

ÖĞRETMEN ADAYLARININ KODLAMA EĞİTİMİNE YÖNELİK GÖRÜŞLERİ: BİR DURUM ÇALIŞMASI

PRE-SERVICE TEACHERS' VIEWS RELATED TO COMPUTING EDUCATION: A CASE STUDY

Aslı GÖNCÜ¹

İbrahim ÇETİN²

Ercan TOP³

Başvuru Tarihi: 14.08.2017 Yayına Kabul Tarihi: 09.06.2018 DOI: 10.21764/maeuefd.334560

Özet: Kodlama eğitiminde son yıllarda hızlı değişimler gözlenmektedir. Birçok ülke müfredatlarına kodlama ile ilgili dersler eklemektedir. Gerçekleşen hızlı değişim araştırmacılar için birçok soruyu beraberinde getirmektedir. Bu bağlamda öğretmenlerin ve öğretmen adaylarının kodlama eğitimine yönelik görüşlerinin belirlenmesi önem arz etmektedir. Alanyazında öğretmen adaylarının kodlama eğitimine yönelik görüşleri hakkında yeteri kadar çalışma bulunmamaktadır. Bu doğrultuda bu çalışmada Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümü öğretmen adaylarının kodlama eğitimine yönelik düşüncelerini keşfetmek amaçlanmaktadır. Çalışmanın amacını gerçekleştirmesine yönelik olarak 12 (3 birinci sınıf, 3 ikinci sınıf, 3 üçüncü sınıf ve 3 dördüncü sınıf) Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümü öğretmen adayından durum çalışması yöntemi ile nitel veri elde edilmiştir. Çalışmada veriler araştırmanın amacı dikkate alınarak geliştirilen yarı-yapılandırılmış görüşme tekniği ile toplanmıştır. Elde edilen verilerin analizinde nitel içerik analizi kullanılmıştır. Araştırma sürecinde elde edilen bulgular 9 farklı kategori altında değerlendirilmiştir. Araştırma bulgularına genel olarak bakıldığında öğretmen adaylarının kodlama eğitimine yönelik görüşlerinin sınırlı olduğu gözlenmiştir. Öğretmen adaylarının kodlama eğitiminin temel yapılarından sadece problem çözme ve algoritmik düşünmeden bahsettikleri görülmüştür. Buna ek olarak öğretmen adaylarının problem çözme ve algoritmik düşünmeden bahsederken onları bütün bileşenleri ile ele aldıklarını iddia etmek güçtür.

Anahtar Sözcükler: *Bilgi-işlemsel düşünme, kodlama eğitimi, öğretmen adayları, durum çalışması*

Abstract: There have been important advancements in computing education. Many countries have added computing related courses to their curriculum. This rapid change brings many questions for researchers. In this context, it is important to determine the views of teachers and pre-service teachers towards computing education. The literature seems void of studies reporting on pre-service teachers' views on computing education. In this study, it is aimed to explore information technology and software pre-service teachers' views related to computing education. For this purpose qualitative data have been obtained from 12 (3 first grade, 3 second grade, 3 third grade and 3 fourth grade) pre-service teachers through case study. In the study the data were collected by a semi-structured interview protocol. Qualitative content analysis was used in the analysis of the obtained data. Findings obtained in the research process resulted in under 9 different categories. Depending on research findings it can be said that opinions of pre-service teachers related to computing education are limited. It has been seen that pre-service teachers mentioned only about problem solving and algorithmic thinking related to computing education.

Keywords: *Computational thinking, computing education, pre-service teachers, case study.*

¹ Yüksek Lisans Öğrencisi, Bolu Abant İzzet Baysal Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü, goncuasl@gmail.com, ORCID ID: 0000-0003-1696-5248

² Doç. Dr., Bolu Abant İzzet Baysal Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü, ibretin@ibu.edu.tr, ORCID ID: 0000-0001-5127-0471

³ Doç. Dr., Bolu Abant İzzet Baysal Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü, ercantop@gmail.com, ORCID ID: 0000-0001-7175-8677

Giriş

Bilgi-işlemsel düşünme kavramını ilk kullanan Seymour Papert'tır (Papert ve Harel, 1991) ancak bu kavramın tanımını ilk yapan kişi olarak Wing'ten (2006) sıklıkla bahsedilmektedir. Wing'e (2006, s.33) göre "bilgi-işlemsel düşünme, bilgisayar biliminin kavramlarını kullanarak problem çözme, insan davranışlarını anlama ve sistem tasarlama üzerinde durur. Cundy, Synder ile birlikte 2010 yılında bu tanımı değiştirerek bilgi-işlemsel düşünmeyi, "çözümlerin bir bilgi-işleme birimi tarafından etkili şekilde yerine getirilebilecek bir formda sunulması amacıyla problemleri ve çözümlerini formülleştirmeyi içeren düşünce sürecidir" şeklinde tanımlamışlardır (Cuny, Snyder & Wing, 2010).

Bilgi-işlemsel düşünmeyle ilgili çalışmalar, Logo programlama dilinin inşa edildiği 1960'lı yıllara kadar uzanır (Feurzeig ve Papert, 2011). Bu dilin temel amacı, öğrencilerin matematiksel ve mantıksal düşünme yollarını öğrenmelerine yardımcı olmaktır. Logo programlama dilinin öğrenciler tarafından kolaylıkla öğrenileceği varsayılmıştır. Logo kullanımının arkasındaki eğitim teorisi olan inşacılık (constructionism), yapılandırmacılığın bir yorumudur. Yapılandırmacılık bireylerin tecrübe yoluyla bilgiyi inşa ettiğini ileri sürmektedir. Piaget (1964, s.176-186) bu konudaki düşüncelerini "bilmek, nesneyi dönüştürmek ve bu dönüşüm süreci ile nesnenin nasıl oluşturulduğunu anlamanın bir sonucudur" şeklinde ifade etmiştir. Logo dili eğitimi dönüştürmek için potansiyel bir araç olarak görülmesine rağmen, bu hedefi gerçekleştirilmede başarılı olamamıştır (Agalianos, Noss, Whitty, 2001).

Wing aynı zamanda bilgi-işlemsel düşünmenin sadece bilgisayar bilimcileri tarafından değil herkes tarafından bilinmesi gerektiğini iddia etmiştir (Wing, 2006). Wing'in iddiaları bilgi-işlemsel düşünme için yeni bir ivme kazandırmıştır. Öğrencilerde bilgi-işlemsel düşünmeyi geliştirmek amaçlı kullanılan bilgisayar programlama ve kodlama dersleri A-12 eğitim düzeyinde giderek popüler hâle gelmiştir (Kafai, Burke ve Resnick, 2014). Bilgisayar bilimi öğretmenleri derneği 2011'de ABD'de A-12 bilgisayar bilimleri standartlarını yayımlamıştır (CSTA Standards Task Force, 2011) ve bu standartları 2017 yılında güncellemiştir (CSTA Standards Task Force, 2017). Kodlama eğitimi dünyada birçok gelişmiş ülkede temel alanlardan biri olarak düşünölmeye başlanmıştır ve zorunlu veya seçmeli ders olarak müfredatlarında yer almaya başlamıştır (Department for Education, 2013). Türkiye'de ise 2013 yılından itibaren 5. ve 6. sınıflarda bilişim teknolojileri ve yazılım dersi içerisinde zorunlu olarak verilmektedir (MEB, 2017). A-12 eğitim müfredat değişikliklerine

ek olarak, programlamayı teşvik eden bazı uluslararası etkinlikler de vardır. Örneğin “Hour of Code ” etkinliğine 2016 yılında 180’den fazla ülkeden 100 milyondan fazla öğrenci ulaşmıştır (Hour of code, 2016).

Öğrencilerin teknolojiyi kullanmayı öğrenmesi artık yeterli olarak görülmemektedir. Öğrencilerin teknolojinin nasıl oluşturulduğunu bilmelerinde bilgi-işlemsel düşünmeyi anlamalarının önemli olduğu iddia edilmektedir (Kafai, Burke ve Resnick, 2014). Peki neden bilgi-işlemsel düşünmeyi anlamamız önemlidir? Öğrencilere kodlama öğretilmesinin sebebi onların hepsini bilgisayar mühendisi olmaları için yönlendirmek değildir. Amaç öğrencilerin sadece kodlamanın nasıl yapılacağını öğrenmekle kalmayıp, öğrenme için kodlamayı kullanabilmelerini sağlamak, bilgi-işlemsel düşünme becerisini geliştirip uygulayabilme yeteneğini kazandırmak, teknoloji çağına ayak uydurmalarını sağlamaktır (Çetin ve Toluk Uçar, 2017; Wing, 2011; Selby ve Woollard; 2013). Avustralya ve Yeni Zelanda mühendislik direktörü bu durumu, “bir akıllı telefon kullanma ve milyonlarca insana ulaşan bir uygulama oluşturma arasında fark vardır” sözleri ile açıklamıştır (Noble, 2012).

