

T.C.
BOLU ABANT İZZET BAYSAL ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ
ANABİLİM DALI
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ
BİLİM DALI

BİLİŞİM TEKNOLOJİLERİ VE YAZILIM DERSİ
ÖĞRETMENLERİNİN KODLAMA EĞİTİMİ HAKKINDAKİ
GÖRÜŞLERİ

Yüksek Lisans Tezi

Hazırlayan
Ash GÖNCÜ

Danışman
Doç. Dr. İbrahim ÇETİN

BOLU, OCAK, 2019

YÜKSEK LİSANS TEZ ONAY FORMU

Aslı GÖNCÜ tarafından hazırlanan “Bilişim Teknolojileri Ve Yazılım Dersi Öğretmenlerinin Kodlama Eğitimi Hakkındaki Görüşleri” adlı çalışma Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir. (10.01.2019)

Akademik Unvan ve Adı Soyadı**İmza**

Üye (Tez Danışmanı) :Doç. Dr. İbrahim ÇETİN
Üye :Prof. Dr. Erkan TEKİNARSLAN
Üye :Prof. Dr. Süleyman Sadi SEFEROĞLU

**Eğitim Bilimleri Enstitüsü'nün Onayı****Prof. Dr. Türkan ARGON****Eğitim Bilimleri Enstitüsü Müdürü**

ETİK İLKELERE UYULDUĐUNA İLİŐKİN BEYAN

Yüksek Lisans Tezi olarak sunduđum, *BiliŐim Teknolojileri ve Yazılım Dersi Öğretmenlerinin Kodlama Eğitimi Hakkındaki Görüşleri* başlıklı çalışmanın yazılmasında bilimsel ve etik kurallara uyduđumu, başkalarının eserlerinden yararlanılması durumunda atıfta bulunduđumu, kullanılan verilerde herhangi bir tahrifat yapmadıđımı, tezin tamamının ya da bir kısmının bu üniversite veya başka bir üniversitede bir tez çalışması olarak sunulmadıđımı beyan ederim. 10/01/2019



Aslı GÖNCÜ



Babama ve anneme...

TEŞEKKÜR

Tez çalışmam sırasında kıymetli, birikim ve tecrübeleri ile bana yol gösterici ve destek olan, sabırla ve büyük bir ilgiyle bana faydalı olabilmek için elinden gelenen fazlasını sunan, güler yüzünü ve samimiyetini benden esirgemeyen, yaşadığım her sorunda yanına çekinmeden gidebildiğim, gelecekteki mesleki hayatımda da bana verdiği değerli bilgilerden faydalanacağımı düşündüğüm kıymetli danışman hocam sayın Doç. Dr. İbrahim ÇETİN'e gönülden teşekkür ve saygılarımı sunarım.

Lisans ve yüksek lisans dönemim boyunca ders aldığım, bana kazandırmış oldukları her şey için ve beni gelecekte söz sahibi yapacak bilgilerle donattıkları için değerli hocalarım Prof. Dr. Erkan TEKİNARSLAN, Doç. Dr. Ercan TOP, Dr. Öğr. Üyesi Dr. Nuh YAVUZALP, Dr. Öğr. Üyesi Melih Derya GÜRER, Doç. Dr. Bahadır ALTINTAŞ, Dr. Öğr. Üyesi Orhan CURAOĞLU'na teker teker teşekkürlerimi sunuyorum. Tezime sağlamış olduğu katkılarından dolayı Ondokuz Mayıs Üniversitesi öğretim üyesi değerli hocam Dr. Öğr. Üyesi Polat ŞENDURUR'a; yüksek lisans döneminde benden manevi desteğini esirgemeyen sayın Mustafa DEMİREL hocama sevgi ve minnetlerimi sunuyorum.

Çalışmalarım boyunca yardımını benden hiç esirgemeyen değerli arkadaşım Canan CEBECİ başta olmak üzere bütün arkadaşlarıma teşekkürü bir borç bilirim.

Ayrıca, hayatımın her döneminde yanımda olan maddi manevi destekleriyle beni hiçbir zaman yalnız bırakmayan babama, anneme ve kardeşlerime de sonsuz sevgi ve minnetlerimi sunarım.

Ocak, 2019

Aslı GÖNCÜ

İÇİNDEKİLER

YÜKSEK LİSANS TEZİ ONAY FORMU	ii
ETİK İLKELERE UYULDUĞUNA İLİŞKİN BEYAN.....	iii
TEŞEKKÜR.....	v
İÇİNDEKİLER	vi
TABLolar DİZİNİ.....	ix
ŞEKİLLER DİZİNİ	x
GRAFİKLER DİZİNİ.....	xi
ÖZET	xii
ABSTRACT.....	xiv
I. BÖLÜM.....	1
1. Giriş	1
1.1. Problem Durumu	3
1.2. Araştırmanın Amacı	4
1.3. Araştırmanın Önemi	4
1.4. Araştırmanın Varsayımları	6
1.5. Araştırmanın Sınırlılıkları	6
II. BÖLÜM	8
2. Kuramsal Çerçeve ve İlgili Alanyazın.....	8
2.1. Bilgi İşlemsel Düşünme Tarihinde İki Önemli Figür	8
2.2. Bilgi İşlemsel Düşünmenin Tanımı	9
2.3. Bilgi İşlemsel Düşünmenin Bileşenleri.....	13
2.3.1. Düşünme süreci.....	13
2.3.2. Problem çözme	14
2.3.3. Soyutlama	15
2.3.4. Ayırıştırma	16
2.3.5. Algoritmik düşünme	17
2.3.6. Genelleme	18
2.3.7. Değerlendirme	19

2.4. Bilgisayar Bilimi Eğitimde Kullanılan Öğretim Yöntemleri	20
2.4.1. Blok tabanlı görsel programlama ortamları	20
2.4.2. Robot uygulamaları.....	22
2.4.3. Metin tabanlı ortamlar	25
2.4.4. Bilgisayarsız bilgisayar bilimi öğretimi.....	27
2.5. Teknolojik Pedagojik Alan Bilgisi (TPAB).....	28
III. BÖLÜM.....	32
3. Yöntem.....	32
3.1. Araştırma Deseni.....	32
3.2. Katılımcılar	33
3.3. Veri Toplama Süreci	34
3.4. Verilerin Analizi.....	36
IV. BÖLÜM.....	39
4. Bulgular	39
4.1. Kodlama Eğitimi Tanımlamaları.....	40
4.2. Kodlamanın Amacı	45
4.3. Herkes İçin Kodlama.....	50
4.4. Kodlama Eğitimine Başlangıç Yaşı	54
4.5. Kodlama İçin Temel Kavram ve Beceriler	56
4.6. Programlama için Yazılım Donanım Gereksinimi.....	60
4.7. Öğretmenlerin Bilgisayarsız Bilgisayar Bilimi Hakkındaki Düşünceleri	67
4.8. Öğretmen Eksiklikleri	71
4.9. Öğretmen Eksiklikleri İçin Yapılması Gerekenler.....	74
V. BÖLÜM	79
5. Tartışma, Sonuç ve Öneriler	79
5.1. Tartışma ve sonuç	79
5.1.1. Kodlama eğitimi	79
5.1.2. Kodlamanın amacı	82
5.1.3. Herkes için kodlama	84
5.1.4. Kodlama eğitimine başlangıç yaşı	86
5.1.5. Kodlama için temel kavram ve beceriler	89
5.1.6. Programlama için yazılım donanım gereksinimi	92

5.1.7. Öğretmenlerin bilgisayarsız bilgisayar bilimi hakkındaki görüşleri.....	95
5.1.8. Öğretmen eksiklikleri ve eksikliklerin giderilmesi için yapılması gerekenler	97
5.2. Öneriler	99
KAYNAKÇA.....	101
EKLER.....	120
Ek 1. Görüşme Formu.....	120
Ek 2. Etik Kurul Onay Formu.....	123
Ek 3 İl Milli Eğitim Müdürlüğü Araştırma İzin Belgesi	124
Ek 4. Tez Başlığında Değişiklik Tutanağı	125
ÖZGEÇMİŞ	126

TABLÖLAR DİZİNİ

Tablo 3.1 Katılımcıların demografik özellikleri 34

Tablo 4.1 Araştırma bulguları sonucunda elde edilen kategoriler 39



ŞEKİLLER DİZİNİ

Şekil 2.1. Scratch programlama ortamı	22
Şekil 2.2. Makeblock mBot (2018)	24
Şekil 2.3. Ozobot (2018).....	24
Şekil 2.4. “Hello World” kod parçaları gösterimi (Kandemir, 2017).....	27
Şekil 2.5. TPAB yapısının bileşenleri (Koehler ve Mishra, 2008).....	30



GRAFİKLER DİZİNİ

Grafik 4.1. Kodlama eğitimi ile kazandırılmak istenen temel kavram ve beceriler 56



ÖZET

BİLİŞİM TEKNOLOJİLERİ VE YAZILIM DERSİ ÖĞRETMENLERİNİN KODLAMA EĞİTİMİNE YÖNELİK GÖRÜŞLERİ

Göncü, Aslı

Yüksek Lisans Tezi

Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı

Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalı

Tez Danışmanı: Doç. Dr. İbrahim ÇETİN

Ocak-2019, xv+126 Sayfa

Bu çalışmanın amacı, bilişim teknolojileri ve yazılım dersi öğretmenlerinin kodlama eğitimine yönelik düşünce ve görüşlerini incelemektir. Bu doğrultuda 2016-2017 öğretim yılı bahar dönemi Bolu il merkezi, Mengen, Dörtdivan, Mudurnu, Yeniçağ, Seben, Göynük ve Gerede'den 22 bilişim teknolojileri ve yazılım öğretmeninden durum çalışması yöntemi ile nitel veri elde edilmiştir. Öğretmenlerle gerçekleştirilen görüşmelerde yarı-yapılandırılmış görüşme tekniği kullanılmıştır. Bu bağlamda öğretmenlerin kodlama eğitimi ile ilgili bilgi ve düşüncelerini ortaya çıkarmak amacıyla toplamda 10 sorudan oluşan bir görüşme formu hazırlanmıştır. Görüşme formunda yer alan sorular araştırmacılar tarafından araştırmanın amacı dikkate alınarak oluşturulmuştur. Araştırma verilerini toplamaya başlamadan önce yapılan pilot uygulama sonucunda iki soru üzerinde düzenleme yapılmış ve görüşme sorularına son şekli verilmiştir. Yüz yüze gerçekleştirilen ve ortalama 15 dakika süren görüşmeler ses kayıt cihazı ile kayıt altına alınmıştır. Elde edilen verilerin analizi için nitel içerik analizi kullanılmıştır.

Araştırma sürecinde elde edilen bulgular 9 kategoride değerlendirilmiştir. Çalışmanın bulgularına genel olarak bakıldığında öğretmenlerin kodlama eğitimine yönelik düşünce ve görüşlerinin sınırlı olduğu gözlenmiştir. Öğretmenlerin kodlama eğitiminin temel yapılarından sadece düşünme süreci, problem çözme, yaratıcılık ve

algoritmik düşünmeden bahsettikleri görülmüştür. Buna ek olarak öğretmenlerin büyük bir çoğunlunun bilgisayarsız bilgisayar bilimi etkinliklerinden haberdar olmadıkları ve haberdar olanların da alanyazında söylenen bilgisayar bilimi etkinliklerinin katkılarından hiç bahsetmedikleri görülmüştür.

Anahtar Kelimeler: Bilgi İşlemsel Düşünme, Kodlama Eğitimi, Öğretmenler, Bilgisayarsız Bilgisayar Bilimi, Durum Çalışması.



ABSTRACT**PERCEPTIONS OF INFORMATION TECHNOLOGIES AND SOFTWARE
COURSE TEACHERS TOWARD CODING EDUCATION**

Göncü, Aslı

M. Sc. Thesis

Department of Computer and Instructional Technologies Discipline

Supervisor: Assoc. Prof. İbrahim ÇETİN

January-2019, xv+126 Pages

The purpose of this study was to explore the thoughts and perceptions information technologies and software course teachers about computing education. For this purpose, qualitative data were obtained from 22 information technologies and software teachers from Bolu city center, Mengen, Dörtdivan, Mudurnu, Yeniçağ, Seben, Göynük and Gerede by using case study method. Semi-structured interview protocol was used to handle the interviews. In this context, an interview form consisting of 10 questions was prepared in order to reveal the teachers' opinions about computing education. The questions included in the interview form were formed by the researchers considering the purpose of the research. Before the data collection process, two questions were arranged on the pilot application and the interview questions were finalized. Face-to-face interviews that lasted about 15 minutes were recorded with a voice recorder. Qualitative content analysis was used to analyze the obtained data. The findings of the research were evaluated in 9 categories. The findings of the study revealed that the thoughts and opinions of teachers about coding education were limited. It was seen that the teachers only mentioned about the thinking process, problem solving, creativity and algorithmic thinking as the components of the computing education. Lastly, it was observed that most of the teachers were not aware of computer science unplugged activities and those who were informed did not mention about the value of computer science unplugged activities that are expressed in the literature.

Keywords: Computational Thinking, Computing Education, Teachers, Computer Science Unplugged, Case Study.



I. BÖLÜM

1. Giriş

Bu bölümde, bilgi işlemsel düşünmenin tanımı, operasyonel tanımı ve programlama dersi ile bilgi işlemsel düşünmenin ilişkisine yer verilmiştir. Ayrıca yapılan araştırmaya ilişkin problem durumu, araştırmanın amacı, önemi, sayıtlar ve sınırlılıklar belirtilmiştir.

