğun bir problemin algoritmasını oluşturma konusunda olduğu görülmüştür. Bu zorluğun sebeplerinin ise algoritma oluşturma bir probleme çözüm getirme süreci olması; yani mantıksal düşünme, farklı açılardan bakabilme ve sorgulama gibi becerileri gerektirmesi, programlamanın karmaşık yapısı, ezberci yaklaşımlar olarak belirtilmiştir. Bu durum, günümüzde programlama eğitimi sürecinde alışlagelmişin dışında yöntem ve teknolojilere ihtiyaç duyulduğunun bir göstergesi olarak değerlendirilmiştir. Nitekim günümüzde bu duruma çözüm niteliğinde olan ve programlamayı basitleştirerek öğretmeyi amaçlayan çok sayıda yazılımın geliştirildiği görülmüştür. Yazılımlarla ilgili yapılan değerlendirmelerde, okul öncesi dönemden üniversite düzeyine kadar kullanılabilir çeşitlilikte oldukları belirtilmiştir. Bu yazılımların programlama eğitimine ve öğrencilerdeki beceri gelişimine olumlu katkılarından hareketle çok sayıda ülkenin eğitim müfredatında yer edinmeye başladığı belirtilmiştir. Ayrıca, ulusal ve ulusla-

rarası düzeyde platformların oluşturulduğu belirtilerek bu platformlar üzerinden çok sayıda öğrenene programlama eğitimlerinin sunulduğu görülmüştür.

Özetle, gerek programlama eğitimini yaygınlaştırmak amacıyla yapılan çalışmalardan, gerek programlama eğitimi üzerine evrensel çerçevede yürütülen projelerden hareketle şu yargıda bulunulabilir: “Programlama, düşünmenin ve düşünceyi ürüne dönüştürebilmenin yollarından biridir”.

Yansıtma Soruları

1. Programlama eğitimine hangi yaş itibariyle başlanması gerektiğini gerekçeleriyle birlikte tartışınız.
2. Programlama eğitiminin önemini 21. yy becerileri bağlamında tartışınız.
3. Okul öncesi, ilkokul ve ortaokul düzeyinde yapılacak programlama eğitimlerinin öğrencilerin diğer derslerine nasıl bir yansıması olabileceğini tartışınız.