Araştırmacılar, ülke ekonomilerine önemli katkılar sağlayan (Zyda, 2009) bu bilim dalının öğrenciler tarafından öğrenilmesi konusunda farklı bakış açılarına sahiptirler. Ancak bu araştırmacılar öğrencilerin programlama öğrenirken birtakım zorluklar ile karşılaştıkları konusunda hemfikirdirler (Çetin, 2013). Öğrencilerin yaşadıkları bu zorluklar DuBoulay (1986) tarafından beş başlık altında toplanmıştır. Birincisi programlama öğrenmeye yeni başlamış olan bireylerin bu yeni duruma adaptasyonu (orientation) ile ilgilidir. Öğrenciler burada programlamanın ne olduğu ve ne işe yaradığı ile ilgilenir. İkincisi kuramsal makinedir. Kuramsal makine öğrencinin beyinde oluşturduğu ve bir programın bilgisayar donanımı tarafından nasıl çalıştırıldığı ile ilgilenen bir modeldir. Üçüncü olarak belirli bir programlama dilinin sözdizimi (syntax) ve semantiği (semantics) bulunur. Dördüncüsü küçük çaplı problemlerin standartlaşmış çözümleridir. Son başlık ise programlama sürecini genel olarak ele alır; örneğin planlama, geliştirme, test etme ve hata ayıklama.

Son zamanlarda bilgi-işlemsel düşünmeye artan ilgi araştırmacılara önemli bir soru yöneltir: *Öğrencilere temel bilgisayar kavramlarını öğretmenin en iyi yolları nelerdir?* Literatürün öngördüğü şekilde kodlama eğitime giriş için üç önemli yönelim bulunmaktadır; görsel programlama, robotik ile programlama ve bilgisayarı hiç kullanmadan kodlama kavramlarını anlatmak olan bilgisayarsız bilgisayar bilimi (computer science unplugged)(Bower & Falkner, 2015).

A-12 düzeyindeki öğrenciler, görsel programlamada blok tabanlı kodlama ortamlarında (Scratch, Code.org gibi programlama yazılımlarıyla) bir program oluşturmak için görsel blokları sürükleyip bırakarak oyun, animasyon ve dijital hikâye gibi ürünler oluştururlar. Robotik yaklaşımında öğrenciler MakeBlock, Arduino gibi robotik donanımlarını farklı programlama dilleri kullanarak işlevsel hâle getirirler. Robotik donanımlar öğrencilerin yaş seviyesine göre değişebilen, tak çıkar parçalar içeren veya mikro denetleyici kartlara sahip, uzaktan kumandalı, kolay sensör monte edilebilme ve programlanabilme gibi özelliklere sahiptirler (Kalelioğlu ve Keskinliç, 2017). Bilgisayarsız bilgisayar biliminde öğrenciler, genellikle bilgisayarlarda karşılaştığımız dikkat dağıtıcılar ve teknik detaylardan arındırılmış bir şekilde, ikili sayı, algoritma ve veri sıkıştırma gibi temel kavramları ilgi çekici oyunlar, bulmacalar, kartlar ve boya kalemleri gibi çeşitli öğrenme etkinlikleri ile öğrenirler. Öğrenenler bilgisayarsız bilgisayar bilimi etkinliklerinin hiçbirinde bilgisayar kullanmadan, aktif öğrenme gerçekleştirirler. Ayrıca öğrenenlerin yapılan bütün etkinliklerde bilgisayar bilimi kavramlarını öğrenmeleri sağlanır (Kalelioğlu ve Keskinliç, 2017).

Bilgisayar bilimi eğitiminde Dünya’da ve Türkiye’de hızlı bir değişim yaşanmaktadır. Bu yaşanan değişim içinde yeni müfredatların hazırlanması, birçok öğretmenin yetiştirilmesi ve sınıf ortamlarında değişiklikler yapılması gerekmektedir. Bu yüzden bilgi-işlemsel düşünmeyi öğreten öğretmenleri eğitmek bulmacanın önemli bir parçasıdır. Öğretmenlerin bilgi-işlemsel düşünme pedagojilerini etkili bir şekilde geliştirmeleri için mesleki gelişim son derece önemlidir (Barr ve Stephenson, 2011). Bu durumda bilgisayar bilimi araştırmacılarına öğretmen eğitime yönelik roller düşmektedir. Henüz kodlama eğitimi alanında Türkiye’de çalışmalar yeni yeni yapılmaktadır (Yecan, Özçınar ve Tanyeri, 2016). Alanyazında bilişim teknolojileri ve yazılım dersi öğretmenlerinin, görsel programlama öğretimi deneyimleri ve bilgisayar programlama öğretimine yönelik görüşleri yönüyle çalışmalar bulunmaktadır (Yükseltürk ve Altıok, 2015; Uzgur ve Aykaç, 2016). Derse girecek olan ve hazırlanan müfredatları uygulayacak olan kişiler öğretmenlerdir. Bu durumda öğretmenlerin bu konuda ne düşündüklerini ortaya çıkarmak önem taşımaktadır. Bu çalışmada Abant İzzet Baysal Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümü öğretmen adaylarının kodlama eğitime yönelik düşüncelerini ortaya çıkarmak amaçlanmaktadır. Bu amaca yönelik olarak şu alt boyutlar incelenmiştir:

- (i) bilişim teknolojileri ve yazılım dersi öğretmen adaylarına göre kodlama eğitiminin temelleri,

- (ii) bilişim teknolojileri ve yazılım dersi öğretmen adaylarına göre kodlama eğitiminin bileşenleri,
- (iii) bilişim teknolojileri ve yazılım dersi öğretmen adaylarına göre kodlama eğitiminin bireye ve topluma katkısı,
- (iv) bilişim teknolojileri ve yazılım dersi öğretmen adaylarına göre öğrencilere kazandırılması gereken kavram ve beceriler ve bu kavram ve becerilerin kazandırılmasında kullanılacak araçlar.

Yöntem

Nitel araştırma yöntemlerinden durum çalışması, belirli bir zaman diliminde gerçekleşen durumların, kendi bağlamıyla bir bütün hâlinde olduğu kesin sınırlarının çizilmesinin zor olduğu, çoklu kaynakları içeren veri toplama araçları (görüşmeler, gözlemler, dokümanlar, raporlar) ile derinlemesine incelenen durumların tanımlandığı bir araştırma yöntemidir (Yin,1984; Creswell ve Plano Clark, 2007; Yıldırım ve Şimşek, 2013). Bu yöntemin en belirgin özelliği incelenen kişinin, topluluğun ya da durumun kendisine has özellikleri nedeniyle kendi bağlamı içerisinde incelenmesidir (Ersoy, 2016). Bu nedenle araştırmanın amacı evren üzerinde genelleme yapmak değil; kişinin, topluluğun ya da durumun ayırt edici özelliklerini anlayabilmek ve açıklayabilmektir. Bu doğrultuda durum çalışması bu araştırmanın hedefini gerçekleştirmede uygun bir yöntem olarak görülmektedir. Durum çalışmaları, ‘ne’, ‘nasıl’ ve ‘niçin’ sorularını temel alıp, araştırmacının kontrol edemediği bir olgunun ya da olayın ayrıntılı bir şekilde incelenmesine olanak verdiğinden (Yıldırım ve Şimşek, 2013), gerçekleştirilmekte olan çalışmada yorumlayıcı bir yaklaşımla “ne oluyor?” sorusuna cevap aranmıştır.

Katılımcılar

Araştırma verileri, Abant İzzet Baysal Üniversitesi’nden 12 (3 birinci sınıf, 3 ikinci sınıf, 3 üçüncü sınıf ve 3 dördüncü sınıf) Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümü (5 kız ve 7 erkek) öğretmen adayından elde edilmiştir. Katılımcıların seçiminde maksimum çeşitliliğin sağlanması temel alınmıştır. Burada amaç “görelilik olarak küçük bir örneklem oluşturmak ve bu örnekte çalışılan probleme taraf olabilecek bireylerin çeşitliliğini maksimum derecede yansıtmaktır” (Yıldırım ve Şimşek, 2013, s.136). Bunun için gönüllüler arasından her sınıf kademesinden 3 öğretmen adayı tamamen rastgele bir yöntem ile seçkisiz olarak belirlenmiştir. Amaçlı örnekleme zengin veriye sahip olduğu düşünülen olgu, olay ve durumların derinlemesine çalışılarak açıklanmasına olanak vermektedir (Yıldırım ve Şimşek,

2013).

Veri Toplama Aracı

Alanyazında görüşme türleri araştırmacılar tarafından çeşitli şekillerde gruplandırılmıştır. Ancak genel olarak görüşmeler yapılandırılmış, yarı-yapılandırılmış ve yapılandırılmamış olmak üzere üç ana kategori altında toplanmıştır (Merriam, 1998). Araştırmada veriler yarı-yapılandırılmış görüşme tekniği ile toplanmıştır. Kullanılan yarı-yapılandırılmış görüşme tekniğinde araştırmacı görüşme esnasında önceden hazırlamış olduğu soruları sorarken konu ile ilgili daha ayrıntılı bilgi almak amacıyla ek sorular sorma özgürlüğüne sahiptir.