Bilgi işlemsel düşünme kavramının kökeni daha eskilere dayansa da bu kavramı ilk kullanan Seymour Papert'tır (Papert ve Harel, 1991). Ancak kavramın tanımını ilk yapan kişi olarak çoğunlukla Wing'ten (2006) bahsedilmektedir. Wing'e (2006, s.33) göre "bilgi işlemsel düşünme, bilgisayar biliminin kavramlarını kullanarak problem çözme, insan davranışlarını anlama ve sistem tasarlama üzerinde durur." Wing aynı makalesinde, bilgi işlemsel düşünmenin sadece bilgisayar bilimcileri için değil herkes için temel bir beceri olduğunu dile getirerek, bilgi işlemsel düşünmenin sadece bilgisayar bilimcileri tarafından değil herkes tarafından geliştirilmesi gerektiğini vurgulamıştır. Bunun yanında okuma, yazma, aritmetik gibi temel becerilerin yanına bilgi işlemsel düşünmenin de temel bir beceri olarak eklenilmesi gerektiğini söylemiştir. Wing'in iddiaları bilgi işlemsel düşünme için yeni bir ivme kazandırmış ve akademisyenler, eğiticiler, politikacılar, bilgisayar ve iletişim teknolojileri endüstrisini de içine alan topluluğun geniş bir yelpazesinde etkili olmuştur.

Bilgi işlemsel düşünmenin teorik tanımı okullarda temel olarak bilgi işlemsel düşünme ile ilgili neler öğretilmesi gerektiği konusunda yeterli değildir. A-12 seviyesindeki derslerde bilgi işlemsel düşünmenin öğretimini gerçekleştirmede operasyonel bir tanıma ihtiyaç vardır (Çetin ve Toluk Uçar, 2017). Uluslararası Eğitimde Teknoloji Topluluğu (ISTE – The International Society for Technology in

Education) ve Bilgisayar Öğretmenleri Derneği (Computer Science Teachers Association – CSTA) öğrencilerin bilgi işlemsel düşünürler olması ve günümüz dijital teknolojilerinin geleceğin problemlerini çözmeye nasıl yardımcı olabileceğini anlamaları için bir araya gelmişler ve bilgi işlemsel düşünmeye yönelik operasyonel bir tanım geliştirmek için bir çalışma yapmışlardır (ISTE, 2011). 700'e yakın eğitimci ve araştırmacının görüşlerini belirttiği bu çalışma sonucu tanım aşağıdaki şekilde ifade edilmiştir.

Bilgi-işlemsel düşünme aşağıdaki süreçlerle sınırlandırılmamakla birlikte bu süreçleri içeren bir problem çözme sürecidir:

- Problemleri, bilgisayar veya diğer araçlarla çözmeye yönelik olarak formüle etme,
- Verileri mantıksal olarak organize ve analiz etme,
- Verileri model ve simülasyonlar gibi soyut biçimlerde sunma,
- Algoritmik düşünme ile otomatik çözümler geliştirme,
- Muhtemel çözümleri tanımlarken, analiz ederken ve uygularken kaynakları ve çözüm yollarını en etkili ve verimli bir şekilde seçmeyi amaçlama,
- Bu problem çözüm sürecini çeşitli problemlere genelleme ve transfer etme, uyarılama.

ISTE ve CSTA tarafından yapılan bu tanımlamada problemi bileşenlerine ayırma, algoritmik düşünme, veri analizi ve temsili, modelleme ve simülasyon gibi soyutlama, otomasyon, problem çözme, çözümlerin genelleştirilmesi gibi kavram ve becerilerden bahsedilmiştir. Bu kavram ve beceriler, eğitimcilere A-12 eğitim seviyesinde öğrencilere bilgi işlemsel düşünme ile ilgili öğretilmesi gerekenler konusunda çerçeve sunmaktadır (Çetin ve Toluk Uçar, 2017).

Yeni teknoloji müfredatının ana kısımlarından birisi olan kodlama eğitiminin amacı öğrencilerin bilgi işlemsel düşünme becerisini geliştirmelerine destek olmaktır. Öğrencilere kodlama öğretilmesinin sebebi onların hepsinin bilgisayar mühendisi ya da programcı olmalarını sağlamak değildir. Genel anlamda amaç öğrencilerde bilgi işlemsel düşünme becerisini geliştirerek uygulayabilme yeteneğini kazandırmak ve

teknoloji çağına ayak uydurmalarını sağlamaktır (Çetin ve Toluk Uçar, 2017; Selby ve Woollard, 2013; Wing, 2011).

Bilgi işlemsel düşünme, bir probleme yaklaşım açısıyla ilgilidir. Bilgi işleyen birimler yardımıyla problemler için çözüm üretilmektedir. Bu süreç iki aşamalıdır; ilk aşamada problemi çözmek için gerekli adımlar hakkında düşünülür, ikinci aşamada ise teknik bilgiler kullanılarak bilgisayar ile çözüm uygulamaya konulur. Öğrenciler, bilgi işlemsel düşünme ile hem okul içindeki hem de okul dışındaki hayatlarında kullanabilecekleri birçok beceriyi geliştirme şansına sahip olurlar. Bu becerilere örnek olarak şunlar verilebilir: yaratıcılık, problem çözme, hata bulup düzeltmek, bir şeyi denemeler yaparak düzeltmek veya geliştirmek, planlama yapmak ve takım halinde çalışmak. Fakat amaçlarından biri öğrencilere bilgi işlemsel düşünme becerisini kazandırmak olan kodlama eğitimine yönelik öğretmen düşünceleri henüz Türkiye’de yeteri kadar incelenmemiştir.

1.1. Problem Durumu

Bilişim teknolojileri ve yazılım dersi öğretmenlerinin kodlama eğitimine yönelik görüşleri nelerdir?

Araştırma probleminin çözümünde aşağıda belirlenen alt problemlere de yanıt aranmıştır:

1. Bilişim teknolojileri ve yazılım dersi öğretmenlerine göre kodlama eğitimi ne ifade etmektedir?
2. (Bilişim teknolojileri ve yazılım dersi öğretmenlerine göre) Kodlama eğitiminin bileşenleri nelerdir?
3. (Bilişim teknolojileri ve yazılım dersi öğretmenlerine göre) Kodlama eğitimine ilişkin öğrencilere kazandırılması gereken kavram ve beceriler nelerdir?

4. (Bilişim teknolojileri ve yazılım dersi öğretmenlerine göre) 3. alt problemde belirtilen bu kavram ve beceriler hangi araçlar kullanılarak öğrencilere kazandırılmalıdır?
5. (Bilişim teknolojileri ve yazılım dersi öğretmenlerine göre) 4. alt problemde belirtilen öğrencilere kazandırılacak olan kavram ve beceriler bu araçlar kullanılmadan kazandırılabilir mi?
6. Bilişim teknolojileri ve yazılım dersi öğretmenlerinin öğrencilerin kodlama becerilerini geliştirme konusunda hangi konularda eksiklikleri vardır?

1.2. Araştırmanın Amacı

Bu araştırmanın amacı bilişim teknolojileri ve yazılım dersi öğretmenlerinin kodlama eğitimine yönelik düşünce ve görüşlerini ortaya çıkarmaktır. Bu amaca yönelik olarak şu alt boyutlar incelenmiştir:

- (i) bilişim teknolojileri ve yazılım dersi öğretmenlerine göre kodlama eğitiminin temelleri,
- (ii) bilişim teknolojileri ve yazılım dersi öğretmenlerine göre kodlama eğitiminin bileşenleri,
- (iii) bilişim teknolojileri ve yazılım dersi öğretmenlerine göre öğrencilere kazandırılması gereken kavram ve beceriler ve bu kavram ve becerilerin kazandırılmasında kullanılacak araçlar,
- (iv) bilişim teknolojileri ve yazılım dersi öğretmenlerine göre diğer bilişim teknolojileri ve yazılım dersi öğretmenlerinin, öğrencilerin kodlama becerilerini geliştirme konusunda eksik noktalarının olup olmadığı

1.3. Araştırmanın Önemi

Kodlama eğitimi Dünya’da birçok gelişmiş ülkede temel alanlardan biri olarak değerlendirilmektedir. Bu ders zorunlu veya seçmeli ders olarak verilmektedir (Balanskat ve Engelhardt, 2014; Balanskat ve Engelhardt, 2015; Sayın, 2017; Tebliğler

Dergisi, 2016). Estonya kodlama eğitimine yer veren ilk ülkelerdendir. 2011’de ortaokul öğrencileri için ders olarak müfredata alınan kodlama daha sonra ilkokul öğrencileri için de uygulanmaya başlamıştır. İngiltere 2013 yılında yaptığı değişiklikle kodlamayı müfredatına eklemiştir. İngiltere Eğitim Bakanlığı, bilgi ve iletişim teknolojileri dersinin müfredatını 5-6, 7-11 ve 11-14 yaş grubu için 3 ayrı basamakta düzenlemiştir. ABD’de, eyalet bazında farklı uygulamalar olmakla birlikte, kodlama eğitimi ilkokul, ortaokul ve liselerde verilmektedir. Türkiye’de ise kodlama eğitimi 2013 yılından itibaren 5. ve 6. sınıflarda Bilişim Teknolojileri ve Yazılım dersi içerisinde zorunlu olarak verilmektedir. Diğer ülkelerde olduğu gibi kodlama eğitimi hızlı bir şekilde Türkiye’nin de gündemine girmiştir. Bu durum akademisyenler için birçok soru ve sorunu beraberinde getirmektedir. Öğretmen eğitimi temel sorunlardan bir tanesidir. Bu sebeple bu çalışmada öğretmenlerin kodlama eğitimine yönelik düşünceleri araştırılacaktır.

Kodlama eğitimine olan ilginin artmasıyla araştırmacılar şu soru üzerinde daha fazla kafa yormalıdır: Öğrencilere temel bilgisayar bilimi kavramlarını öğretmenin en iyi yolları var mıdır eğer varsa bunlar nelerdir (Karadeniz, 2017)? Öğretim esnasında, öğretmenlerin içerik bilgisini ön planda tutarak pedagojiyi geri plana aldıkları düşünülmektedir (Karadeniz, 2017). Öğretmenler konu ile ilgili alan bilgisini, öğrenci ihtiyaçları ve sınıf durumlarına yönelik uygulama ve yöntem bilgisiyle birlikte öğrencilerin öğrenmesini destekleyici teknolojileri kullanarak etkili bir öğretim yapabilmelidir (Niess, 2008, s.224). Alanyazında küçük yaşlarda kodlama eğitimine giriş konusunda üç önemli yönelim ile karşılaşmaktadır; görsel programlama, robot programlama ve bilgisayarsız bilgisayar bilimi (computer science unplugged) (Bower & Falkner, 2015). Görsel programlamada öğrenciler blok tabanlı kodlama ortamlarında, hazır olarak bulunan kodları kullanarak oyun, animasyon ve dijital hikâye gibi ürünler oluştururlar. Robot yaklaşımında öğrenciler, robot donanımlarını farklı programlama dilleri kullanarak işlevsel hale getirirler. 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 çok sayıda öğrenme etkinliği ile öğrenirler.

Bilgisayar bilimi eğitiminde Dünya’da ve Türkiye’de yaşanan değişim içerisinde 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 eğitimi araştırmacılarına öğretmen eğitimine yönelik roller düşmektedir. Henüz kodlama eğitimi alanında Türkiye’de çalışmalar yeni yeni yapılmaktadır (Yecan, Özçınar ve Tanyeri, 2016). Derse girecek olan ve hazırlanan müfredatları uygulayacak olan kişiler öğretmenlerdir. Bu durumda öğretmenlerin bu konuda ne düşündüklerini ortaya çıkarmak önem taşımaktadır. Bu çalışmanın amacı Türkiye’deki öğretmenlerin bahsedilen değişim sürecinde nerede olduklarını belirlemeye dair bir girişimde bulunmaktır. Bu çalışmayla ortaya çıkacak sonuçların, bilişim teknolojileri ve yazılım dersi öğretmenlerinin kodlama eğitimine yönelik görüşleri hakkında yetersiz sayıda çalışmanın olması bakımından alanyazına katkısı olacağı düşünülmektedir. Kodlama eğitimi ile ilgili yaşanan hızlı değişimler içerisinde öğretmenlerin güncel durumunun tespit edilmesi öğretmenlerin ve öğretmen adaylarının eğitimi konusunda faydalı olabilir.

1.4. Araştırmanın Varsayımları

Bu araştırma kapsamında katılımcılarla görüşmeler yapılmıştır. Dolayısıyla, katılımcıların görüşme formunda yer alan soruları, düşüncelerini doğru yansıttıkları şekilde içtenlikle yanıtladıkları varsayılmıştır.

1.5. Araştırmanın Sınırlılıkları

Bu araştırma Bolu ili ve ilçelerindeki bilişim teknolojileri ve yazılım dersi öğretmenleri ile yapılmıştır. Dolayısıyla araştırmanın bulguları ve sonuçları, çalışmaya katılan 22 bilişim teknolojileri ve yazılım dersi öğretmenin 2016-2017 eğitim öğretim

yılı bahar dönemindeki kodlama eğitimine ilişkin düşünce ve görüşleri için geçerlidir. Ayrıca öğretmenlerin düşünce ve görüşleri görüşme formunda yer alan soruların kapsamı ile sınırlıdır.