Kaynaklar

- Aşkar, P. (1989). *Etkileşimli problem çözme*. Problem Çözme Yöntemleri Sempozyumu. ODTÜ, Ankara. [Çevrim-içi: <https://goo.gl/HkByMq>, Erişim tarihi: 12.09.2017.]
- Balm, M. A. (2013). *Çocuklara neden programlama öğretmeliyiz?* [Çevrim-içi: <http://www.elektrikport.com/teknik-kutuphane/cocuklara-neden-programlama-ogretmeliyiz/8832#ad-image-0>, Erişim tarihi: 10.08.2017.]
- Bayman, P., & Mayer, R. E. (1983). A diagnosis of beginning programmers' misconceptions of Basic programming statements. *Communications of the ACM*, 26(9), 677-679.
- BTE Derneği (2013). *Türkiye'de ilk ve ortaokullarda (ilköğretim) okutulan bilişim teknolojileri derslerinin tarihi*. [Çevrim-içi: <http://bte.org.tr/bte-derneği/bt-derslerinin-tarihcesi/>, Erişim tarihi: 01.08.2017.]
- Byrne, P., & Lyons, G. (2001). *The effect of student attributes on success in programming*. Proceedings of The 6th Annual Conference On Innovation And Technology In Computer Science Education. Canterbury, UK, June 25-27, 2001. [Çevrim-içi: <http://dl.acm.org/citation.cfm?id=377467>, Erişim tarihi: 19.09.2017.]
- Cevahir, H., & Özdemir, M. (2017). *Programlama öğretiminde karşılaşılan zorluklara yönelik öğretmen görüşleri ve çözüm önerileri*. 11. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu, İnönü Üniversitesi, Malatya. [Çevrim-içi: <https://goo.gl/BnVLQL>, Erişim tarihi: 01.01.2018.]
- Code.org (2017). *Expanding computer science through partnerships*. [Çevrim-içi: <https://code.org/3rdparty-partners>, Erişim tarihi: 17.05.2017.]
- Çağltay, N. E., & Fal, M. (2014). *Scratch ile programlama öğreniyorum*. Ankara: ODTÜ Yayıncılık.
- Derus, S. R. M., & Ali, A. Z. M. (2012). *Difficulties in learning programming: Views of students*. In 1st International Conference on Current Issues in Education (ICCIE 2012) (pp. 74-79).
- DuBoulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, 2(1), 57-73.
- Dunican, E. (2002). *Making the analogy: Alternative delivery techniques for first year programming courses*. 14th Workshop of the Psychology of Programming Interest Group, Brunel University. [Çevrim-içi: <http://eprints.teachingandlearning.ie/3991/1/Dunican%202002.pdf>, Erişim tarihi: 15.02.2018.]
- EARGED, (2011). *MEB 21.yy öğrenci profili*. [Çevrim-içi: <https://goo.gl/TUVDPA>, Erişim tarihi: 14.05.2016.]
- Erdoğan, B. (2005). *Programlama başarısı ile akademik başarı, genel yetenek, bilgisayara karşı tutum, cinsiyet ve lise türü arasındaki ilişkilerin incelenmesi*. Yayınlanmamış Yüksek Lisans Tezi, Marmara Üniversitesi, Eğitim Bilimleri Enstitüsü, İstanbul.
- Ersoy, H., Madran, R. O., & Gülbahar, Y. (2011). *Programlama dilleri öğretimine bir model önerisi: Robot programlama*. Akademik Bilişim'11-XIII. Akademik Bilişim Konferansı Bildirileri. Malatya İnönü Üniversitesi, 2-4 Şubat 2011. [Çevrim-içi: <https://goo.gl/pZvmFe>, Erişim tarihi: 5.07.2017.]
- Esteves, M., & Mendes, A., (2004). *A simulation tool to help learning of object oriented programming Basics*. In Proceedings of the 34th ASEE/IEEE Frontiers in Education Conference. Savannah, Georgia, 20-23 Ekim 2004. [Çevrim-içi: <https://goo.gl/8VA6Rz>, Erişim tarihi: 03.08.2016.]
- European Schoolnet (2015). *Computing our future Computer programming and coding-Priorities, school curricula and initiatives across Europe*. [Çevrim-içi: <http://goo.gl/QIOkGw>, Erişim tarihi: 11.04.2017.]
- 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.
- Genç, Z., & Karakuş, S. (2011). *Tasarımla öğrenme: Eğitsel bilgisayar oyunları tasarımında Scratch kullanımı*. 5th International Computer & Instructional Technologies Symposium (ICITS), Elazığ, 22-24 Ekim 2011. [Çevrim-içi: <https://goo.gl/ik59o9>, Erişim tarihi: 15.07.2016.]