Görüşmeler araştırmacının amacı dikkate alınarak geliştirilen yarı-yapılandırılmış görüşme formu kullanılarak gerçekleştirilmiştir. Görüşme formu Bilişim Teknolojileri ve Yazılım dersi öğretmen adaylarının kodlama eğitimine yönelik bilgi ve görüşlerini belirlemek amacıyla oluşturulmuştur. Görüşme formunda yer alan toplamda 11 soru oluşturulurken iki kaynaktan yararlanılmıştır. Bunlar alanyazın taraması sonucunda elde edilen ve araştırmacıların kendi ürettikleri sorulardır. Hazırlanan sorular bir alan uzmanı tarafından incelendikten sonra sorular üzerinde gereken değişiklikler yapılmıştır. Ardından Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümünden 3 öğrenci ile pilot çalışma yapılarak görüşme soruları üzerinde gerekli düzenlemeler yapılmış ve veri toplamada kullanılan görüşme sorularına son hali verilmiştir. Aşağıda görüşme formundan iki örnek soru bulunmaktadır:

Örnek1: Sizce kodlama eğitimi nedir? Size göre kodlama eğitimi ne ifade ediyor?

Örnek2:Kodlama eğitiminin bütün öğrencilere zorunlu olarak verilmesi konusunda ne düşünüyorsunuz?

Sonda: Kodlama eğitimine başlangıç yaşı hakkında ne düşünüyorsunuz?

Sonda: Sizce şu anda verilen programlama / kodlama eğitimi yeterli midir?

Veri toplama sürecinde her bir katılımcı ile ortalama 15 dakika süren görüşmeler ses kayıt cihazı ile kayıt altına alınmıştır. Ardından görüşmeler yazılı dökümleri çıkarılarak deşifre edilmiştir.

Veri Analizi

Görüşmeler sonrasında toplanan verilerin analizi için nitel içerik analizi kullanılmıştır. Nitel içerik analizinde amaç elde edilen verileri betimleyebilecek kavramlara ve ilişkilere ulaşmaktır. Nitel içerik analizinde verilerin kodlanarak kategorileştirilmesi, temaların bulunması, verilerin kodlara ve temalara göre düzenlenmesi ve tanımlanması ve bulguların

yorumlanması aşamaları birbirini izlemektedir (Yıldırım ve Şimşek 2013). Nitel içerik analizi sistematik bir yapıya sahiptir ancak evrensel kabul edilmiş bir yapısı olmadığı için araştırmacılar farklı adımlar izleyerek veri analizini gerçekleştirebilirler (Çetin, 2016a).

Elde edilen verilerin analizine görüşme kayıtlarının çözümlenerek yazılı hale getirilmesi ile başlanılmıştır. Yazılı dökümü yapılan veriler incelenerek üç araştırmacı tarafından kodlama tablosu oluşturulmuştur. Ardından iki araştırmacı tarafından kodlama tablosu kullanılarak verinin yüzde onluk bir kısmı ayrı ayrı kodlanmıştır. Bu süreç sonunda %91 oranında araştırmacılar arası kodlama uyumu (Miles ve Huberman, 1994) elde edildikten sonra ortak olmayan kodlar üzerinde tartışarak araştırmacılar arası uyum %100'e çıkarılmıştır. Bu aşamadan sonra ise verinin tamamı bir araştırmacı tarafından kodlanmıştır.

Bulgular

Araştırma sürecinde elde edilen bulgular 9 farklı kategori altında değerlendirilmiştir. Bu kategoriler; kodlamanın amacı, kodlamanın bireye katkısı, kodlama eğitiminin topluma yansımaları, herkes için kodlama, kodlama eğitiminin yeterliliği, kodlama eğitimine başlangıç yaşı, kodlama için temel kavram ve beceriler, programlama için yazılım/donanım gereksinimi, programlama gelişmelerinin takibi şeklinde sıralanmıştır. Tablo 1'de kategoriler ve alt kategorilerdeki yanıt sayısı bulunmaktadır. Çalışmaya 12 öğretmen adayı katılmıştır fakat bazı durumlarda bir öğretmen adayı birden fazla kategori altında gruplandırılmıştır. Bu durumun bulunduğu kategorilerde toplam cevap sayısı 12'yi geçmiştir.

Tablo1

Kategoriler ve alt kategoriler

Kodlamanın amacı	Kodlama eğitimine başlangıç yaşı
Kodlamanın bilişsel becerilere katkısı	Küçük yaşlarda başlanılmalı
Kodlamanın bireyler için amacı	6. veya 7. Sınıfta başlanılmalı
Kodlamanın mantığını öğretmek	
Kodlamanın bireye katkısı	Kodlama için temel kavram ve beceriler
Kodlamanın bilişsel becerilere katkısı	Bilişsel beceriler
Kodlamanın iş hayatına katkısı	Problem çözme becerisi
Kodlamanın günlük yaşama katkısı	Derste uygulama becerisi
Kodlama eğitiminin yeterliliği	Programlama gelişmelerinin takibi
Yeterli değil	Yeri
Yeterlidir	Sıklığı
Bilgim yok	Nedeni
Programlama için yazılım/donanım gereksinimi	Herkes için kodlama
Görsel kodlama yazılımları	Herkese verilmeli
Robotik yönelimli yazılımlar	Zorunlu değil seçmeli
Donanımlar	

Aşağıda araştırma sürecinde elde edilen bulgular araştırmanın temaları doğrultusunda sunulmaktadır.

Kodlamanın Amacı

Öğretmen adaylarının kodlama eğitiminin temel amacına yönelik görüşlerinden; *kodlamanın bilişsel becerilere katkısı*, *kodlamanın bireyler için amacı*, *kodlamanın mantığını öğretmek* alt kategorileri ortaya çıkmıştır. Öğretmen adaylarının bu başlıkta en çok altında toplandığı kategori, kodlamanın bilişsel becerilere katkısıdır. Kodlama eğitiminin amacını *kodlamanın bilişsel becerilere katkısı* başlığı altında dile getiren 7 öğretmen adayı kodlamayı insanın bilişsel becerilerini geliştirme bağlamında düşünmüşlerdir. Bu konuda öğretmen adayları; soyut düşünceyi somutlaştırma, algoritmik düşünme ve stratejik düşünme becerisi kazandırma gibi ifadelerden bahsetmişlerdir. Aşağıdaki alıntıda Ö12, kodlamanın amacının günlük yaşam problemlerini çözüme farklı bakış açısı sağlamak olduğunu ifade etmektedir.

Ö12: "...sadece derslerle ya da işte dersteki sorunlarla ilgili değil hani normal yaşamlarında da karşılaştıkları problemle ilgili stratejik düşünme, yani çok farklı yönlerden düşünerek çözüme ulaşma yöntemleri kazanılabilir. Hani bu tip sorunun bir çözümü var yerine hani şöyle de çözebilir miyim şeklinde düşünme becerisi kazandırdığını düşünüyorum."

Kodlamanın bireyler için amacı başlığı altında toplanan 4 öğretmen adayı kodlamanın öğrenciler tarafından neden bilinmesi gerektiğini dile getirmişlerdir. Aşağıdaki alıntı da (Ö8) kodlamanın amacı kendi açısından ifade edilmektedir.

Ö8: "Kodlama eğitimi küçükler için daha çok programlama mantığını öğretmek olmalı, büyükler içinse temel bir yani programlamanın temelini oluşturmak olmalıdır."

Kodlamanın mantığını öğretmek başlığı altında toplanan 4 öğretmen adayı kodlamanın amacını, bireylerin kodlama ile neler yapabileceğinin farkına varmasını sağlamak için kodlamanın mantığını öğretmek şeklinde düşünmüşlerdir. Aşağıdaki alıntı da (Ö10) kodlamanın amacının çocukların bilgisayar ile neler yapabileceğinin farkına varmasını sağlamak olduğunu ifade ediyor.

Ö10: "Bence kodlama eğitiminin temel amacı öğrencilerin ... bilgisayar sadece bir oyun ne bileyim vakit geçirme aracı olarak kullanmalarını değil de hani

bilgisayarda neler yapabileceklerini göstermeye yarar, bence kodlama dersi mesela oyunun yerine bilgisayarda bir çocuk karşısındaki oyuncağı hareket ettirebilir.”

Kodlamanın Bireye Katkısı

Veriler incelendiğinde öğretmen adaylarının kodlama eğitiminin bireye katkısına ilişkin görüşlerinden; *kodlamanın bilişsel becerilere katkısı*, *kodlamanın iş hayatına katkısı* ve *kodlamanın günlük yaşama katkısı* alt kategorileri ortaya çıkmıştır. Öğretmen adaylarının en çok altında toplandığı kategori kodlamanın bilişsel becerilere katkısı kategorisidir. *Kodlamanın bilişsel becerilere katkısı* başlığı altında toplanan 9 öğretmen adayı kodlamanın bireye katkısını insanın bilişsel becerilerine katkısı bağlamında düşünmüşlerdir. Bu konuda öğretmen adayları; problem çözme becerisi, analitik düşünme ve sayısal düşünme gibi ifadelerden bahsetmişlerdir. Aşağıdaki alıntı da (Ö4) kodlamanın bireylere katkısının, problem çözme becerisini geliştirme ve mantıklı düşünme konularında olduğu görüşündedir.