II. BÖLÜM

2. Kuramsal Çerçeve ve İlgili Alanyazın

Bu bölümde, bilgi işlemsel düşünmenin bileşenleri ve bilgisayar bilimi eğitiminde kullanılan öğretim yöntemlerinin yanı sıra bilgi işlemsel düşünme tarihinde bahsedilen iki önemli figüre ve bilgi işlemsel düşünmenin tanımına yer verilmiştir.

2.1. Bilgi İşlemsel Düşünme Tarihinde İki Önemli Figür

Bilgi işlemsel düşünme kavramı eğitim topluluğu için yeni değildir. 1960'larda Logo programlama dili ilk kez geliştirildiğinde eğitimde bilgi işlemsel düşünme ile ilgili çalışmalar görülmekteydi (Feurzeig & Papert, 2011). Bu dil tasarlanırken amaç, öğrencilerin matematiksel ve mantıksal düşüncelerini geliştirmelerine yardımcı olmaktı. Logo programlama dilinin öğrenciler tarafından kolaylıkla öğrenileceği varsayılmıştır. Logo kullanımının arkasındaki yaklaşım olan inşacılık (constructionism), yapılandırmacılığın (constructivism) bir yorumudur (Papert, 1990). Yapılandırmacılık, bireylerin aktif yaşantı yoluyla bilgi inşa ettiğini ileri sürmektedir. İnşacılıkta bireylerin düşünme, analiz etme gibi zihinsel aktiviteler yoluyla kendi bilgilerini kendilerinin inşa ettiğini savunur (Marlowe ve Page, 1998). Piaget (1964, s.176-186) bu konudaki düşüncelerini "bilgi oluşturulurken birey nesneyi değiştirir, dönüştürür ve bu dönüşüm sürecini anlayarak nesnenin nasıl oluşturulduğunun farkına varır" şeklinde ifade etmiştir. Papert (1993), öğrenme sürecinde mevcut araçları kullanarak ürün oluşturma ile zengin öğrenme ortamlarının sağlanması gerektiğini vurgulamıştır. Bireyler daha sonra oluşturdukları ürünü diğerleri ile paylaşarak ve onlarla iletişim kurarak bilgi alışverişinde bulunabilirler. Burada bahsedilen ürünler Logo dilinde yazılan kodlar

olabileceği gibi elektronik olmayan el ile üretilmiş materyaller de olabilir. Matematiksel ve mantıksal kavramlar soyut yapıdayken bireyler tarafından oluşturulan ürünler somut olduğu için soyut kavramlar somut ürünler aracılığıyla öğrenilebilir. Kodlama, öğrencilerin matematiksel ve mantıksal düşünme şekillerini geliştirmelerine yardımcı olabilir (Çetin ve Toluk Uçar, 2017).

Benzer bir yaklaşımla SET dili öğrencilerin matematik konularını öğrenmesine yardımcı olmak için kullanılmıştır. Dubinsky (1995), çok sayıda insanın programlama öğrenmeye istekliken az sayıda insanın matematik öğrenmede başarılı olduğunu savunmuştur. Dubinsky matematik öğretmek için bilgi işlemsel düşünmenin gücünden faydalanmaya karar vermiştir. Bu amaç doğrultusunda SET dili yapılandırıcı bir yaklaşım olan APOS (Action Process Object Schema) teorisine dayandırılarak, matematik öğretiminde kullanmak üzere ISETL olarak geliştirilmiştir (Çetin, 2016). APOS teorisi Piaget'nin yansıtıcı soyutlama kavramını ileri seviye matematik eğitimi bağlamında yeniden düzenler. Eylem (Action), Süreç (Process), Nesne (Object) ve Şema (Schema) matematiksel bir konuyu öğrenirken bir bireyin oluşturabileceği zihinsel yapılarıdır. Bu zihinsel yapıları geliştirecek zihinsel mekanizmalara yansıtıcı soyutlama denir. APOS teorisi öğretimde ISETL dilini kullanarak öğrencilerin Eylemden Süreç, Süreçten Nesne aşamasına geçmelerine yardımcı olunabileceğini söyler (Çetin ve Toluk Uçar, 2017). APOS teorisi üzerine çalışanlar matematiksel öğrenmenin bilgi işlemsel düşünme ile nasıl geliştirilebileceği konusunda çalışmalar yapmıştır. Buna ek olarak, APOS teorisi, öğrencilerin bilgisayar bilimi kavramlarını nasıl anladıklarını açıklamak için de kullanılmıştır (Çetin, 2015).

2.2. Bilgi İşlemsel Düşünmenin Tanımı

Bilgi işlemsel düşünmenin kesin bir tanıma ihtiyacı olmadığını düşünenler olduğu gibi, eğitim ortamına dâhil edildiği zaman ortak bir tanıma ihtiyaç duyulacağından bahseden yazarlar da vardır (Barr ve Stephenson, 2011; Guzdial, 2011; National Research Council, 2010; National Research Council, 2011). Çünkü bilgi işlemsel düşünme yeteneğini ölçen değerlendirme araçlarını geliştirme gibi konularda

sıkıntı yaşanabileceği düşünülmektedir. Farklı yazarlar tarafından farklı şekillerde tanımlanan bilgi işlemsel düşünmenin tanımı hakkında ortak bir görüş bulunmamaktadır. Seymour Papert bilgi işlemsel düşünme kavramını ilk kullanan kişidir (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".

Wing (2006), bilgi işlemsel düşünme teriminin sadece bilgisayar bilimcileri tarafından değil herkes tarafından bilinmesi gerektiğini önermiştir. Çünkü Wing'e göre bilgi işlemsel düşünme sadece bilgisayar bilimcileri için değil herkes için temel bir beceridir.

Çocukların okuma, yazma, aritmetik gibi analitik becerilerinin yanına bilgi işlemsel düşünmenin de eklenmesini önermiştir (Wing, 2006). Wing'in iddiaları bilgi işlemsel düşünmeye yeni bir ivme kazandırmıştır.

Wing (2006) makalesinde, bilgi işlemsel düşünmeyi tanımlarken; görünürde zor olan bir problemi çözülebilir bir problem olarak yeniden şekillendirmek, özyinelemeli düşünmek, karmaşık yapılarda soyutlama ve ayırıştırma kullanmak, çözüm üretmede sezgisel akıl kullanmak ve bilgisayar bilimcisi gibi düşünmekten bahsetmiştir. Wing aynı makalesinde bilgi işlemsel düşünmenin hangi niteliklere sahip olup olmadığını ifade etmiştir:

- Bilgisayar bilimi bilgisayar programlama ile aynı şey değildir. Bir bilgisayar bilimcisi gibi düşünmek, bir bilgisayarı programlayabilmekten daha fazlası anlamına gelir.
- Nasıl biyoloji için mikroskop ne ifade ediyorsa, bilgi işlemsel düşünme içinde bilgisayar onu ifade etmektedir. Yani bilgisayar, bilgi işlemsel düşünme için çok önemlidir olmazsa olmazlarından ama her şey demek değildir.

- Bilgisayarlar için değil insanlar için düşünmenin bir yoludur. Bilgi işlemsel düşünme insanlar için sorunları çözenin bir yoludur, insanları bilgisayar gibi düşündürmeye çalışmak değildir.
- Matematik ve mühendislik düşüncelerini tamamlar ve birleştirir. Bilgisayar bilimi, bütün bilimlerde olduğu gibi temellerinin matematiğe dayanması nedeniyle matematiksel düşünmeyi doğal olarak barındırır. Bilgisayar biliminde gerçek dünyayla etkileşime giren sistemler kurduğumuz için mühendislik düşüncesini de içinde barındırır.
- Bilgi işlemsel düşünme sadece ürettiğimiz yazılım ve donanım eserleri değil, fiziksel olarak her yerde olabilecek ve hayatımızda her zaman yer alabilecek eserlerdir. Problemleri çözmek, günlük yaşamlarımızı yönetmek ve diğer insanlarla iletişim kurmak, etkileşim içinde olmak için bilgi işlemsel kavramlar olacaktır.
- Bilgi işlemsel düşünme hayatımızın ayrılmaz bir parçası olduğu zaman herkes tarafından bilinen ve kabul edilen bir gerçeklik olacaktır.

ACARA bilgi işlemsel düşünmeyi “mantıksal olarak veri organize etmek, algoritmaları, modelleri tasarlayarak ve kullanarak problemleri bileşenlerine ayırmak gibi çeşitli teknik ve stratejileri içeren problem çözme yöntemi olarak tanımlamaktadır” (ACARA, 2012). Brennan ve Resnick (2012) ise Scratch deneyimlerini kullanarak bilgi işlemsel düşünmenin üç boyutunu tanımlamıştır: bilgi işlemsel kavramlar (programlama için kullanılan kavramlar), bilgi işlemsel uygulamalar (problem çözme uygulamaları), bilgi işlemsel bakış açıları (hesaplama uygulamaları üzerine düşünme ile farklı bakış açıları oluşturmak).

Wing, Cuny ve Snyder ile birlikte 2010 yılında bilgi işlemsel düşünmeyi; “çözümlerin bir bilgi işleme birimi tarafından etkili şekilde yerine getirilebilecek bir formda sunulması amacıyla problemleri ve çözümlerini formülleştirmeyi içeren düşünme sürecidir” şeklinde yeniden tanımlamıştır (akt. Wing, 2011; s. 20). Wing meslektaşları ile yaptığı tanımda bilgi işlemsel düşünmeyi, düşünme süreci olarak nitelendirmiştir. Burada problem ve çözümleri formülleştirilerek bunların bir bilgi işleme biriminin (robot, insan veya bilgisayar) anlayacağı şekilde ifade edilmesine

dikkat edilmelidir. Bilgi işleme birimi denilince aklımıza sadece bilgisayar gelmemelidir. Bilgi işleme her yerdedir, örneğin; zihnimizde veya biyolojide. Biyolojide Adenin, Timin, Guanin ve Sitozin sıralamasının değişmesiyle DNA'nın yapısı değişmekte ve farklı bilgiler ortaya çıkmaktadır. Fizik, kimya, biyoloji, ekonomi ve daha birçok alanda yapılan bilgi işlemsel düşünme çalışmaları bu alanlarda yapılan çalışmaların düşünme yollarını değiştirmektedir.

Sadece bilgi işlemsel düşünmeyi tanımlamak, öğrencilerin öğrenme doğasını ve onların öğrenme ortamlarındaki potansiyellerini ortaya çıkarmada yardımcı olacak yolları açıklamak için yeterli değildir. A-12 seviyesinde derslerde bilgi işlemsel düşünme ile ilgili temel olarak neler öğretilmesi gerektiğini belirlemek için bilgi işlemsel düşünmenin operasyonel tanımının yapılması gerekmektedir (Çetin ve Toluk Uçar, 2017). Öğrencilerin başarı seviyelerinin yükseltilmesinde, öğrencileri küresel dünya ile rekabete hazırlamada ve akademik becerileri günlük yaşamla harmanlamada birtakım becerilerin gerekliliğine inanan Uluslararası Eğitimde Teknoloji Topluluğu (ISTE – The International Society for Technology in Education) ve Bilgisayar Öğretmenleri Derneği (Computer Science Teachers Association – CSTA) bilgi işlemsel düşünmeye yönelik operasyonel bir tanım geliştirmek için bir çalışma yapmıştır. Bilgi işlemsel düşünme aşağıdaki süreçlerle sınırlandırılmamakla birlikte bu süreçleri içeren bir problem çözme sürecidir:

- Problemleri, bilgisayar veya diğer araçlarla çözmeye yönelik olarak formüle etme,
- Verileri mantıksal olarak organize ve analiz etme,
- Verileri model ve simülasyonlar gibi soyut biçimlerde sunma,
- Algoritmik düşünme ile otomatik çözümler geliştirme,
- Muhtemel çözümleri tanımlarken, analiz ederken ve uygularken kaynakları ve çözüm yollarını en etkili ve verimli bir şekilde seçmeyi amaçlama,
- Bu problem çözüm sürecini çeşitli problemlere genelleme ve transfer etme, uyarılama.

Bu beceriler bilgi işlemsel düşünmenin olmazsa olmaz boyutları olan bir dizi tutum ve yetenek tarafından desteklenmektedir. Bunlar:

- Karmaşıklıkla mücadele etmede güven,
- Zor problemlerle çalışırken sabır,
- Belirsizliğe dayanabilme,
- Açık uçlu problemlerle mücadele edebilme,
- Ortak bir amaç veya çözüm için diğerleriyle iletişim kurabilme ve çalışabilme.

Operasyonel tanımda problem çözme, problemi bileşenlerine ayırma, algoritmik düşünme, verileri analiz etme, verilerin temsili, modelleme ve simülasyon, otomasyon, çözümlerin verimi ve geliştirilmesi gibi kavram ve becerilerden bahsedilmiştir. Bu kavram ve beceriler, A-12 eğitim seviyesindeki okullarda bilgi işlemsel düşünmeye dair neler öğretilmesi gerektiği konusunda eğitimciler için bir çerçeve oluşturmaktadır (Çetin ve Toluk Uçar, 2017). Bu sayede öğretmenler öğrencilerine örneğin Code.org ile eğitim verirken amaçlarının sadece programlama öğretmek olmadığını bilir.