- Giordano, D., & Maiorana, F. (2015). *Teaching algorithms: Visual language vs flowchart vs textual language*. IEEE Global Engineering Education Conference. Tallinn University of Technology, Tallinn, Estonia, 18-20 March 2015. [Çevrim-içi: <https://goo.gl/yMyEh2>, Erişim tarihi: 19.11.2016.]
- Glass, B. (2017). *Fast company: How you'll search for a job in 2017*. [Çevrim-içi: <http://burning-glass.com/fast-company-youll-search-job-2017>, Erişim tarihi: 01.05.2017.]
- Gomes, A., & Mendes, A. (2014). *A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations*. In Frontiers in Education Conference (FIE), (pp. 1-8).
- Gomes, A., & Mendes, A. J. (2007). *Learning to program difficulties and solutions*. International conference on Engineering Education, Portugal, Coimbra, 3-7 Ekim 2007. [Çevrim-içi: <http://icee2007.dei.uc.pt/proceedings/papers/411.pdf>, Erişim tarihi: 05.08.2017.]
- Hatsırsu, S. (2016). *Vodafone'dan çocuklara kod yazma dersi*. [Çevrim-içi: <https://goo.gl/L97Umt>, Erişim tarihi: 01.08.2016.]
- 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.
- Hepner, P. P., & Petersen, C. H. (1982). The development and implications of a personal problem-solving inventory. *Journal of Counseling Psychology*, 29(1), 66-75.
- HITSa (2015). *ProgeTiger programme 2015-2017*. [Çevrim içi: <http://www.hitsa.ee/it-education/educational-programmes/progetiger>, Erişim tarihi: 01.03.2018.]
- Hongwarittorn, N., & Krairit, D. (2010). *Effects of program visualization (jeliot3) on students' performance and attitudes towards java programming*. In The spring 8th International conference on Computing, Communication and Control Technologies, Orlando, Florida USA, 6-9 April 2010. [Çevrim-içi: <https://goo.gl/pasiEz>, Erişim tarihi: 11.09.2016.]
- İmal, N., & Eser, M. (2009). *Programlama dili öğrenmedeki zorluklar ve çözüm yaklaşımları*. Elektrik Elektronik Bilgisayar Biyomedikal Mühendislikleri Eğitimi IV. Ulusal Sempozyumu. Eskişehir Osmangazi Üniversitesi Kongre ve Kültür Merkezi, Eskişehir, 22-24 Ekim 2009. [Çevrim-içi: <https://goo.gl/gPuRB3>, Erişim tarihi: 11.08.2016.]
- 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*, 4, 53-58.
- Kahraman, B. (2015). *Avustralya'da ilkokullarda programcılık eğitimi veriliyor*. [Çevrim-içi: <https://goo.gl/d977rN>, Erişim tarihi: 01.08.2016.]
- Kaya, M. (2011). *Problem ve problem çözme*. [Çevrim-içi: <https://goo.gl/ywxXvx>, Erişim tarihi: 11.08.2016.]
- Kereki, I. F. (2008). *Scratch: Applications in Computer Science 1*. 38th Annual Frontiers in Education Conference. Saratoga Springs, NY. [Çevrim-içi: <https://goo.gl/ELXM8r>, Erişim Tarihi: 20.09.2016.]
- Kesici, T., & Kocabaş, Z. (2001). *Liseler için bilgisayar 2*. Ankara: MEB Yayınları.
- Kinnunen, P., & Malmi, L. (2008). *CS minors in a CS1 course*. In Proceeding of the Fourth international Workshop on Computing Education Research Sydney, Australia, 06-07 Ekim 2008. [Çevrim-içi: <https://goo.gl/D9ZrPp>, Erişim tarihi: 11.09.2016.]
- 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.
- Kodlamaogreniyorum (2015). *Türkiye ve dünyada kullanılan kodlama programları*. [Çevrim-içi: <http://kodlamaogreniyorum.weebly.com/>, Erişim tarihi: 02.04.2017.]
- Küçükkoğlu, Ö. (2014). *21.yüzyıl becerileri için en etkin oluşumlar: "Kod okur-yazarlığı" ve "FLL"*. [Çevrim-içi: <https://goo.gl/44uuWm>, Erişim tarihi: 10.08.2016.]