Ö4: “...daha çok mantıklı düşünebiliyorum. Kodlamam işte lisede başladım öğrenmeye, temelimi oturtuktan sonra hani çoğu programı çok daha kolay yapabiliyorum, problemleri daha kolay çözüyorum, bulmacaları falan”.

Kodlamanın bireye katkısını *kodlamanın iş hayatına katkısı* başlığı altında dile getiren 4 öğretmen adayı iyi bir programcı olarak yazılım sektöründe çeşitli iş alanlarında çalışma olanağına sahip olunabileceğinden bahsetmişlerdir. Bu konuda öğretmen adayları; şirketlerde yönetici olunabileceği, program yazarak ciddi paralar kazanılabileceği, siber saldırılara karşı güvenlik alanında çalışılabileceği şeklinde ifadelerde bulunmuşlardır. Aşağıdaki alıntı da (Ö10) kodlamanın bireye katkısını, kodlama eğitimi sayesinde birçok alanda programcı olarak çalışılabileceği ve programlama ile ciddi paralar kazanılabileceği şeklinde ifade etmektedir.

Ö10: “siber güvenlik; yani kodlamayı çocuklardan temelden öğrendikçe ülkenin ... siber saldırılara karşı bir güvenlik açığı oluyor ya çocuklar o alanlarda çalışabilirler mesela ilk şeyi bu. Ondan sonra çocuklar gelecekte herhangi bir program yazıp bunla ilgili ticarî bir şeyler yapabilirler, yani para kazanabilirler çok ciddi paralar da kazanabilirler.”

Kodlamanın bireye katkısını, *kodlamanın günlük yaşama katkısı* başlığı altında ifade eden 3 öğretmen adayı, bu katkıyı insan hayatını kolaylaştırma şeklinde düşünmüşlerdir. Aşağıdaki alıntı da (Ö3) yazılan programlar ile insanların hayatının olumlu yönde etkilenebileceğini

ifade etmektedir.

Ö3: “İşte günlük hayatta kullandığımız hastane otomasyon programlarıdır ya da büyük şirketlerin kullandığı programlardır ya da tıpta kullanılan programlardır falan. Bunların gelişmesi tabii insan hayatını, hem bireyin kendini geliştirmesi hem de ilerisinde diğer insanların hayatını olumlu yönde etkileyecektir.”

Kodlama Eğitiminin Toplumsal Katkısı

Kodlama eğitiminin bireye olan katkısının topluma nasıl yansıtılacağına dair 7 öğretmen aday görüşlerini dile getirmiştir. Öğretmen adaylarının görüşlerine bakıldığında, genel olarak programlama, öğrenen bireylerin kendi gelişimlerine katkı sağlayarak topluma faydalı bireyler olabileceklerinin vurgulandığı görülmektedir. Kodlama eğitimi alan bireylerin günlük yaşamı kolaylaştıracak teknolojik gelişmelerde rol alarak toplumun gelişmesine katkı sağlayacaklarından bahsetmişlerdir. Örneğin; kodlamanın günlük yaşama etkisi, başkalarına kodlama öğretiminde yardımcı olma, yazılım, araç-gereç vb. kendimiz üretiriz, bir problem durumunda bilinen-bilinmeyen sorunlar ortaya konur. Aşağıdaki alıntı da (Ö7) kodlama eğitiminin topluma teknolojik gelişmeler olarak yansıtıp insanların hayatını kolaylaştıracağından bahsetmektedir.

Ö7: “Eğer ben kendi problemimi çözersem topluma da faydalı olacağımı düşünüyorum. Ya da mesela şöyle düşüncem, biraz mekatroniğe kayıcam tamam mı. Mesela sonuçta orda da bir programlama var. Orda nıpiyosun sen insanı hayatını ya da toplumun hayatını kolaylaştırmak için icatlarda bulunuyosun. Hani nedir? Bir bilgisayar işte bi buzdolabı, fırın ee bunların hepsinin algoritması var, hepsinin programlaması var ve bunlar topluma nasıl yansıyor? Kolaylık olarak yansıyor teknoloji olarak yansıyor, hadi biraz daha ileri boyuta bakıyorum mesela toplumsal gelişmeyi yansıtıyor bunların hepsi programlamadan geçiyo...”

Herkes İçin Kodlama

Kodlama eğitiminin bütün öğrencilere zorunlu olarak verilmesi konusunda öğretmen adaylarının görüşlerinden; *herkese verilmeli ve zorunlu değil seçmeli olmalı* alt kategorileri ortaya çıkmıştır. Kodlama eğitiminin herkese verilmesi gerektiği kategorisi 8, kodlama eğitiminin zorunlu değil seçmeli olması gerektiği kategorisi ise 4 defa vurgulanmıştır.

Kodlama eğitiminin bütün öğrencilere zorunlu olarak verilmesi konusunda *herkese verilmesi gerektiğini* savunan 8 öğretmen adayı bu konuda; kodlamanın mantığını anlamak bireylerin

birçok şeyi anlamasını sağlıyor, dünyanın eksenini bilgisayara kayıyor o yüzden temel derslere verilen önem kodlamaya da verilmelidir. Kodlama eğitimiyle ilgili temel bilgiler, algoritma mantığı zorunlu olarak verilmelidir gibi ifadelerden bahsetmişlerdir. Aşağıdaki alıntı da (Ö3) kodlama eğitiminin öncelikle sayısal zekâyâya sahip olan insanlara verilmesi gerektiğini daha sonra Scratch gibi programların sağladığı görsellik sayesinde sözel zekâyâya sahip olan insanlara da verilebileceğinden bahsediyor. Son olarak da kodlama eğitiminin herkese verilmesi gerektiğini, çünkü kimin nasıl bir farkındalık yaratacağının belli olmadığını için savunuyor.

Ö3: “Kodlama eğitimi yani kodlamanın, asıl mantığı sayısal yani daha önce okuduğumuz bilgilerden anladığımız kadarıyla sayısal düşünme yeteneği olan ya da sayısal zekâsı olan insanların kodlama derslerini ya da programlama derslerini daha iyi yaptıkları yönünde ... tabii sosyal becerilere ya da sosyal ne diyorduk biz ona... sözel düşünme becerisine sahip insanlar kodlama dersini alabilir mi? Şimdi kodlama dersi Scratch’larla ya da bugünkü kullanılan Scratch’lardır ya da farklı kullanılan programlar var hani hem görsel boyutta insanlara düşünme kabiliyeti sağlıyor hem de o masa başında sürekli şeyler yazma korkusunu önlerinden kaldırıyor. Önceye göre evet bütün insanlara verilmeli çünkü kimin nasıl bir farkındalık yaratacağı belirsiz...”

Kodlama eğitiminin bütün öğrencilere zorunlu olarak verilmesi konusunda *zorunlu değil seçmeli olması* gerektiğini savunan 4 öğretmen adayı kodlama eğitiminin ilgi duyan öğrencilere verilmesi gerektiğini belirtmişlerdir. Aşağıdaki alıntıda (Ö4) kodlama eğitiminin zorunlu değil seçmeli olması gerektiğini ve kodlama eğitimine ilgi duymayan bireylerin sıkılacağı ifade edilmektedir.

Ö4: “Ya zorunlu olarak değil de ilgisini çeken birisinin mesela seçmeli ders olarak verilebilir ilköğretimde artık seçmeli ders koydular, herhalde ortaokullarda falan mesela ortaokulda seçmeli ders olarak verilebilir... Yani zorunlu belki ya birey bilgisayarı sevmiyordur başka bir şey de mesela astronomi koymuşlar galiba, astronomi okumak istiyordur. Astronomi alıp da ona programlama dersi vermek anlamsız olabilir.”

Kodlama Eğitiminin Yeterliliği

Öğretmen adaylarının, Millî Eğitim müfredatına bağlı olarak okullarda verilen kodlama eğitiminin yeterli olup olmadığına ilişkin görüşlerine bakıldığında; 9 öğretmen adayı yetersiz, 1 öğretmen adayı yeterli ve 2 öğretmen adayı da bu konuda bilgilerinin olmadığını ifade

etmişlerdir.

Kodlama eğitiminin yeterli olup olmadığı konusunda *yeterli değil* başlığı altında düşüncelerini ifade eden 9 öğretmen adayı verilen kodlama eğitimini yetersiz görmelerinin nedenini ve bu konuda neler yapılması gerektiği açıklamışlardır. Öğretmen adayları kodlama eğitimini yeterli görmemelerinin nedeni olarak; derse gereken önemin verilmediğinden, müfredat ya da öğretmenden kaynaklı yetersizliklerden, dersin mantığının verilmeyip basitçe işlendiğinden bahsetmişlerdir. Bu konuda yapılması gerekenler konusunda ise dersin daha geniş kapsamlı anlatılması, güncel programların öğretilmesi, örneklerin günlük yaşamla ilişkilendirilmesi, “nedir?” değil, “ne için kullanıyoruz?” öğretilmeli şeklinde düşüncelerini dile getirmişlerdir. Verilen kodlama eğitiminin yeterli olduğunu dile getiren 1 öğretmen adayı Scratch gibi programlarla verilen kodlama eğitiminin 5. ve 6. sınıflar için yeterli olduğunu ifade etmiştir. 2 öğretmen adayı ise kodlama eğitiminin yeterli olup olmadığı konusunda bir bilgilerinin olmadığını söylemişlerdir. Aşağıdaki alıntıda (Ö8) müfredat ya da öğretmenden kaynaklı bir nedenden dolayı derse gereken önemin verilmediğini bu yüzden verilen kodlama eğitiminin yetersiz olduğu şeklinde bir açıklamada bulunulmuştur.