2.3. Bilgi İşlemsel Düşünmenin Bileşenleri

Birçok müfredat bilgi işlemsel düşünmenin tanımının belirsizliğini kabul ederken bilgisayar bilimlerindeki kavramlar ve teknikler üzerine odaklanmaya devam etmektedir. Müfredatlar bu kavram ve teknikleri sunarken, bilgi işlemsel düşünmenin tanımlarında sıklıkla bulunan terimlere yer vermektedir. Alanyazın incelendiğinde bilgi işlemsel düşünmenin düşünme süreci, problem çözme, algoritmik düşünme, ayrıştırma, soyutlama, genelleme ve değerlendirme kavramlarını içermesi konusunda ortak bir görüş vardır (Aho, 2011; Kalelioğlu, Gülbahar ve Kukul, 2016; Selby ve Wollard, 2013; Wing, 2006;). Bu kavramlar aşağıda daha detaylı olarak incelenecektir.

2.3.1. Düşünme süreci

Wing (2006), bilgi işlemsel düşünmeyi tanımlarken insanların problem çözme konusunda düşünmesinin bir yolu olarak nitelendirir. Problem çözme, bilgisayar bilimlerinde kullanılan zihinsel araç kümesini içermektedir. Bu araçlar zor bir sorunu

daha kolay çözülebilen bir hale dönüştürmek için kullanılır. Guzdial (2011) bilgi işlemsel düşünmenin öğretimi ile ilgili Wing'in açıklamalarına ek olarak bilgi işlemsel düşünmeyi bilgi işlem hakkında düşünmenin bir yolu olarak görmektedir. Bilgi işlemsel düşünmenin kapsamı ve niteliği üzerine çalışmalara katılanlar, bilgisayar bilimlerinden bir dizi zihinsel araç ve kavramların birleşmesinde hemfikir olmuşlardır. Aho (2011), algoritma olarak ifade edilebilen problemleri ve çözümlerini formülleştirmede düşünme süreçleri olarak bilgi işlemsel düşünmeyi tanımladığında problem dönüşümü fikrini ele almıştır. Bu düşünme süreçlerinin odak noktası çoğunlukla problem çözme olarak tanımlanır.

2.3.2. Problem çözme

Problem çözme, Anderson (1980) tarafından bilişsel işlemleri bir düzen içerisinde hedefe yöneltmek ve Morgan (1999) tarafından ise bir hedefe ulaşmada karşılaşılan engellerin üstesinden gelmenin en iyi yolunu keşfetmek olarak tanımlanmaktadır. Problem çözme sürecinde yapılması gerekenler, karşılaşılan probleme ve problemle karşılaşan kişiye göre farklı olsa bile problem çözme sürecinin belirli aşamaları bulunmaktadır. D'Zurilla ve Goldfried (1971) problem çözme sürecinde takip edilmesi gereken beş aşamadan bahsedilmektedir. Bunlar; problemi değerlendirme, problemi yönetme, karar verme, problemi çözme ve sonuçları inceleme.

Problem çözme yeteneğini geliştirmek, bilgi işlemsel düşünme yeteneğini geliştirme için önemlidir (Huang, Deng ve Rongsheng, 2009). Bilgi işlemsel düşünme bir problemin çözümünde, sezgisel akıl kullanır. Problemin çözümüne dair soracağımız soruların cevaplarını destekleyici teoriler üzerine dayanır. Görünüşte zor bir problemin çözümünde bilgi işlemsel düşünme sayesinde problemi gömme, dönüşüm, benzeşim yoluyla yeniden formüle ederek nasıl çözeceğimizi biliriz (Selby ve Woollard, 2013). Problem çözme, programlama öğretimi için bir temel olarak da tanımlanmaktadır. Mannila ve diğerleri (2014), bilgi işlemsel düşünmeyi çeşitli alanlardaki problemleri formülleştirmek ve çözüm üretmek için bilgisayar bilimindeki kavramları ve süreçleri kullanmak olarak tanımlamıştır. Alanyazında bilgi işlemsel düşünmeyi bir problem

çözme süreci olarak tanımlama üzerine bir fikirbirliği vardır. Wing'in bilgi işlemsel düşünme tanımı bilgisayar bilimi kavramlarını kullanarak problem çözmeyi içerir (Wing, 2006; Wing, 2008).

2.3.3. Soyutlama

Soyutlama, belirli bir durumdan ilgisiz detayları veya karmaşıklığı ortadan kaldırarak durumun özüne odaklanmayı gerektirir (Çetin ve Toluk Uçar, 2017). Örneğin, bir arabanın motorunun nasıl çalıştığının ayrıntıları ne sürücü ne de yolcularla alakalıdır. Yine de, bu bilgi olmadan, bir sürücü bir arabayı başarılı bir şekilde çalıştırabilir. Bilişim teknolojileri kavramı olarak soyutlama, problem ayrışması ile yakından ilgilidir. Daha büyük bir problemi bileşenlerine ayırırken, sorunun nasıl yapılandırıldığına dair daha iyi bir anlayış elde etmek için soyutlama yeteneği gerekebilir. Matematikle ilgili olarak soyutlama bir düşünme biçimi veya zihinsel yapıları ya da düşünme biçimlerini sınıflandırmak için bir araç anlamına gelmektedir (Kramer, 2007).

Frorer, Manes ve Hazzan (1997) soyutlamayı üç ayrı anlamda değerlendirmişlerdir. Birincisi, dikkati odaklamak için ayrıntının göz ardı edilmesi sürecini vurgular. Karmaşık nesnelerin bir veya birden fazla özelliğini göz ardı ederek, belirli özellikleri ön plana çıkartılır. Örneğin birbirinden farklı kuşlar gören bir çocuk kuşların iki kanadının olması, gagalarının olması, vücutlarının tüylerle kaplı olması gibi özellikleri üzerinde durarak ve onların göçebe olup olmadıkları, renkleri ve boyutlarını göz ardı ederek bir kuş şeması oluşturur. İkincisi, soyut veya somutun göreceli olmasıdır. Bir kavramın soyut veya somut olarak algılanması kişiye göre hatta aynı kişiye göre farklı durumlarda farklılık gösterebilir. Piaget (1970)'nin bu duruma verdiği örneklerden biri şu şekildedir; şeklinden bağımsız olarak madde miktarının korunması belirli yaş grubu için anlaşılması zor, soyut bir kavram olsa da çocuklar belirli bir yaş aralığından sonra bu kavramı somut halde anlayacaklardır. Üçüncüsü ise, özelliklerdir. Nesnelere ve işlemlerden oluşan bir yapıda nesnelere ve işlemlerden daha çok yapının özelliklerine bakılır. Örneğin tam sayılar ve dört işlemde oluşan bir yapıda tam sayılar

ve dört işlemde çok özelliklerine odaklanılır. Tam sayıların sıfıra bölümünün tanımsız olması, iki tam sayının toplamının yine bir tam sayı olması, en büyük negatif tamsayının -1 olması soyutlamanın özellikler anlamına örnektir.

Soyutlama fikri Wing (2006)'in ilk makalesinde karmaşıklığı gizleme olarak tanımlanmıştır ama tanım sonraki yıllarda gelişmiştir. Wing (2007), tanımı katmanlar arasındaki arayüzleri tanımlamak için birden fazla soyutlama katmanı ve düşünmenin eşzamanlı olarak dikkate alınması şeklinde değiştirir. Peter Denning (2007) de soyutlamanın programlama da dâhil olmak üzere bilgi işleminde önemli bir rol oynadığını kabul eder. Bununla birlikte soyutlama eyleminin bilgisayar bilimlerine özgü olmadığını belirtmiştir. Bir sonraki yıl Wing (2008) soyutlamayı bilgi işlemsel düşünmenin temel taşı olarak tanımlamıştır. Bilgi işlemsel düşünmenin kapsamı ve niteliği üzerine çalışmalara katılanlar bilgi işlemsel düşünmenin soyutlama, onları oluşturma ve onların arasındaki ilişkileri tanımlama süreçleri etrafında yoğunlaştığını kabul etmişlerdir. ISTE ve CSTA A-12 müfredatı ile ilgili raporları, ve Barr ve Stephenson (2011)'in bilgi işlemsel düşünmenin tanımı soyutlama kavramını içermektedir. Soyutlama kavramı, L'Heureux ve arkadaşları (2012) tarafından bilgi işlemsel düşünmeye yönelik bilgi teknolojisi yaklaşımlarının altı yönünden biri olarak keşfedilmiştir.

2.3.4. Ayırıştırma

D'Zurilla ve Goldfried (1971)'e göre ayırıştırma, karmaşık bir problemin kolaylıkla organize edilebilmesi için küçük, anlaşılabilir ve görece kolay çözümlenebilir bileşenlere ayrılmasıdır. Ayırıştırma, bir eseri bileşenleri açısından düşünmenin bir yoludur. Bu bileşenler ayrı ayrı anlaşılabilir, çözümlenebilir, geliştirilebilir ve değerlendirilebilir. Bu durum karmaşık problemlerin çözülmesini kolaylaştırır, karşılaşılan yeni problemlerin daha iyi anlaşılmasını sağlar ve büyük sistemlerin tasarımını kolaylaştırır (Csizmadia, Curzon, Dorling, Humphreys, Selby and Woollard, 2015). Örneğin kahvaltı hazırlamak ayrı etkinliklere ayrılabilir veya parçalanabilir; çay yap, yumurta haşla; vb. Bunların her biri sırayla bir dizi aşamaya da bölünebilir. Özgün

görevin ayrışmasıyla her parça, süreçte daha sonra geliştirilebilir ve entegre edilebilir. Basit bir oyun seviyesi, bir karakterin hareketleri, arkaplanı kaydırma ve karakterlerin etkileşimi hakkındaki kuralları belirleme gibi çeşitli parçalara ayrılabilir.

Ayrıştırma kavramı Wing (2006) tarafından, bilgi işlemsel düşünmenin bir parçası olarak problemleri parçalama şeklinde tanımlanmıştır. Karmaşık sistemler, büyük problemler veya karmaşık görevlerle uğraşırken ayrıştırma gerekir. Bilgi işlemsel düşünmenin kapsamı ve niteliği üzerine çalışmalara katılanlar da problemi ayrıştırma ihtiyacını belirtmişlerdir. Robert Tinker (2011), büyük problemlerin parçalarını bilgi işlemsel düşünmenin çekirdeği olarak görür. Danny Edilson (2011), çözümlerin oluşturulması için problemlerin belirli işlevsellik parçaları halinde parçalanmasını ve parçaların sıralanması gerektiğini belirtmiştir. En son olarak Guzdial (2012), bilgi işlemsel düşünmenin tanımını geliştirirken soyutlama ve ayrıştırmayı kapsayan araçların kullanılmasını içerir şeklinde bir tanımlamada bulunmuştur.

2.3.5. Algoritmik düşünme

Algoritma alanyazında farklı şekillerde tanımlanmaktadır, Olsen (2000) algoritmayı, bir problemi çözmek için tam olarak tanımlanmış talimatlardan oluşan bir yöntem olarak tanımlamıştır. Algoritmik düşünme, ihtiyaç duyulan adımların açık tanımlanması yoluyla çözüme ulaşmanın bir yoludur. Bir probleme tek bir çözüm bulmak yerine, algoritmalar geliştirerek bu probleme ve benzer problemlere cevap verebilecek talimatlar ve kurallar geliştirilir (Csizmadia ve diğerleri, 2015). Örneğin, hepimiz okulda çarpma işlemi yapabilmek için algoritmalar öğreniyoruz. Eğer çarpma işlemi yaparken öğretilen kuralları takip edersek, herhangi bir çarpma işlemine cevap bulabiliriz. Algoritmaya sahip olduğumuzda, yeni bir sorunla karşılaştığımızda her zaman sıfırdan çarpmanın nasıl yapılacağını öğrenmek zorunda değiliz. Algoritmik düşünmenin gücü, çözümlerin otomatikleştirilmesine izin vermesidir. Algoritmik düşünme, problem çözme ya da durumları anlama yolu olarak diziler ve kurallar açısından düşünebilme yeteneğidir (Csizmadia ve diğerleri, 2015).

Algoritmik düşünme, bilişimde eğitim ile elde edilebilecek en önemli yeterliliklerden biri olarak sıklıkla kullanılan bir terimdir. Futschek (2006, s. 160) algoritmik düşünmeyi verilen problemleri çözen yeni algoritmalar inşa etmek ve bunu yapabilmek için algoritmik düşünebilme becerisi olarak tanımlamıştır. Bu beceriler;

- Verilen problemleri analiz etme becerisi
- Bir problemi açık bir şekilde ortaya koyma becerisi
- Verilen problem için yeterli olan temel eylemleri bulma becerisi
- Temel adımları kullanarak belirli bir soruna doğru bir algoritma oluşturma becerisi
- Bir problemin tüm özel ve normal durumlarını yorumlama becerisi
- Bir algoritmanın verimliliğini artırma becerisi

Brown (2015)'a göre algoritmik düşünme, algoritmayı anlama, uygulama, değerlendirme ve oluşturma becerisidir. Algoritmik düşünme öğretiminde programlama dili kullanıldığında öğrenciler, programlama dilinin özelliklerine odaklandıklarından, algoritmik düşünme tasarımına yeteri kadar vakit ayıramamaktadırlar (Futschek, 2006). Bu nedenle Futschek (2006), algoritmik düşünme öğretiminde bir programlama diline bağlı kalmak yerine, sözde kod kullanılmasının ve öğrenci seviyesine uygun algoritmalar oluşturulmasının daha doğru olacağından bahsetmiştir.