- Lahtinen, E., Ala-Mutka, K., & Jarvinen, H. M. (2005). *A study of the difficulties of novice programmers*. Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education. New York, USA, 26-29 Haziran. [Çevrim-içi: <http://dl.acm.org/citation.cfm?id=1067453>, Erişim tarihi: 11.07.2016.]
- Lai, A. F., & Yang, S. M. (2011). *The learning effect of visualized programming learning on 6 th graders' problem solving and logical reasoning abilities*. Electrical and Control Engineering (ICECE), 2011 International Conference. Ramada Yichang Hotel Yichang, China, September 16-18 2011. [Çevrim-içi: <https://goo.gl/K4fc3a>, Erişim Tarihi: 14.07.2016.]
- Lai, C. S., & Lai, M. H. (2012). *Using computer programming to enhance science learning for 5th graders in Taipei*. In Computer, Consumer and Control (IS3C), 2012 International Symposium. 4-6 June 2012. [Çevrim-içi: 10.1109/IS3C.2012.45, Erişim tarihi: 19.07.2016.]
- Linn, M. C., & Clancy, M. J. (1992). The case for case studies of programming problems. *Communications of the ACM*, 35(3), 121-132.
- McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B. D., & Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin*, 33(4), 125-180.
- Menge, G. (2012). *Estonya'da mini mini birlere bile programlama öğretilecek*. [Çevrim-içi: <https://goo.gl/CcvHCs>, Erişim tarihi: 10.05.2017.]
- Monroy-Hernandez, A., & Resnick, M. (2008). Empowering kids to create and share programmable media. *ACM Digital Library*, 15(2), 50-53.
- Nam, D., Kim, Y., & Lee, T. (2010). *The effects of scaffolding-based courseware for the scratch programming learning on student problem solving skill*. 18th International Conference on Computers in Education. Putrajaya, Malaysia, November 29- December 3, 2010. [Çevrim-içi: <https://goo.gl/ZylMej>, Erişim tarihi: 30.08.2016.]
- Nikiforos, S., Kontomaris, C., & Chorianopoulos, K. (2013). MIT Scratch: A Powerful tool for improving teaching of programming. *Conference on Informatics in Education*, 1-5.
- Nikou, S. A., & Economides, A. A. (2014). *Transition in student motivation during a scratch and an app inventor course*. In 2014 IEEE Global Engineering Education Conference (EDUCON). Military Museum and Cultural Center, Harbiye, Istanbul, Turkey, 3-5 April 2014. [Çevrim-içi: <https://goo.gl/xT8X8C>, Erişim tarihi: 03.09.2016.]
- Olson, P. (2012). *Why Estonia has started teaching its first-graders to code*. [Çevrim-içi: <https://goo.gl/qx6aN4>, Erişim tarihi: 02.08.2016.]
- Ozoran, D., Çağiltay, N. E., & Topallı, D. (2012). Using scratch in introduction to programming course for engineering students. *2nd International Engineering Education Conference (IEEC2012)*, 2, 125-132.
- Öçalan, H. (2015). *Avustralya'da kodlama dersi ilköğretim müfredatına girdi*. [Çevrim-içi: <https://goo.gl/nMRc7K>, Erişim tarihi: 01.06.2017.]
- Öndeş, Ö. (2016). İngiltere ve ABD'de kodlama eğitimi. [Çevrim-içi: <https://goo.gl/Queu9Hd>, Erişim tarihi: 02.08.2016.]
- Özdemir, A. (2015). *Kodlama, okuma yazma bilmek kadar önemli!* [Çevrim-içi: <https://goo.gl/jLSyXp>, Erişim tarihi: 07.10.2016.]
- Özmen, B., & Altun, A. (2014). Undergraduate students' experiences in programming: difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry*, 5(3), 9-27.
- Pae, R. D., & Kurland, D. M. (1983). *On the cognitive prerequisites of learning computer programming (Tech. Rep. No. 18)*. New York: Bank Street College of Education, Center for Children and Technology.