Ö8: “...ben yeterli olduğunu düşünmüyorum artık öğretmenin kalitesinden midir desem müfredattan mıdır desem bilemeyeceğim, ama yani şu andaki durumu beğenmiyorum... Müfredatta olanı da bilmiyorlar olmayanı zaten bilmiyorlar. Yani .. bu dersin biraz daha Matematik, Türkçe gibi önemsenmesi, biraz daha üstüne gidilmesi lazım. Sınıflara gidip te eskideki gibi böyle serbest bırakılıp oyun oynatılmaması lazım. Yani bu dersin birazcık daha üstüne gidilmesi önem verilmesi lazım.”

Kodlama Eğitime Başlangıç Yaşı

Öğretmen adaylarının kodlama eğitime başlangıç yaşı hakkındaki görüşlerinden; *küçük yaşlarda başlanılmalı* ve *6. veya 7. sınıfta başlanılmalı* olmak üzere 2 alt kategori ortaya çıkmıştır. Kodlama eğitime başlangıç yaşını, *küçük yaşlarda başlanılmalı* başlığı altında dile getiren 7 öğretmen adayı kodlama eğitime küçük yaşlarda başlanması gerektiğini; çocukların uzmanlaşmak istedikleri alanlara erken karar vermelerini sağlama, ilerleyen yaşlarda ülke gelişimine daha büyük katkılar sağlama ve bireyin kendini daha çok geliştirmesi gibi düşüncelerden bahsetmişlerdir. Aşağıdaki alıntı da (Ö3) kodlama eğitime küçük yaşlarda başlanması gerektiğinin nedenini öğrencinin farklı düşünce yapısına sahip olduğu erken yaşlarda fark edilirse becerisi olduğu alana yönlendirilmesinin daha kolay olacağı ve bu

sayede ilerleyen yaşlarda kendini daha iyi geliştirebileceği şeklinde ifade etmektedir.

Ö3: “Bence okumayı yazmayı çözdükten sonra yani 3. sınıftan sonra 4. sınıftan başlanabilir. Çünkü o yaşlardaki çocukların neler düşünebildikleri bizim için önemli ve yönlendirilmeleri daha kolay olacaktır yani öğrenci eğer bir şeyleri yapabiliyorsa ya da farklı düşünebiliyorsa diğer arkadaşlarına göre ... bu öğrenci kodlama dersini aldıktan sonra eğer öğretmenleri bunun farkına varırsa farklı düşündüğünün, bu yönde eğitime devam edilirse çok başarılı yani ilerde çok başarılı birey ortaya çıkabilir.”

Kodlama eğitimine başlangıç yaşını, 6. veya 7. sınıfta başlanmalı başlığı altında dile getiren 4 öğretmen adayı, 6. veya 7. sınıftaki öğrenciler somut ile soyutu ayırt edebildikleri dönemde olduklarından dolayı bu dönemdeki bireyleri düşünmüşlerdir. Bu konuda öğretmen adayları; matematiksel zekâ ile ilişkisinden dolayı 6. sınıfta başlanması, belirli becerileri yerine getirebilecek seviyede olmaları gerektiği, kodlama mantığını üniversiteye kadar unutmamak için 7. sınıfta başlanması gerektiğinden bahsetmişlerdir. Aşağıdaki alıntı da (Ö5) kodlama eğitimine 6. veya 7. sınıfta başlanılmasının nedenini öğrencilerin kodlamayı öğrenebilmesi için somut ile soyutu ayırt edebildiği zihinsel gelişim döneminde olması gerekir şeklinde ifade etmektedir.

Ö5: “...ortaokul döneminde verilmesi gerekiyor çünkü öğrenci somutla soyutu ayırt edemediğinde... çoğu şeyi muhtemelen hani zor yaşayacaktır ya da ne bileyim algılamakta çok zorluk çekecektir ve algılamakta çok zorluk çektiğin bir şey de biktirir ya da ona olan ilgin olsa da kaybolur.”

Kodlama İçin Temel Kavram Ve Beceriler

Öğretmen adaylarının kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve becerilerle ilgili görüşlerinden; *bilişsel beceriler*, *problem çözme becerisi* ve *derste uygulama becerisi* gerektiği alt kategorileri ortaya çıkmıştır. Öğretmen adaylarının kodlama eğitimi aracılığıyla öğrencilere kazandırılması gereken kavram ve becerilerle ilgili görüşlerini *bilişsel beceriler* kategorisi altında dile getiren 8 öğretmen adayı kazandırılması gereken kavram ve becerileri bireylerin bilişsel gelişimine katkı sağlayacak beceriler olarak düşünmüşlerdir. Bu konuda öğretmen adayları; analitik düşünme, soyut düşünme, algoritmik düşünme gibi ifadelerden bahsetmişlerdir. Aşağıdaki alıntı da (Ö7) algoritmik düşünme becerisinden bahsetmiştir.

Ö7: “Kavram ve beceriler kesinlikle algoritma mantığını öğrenmesi lazım çocuk... mesela algoritmanın mantığını kesinlikle öğrenmesi lazım. Algoritmik düşünme. Bunu kesinlikle yapması lazım...”

Öğretmen adaylarının kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve becerilerle ilgili görüşlerini *problem çözme becerisi* kategorisi altında dile getiren 5 öğretmen adayı bu becerinin bireylerin karşılaştıkları problemler ile başa çıkmada faydalı olacağı görüşündedirler. Bu konuda öğretmen adayları; alternatif çözümler, hız kazanma, problemin matematiksel kurgusu, farklı bakış açıları gibi ifadelerden bahsetmişlerdir. Aşağıdaki alıntı da (Ö10) probleme bakış açılarının değişebileceğini ifade etmektedir.

Ö10: “...problemlere bakış açıları değişir yani daha kolay çözebilir ya da...”

Öğretmen adaylarının kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve becerilerle ilgili görüşlerini *derste uygulama becerisi* kategorisi altında dile getiren 3 öğretmen adayı genel olarak derste öğrenilenleri uygulayabilme becerisinden bahsetmişlerdir. Aşağıdaki alıntı da (Ö7) bireylerin derste öğrendiklerini uygulayabilmesi gerektiğini ifade etmektedir.

Ö7: “...Zaten bizim programlama derslerimizde de aynı mantık var ilk önce gösteriliyor, anlatılıyor ondan sonra da biz uyguluyoruz. Ama bence en önemli burada ya dersin sonunda ya da ünitenin sonunda çocuğun elinde ya da bireyin elinde bir tane materyalin olması gerekiyor. Hani ben bunu öğrendim ama öğrendiklerimle ne yapabiliyorum ya da ben bir şey yapmak istiyorum öğrendiklerim buna yeterli mi? Hani şey olabilir dersten önce bi proje gösterilir hani bu haftaki dersten sonra bunu yapabileceksiniz öğrendiğiniz komutlarla. Öğrenci güdülenecek ondan sonra ona göre uygulama şansı olacak. Kazanılması gereken davranışların en önemlisi bence öğrendiklerini uygulayabilmesi becerilerden bu.”

Kodlamaya Giriş

Öğretmen adaylarının kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler için hangi yazılım ve donanımları kullanacakları ile ilgili görüşlerinden; *görsel kodlama yazılımları*, *robotik yönelimli yazılımlar*, *donanımlar* ve *metin tabanlı yazılımlar* kategorileri ortaya çıkmıştır. Öğretmen adayları programlama için kullanılacak, 7 farklı görsel yazılımdan, 1 robotik yönelimli yazılım, 4 donanım ve 3 adet metin tabanlı yazılımdan

bahsetmektedir.

Öğretmen adaylarının kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler için kullanılacak *görsel yazılımlar* olarak; Alice, Code.org ve Scratch'tan bahsetmişlerdir. Aşağıdaki alıntı da (Ö5) basit ve insan diline daha yakın yazılımlar olduğu için Scratch ve Code.org'un kullanılabilceğini ifade etmiştir.

Ö5: "Scratch bence gayet iyi, bir de Code.org...Basit daha anlaşılır ya benim dilime daha hani insan diline daha yakın."

Öğretmen adaylarının kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler için kullanılacak *robotik yönelimli yazılımlar* olarak Arduino; *donanım* olarak Lightboard, Codeon, Ales; *metin tabanlı yazılımlar* olarak ise Java Script, Dreamweaver, .Net, C, C++'dan bahsetmişlerdir. Aşağıdaki alıntı (Ö7) öğrenci seviyesine göre 7-11 yaşları arasındaki öğrenciler için donanım olarak Lightboard'un kullanılabilceğini ifade etmektedir.