2.3.6. Genelleme

Genelleme, problem çözme evrelerinden sonuçları inceleme aşamasında bireylerin ileride benzer durumları göz önünde bulundurarak problemi ve problem çözme sürecini gözden geçirebilme ve benzer problem durumlarına uyarlayabilme becerisidir (D'Zurilla ve Goldfried, 1971). Genelleme kalıpları, benzerlikleri ve bağlantıları belirleyerek bu özellikleri kullanma ile ilgilidir. Modelleri tanımlamayı ve kullanmayı içerir (Csizmadia ve diğerleri, 2015). Genelleme, daha önce geliştirilen çözümleri kullanarak ve önceki bilgi ve deneyimlere dayanarak yeni problemleri daha hızlı çözmeye yardımcı olur. "Bu, daha önce çözdüğüm bir probleme benziyor mu?" ve "Hangi yönlerden farklıdır?" gibi sorular sormak, hem kullanılan verilerde hem de

kullanılan süreçler / stratejilerdeki kalıpları tanıma işlemi gibi burada da önemlidir. Bazı spesifik problemleri çözen algoritmalar, benzer türde problemleri çözmek için uyarlanabilir (Csizmadia ve diğerleri, 2015). Yalnızca önceden karşılaşılan bir problemin çözümünde kullanılan bütün gerçeklikleri kullanarak genelleme yapılmış olmaz. Önceden karşılaştığımız problemlerin çözümünde kullanılan stratejilerin bir bölümünden faydalanarak, karşılaşılan bazı probleme çözüm bulunabilir (D’Zurilla ve Goldfried, 1971). Genelleme yapabilen bir öğrenci; eserlerdeki örüntü ve ortaklıkları belirleyebilir, çözümleri uyarlayarak, benzer problem durumlarında uygulayabilir ve fikir ve çözümleri bir problem alanından diğerine aktarabilir.

2.3.7. Değerlendirme

Bilgisayar biliminde üretilen bilginin günlük yaşam içerisinde, bilgisayar bilimi içerisinde veya başka alanlarda sistem kaynaklarını verimli bir şekilde kullanarak uygulanması istenebilir (Çetin ve Toluk Uçar, 2017). Sadece bir problem çözümü için geliştirilen bir algoritma ihtiyacı olandan fazla sistem kaynağı kullanıyor veya çok fazla zaman harcıyor olabilir. Bu nedenle üretilen algoritmaların verimlilik yönünden değerlendirilmesi gerekmektedir (Çetin ve Toluk Uçar, 2017).

Değerlendirme, bir algoritma sistem veya süreç için iyi bir çözüm üretmek amacıyla kullanılır (Csizmadia ve diğerleri, 2015). Çözümlerin çeşitli özelliklerinin değerlendirilmesi gerekir. Doğru mu? Yeterince hızlı mı? Kaynakları ekonomik olarak kullanıyorlar mı? İnsanlar için kullanımı kolay mıdır? Algoritmanın verimli olması, hafızada kullandığı alan ve işlemleri zamanında yerine getirip getirmemesi yönünden değerlendirilebilir (Sedgewick ve Wayne, 2011). Kullanıcılar tarafından çalıştırıldığında işlemleri olması gerekenden fazla sürede yapan ve bilgisayarı yavaşlatan programlar kullanılmak istenmez (Çetin ve Toluk Uçar, 2017). Örneğin, bir hastaya otomatik olarak ilaç vermek için tıbbi bir cihaza ihtiyaç duyulursa, hatasız, hızlı, basit ve güvenli bir şekilde programlanabilir olmalıdır. Yapılması gereken tasarımda, verimlilik ile etkililik ve kullanılabilirlik arasında bir etkileşim olmalıdır (Csizmadia ve diğerleri, 2015).

2.4. Bilgisayar Bilimi Eğitimde Kullanılan Öğretim Yöntemleri

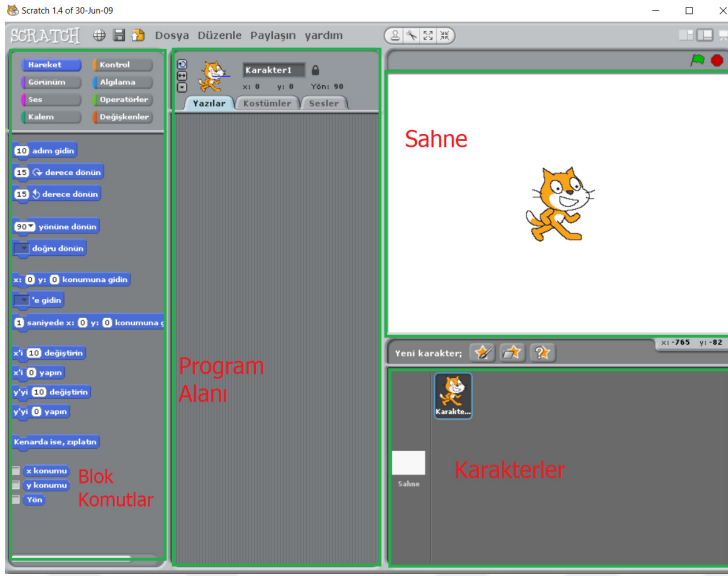
Bilgisayar bilimi günümüz okullarında yaygınlaşırken öğrencilerin bilgisayar bilimleri ile ilgili kavramları öğrenme zorlukları ile ilgili araştırmalar alanyazında sıklıkla yer almaktadır (Çetin, 2013; DuBoulay, 1986; Denner, Werner ve Ortiz, 2012; Robins, Rountree ve Rountree, 2003). DuBoulay (1986) programlamaya yeni başlayan birinin karşılaşacağı 5 tür zorluğu kategorize etmiştir: (1) programlamaya uyum sağlama, (2) bilgisayar donanımının bir zihinsel modelinin oluşturulması (kavramsal makine olarak adlandırılır), (3) sözdizimi ve anlambilim, (4) programlamada standart problemlerle ilgili standart yapılar, (5) programlama pragmatikleri. Öğrencilerin zorlukların üstesinden gelmelerine yardımcı olmak için, bilgisayar bilimi eğitimi araştırmacıları tarafından iki soru ele alınmalıdır: Yeni öğrenenler programlama kavramlarını nasıl öğrenirler ve eğitimciler yeni başlayanların programlama kavramlarını öğrenmelerine nasıl yardımcı olabilirler? Bilgi işlemsel düşünme becerilerinin öğretimi sınıf içinde ve dışında yapılan uygulamalar, programlama aktiviteleri ve oyunlar ile yapılabilmektedir (Apostolellis, Stewart, Frisina ve Kafura, 2014; Basawapatna, Repenning, Koh ve Savignano, 2014; Lee, Martin ve Apone, 2014; Prater ve Mazur, 2014). Bilgi işlemsel düşünme becerilerinin öğretiminde kullanılacak yaklaşımlardan en çok tercih edilenleri; blok tabanlı uygulamalar, robot uygulamaları, bilgisayarsız bilgisayar bilimi ve metin tabanlı ortamlardır (Berry, 2014; Bers, Flannery, Kazakoff ve Sullivan, 2014; Nishida, Kanemune, Idosaka, Namiki, Bell ve Kuno, 2009; vanDam, 2000; Weinberg, 2013).

2.4.1. Blok tabanlı görsel programlama ortamları

Son zamanlarda bilgisayar biliminin nasıl öğretilmesi gerektiğine yönelik yapılan çalışmalar, dijital okuryazarlığa, yeni teknolojilerin oluşturulmasına ve teknolojilerin oluşturulmasının nasıl sağlanacağı konusuna odaklanmıştır (Bower ve Falkner, 2015). Öğrencilerin günümüzün dijital okuryazarları olmalarını sağlamak, üretilen teknolojileri kullanmak yerine üreten bireyler olmalarını sağlamak için bilgisayar bilimi kavramlarını öğrenmeye erken yaşta başlamaları önerilmektedir

(Yükseltürk ve Altıok, 2015; Wilson ve Guzdial, 2010). Bilgisayar bilimi kavramlarını erken yaşta öğrenmek öğrencilerde bilgi işlem öz yeterliliği artırarak bilgi işlem konusundaki ilgiyi artırdığı düşünülmektedir (Akbulut ve Looney, 2009). Öğrencilerin bilgi işlemsel düşüncelerini geliştirmek için görselleştirme araçlarının kullanımının etkili olacağına dair bazı kanıtlar bulunmaktadır (Korkmaz, 2016). Blok tabanlı görsel programlama ortamları, öğrencilerin kodlama yaparken ezberlemek yerine mantığı öğrenmesine (Byrne ve Lyons, 2001; İmal ve Eser, 2009), söz dizimi ile yaşadığı sıkıntıları gidermesine (Kinnunen ve Malmi, 2008; Lahtinen, Ala-Mutka ve Jarvinen, 2005) ve algoritma oluştururken yaşadığı zorlukları aşmasına yardımcı olduğu düşünülmektedir. Bu ortamlar öğrencilere programlama konusunda önbilgiye ihtiyaç duymadan programlama yapabilme olanağı sunmaktadır. Blok tabanlı kodlama ortamları programlamanın anlaşılması zor olan yapısını ve öğrenirken karşılaşılan sorunları en aza indirmek amacıyla, öğrencinin öğrenmesini destekleyici bir yol olarak önerilmektedir (Yükseltürk ve Altıok, 2015). Bu blok tabanlı araçlar arasında ilkökul ve ortaokul düzeyindeki öğrenciler için Scratch, Alice ve App Inventor gösterilebilir.

Günümüzde oldukça popüler olan Scratch programlama ortamı, inşacılık ilkelerine ve Logo dilinden öğrenilen derslere bağlı olarak MIT’de (Massachusetts Institute of Technology) geliştirilmiştir (Resnick, 2012). Scratch, alçak zemin (öğrenmesi kolay) ve yüksek tavan (kullanıcıların göreceli olarak karmaşık projeler oluşturmasına izin verir) özelliklerine sahip bir programlama ortamıdır. Program öğrencilerin görsel kodlama bloklarını bir araya getirerek kod yazmalarına olanak sağlar. Öğrenciler kod bloklarını birleştirebilir ve sonuçları hemen gözlemleyebilir. Daha sonra bloklar ayrılabilir ve istenilen şekilde yeniden birleştirilebilir. Ayrıca Scratch sağladığı geniş duvarlar sayesinde öğrenciler tarafından anlamlı projeler geliştirmeyi destekler. Kullanıcılar Scratch ile kodlayarak oyunlar, animasyonlar, simülasyonlar ve interaktif hikâyeler yapabilir. Şekil 2.1’de Scratch programının ara yüzü sunulmuştur.



Şekil 2.1. Scratch programlama ortamı

Blok tabanlı görsel programlama araçları ile yapılan programlama eğitimi araştırmalarının sonuçlarına bakıldığında genel olarak öğrenciler üzerinde olumlu etkilerden bahsedilmektedir. Kalelioğlu ve Gülbahar (2014) ilkokul öğrencileri ile yapmış oldukları araştırmada öğrencilerin Scratch kullandıktan sonra programlamaya karşı olumlu tutum sergilediklerini gözlemlemişlerdir. Kelleher, Pausch ve Kiesler (2007) ortaöğretim öğrencileri ile yaptıkları araştırmada, Alice programı ile 3 boyutlu ortamlar oluşturarak bilgisayar programlamanın temellerini öğrenirken öğrencilerin daha iyi motive oldukları sonucuna ulaşmışlardır.

2.4.2. Robot uygulamaları

Eğitimde robotlar öğretmen, öğretmene yardımcı olacak bir rehber, öğrenme sürecinde öğrenenlere rehberlik eden konu alanı uzmanı veya öğretim materyalleri rolü gibi birçok amaçla kullanılmaktadır. Eğitimde robotların kullanımına yönelik olarak iki yaklaşım bulunmaktadır. Birincisi, öğretimin hedefinde robot kontrolü, robot tasarımı, yapay zekâ gibi robot ve robotla ilgili konuların yer aldığı, üniversite düzeyindeki öğrenenlere yönelik eğitim faaliyetleridir. İkincisi, robotların bir öğretim aracı olarak matematik, fen gibi derslerin öğretiminde araç olarak, A-12 seviyesindeki öğrenenler

için kullanılmasıdır (Malec, 2001). A-12 seviyesinde kullanılan robotlar eğitsel robotlar olarak adlandırılmaktadır. Eğitsel robotlar çocuklara bilgisayar bilimi alanında yaptıkları uygulamaları test etme ve problem çözme becerilerini uygulama imkânı sunmaktadır (Bers, Flannery, Kazakoff ve Sullivan, 2014). Robotların eğitim amaçlı kullanımlarını konu alan çalışmalarda, robotların öğrencilerin problem çözme becerilerine olumlu katkısı olduğu (Nourbakhsh, Hamner, Crowley ve Wilkinson, 2004; Robinson, 2005; Rogers ve Portsmore, 2004) ve öğrencilerin fen ve matematik derslerindeki motivasyonunu arttırdığı (Robinson, 2005; Rogers ve Portsmore, 2004) görülmektedir.