- Pea, R. D., & Kurland, D. M. (1987). On the cognitive effects of learning computer programming: A critical look. *New Ideas Psychology*, 2(2), 137-168.
- Pinto, A., & Escudeiro, P. (2014). *The use of Scratch for the development of 21st century learning skills in ICT*. In Information Systems and Technologies (CISTI), 9th Iberian Conference. [Çevrim-içi:10.1109/CIS-TI.2014.6877061, Erişim tarihi: 11.06.2016.]
- Piteira, M., & Costa, C. (2013, July). *Learning computer programming: study of difficulties in learning programming*. In Proceedings of the 2013 International Conference on Information Systems and Design of Communication (pp. 75-80).
- Ruf, A., Muhling, A., & Hubwieser, P. (2014). *Scratch vs. Karel: Impact on learning outcomes and motivation*. Paper presented at the Proceedings of the 9th Workshop in Primary and Secondary Computing Education, Berlin, Germany, 5-7 November 2014. [Çevrim-içi: <http://dl.acm.org/citation.cfm?id=2670772>, Erişim tarihi: 15.07.2016.]
- Shin, S., Park, P., & Bae, Y. (2013). The effects of an information-technology gifted program on friendship using scratch programming language and clutter. *International Journal of Computer and Communication Engineering*, 2(3), 246.
- Sivasakthi, M., & Rajendran, R. (2011). Learning difficulties of object-oriented programming paradigm using Java: students' perspective. *Indian Journal of Science and Technology*, 4(8), 983-985.
- Soylu, Y., & Soylu, C. (2006). Matematik derslerinde başarıya giden yolda problem çözenin rolü. *Eğitim Fakültesi Dergisi*, 7(11), 97-111.
- SSRVM (2007). *Model curriculum and teaching material for K-12 Indian schools*. [Çevrim-içi: <http://www.ititb.ac.in/~sri/papers/SSRVM-CS-March07.pdf>, Erişim tarihi: 01.08.2017.]
- Tan, P. H., Ting, C. Y., & Ling, S. W. (2009). Learning difficulties in programming courses: undergraduates' perspective and perception. *Computer Technology and Development*, 1, 42-46.
- Taylan, E. (2013). *Kod okuryazarlığında genç yaşta eğitimin önemi ve online platformların etkisi*. [Çevrim-içi: <http://webrazzi.com/2013/03/05/kod-okuryazarligi/>, Erişim tarihi: 10.08.2016.]
- TBD (2014). *Programlama çocuk oyuncağı*. [Çevrim-içi: <https://goo.gl/B4DRk5>, Erişim tarihi: 01.08.2017.]
- Türkiyevodafonevakfı (2016). *Türkiye Vodafone Vakfı ve Habitat "Yarını Kodlayanlar" projesine hız verdi*. [Çevrim-içi: <http://www.turkiyevodafonevakfi.org.tr/tr/haberler/yarini-kodlayanlar>, Erişim tarihi: 10.04.2017.]
- Wang, H. Y., Huang, I., & Hwang, G. J. (2014). *Effects of an integrated Scratch and project-based learning approach on the learning achievements of gifted students in computer courses*. In Advanced Applied Informatics (IIAIAI), 2014 IIAI 3rd International Conference. 31 Aug.-4 Sept. 2014. [Çevrim-içi: <https://goo.gl/mqp9Pt>, Erişim tarihi: 20.10.2017.]
- Wyffles, F., Martens, B., & Lemmens, S. (2014). *Starting from scratch: Experimenting with computer science in flemish secondary education*. Proceedings of the 9th Workshop in Primary and Secondary Computing Education ACM. Berlin, Germany, November 05-07 2014. [Çevrim-içi: 10.1145/2670757.2670763, Erişim tarihi: 15.07.2017.]
- Yüzak, Ö. (2016). *Kodlama yapacak çocuk yetiştirecek*. [Çevrim-içi: <https://goo.gl/4zdSk3>, Erişim tarihi: 01.08.2016.]
- Zekaküpü (2016). *Türkcell zekâ küpü projesi*. [Çevrim-içi: <http://www.zekâ-kupu.com>, Erişim Tarihi: 08.04.2017.]