Ö7: "Mesela eğer seviyelerine göre 7 yaş ile kaç yaş 10 yaş ya da 11 yaşa kadar Lightboard'ı kullanırım. Daha çok tetris programına benziyor hani o tarz, bi ara yüzü var....O yüzden gelişim düzeylerine göre psikolojilerine göre hani eğitim psikolojisinde oluyor ya hani şunları şunları yapabilir seviyelerine göre bu programları önerdim."

Kodlama Gelişmelerinin Takibine Yönelik Bulgular

Programlama ile ilgili gelişmelerin takibine yönelik öğretmen adaylarının görüşlerinden, programlama eğitimini nereden takip ettikleri, hangi sıklıkla takip ettikleri ve takip etme nedenlerinin yer aldığı 3 kategori ortaya çıkmıştır. Programlama ile ilgili gelişmeleri nereden takip ettiklerine yönelik 7 farklı yer belirten 9 öğretmen adayı; programlama ile ilgili gelişmeleri takip etme sıklığı konusunda 9 farklı ifade bulunan 9 öğretmen adayı ve programlama ile ilgili gelişmelerin takibini 8 öğretmen adayı toplamda 11 nedenle açıklamıştır.

Programlama ile ilgili gelişmeleri nereden takip ettikleri konusunda 9 öğretmen adayı; sosyal medya, akran, web sitesi, dersler, konferanslar, eğitim seminerleri, teknoloji dergilerinden bahsetmişlerdir. Programlama gelişmelerini takip etme sıklığı konusunda yine 9 öğretmen adayı; web sitelerinde ilgi çekici haber gördükçe, sosyal ağları her gün, paylaşım yapıldıkça, her gün, rastgele, ara sıra/denk gelirse, haftada 3-5 defa, yeni haber çıktıkça, haftada en fazla 1 saat ifadelerini dile getirmişlerdir. Programlama gelişmelerini takip etme nedenini 8

öğretmen adayları; eğitimlere katılmak için, sektördeki insanları tanımak, merak, önemsemek, ilgi çekici bir paylaşım sonucu ya da duyular aracılığıyla, işe girme ile ilgili bilgilerin bulunması şeklinde açıklamışlardır. Aşağıdaki alıntıda (Ö8) programlama ile ilgili gelişmeleri her gün takip ettiğini, Chip, Shift Delete, Donanım haber gibi popüler olan dergilerden, web sitelerinden merak edip önemseydiği için takip ettiğini ifade etmektedir.

Ö8: “Her zaman takip ediyorum her gün takip ediyorum. İnternet sitelerinden daha çok dergi sitelerinden, sosyal medyadan da takip ediyorum... Chip, Shift Delete, Donanım haber, Web Tekno yani bilgisayarla ilgili bir sürü dergi web sitesi hepsini takip ediyorum. Genellikle popüler olanları seçiyorum... Aslında bir ... kendi alanımla ilgili olduğu için merak ediyorum ve önemsiyorum o yüzden takip ediyorum. Zaten her gün bakıyorum yeni bir şey varsa içeriğini açıp bakıyorum.”

Tartışma, Sonuç ve Öneriler

Bu araştırma öğretmen adaylarının kodlama eğitimine yönelik düşüncelerini ortaya çıkarmayı amaçlamaktadır. Bu amaca yönelik olarak nitel bir yaklaşımla yarı-yapılandırılmış görüşme formu kullanılarak Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümü öğretmen adaylarından veriler toplanmıştır. Araştırma bulgularına genel olarak bakıldığında öğretmen adaylarının kodlama eğitimine yönelik görüşlerinin sınırlı olduğu gözlenmiştir. Kodlama derslerinde öğrencilere verilmek istenen sadece kod yazımının öğretilmesi değildir. Bu derslerde öğrencilerin bilgi-işlemsel düşünme becerilerini geliştirmesi gerekmektedir (Brennan ve Resnick, 2012). Bilgi-işlemsel düşünmenin problem çözme, algoritmik düşünme, genelleme, soyutlama ve ayrıştırma olmak üzere 5 temel yapısı vardır (Çakır, Korkmaz ve Özden, 2017; Çetin ve Toluk Uçar, 2017; Gülbahar, Kalelioğlu ve Kukul, 2016; Selby ve Woollard, 2013). Öğretmen adaylarının bu temel yapılardan sadece problem çözme ve algoritmik düşünmeden bahsettikleri görülmüştür. Buna ek olarak öğretmen adaylarının problem çözme ve algoritmik düşünmeden bahsederken onları bütün bileşenleri ile ele aldıklarını iddia etmek oldukça güçtür.

Literatürün öngördüğü şekilde kodlama eğitimine giriş dört farklı şekilde yapılmaktadır; görsel programlama, robotik ile programlama, metin tabanlı programlama ve bilgisayarı hiç kullanmadan kodlama kavramlarını anlatmak olan bilgisayarsız bilgisayar bilimi (computer science unplugged) (Weinberg, 2013). Öğretmen adayları genel olarak Scratch, Code.org, Alice gibi görsel ve C, C++ gibi metin tabanlı programlama ortamlarından bahsetmişlerdir.

Fakat, robotik ile programlamadan daha az bahsettikleri ve bilgisayarsız bilgisayar biliminden ise hiç bahsetmedikleri görülmüştür. Scratch, Code.org ve Alice gibi programların öğretmen adayları arasında popüler olmasının nedenlerine bakacak olursak; Scratch EBA'nın sitesinde kodlama eğitime dair başlıca program olarak yer almaktadır, bilişim atölyelerinde (örneğin, BTE-Atölyeler, 2017) Scratch, Code.org, Alice programlarından bahsedilmektedir. Özellikle Scratch kullanım kolaylığı sağladığı için öğretmen adayları tarafından en çok tercih edilen programlar arasında yer almaktadır (Saltan ve Kara, 2016). Ayrıca Çetin (2016b) öğretmen adaylarının kodlamayı Scratch ile daha anlamlı öğrendiklerini iddia etmiştir. Metin tabanlı diller, meslek liseleri ve üniversitelerde bahsedildiği için öğretmen adayları tarafından bilinmektedir. Robotik programlama aracı olan Arduinio maliyeti yüksek olduğu için popülerleşmemiş olabilir. Öğretmen adayları gelişmiş teknolojilerden bir şekilde haberdar olmuş ve kullanmak istediklerini dile getiren ifadelerde bulunmuşlardır (Yükseltürk ve Altıok, 2015). Ancak bilgisayarsız bilgisayar bilimi öğretmen adayları tarafından bilinmemektedir.

Öğretmen adayları kodlama eğitiminin bireye ve topluma katkısı konusunda alanyazına paralel olarak, kodlama eğitime küçük yaşta başlanılmasının gerektiğini dile getirdikleri ifadelerinde bazı öğrencilerin ilgileri doğrultusunda kendilerini keşfetme ve geliştirme fırsatı bulabileceklerinden bahsetmişlerdir. Aynı zamanda öğretmen adayları kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler doğrultusunda analitik düşünme, soyut düşünme, algoritmik düşünme, problem çözme ve derste öğrenilenleri uygulama yani bir ürün ortaya çıkarma becerilerinin gelişimine katkıda bulunabileceğinden bahsetmişlerdir (Kalelioğlu ve Gülbahar, 2014; Akpınar ve Altun, 2014). Literatürde bahsedilen, teknolojik bir toplumda ayakta kalmak için bireylerin kodlama bilmesi gerekliliği, ulusların ekonomik gücüne katkısı, öğrencilerin bilişim teknolojilerini sadece kullandığı değil aynı zamanda bunları ürettiği yani pasif değil aktif teknolojik araç üretimine katıldığı bir ortam olarak kodlama eğitimi ifadelerinden ise bahsetmemişlerdir (Sayın ve Seferoğlu, 2016).

Bilgi-işlemsel düşünmenin ülkelere ekonomik katkısı, bireyler için oluşturduğu iş imkânları, dünyayı anlama çabası, öğrencilere kazandırdığı temel beceriler ve teknolojik vatandaşlığın gereği olarak bilgi-işlemsel düşünme gibi faktörlerden dolayı birçok Avrupa ülkesi kodlama derslerini ilköğretim ya da ortaöğretimde zorunlu veya seçmeli ders olarak müfredatlarına eklemişlerdir veya eklemeyi düşünmektedirler (Bocconi, Chiocciariello, Dettori, Ferrari ve Engelhardt, 2016). Bulgular kısmında da belirtildiği gibi öğretmen adaylarının çoğu kodlamanın mantığını anlayarak problem çözme, soyut düşünme, algoritmik düşünme gibi becerilerin geliştirilebileceğinden bahsetmişlerdir. Öğretmen adayları ayrıca ulusların

ekonomik gelecekleri açısından öğrencilerin bilgisayar bilimini temel olarak bilmeleri gerektiğinden dolayı kodlama eğitiminin zorunlu olarak verilmesini gerekli görmektedirler (Kafai, Burke ve Resnick 2014). Kodlama eğitiminin seçmeli ders olarak müfredatta yer almasını dile getiren öğretmen adayları ise öğrencilerin ilgileri doğrultusunda kodlama eğitiminin getirilebileceğini düşünmektedirler.