A-12 seviyesindeki öğrenciler basit montaj setleri ile yapay organizmalar ve hayvan davranışlarını taklit eden robot projeleri yapabilirler (Miglino, Lund ve Cardaci, 1999). Ancak robotik projeler yapabilmek için programlama bilgisinin yanı sıra mekanik ve elektronik bilgisine de sahip olunması gerekmektedir. Çocukların bir mühendis seviyesinde bu bilgilere sahip olması beklenemez (Kalelioğlu ve Keskinçılıç, 2017). Bu nedenle çocukların seviyesine uygun, programlanabilir, kolayca sensör monte edilebilen, tak çıkar parçalar içeren, uzaktan kumandalı Makeblock, Mbot, Lego, Ozobot ve Makey makey gibi robotik kitler kullanılmaktadır. Şekil 2.2 ve Şekil 2.3'te sırasıyla Makeblock Mbot ve Ozobot robot kitleri gösterilmiştir. Bu ve benzeri lehimleme gerektirmeyen kitler ilköğretim seviyesindeki öğrenciler için tercih edilmektedir. Ortaöğretim ve lise düzeyindeki öğrenciler elektrik bilgisi içeren dersler almaktadırlar. Robotik bir sistem tasarlamada bu öğrencilere yönelik olarak Arduinio gibi mikro denetleyici kartlar kullanılan robotik kitler de tercih edilebilir. Mikro denetleyici kartlar ve sensörler, soyut girdi ve çıktılara dayanan programlama ortamlarının çocuklar için somut girdi ve çıktı sağlayan programlama ortamlarına dönüştürmede kullanılmaktadır. Mikro denetleyicilerin programlanması Scratch for Arduinio ve mBlock gibi görsel blok programlama yazılımları ile yapılmaktadır. Bu yazılımlar sayesinde öğrenciler daha zevkli ve kolay bir şekilde, sadece sürükle bırak mantığı mikro denetleyicileri programlayabilmektedir (Kalelioğlu ve Keskinçılıç, 2017).

diğerlerini göz ardı etmesi soyutlama, benzerlikleri ve farklılıkları algılama, örüntüyü algılama işleminde genelleme ve robotun tasarlanan robotun gerekli kriterlere uyup uymadığını değerlendirme becerisini kullanmaktadırlar.

2.4.3. Metin tabanlı ortamlar

Günümüzde bilgisayar bilimcileri tarafından geliştirilmiş birçok programlama dili bulunmaktadır. Bilgisayar bilimcilerinin karşılaştıkları problem türü ve problemin çözümünde kullandıkları paradigmaların her biri farklı programlama dili tarafından desteklenmektedir (Kandemir, 2017). Bilgisayar bilimlerinde temel olarak dört farklı programlama paradigması kabul edilmektedir (Vujosevic – Janicic & Tosic, 2008). Bunlar; prosedürel paradigma, fonksiyonel paradigma, nesne yönelimli paradigma ve mantıksal paradigmadır.

Prosedürel programlama, bir durum üzerine programlama ve durumu değiştiren komutlar üzerine kuruludur. Bu paradigmaya dayalı programlar bilgisayarın gerçekleştireceği sıralı komutlardır. Önemli olan hesaplamaların nasıl yapılacağıdır. Hemen hemen tüm bilgisayarların donanımı makine kodunu çalıştıran tarzda tasarlanmıştır. Programlar arka arkaya yürütülen komutlardan oluşur. Hafızada yer alan veriler kullanılarak işlemler yapılır ve sonuçlar yine hafızanın bir bölgesine konulur. Bu nedenle programın hangi adıma ulaştığı hafızanın içeriğinden anlaşılabilir. C (1971), Pascal (1970), Algol 60 (1960), Basic (1964), Fortran (1954) prosedürel paradigmaya dayalı dillerdendir (Kandemir, 2017).

Fonksiyonel programlama paradigması matematikteki fonksiyon mantığını temel almaktadır. Programlar değerlendirilerek yürütülen tek bir deyimden oluşur. Veri üzerinde değişiklik yapmaktan kaçınır. Otomatik bir hafıza yönetimi vardır. Başka fonksiyonları argüman olarak alan üst düzey fonksiyonlar kullanılır. İşlemci kullanımı ve hafıza yönetimi açısından daha az verimlidir (Demirci, 2007). Fonksiyonel paradigmaya dayalı dillere örnek verecek olursak; Scheme (1975), ML (1973), APL (1960), LIPS (1958) (Kandemir, 2017).

Nesne yönelimli paradigma her bilgisayar programının etkileşim içerisinde bulunduğu birimler ve nesnelere kümesinden oluştuğunu varsayan bir oluşumdur. 1960'lı yıllarda donanım ve yazılımın karmaşıklaşmaya başlamasıyla birlikte, yazılım kalitesinin korunabilmesi ve yeniden kullanılabilirliği artırmak için düşünülmüştür (Demirci, 2007). Bu paradigmada esas olan şey nesnedir ve bu nesne üzerine çeşitli fonksiyonlar çağrılarak nesnede istenilen değişikliklerin yapılması sağlanır. Tekil nesnelere genel halini tarif eden sınıflar oluşturulur. Bu sınıflar içerisinde veri yapılarını ve bu veri yapıları üzerinde değişiklik yapmamızı sağlayacak fonksiyonlar tanımlanır. Bu şekilde bir mantık izleyerek programlama yapılır. Nesne yönelimli paradigmaya dayalı dillere örnek verecek olursak; Java (1995), Python (1990), Visual Basic (1991), C++ (1983), Smalltalk (1969).

Mantıksal programlama paradigması genel anlamda matematiksel mantığı esas alır. Bir problem çözümü algoritmalarla değil mantıksal ifadeler ile tanımlanır. Bu paradigmanın diğerlerinden farkı yorumlama da yapabilmesidir. Mantıksal programlama paradigmasını destekleyen diller, belirli bir koşulun varlığını kontrol ederek ve koşul sağlanıyorsa, uygun bir işlem gerçekleştirerek çalışırlar (Demirci, 2007). Programlamacının görevi problemle alakalı temel mantıksal ilişkileri belirlemektir (Kandemir, 2017). Bu modeldeki dillere en tanınmış örnek Prolog (1970) programlama dilidir.

İlk zamanlarda programlamayı yeni öğrenenler için kullanılan araçlarla bilgisayar bilimcilerinin kullandığı araçlar arasında büyük bir farklılık bulunmamaktaydı (Kölling, 2016). Değişken ve fonksiyon tanımlamaları, matematik ve operatör sembelleri ve semboller, virgül, noktalı virgül gibi metin tabanlı programlama dillerindeki söz dizimi kuralları, programlama dili öğretiminde karşılaşılan problemlerden birisiydi (Kandemir, 2017). Öncelikli hedefi programlamaya yeni başlayanlar olan, pedagojik yaklaşıma sahip ilk metin tabanlı programlama dilleri arasında Pascal (1970), Logo (1967) ve BASIC (1964) en çok kullanılanlardır (Kölling, 2016). Bu programlama dillerinin amacı diğer programlama dillerinde meydana gelebilecek karmaşıklık ortadan kaldırarak basit bir hale getirmektir (Kandemir, 2017). Programlamaya yeni başlayanlara yönelik olarak veya erken yaşta programlama öğrenimini sağlamak için, eğitsel metin

tabanlı programlama dili olarak Python örnek verilebilir. Python dilinin diğer metin tabanlı programlama dillerine göre avantajı, blok tabanlı programlama dilleri gibi söz dizimi basitliği nedeniyle programın çalışmasını takip etmek ve programı yorumlamak gibi işlemlerin kolay bir şekilde yapılmasıdır (Kandemir, 2017). Bu konuda Python dilinin özellikleri Pascal ve Logo dili ile karşılaştırılabilir (Kandemir, 2017). Aşağıda “Merhaba Dünya” kodunun Pascal, Java ve Python metin tabanlı programlama dillerinde yazılışı gösterilmiştir.

<pre> Public class Hello World { Public static void main (String [] args) { System.out.println("Hello World."); } } </pre>	<pre> print ("Hello World") </pre>	<pre> Program Hello World (output) begin WriteLn ('Hello World'); end. </pre>
i. Java	ii. Python	iii. Pascal

Şekil 2.4. “Hello World” kod parçaları gösterimi (Kandemir, 2017)

2.4.4. Bilgisayarsız Bilgisayar Bilimi Öğretimi

Bilgisayar bilimi öğretiminde kullanılan bir diğer yaklaşım ise bilgisayarsız bilgisayar bilimi (B³) öğretimidir (Kalelioğlu, 2017). B³ öğretim etkinlikleri kartlar, boya kalemleri, bulmacalar ve hoşça giden çok sayıda oyun aracılığıyla bilgisayar bilimi ve bilgi işlemsel düşünme kavram ve yaklaşımlarını öğreten ücretsiz öğrenme etkinliklerinin bir koleksiyonudur (Celaya, 2016; Kalelioğlu, 2017). Öğrenciler bilgisayar kullanımına gerek kalmadan, dikkat dağıtan ve teknik detaylardan arındırılmış etkinlikler aracılığıyla, ikili sayılar, algoritmalar, veri sıkıştırma ve arayüz tasarımı gibi konular ile teknik olarak deneyimlemeden tanışabilirler (Bocconi, Chiocciariello, Dettori, Ferrari, Engelhardt, Kampylis ve Punie, 2016). B³ etkinlikleri sadece programlama öğretimini değil daha geniş bir alanı kapsayan konuları örneğin; yapay zekâ, insan bilgisayar etkileşimi, bilgisayar grafikleri gibi konuları da ele almaktadır (Bell, 2014). Pek çok ülkede yaygın olarak kullanılan ve son derece popüler bir yaklaşım olan B³ etkinlikleri, problem çözme becerisini, grup çalışmasını ve yaratıcılığı teşvik etmektedir

(Cortina, 2015). Öğrencilerin fiziksel olarak problem çözümünde yer almaları etkinlikleri ilgi çekici hale getirmektedir (Bocconi ve diğerleri, 2016).

B³ etkinliklerinden bir örnek verecek olursak, metin sıkıştırma etkinliğinde öğrenciler tekrar edilen harfleri bularak metnin sıkışmasını sağlarlar. Öğrenciler bu etkinlik ile bilgisayarın verileri sabit diske sıkıştırma işlemini nasıl gerçekleştirdiği hakkında bilgi sahibi olabilir. Etkinlik sırasında öğrenciler arkadaşları ile birlikte, büyük paragrafları daha küçük hale getirmek için yarışabilirler (Kalelioğlu, 2017).

B³ etkinlikleri üzerine pek çok proje bulunmaktadır. Bu projelerden, Bilge Kunduz, Keşf@ - Kodlamayı Keşfediyorum, Eğlence için Bilgisayar Bilimi (CS4FN), Bilgisayarsız Code.org (Code.org unplugged) ve Bilgisayarsız Bilgisayar Bilimi (CS Unplugged) projeleri popüler olanlardır (Kalelioğlu, 2017). Bu projelere bakıldığında ortaya çıkış amaçlarının bilgisayar bilimine olan farkındalığı arttırmak olduğu görülmüştür. Ayrıca etkinliklerin bilgisayarsız olarak uygulanması teknolojik alt yapı sorunu yaşayanlar adına fırsat eşitliği sağlayabilir (Kalelioğlu, 2017). Alanyazında B³ etkinlikleri adına yapılan çalışmalarda (Nishida, Kanemune, Idosaka, Namiki, Bell ve Kuno, 2008; Mano, Allan ve Cooley, 2010; Taub, Ben-Ari ve Armoni, 2009; Taub, Armoni ve Ben-Ari, 2012; Thies ve Vahrenhold, 2013) öğrencilerin etkinlikler ile hayal gücünün ve düşünme becerilerinin arttığı, tutumlarının olumlu yönde değiştiği, bilgisayar bilimine olan ilgilerinin arttığı, bilgisayar bilimi kavramlarını daha iyi anladıkları ve matematik ile ilgili kendilerine olan güvenlerinin arttığı görülmüştür (Kalelioğlu, 2017).

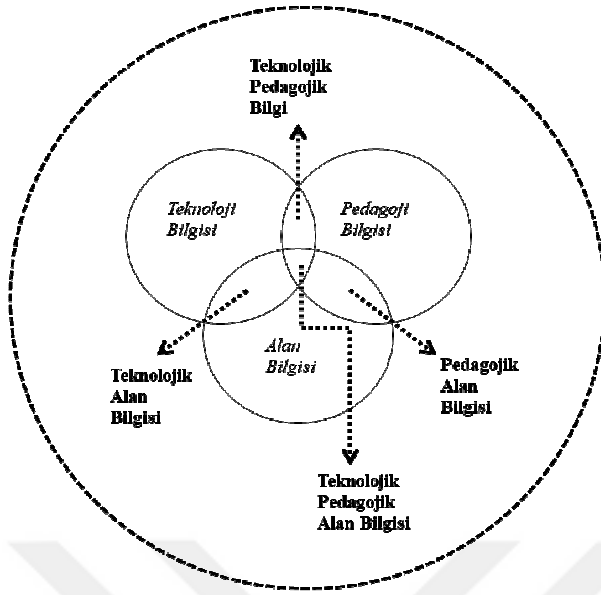
2.5. Teknolojik Pedagojik Alan Bilgisi (TPAB)

Geçmişte öğretmen yeterliliği alanı ile ilgili konuyu aktarabilmesi ile belirlenmekteydi, pedagojik bilgileri ikinci plandaydı (Fullan & Langworthy, 2014). Shulman (1986) ise pedagoji bilgisi ile alan bilgisi arasındaki ilişkiyi, bir öğretmenin alanına yönelik konu hakkında bilgi verirken bilgiyi öğrencilere nasıl öğretebileceği hakkındaki bilgisinin yalnızca bilgiye hâkim olmaktan daha önemli olduğunu dile

getirerek açıklamıştır. Shulman tarafından Pedagojik Alan Bilgisi (PAB) olarak adlandırılan bu yapı konunun en verimli şekilde öğretimini sağlayacak yöntem, teknik ve stratejileri bilmekle ilgilidir (Göl, 2016).

Koehler ve Mishra (2008) günümüzde teknolojinin günlük hayatımızın olduğu gibi eğitiminde bir parçası olması gerektiğini savunarak teknoloji bilgisinin de PAB'a eklenilmesi gerektiğini düşünmüştür. Teknolojinin, pedagojiden ve alan bilgisinden bağımsız bir şekilde, öğrencinin nasıl öğrendiğini bilmeden eğitime entegre edilemeyeceği anlaşılmış (Graham, Burgoyne, Pamela Cantrell, Smith, Clair ve Harris, 2009) ve Shulman (1986)'ın modeline (PAB) teknoloji bilgisinin eklenmesi ile Teknolojik Pedagojik Alan Bilgisi (TPAB) modeli geliştirilmiştir (Şekil 2.5). TPAB, teknoloji, pedagoji ve alan bilgisi olmak üzere üç temel bileşenin ötesinde bu bileşenlerin arasındaki etkileşimle ortaya çıkan bilgi türüdür (Mishra & Koehler, 2008). Teknolojinin eğitime sağlıklı bir şekilde entegre edilebilmesi, üç temel bileşenin birbirinden bağımsız bir şekilde düşünülmesiyle değil; bileşenler arasındaki ilişkinin dikkate alınmasıyla gerçekleşir. Birbirinden ayrı düşünülmemesi gereken bu bileşenlerden birinde meydana gelecek değişiklik diğer iki bileşeni de etkileyecektir (Harris, Mishra ve Koehler, 2009). TPAB'ın gelişigüzel meydana gelmiş bir yapı olmadığından bahseden Niess (2005) tarafından bu sürecin değişkenleri şu şekilde belirtilmiştir;

- Herhangi bir konunun teknoloji ile öğretiminde kullanılması gereken öğretim yöntem ve tekniklerini bilmek,
- Öğrenme sürecine teknoloji entegre edildiğinde öğretimin ne ifade ettiğini anlamak,
- Herhangi bir konunun teknoloji destekli öğretiminde, öğrencilerin öğrenme durumları ile ilgili bilgi sahibi olmak,
- Öğrenme sürecine teknoloji entegre edilmesinde kullanılacak ilgili teknolojileri ve materyalleri bilmek.



Şekil 2.5. TPAB yapısının bileşenleri (Koehler ve Mishra, 2008)

TPAB'ın temel bileşenlerini ve bu bileşenler arasındaki etkileşim ile oluşan kavramları tanımlayacak olursak;

Alan Bilgisi (AB): Öğretmenlerin öğretilen veya öğrenilecek konu hakkındaki sahip olduğu bilgidir (Koehler & Mishra, 2009). Konu alanı hakkında yetersiz bilgiye sahip olan öğretmenler, öğrencilerin konu ile ilgili yanlış kavram geliştirmeleri gibi istenmeyen sonuçlara neden olabilirler (Harris ve diğerleri, 2009).

Pedagoji Bilgisi (PB): Öğretmenlerin öğretme ve öğrenme sürecinde, öğretim yapılacak hedef kitlenin özellikleri, öğretim yöntem ve teknikleri, sınıf yönetimi, uygulama ve değerlendirme stratejileri gibi konularda sahip oldukları bilgilerdir (Koehler ve Mishra, 2008).

Teknoloji Bilgisi (TB): Yeni nesil teknolojiler hakkında sahip olunması gereken bilgidir. TB, bu bilgileri kullanabilme becerilerini de ifade etmektedir. Dijital teknolojiler açısından TB, yazılım ve donanımlar hakkındaki bilgi ve becerilere hâkim olunmasını da içerir (Karataş, 2014). Öğretmenlerden TB'nin yanında, teknolojik gelişmelere ayak uydurmaları ve teknoloji odaklı araçların nasıl kullanılması gerektiği

ile ilgili bilgiye sahip olmaları istenmektedir (Koehler ve Mishra, 2008; Ozgun-Koca, Meagher ve Edwards, 2010).

Pedagojik Alan Bilgisi (PAB): Öğretmenin, öğrencilere belirli bir konu alanını nasıl öğretmesi gerektiği bilmesidir. Öğretmen, farklı ilgi ve yetenekleri olan öğrenen profiline göre konu alanını en verimli şekilde aktarabileceği uygun strateji yöntem ve teknikleri bilmelidir (Harris ve diğerleri, 2009).

Teknolojik Alan Bilgisi (TAB): Öğretmenlerin hangi teknolojilerin öğretim sürecinde daha yararlı olacağını ve alanın teknolojiye nasıl etkilediğini ya da nasıl değiştirdiğini ve alanın teknoloji seçimini nasıl etkilediğini anlamaya ihtiyaçları vardır (Mishra & Koehler, 2008). Örneğin; Geometer's Sketchpad yazılımı ile öğrenciler şekiller oluşturabilmekte, oynayabilmekte ve geometrik ispatlar yapabilmektedirler. Aynı zamanda öğrenciler bilgisayar programı aracılığıyla oyun oynayarak geometri öğrenimini gerçekleştirdiğinden, bilgisayar programı geometri öğrenmenin doğasını değiştirmektedir (Harris ve diğerleri, 2009).

Teknolojik Pedagojik Bilgi (TPB): Öğrenme öğretme sürecinde teknolojinin pedagoji ile birlikte kullanımının süreci nasıl değiştirdiği ile ilgili bilgidir (Koehler ve Mishra, 2009). Cox (2008)'a göre TPB, teknolojilerin pedagojik bağlamda kullanımının fayda ve sınırlılıklarının bilinmesi ve bu teknolojilerin kullanılan pedagojik yöntemi ve öğrencilerin öğrenmesini nasıl etkilediği aynı zamanda bunlardan nasıl etkilendiği ile ilgili bilgidir. Örneğin; Öğretmenlerin derste, öğrencilerin gelişimsel özelliklerini dikkate alarak dijital araçlar kullanması (Öztürk ve Horzum, 2011).

III. BÖLÜM

3. Yöntem

Bu bölümde araştırmanın deseni, araştırmaya katılan katılımcılar, araştırmada kullanılan veri toplama aracı ve geliştirilme süreci, verilerin nasıl toplandığı, yorumlandığı ve çözümlendiği açıklanmıştır.

3.1. Araştırma Deseni

Bilişim teknolojileri ve yazılım dersi öğretmenlerinin bilgi işlemsel düşünme ile ilgili bilgilerini ve düşüncelerini ortaya çıkarmanın amaçlandığı bu araştırmada nitel araştırma yöntemi benimsenmiştir. Nitel araştırma alanyazınında, nitel araştırmaya yönelik genellenebilir bir tanımlama yapmanın güç olduğu görülmektedir. Bunun nedeni ise nitel araştırma çatısı altında bulunan birçok kavramın farklı disiplinlerle bağlantılı olmasından kaynaklanmaktadır. Nitel araştırma desenlerine birkaç örnek verecek olursak; durumsal araştırma, yorumlayıcı araştırma, betimsel araştırma ve kuram geliştirme bunlardan bazılarıdır. Her ne kadar nitel araştırmanın bütün yönlerini ve bütün özelliklerini kapsayacak bir tanım yapmak güç olsa da, alanyazınında farklı şekillerde tanımlamalar mevcuttur. Örneğin, Yıldırım ve Şimşek (2013) nitel araştırmayı, olayların ve algıların gerçekçi ve bütüncül bir şekilde, nitel veri toplama yöntemlerinden görüşme, gözlem ve doküman analizi gibi yöntemler kullanılarak ortaya konulmasında nitel bir sürecin izlendiği araştırma şeklinde tanımlamıştır.

Bu çalışmada nitel araştırma yöntemlerinden durum çalışması kullanılmıştır. Durum çalışması, belirli bir zaman diliminde gerçekleşen durumların, kendi bağlamıyla

bir bütün halinde olduğu kesin sınırlarının çizilmesinin zor olduğu, çoklu kaynakları içeren veri toplama araçları (görüşmeler, gözlemler, dokümanlar, raporlar) ile derinlemesine incelenen durumların tanımlandığı bir araştırma yöntemidir (Creswell ve Plano Clark, 2007; Yıldırım ve Şimşek, 2013; Yin,1984). Bu yöntemin en önemli özelliği incelenen bireyin, topluluğun ya da durumun kendisine özgü nitelikleri nedeniyle doğal ortamı içerisinde incelenmesidir (Ersoy, 2016). Bu nedenle araştırmanın amacı tüm evrene yönelik genelleme yapmak değil; kişinin, topluluğun ya da durumun ayırt edici özelliklerini anlamak ve sistematik biçimde açıklayabilmektir. Bu doğrultuda durum çalışması bu araştırmanın hedefini gerçekleştirmede uygun bir yöntem olarak görülmektedir. Durum çalışmaları, ‘nasıl’ ve ‘niçin’ sorularını temel alarak, araştırmacının kontrol edemediği bir olgunun ya da olayın detaylı bir biçimde incelenmesine imkân sağladığından (Yıldırım ve Şimşek, 2013), araştırmada yorumlayıcı bir yaklaşımla “ne oluyor?” sorusuna yanıt aranmıştır.

3.2. Katılımcılar

Araştırmanın katılımcılarını, 2016-2017 öğretim yılı bahar dönemi Bolu il merkezi ve Mengen, Dörtdivan, Mudurnu, Yeniçağ, Seben, Göynük, Gerede olmak üzere Kırbrısık dışındaki bütün ilçelerinden 22 bilişim teknolojileri ve yazılım dersi öğretmeni oluşturmaktadır. Katılımcıların seçiminde amaçlı örnekleme yöntemlerinden maksimum çeşitlilik örnekleme kullanılmıştır. Amaçlı örnekleme, araştırmanın amacı bağlamında bilgi açısından zengin durumların derinlemesine çalışılmasına imkân sağlamaktadır (Patton, 2002). Maksimum çeşitliliğin temel alınmasında amaç, çeşitliliği sağlayarak araştırılan duruma genelleme yapmak değil, aksine çeşitlilik gösteren durumlar arasındaki ortaklık veya benzerlikleri maksimum derecede yansıtmaktır (Yıldırım ve Şimşek, 2013). Bu sebeple katılımcılar, Bolu merkez ve Mengen, Dörtdivan, Mudurnu, Yeniçağ, Seben, Göynük, Gerede ilçelerinde bulunan ortaokullardaki öğretmenler arasından gönüllülük esasına göre seçilmiştir.

Araştırmaya katılacak olan öğretmenler belirlenirken bilişim teknolojileri ve yazılım dersine hâlihazırda giriyor olmalarına dikkat edilmiştir. Katılımcıların cinsiyeti,

mezun oldukları bölüm, kaç yıldır öğretmen oldukları ve araştırmaya katıldıkları ilçeler Tablo 3.1’de yer almaktadır.

Tablo 3.1. Katılımcıların demografik özellikleri

Öğretmenler	Cinsiyet	Lisans	Deneyim	İl/ilçe
Ö1	Erkek	BÖTE	9	Merkez
Ö2	Kadın	BÖTE	3	Merkez
Ö3	Erkek	BÖTE	3	Merkez
Ö4	Erkek	Görsel Sanatlar	9	Merkez
Ö5	Erkek	BÖTE	13	Merkez
Ö6	Kadın	BÖTE	8	Merkez
Ö7	Kadın	BÖTE	9	Merkez
Ö8	Erkek	BÖTE	2	Merkez
Ö9	Kadın	BÖTE	2	Merkez
Ö10	Erkek	BÖTE	11	Yeniçağ
Ö11	Erkek	BÖTE	4	Merkez
Ö12	Erkek	BÖTE	3	Merkez
Ö13	Kadın	BÖTE	12	Merkez
Ö14	Erkek	BÖTE	10	Merkez
Ö15	Erkek	Sanat Öğretmenliği	10	Merkez
Ö16	Erkek	Aile Ekonomisi ve Beslenme Öğretmenliği	9	Merkez
Ö17	Erkek	BÖTE	8	Seben
Ö18	Erkek	BÖTE	3	Mudurnu
Ö19	Kadın	BÖTE	4	Merkez
Ö20	Erkek	BÖTE	9	Mengen
Ö21	Erkek	BÖTE	8	Dörtdivan
Ö22	Kadın	BÖTE	4	Gerede

3.3. Veri Toplama Süreci

Araştırma verilerinin elde edilmesinde, nitel araştırmalarda yaygın olarak kullanılan yöntemlerden birisi olan görüşme yöntemi kullanılmıştır. Görüşme, en az iki kişi arasında sözel olarak gerçekleştirilen, insanların duygularını, deneyimlerini ve bakış açılarını ortaya çıkarmada kullanılan bir yöntem olarak tanımlanmaktadır (Bogdan ve Biklen, 1992; Büyüköztürk, Çakmak, Akgün, Karadeniz ve Demirel, 2012). Araştırmacının görüşme sırasında, araştırmanın amacı doğrultusunda sorduğu sorulara katılımcılar, konuya yönelik bakış açıları, bilgileri, düşünceleri, tutumları ve şikâyetleri doğrultusunda yanıt verirler (Glesne, 2015; Yıldırım ve Şimşek, 2013).