Öğretmen adaylarının görüşleri değerlendirildiğinde kodlama eğitimine yönelik alan bilgisinden haberdar oldukları, kodlamaya yönelik derslerin üniversitelerde mevcut olduğu, teknolojik boyutta da Scratch, Code.org gibi yeni kodlama araçlarından haberdar oldukları anlaşılmaktadır. Bu bilgileri öğrencilere nasıl aktaracakları konusunda eksik oldukları için öğretmen adaylarına teknoloji, pedagoji ve alan bilgilerinin birbirini destekler nitelikte verildiği tekno-pedagojik alan bilgisi derslerine Bilgisayar ve Öğretim Teknolojileri Eğitimi lisans programlarında yer verilmesi gereklidir. Hâlihazırdaki Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümü mezunları düşünüldüğünde hizmet içi eğitimler onların alan ve pedagoji bilgilerinde önemli yer tutmaktadır. Fakat 2017 yılı hizmet içi eğitim planına bakıldığında Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümü hizmet içi eğitimlerinin sınırlı olduğu anlaşılmaktadır (MEB, 2017). Hizmet içi eğitimlerin sıklık ve içerik açısından geliştirilmesi önemlidir. Eğitimler öğretmenlerin ihtiyaçları olan tekno-pedagojik alan bilgisini içerecek şekilde verilmelidir. Son olarak bu çalışmada, kişisel gelişim kaynakları olarak, öğretmen adaylarının Bilişim Atölyesi gibi dernek faaliyetlerini, bilişim fuarlarını, konferansları ve internet kaynaklarını kullandıkları gözlenmiştir. Öğretmen adaylarının kendilerini geliştirmesi ve yeni ortaya çıkan ihtiyaçları göz önüne alarak bu kaynakların çeşidi, miktarı ve sıklığı artırılmalıdır.

Kaynaklar

- Agalianos, A., Noss, R. ve Whitty, G. (2001). Logo in mainstream schools: The struggle over the soul of an educational innovation. *British Journal of Sociology of Education*, 22(4), 479–500.
- Akpınar, Y. ve Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *İlköğretim Online*, 13(1), 1-4.
- Bakanlığı, M. E. (2017). 2017 yılı öğretmenlerin hizmet içi eğitim planı. http://oygm.meb.gov.tr/www/icerik_goruntule.php?KNO=28 adresinden 21 Şubat 2018 tarihinde indirilmiştir.

- Barr, V. ve Stephenson, C. (2011). Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community?. *Acm Inroads*, 2(1), 48-54.
- Bower, M. ve Falkner, K. (2015). Computational Thinking, the Notional Machine, Pre-service Teachers, and Research Opportunities. *In Proceedings of the 17th Australasian Computing Education Conference (ACE 2015)*, 27, 30.
- Bower, M. ve Falkner, K. (2015). Computational thinking, the notional machine, pre-service teachers, and research opportunities. In D. D'Souza, & K. Falkner (Ed.), *Proceedings of the 17th Australasian Computing Education Conference (ACE 2015)*. 160(10), 37-46. Sydney: Australian Computer Society.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A. ve Engelhardt, K. (2016). *Developing Computational Thinking in Compulsory Education-Implications for policy and practice*. Joint Research Centre (Seville site). doi:10.2791/792158
- Brennan, K. ve Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, Canada. 1-25.
- BTE Atölyeler. (2017). *Bilişim Teknolojileri Eğitimcileri Derneği*. <http://bte.org.tr/atolyeler/> adresinden 23 Şubat 2018 tarihinde ulaşılmıştır.
- Büyüköztürk, Ş., Çakmak, E. K., Akgün, Ö. E., Karadeniz, Ş. ve Demirel, F. (2011). *Bilimsel araştırma yöntemleri*. Ankara: Pegem Yayıncılık.
- Creswell, J. W. & Clark, V. L. P. (2007). *Designing and conducting mixed methods research*. Thousand Oaks, CA:Sage.
- CSTA (2016). K-12 Computer Science Standards. https://www.csteachers.org/page/CSTA_Standards adresinden 23 Şubat 2018 tarihinde ulaşılmıştır.
- Cuny, J., Synder, L. ve Wing, J. M. (2010). Demystifying computational thinking for non-computer scientists. Unpublished manuscript in progress. <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf> adresinden 23 Şubat 2018 tarihinde ulaşılmıştır.
- Çetin, I. (2013). Visualization: A tool for enhancing students' concept images of basic object-oriented concepts. *Computer Science Education*, 23(1), 1-23.
- Çetin, I. (2015). Students' understanding of loops and nested loops in computer programming: An APOS theory perspective. *Canadian Journal of Science, Mathematics and Technology Education*, 15(2), 155-170.

- Çetin, I. (2016a). Nitel İçerik Analizi. M. Y. Özden & Levent Durdu (Ed.), Eğitimde Üretim Tabanlı Çalışmalar İçin Nitel Araştırma Yöntemleri (125-148). Ankara: Anı Yayıncılık.
- Çetin, I. (2016b). Preservice Teachers' Introduction to Computing: Exploring Utilization of Scratch. *Journal of Educational Computing Research*, 54(7), 997-1021.
- Çetin, İ. ve Toluk Uçar, Z. (2017). Bilgi-işlemsel düşünme tanımı ve kapsamı. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya* (41-74). Ankara: Pegem Yayıncılık.
- Department for Education. (2013). National curriculum in England: Framework for key stages 1 to 4. <https://www.gov.uk/government/publications/nationalcurriculum-in-england-framework-for-key-stages-1-to-4> adresinden 23 Şubat 2018 tarihinde ulaşılmıştır.
- Du Boulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, 2(1), 57-73.
- Feurzeig, W. ve Papert, S. A. (2011). Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments*, 19(5), 487-501.
- Glesne, C. (2013). *Nitel Araştırmaya Giriş*. (Çev: Ali Ersoy ve Pelin Yalçinoğlu). Ankara: Anı Yayıncılık.
- Gülbahar, Y. ve Kalelioğlu, F. (2014). The Effects Of Teaching Programming Via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education-An International Journal*, 13(1), 33-50.
- Hour of code. (2016). *Hour of code* <https://code.org/employees/microsoft> adresinden 23 Şubat 2018 tarihinde ulaşılmıştır.
- Kafai, Y. B., Burke, Q. ve Resnick, M. (2014). *Connected code: Why children need to learn programming*. Mit Press. <https://mitpress.mit.edu/connected-code> adresinden 23 Şubat 2018 tarihinde ulaşılmıştır.
- Kalelioglu, F., Gülbahar, Y. ve Kukul, V. (2016). A Framework for Computational Thinking Based on a Systematic Research Review. *Baltic Journal of Modern Computing*, 4(3), 583-596.
- Kalelioğlu, F. ve Keskinılıç, F. (2017). Bilgisayar Bilimi Eğitimi için Öğretim Yöntemleri. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya* (155-178). Ankara: Pegem Akademi.
- Korkmaz, Ö., Çakır, R. ve Özden, M. Y. (2017). A validity and reliability study of the Computational Thinking Scales (CTS). *Computers in Human Behavior*, 72, 558-569. doi: 10.1016/j.chb.2017.01.005

- Merriam, S. B. (1998). *Qualitative Research and Case Study Applications in Education. Revised and Expanded from "Case Study Research in Education."*. San Francisco, CA: Jossey-Bass Publishers.
- Miles, M. B. ve Huberman, A. M. (1994). *Qualitative data analysis: A sourcebook of new methods*. Thousand Oaks, CA: Sage.
- Noble, A. (2012). Science the key to seize control of the future. *Sydney Morning Herald*. <http://www.smh.com.au/federal-politics/political-opinion/science-the-key-to-seize-control-of-the-future-20121225-2bv55.html> adresinden 23 Şubat 2018 tarihinde ulaşılmıştır.
- Özçınar, H., Yecan, E. ve Tanyeri, T. (2016). *Öğretmen gözüyle görsel programlama öğretimi*. 3. Uluslararası Eğitimde Yeni Yönelimler Konferansı, *Proceeding book*, 71-79.
- Piaget, J. (1964). Part I: Cognitive development in children: Piaget development and learning. *Journal of research in science teaching*, 2(3), 176-186.
- Saltan, F. ve Kara, M. (2016). ICT Teachers' Acceptance of "Scratch" as Algorithm Visualization Software. *Higher Education Studies*, 6(4), 146.
- Sayın, Z. ve Seferoğlu, S. S. (2016). *Kodlama Eğitimi ve Kodlamanın Eğitim Politikalarına Etkisi*. XVIII. Akademik Bilişim Konferansı (AB16), 3-5 Şubat 2016, Adnan Menderes Üniversitesi, Aydın.
- Selby, C. ve Woollard, J. (2013). Computational thinking: the developing definition. *University of Southampton (E-prints) 6pp*. https://eprints.soton.ac.uk/356481/1/Selby_Woollard_bg_soton_eprints.pdf adresinden 23 Şubat 2018 tarihinde ulaşılmıştır.
- Taub, R., Ben-Ari, M. ve Armoni, M. (2009). The effect of CS unplugged on middle-school students' views of CS. *ACM SIGCSE Bulletin*, 41(3), 99-103.
- Uzgun, B. Ç., ve Aykaç, N. (2016). Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programının Öğretmen Görüşlerine Göre Değerlendirilmesi (Ege Bölgesi Örneği)/The Evaluation of Information Technologies and Software Course's Curriculum According to the Teacher's Ideas. *Mustafa Kemal Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 13(34), 1-25.
- Weinberg, A. E. (2013). *Computational thinking: An investigation of the existing scholarship and research*. (Doctoral dissertation, Colorado State University). <https://search.proquest.com/openview/42c6d1d14b6347eb574d258a2f2844fb/1?pq-origsite=gscholar&cbl=18750&diss=y> adresinden 23 Şubat 2018 tarihinde ulaşılmıştır.

- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. (2011). *Research Notebook: Computational Thinking - What and Why? The Link*. Pittsburgh, PA: Carneige Mellon.
- Yecan, E., Özçınar, H. ve Tanyeri, T. (2017). Bilişim Teknolojileri Öğretmenlerinin Görsel Programlama Öğretimi Deneyimleri. *Elementary Education Online*. 16(1), 377-393.
- Yıldırım, A. ve Şimşek, H. (2013). *Sosyal Bilimlerde Nitel Araştırma Yöntemleri*. Ankara: Seçkin Yayıncılık.
- Yin, R. (1984). *Case Study Research: Design & Methods*. Beverly Hills, CA: Sage.
- Yükseltürk, E. ve Altıok, S. (2015). Bilişim Teknolojileri Öğretmen Adaylarının Bilgisayar Programlama Öğretimine Yönelik Görüşleri. *Amasya Üniversitesi Eğitim Fakültesi Dergisi*, 4(1), 50-65.
- Zyda, M. (2009). Computer science in the conceptual age. *Communications of the ACM*, 52(12), 66-72.

Extended Abstract

Purpose and Significance

Research related to computational thinking can be traced back to 1960s, when the Logo programming language, which is supposed to be easily learned by students, were constructed. Logo programming language was intended to help students learn mathematical and logical ways of thinking. The Logo has been utilized in the context of constructionism, a variant of constructivism. Constructivism posits that individuals build knowledge through experience.

Wing (2006) defined computational thinking as “involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science.” Wing, Cundy and Synder refined definition as “Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent” (as cited in Wing, 2011). Wing (2006) argued that the computational thinking should be learned by everyone. Her ideas have given a new impetus to computational thinking. Computing courses become increasingly popular at the K-12 level of education. Computing education has begun to be considered as one of the key areas in many developed countries. In Turkey, it is compulsory in the 5th and 6th grades in the information technology and software courses since 2013. In addition to the changes in the K12 curriculum, there are also international events such as the "Hour of Code" that encourages computing.

Today, learning to use computing technology by the students is no longer seen as sufficient. It is contended that students should study on constructing computing tools. Learning computational thinking will serve fundamental role for this purpose. The reason for teaching students computing is not to make all of them computer scientist. The aim is to provide students with the ability to develop and apply the computational thinking skills, to keep them up to the technology age, and to participate in the technological community.

Increasing interest in computational thinking leads researchers to an important question: What are the best ways to teach students basic computing concepts? There are three important trends for introducing computing as the literature suggests; computer science unplugged, visual programming and robotics. In visual programming environments, students create products such as games, animations and digital stories in block-based coding environments. In the robotic approach, students make their robotics equipment functional by using different programming languages. Students in computer science unplugged approach learn basic concepts such as binary numbers, algorithms and data compression with a multitude of learning activities, such as interesting games, puzzles and cards without distractions and technical details that students often encounter in front of computers.

There is a rapid change in computer science education in Turkey and in the world. It is necessary to prepare new curriculums, to train teachers and to make changes in the classroom environment. It is therefore an important part of the puzzle to educate pre-service teachers who will guide students in learning computational thinking and to help teachers develop themselves on the pedagogy of computational thinking. In this context, there are fundamental roles for computer science education researchers in teacher education. Therefore it is important to reveal what pre-service teachers think about computing education.

Method

The aim of this study was to explore pre-service teachers' views related to computing education. In this respect, the case study is a suitable method for realizing the goal of this research. Case study is useful to understand and explain the distinctive characteristics of a person, a community or a situation. Case studies are based on the questions 'what', 'how', and 'why'. In the study, an interpretive approach was used to answer the question "what pre-service teachers think about computing education" (Yin, 1984).

The research data were obtained from 12 pre-service teachers (3 first grade, 3 second grade, 3 third grade and 3 fourth grade) majoring Computer Education and Instructional Technology in

Bolu Abant İzzet Baysal University. 3 pre-service teachers from each grade level among the volunteers were randomly selected. The participants involved 5 girls and 7 boys.

The data were collected through a semi-structured interview protocol. In this research, in addition to the questions that researchers had already prepared before the interview, follow up questions were asked to get more detailed information about the issues in consideration. The interview protocol was created to explore the opinions of pre-service teachers about computing education. There were 11 questions in the interview protocol. A pilot study of interview protocol was conducted with three pre-service teachers from the Computer Education and Instructional Technology Department to refine the interview questions. The interviews that lasted about 15 minutes in average were audio recorded and then interviews were transcribed.

Qualitative content analysis was used for the analysis of collected data. The goal in qualitative content analysis is to reach concepts and associations that can describe the data obtained (Çetin, 2016a). The transcribed records were examined and the coding table was created by three researchers. After that ten-percent of the data was coded separately by two researchers using the coding table. At the end of this process, 91 percent inter-coder agreement ratio was acquired and then researchers negotiated on coding table that resulted in unanimous agreement. After this step, the entire data is coded by one of the researchers.

Results

Findings obtained in the research process were categorized under 9 different headings. These categories included; the purpose of coding, contribution of coding to an individual, contribution of coding to the society, coding for all, efficiency of coding education, beginning age of coding education, basic concepts and skills for coding, software / hardware requirement for programming and how to keep up with developments of programming. Although twelve pre-service teachers participated in the study, the total number of responses exceeds 12 for some categories since in some cases pre-service teachers were grouped under more than one category. Although 9 categories emerged in the study, most significant 3 of them will be presented.

Under purpose of coding education category the following subcategories were obtained from the pre-service teachers' responses; the effect of coding to cognitive skills, the aim of coding for individuals and teaching logic of coding. It is the effect of cognitive skills to the coding subcategory in which the pre-service teachers are mostly accumulated. 7 pre-service teachers

expressed the purpose of coding education under the effect of coding to cognitive skills subcategory. They expressed their view in the context of improvement of cognitive skills. The 4 pre-service teachers who were classified under the aim of coding for individuals subcategory explained why coding should be learned by students. The 4 pre-service teachers who were categorized under the teaching logic of coding subcategory expressed the aim of coding as to teach logic of coding so that individuals become aware of what they can do with coding.

From the data, it was seen the pre-service teachers' opinions about the contribution of coding to an individual is composed of 3 subcategories; effect of coding to cognitive skills, effect of coding to business life, and effect of coding to daily life. Pre-service teachers were mostly accumulated under effect of coding to cognitive skills subcategory. The 9 pre-service teachers classified under the effect of coding to cognitive skills expressed their view in the context of improvement of human cognitive skills. 4 pre-service teachers were categorized under effect of coding to business life. They contended that knowing programming may help individuals find jobs or better jobs. 3 pre-service teachers categorized under effect of coding to daily life argued that coding makes life easier.

There were 3 subcategories under basic concepts and skills for coding; cognitive skills, problem solving skills and know-how skills. There were 8 pre-service teachers under cognitive skills subcategory. They stated that basic concepts and skills for coding should help. There were 5 pre-service teachers under problem solving skills subcategory. They contended that basic concepts and skills for coding should help students to overcome problems encountered. There were 3 pre-service teachers classified under know-how skills. Their main idea was tied with improving individuals' know-how skills.

Discussion and Conclusion

This research aims to reveal the views of pre-service teachers about computing education. From an overall perspective, it can be said that the opinions of pre-service teacher about computing education are limited. In computing courses, the aim is not to teach only the code writing; rather the aim is to help students improve their computational thinking skills. It was seen that the pre-service teachers mentioned only about problem-solving and algorithmic thinking as basic skills related to computing education. Moreover, it can be said that pre-service teachers had limited view related to problem solving and algorithmic thinking.

As the literature suggests there are 4 ways for introducing computing: visual programming,

robotics, computer science unplugged, text based. Pre-service teachers mainly talked about visuals programming environment, e.g., Scratch, Code.org and Alice and text-based programming environments such as C and C ++. Moreover they mentioned less about robotics and they did not mention about computer science unplugged way.

The study revealed that pre-service teachers have computing courses in their departments and they are informed about content knowledge and current coding environments like Scratch and Code.org. However it can be said that their knowledge about how to use these concepts and tools in teaching computing is limited. Therefore it is necessary for pre-service teachers to be given techno-pedagogical content knowledge related courses in which technology, pedagogy and content knowledge are given in a complementary fashion.