Görüşme yöntemi, araştırmacılar tarafından alanyazında farklı şekillerde sınıflandırılmıştır. Durdu (2016)’ya göre görüşmeler, yapısına ve amacına göre iki ana

başlık altında sınıflandırılabilir. Yapısına göre görüşmeler yapılandırılmış, yarı-yapılandırılmış ve yapılandırılmamış olmak üzere üç ana kategori altında toplanmıştır. Araştırmada öğretmenlerle gerçekleştirilen görüşmelerde yarı-yapılandırılmış görüşme tekniği kullanılmıştır. Bu teknikte araştırmacı, görüşme esnasında önceden planlamış olduğu soruları sorarken konu ile ilgili daha detaylı bilgi almak amacıyla katılımcının yanıtlarını açmasını ya da detaylı açıklamasını sağlayacak ek sorular sorabilir. Bu bağlamda öğretmenlerin bilgi işlemsel düşünme ile ilgili bilgileri ve düşüncelerini ortaya çıkarmak amacıyla kodlama eğitimine yönelik görüşleri alındığı yarı yapılandırılmış görüşme formu hazırlanmıştır (Ek 1). Görüşme formunda yer alan toplamda 10 soru, araştırmacılar tarafından araştırmanın amacı dikkate alınarak ve iki kaynaktan yararlanılarak oluşturulmuştur. Bunlar alanyazın taraması ile ulaşılan ve araştırmacıların kendileri tarafından oluşturulan sorulardır. Görüşme soruları hazırlanırken, soruların açık ve anlaşılır olmasına, öğretmenleri yönlendirecek tarzda olmamasına, ayrıntılara ve açıklamalara ihtiyaç duyulacak sorular için sonda sorular hazırlanmasına ve soruların farklı türlerde yazılmasına özen gösterilmiştir.

Araştırma verileri toplanmaya başlamadan önce Bolu Abant İzzet Baysal Üniversitesi'nden rastgele seçilmiş 8 Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümü öğretmen adayı ile pilot uygulama yapılmıştır. Araştırmaya katılmaya gönüllü olan öğretmen adayları ile görüşme yapılmıştır. Pilot uygulama sonrasında, görüşme soruları üzerinde gereken değişiklikler yapılmıştır. Ardından öğretmenlerle görüşmeye başlamadan önce Ankara'da bir ortaokulda görev yapan bilişim teknolojileri ve yazılım dersi öğretmeni ile görüşme yapılmıştır. Yapılan görüşme kaydı araştırmacılar ve bir alan uzmanı tarafından dinlendikten sonra iki soru üzerinde düzenleme yapılmış ve görüşme sorularına son şekli verilmiştir. Aşağıda görüşme formunda yer alan sorulara iki örnek verilmiştir:

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 řu anda verilen programlama / kodlama eđitimi yeterli midir?

Veri toplama sürecinde görüşmeler, arařtırmacı tarafından katılımcılarla önceden belirlenen gün ve saatte gerçekleştirilmiştir. Her bir katılımcı ile ortalama 15 dakika kadar yüz yüze gerçekleştirilen görüşmeler, katılımcıların izni alınarak ses kayıt cihazı ile kayıt altına alınmıştır. Bu sayede verilerin eksik veya yanlış olma ihtimali büyük ölçüde ortadan kalkmıştır. Ardından görüşmelerin yazılı dökümleri çıkarılarak deřifre edilmiştir.

3.4. Verilerin Analizi

Görüşme yöntemi ile elde edilen verilerin analizi için nitel içerik analizi kullanılmıştır. Nitel içerik analizi, tümevarımsal bir yöntemle, başlangıçta belirgin olmayan durumların ortaya çıkarılmasına ve verilerin derinlemesine çözümlenmesine olanak sağlamaktadır (Çetin, 2016; Yıldırım ve Şimşek, 2013). Nitel içerik analizinde asıl amaç elde edilen verileri yorumlayabilecek kavram ve bağlantılara erişebilmektir. Nitel içerik analizi sistematiktir ancak evrensel olarak kabul görmüş bir yapıya sahip olmadığından arařtırmacılar farklı adımları takip ederek veri analizini gerçekleştirebilirler (Çetin, 2016). Örneđin, Schreier (2012) 8 adımdan oluşan bir veri analizi önermiştir;

1. Arařtırma probleminin belirlenmesi
2. Analizi yapılacak verinin seçimi
3. Kodlama şemasının oluşturulması
4. Verinin bölümlere ayrılması
5. Kodlama şemasının denenmesi
6. Kodlama şemasının değerlendirilmesi ve gerekli deđişikliklerin yapılması
7. Kodlamanın yapılması
8. Bulguların yorumlanıp sunulması

Bu arařtırmada elde edilen verilerin analiz aşamaları ařađıdaki adımlardan oluşmaktadır:

1. Elde edilen verilerin analizine, görüşme kayıtlarının yazılı dökümleri çıkarılarak deşifre edilmesi ile başlanılmıştır. Ardından veri, bir kelime işlemci programında oluşturulan iki sütunlu tablonun sol kısmına yerleştirilmiştir.
2. Daha sonra araştırmacılardan biri sol taraftaki metni dikkatli bir şekilde okuyarak ve bahsedilen konu bütünlüklerini dikkate alarak bölümlere ayırmıştır. Sağ sütunda ise her bir bölüm numaralandırılmıştır. Numaralandırmanın amacı veriye daha sonra kolayca erişebilmektir. Schereier, veri analizi adımlarında verinin bölümlere ayrılmasını, kodlama şemasının oluşturulmasından sonra gerçekleştirilmektedir. Ancak veri miktarı az olduğundan ve kodlama şeması oluşturulmadan önce verinin bir bütün olarak araştırmacı tarafından okunup veriye hâkim olması istenildiğinden, kodlama şemasının oluşturulmasından önce bölümlere ayırma işlemi gerçekleştirilmiştir.
3. Kodlama şemasının oluşturulması aşamasında iki araştırmacı verileri birlikte soru soru değerlendirerek, örneğin görüşme formunda yer alan 4. soruya bütün öğretmenler tarafından verilen yanıtları satır satır okuyarak, hem veriden çıkan anlamlara göre kodlar üretmişlerdir hem de doğrudan verilerden kodlar oluşturmuşlardır. Araştırmada kodlama şemalarının nasıl oluşturulduğunu bir örnekle anlatmaya çalışalım. Görüşme formundaki 4. soruya öğretmenler tarafından verilen yanıtların kodlaması yapılmış ve kategoriler oluşturulmuştur. Ardından oluşturulan kategorilerde benzerlik gösteren kategoriler sınıflandırılarak bir üst kategori altında bir araya getirilmiştir.
4. Kodlama şemalarının oluşturulma aşamalarında araştırmacılar veriyi birlikte kodladıklarından dolayı, kodlayıcılar arası uyumsuzluk yaşandığı an aralarında tartışarak uyumsuzluk giderilmiştir. Böylelikle kategori tanımlamaları yapıldıktan sonra kodlayıcılar arası uyumu kontrol etme ihtiyacı doğmamıştır.

5. Bütün veriler kodlandıktan sonra oluşturulan kategori tablolarında kategoriler, alt kategoriler, bu kategorilerden bahseden öğretmenler ve öğretmenlerin bulunduğu kategoriden verisi içerisinde hangi bölümde bahsettikleri gösterilmiştir. Oluşturulan kategori tabloları ile veriye bir bütün olarak bakma ve farklı yönlerden veriyi inceleme şansı elde edilmiştir. Verilerin yorumlanma aşamasında kategori tabloları, veriler arasında neden sonuç ilişkisi kurmak, kategorileri yorumlamak, katılımcıların hangi kategorilerden ne sıklıkta bahsettiklerini görmek gibi çeşitli yönlerden verinin anlamlandırılmasını sağlamaya yardımcı olmuştur.



IV. BÖLÜM

4. Bulgular

Bu bölümde, araştırma verilerinin analizleri sonucunda elde edilen bulgular yer almaktadır. Araştırma sürecinde elde edilen bulgular 9 kategoride değerlendirilmiştir. Bu kategoriler; kodlama eğitimi tanımlamaları, kodlamanın amacı, herkes için kodlama, kodlama eğitimine başlangıç yaşı, kodlama için temel kavram ve beceriler, öğretmenlerin bilgisayarsız bilgisayar bilimi (B³) hakkındaki düşünceleri, programlama için yazılım/donanım gereksinimi, öğretmen eksikliklerinin nedeni, öğretmen eksiklikleri için yapılması gerekenler şeklinde sıralanmıştır. Tablo 4.1’de kategoriler ve alt kategoriler, her bir kategoride bulunan katılımcı sayısı ile birlikte sunulmuştur. Araştırmaya toplamda 22 öğretmen katılmıştır fakat bazı durumlarda bir öğretmen birden fazla kategori altında gruplandırılmıştır. Bu durumun bulunduğu kategorilerde toplam yanıt sayısı 22’yi geçmiştir.

Tablo 4.1. Araştırma bulguları sonucunda elde edilen kategoriler

Kategoriler ve Alt Kategoriler	Katılımcı Sayısı	Kategoriler ve Alt Kategoriler	Katılımcı Sayısı
Kodlama Eğitimi Tanımlamaları		Kodlama İçin Temel Kavram ve Beceriler	
Düşünme becerileri	14	Programlamanın temel kavramları	2
Yazılım geliştiren bireyler yetiştirmek	6	Üretken olma	5
Duyuş	2	Algoritmik düşünme becerisi	3
Diğer	4	Analitik düşünme becerisi	7
Kodlamanın Amacı		Sistematik düşünme becerisi	2
Düşünme becerileri	13	Farklı düşünme becerisi	5
Üretici olma	6	Problem çözme becerisi	9
Öğretim kaynaklı	7	Diğer	7
İş yaşamı kaynaklı	14	Öğretmenlerin Bilgisayarsız Bilgisayar Bilimi (B³) Hakkındaki Düşünceleri	
Diğer	9	B ³ ’ten bahsedener	3

Tablo 4.2. (Devamı) Araştırma bulguları sonucunda elde edilen kategoriler

Kategoriler ve Alt Kategoriler	Katılımcı Sayısı	Kategoriler ve Alt Kategoriler	Katılımcı Sayısı
Kodlama Eğitimi Tanımlamaları		Kodlama İçin Temel Kavram ve Beceriler	
Herkes İçin Kodlama		B ³ 'ten bahsetmeyenler	15
Zorunlu olmalı	16	İnformal B ³	4
Seçmeli olmalı	3	Olumsuz bakanlar	15
Zorunlu ve seçmeli olmalı	3	Olumlu bakanlar	4
Kodlama Eğitimine Başlangıç Yaşı		Öğretmen Eksikliklerinin Nedeni	
Anasınıfı	8	Alan bilgisi eksikliği	12
İlkokul	12	Üniversitedeki pedagojik alan bilgisi eksikliği	7
Ortaokul	2	Diğer	3
Programlama İçin Yazılım/Donanım Gereksinimi		Öğretmen Eksiklikleri İçin Yapılması Gerekenler	
Görsel programlama araçları	22	Kişisel gelişim	3
Mobil uygulamalar	2	Üniversitelerdeki eksikliklerin giderilmesi	5
3B yazılımlar	5	Eğitimler verilmeli	16
Robotlar	9	Diğer	1
Metin tabanlı ortamlar	3		
B ³ etkinlikleri	1		

Aşağıda araştırma sürecinde elde edilen bulgular araştırmanın temaları doğrultusunda sunulmuştur.

4.1. Kodlama Eğitimi Tanımlamaları

Öğretmenlerin “Sizce kodlama eğitimi nedir? Size göre kodlama eğitimi ne ifade ediyor?” sorusuna verdikleri yanıtlardan *düşünme becerileri*, *yazılım geliştiren bireyler yetiştirmek*, *duyuş* ve *diğer* kategorileri ortaya çıkmıştır. Öğretmenlerin bu başlıkta en çok altında toplandığı kategori düşünme becerileri kategorisidir.

Kodlama eğitimini genel anlamda öğrencilerin bilişsel becerilerini geliştirmesi bağlamında düşününen 14 öğretmenin görüşleri ile *düşünme becerileri* kategorisi meydana getirilmiştir. Bu konuda öğretmenler; *çözüm şemaları*, *algoritmik düşünme*, *yaratıcılık*, *mantıksal ve/veya matematiksel düşünme*, *düşünme becerisini geliştirme* ve *pratik problem çözüme*'den bahsetmişlerdir. Çözüm şemaları alt kategorisinde 5 öğretmen görüşü bulunmaktadır. Bu öğretmenler, bir problem durumunda problemi kısa yoldan

basit anlaşılır bir şekilde nasıl çözebileceğini sağlamak, probleme yüzeysel bakmak yerine çözüm üretebilmek ve bunun için planlı çalışmayı sağlamak, olaylara farklı bakış açısı ile bakabilme becerisi sayesinde problemlere farklı çözüm yolları geliştirebilmeleri bunlar için farklı düşünme yöntemleri ve geniş kapsamlı düşünme becerilerinin öğretildiğinden bahsetmişlerdir. Aşağıdaki alıntı çözüm şemaları alt kategorisine örnek olarak verilebilir.

Ö22: *“Kodlama eğitimi öncelikle çocukların daha geniş kapsamlı düşünme ve probleme problem çözmeye dayalı düşüncelerini, yaklaşım olarak da problemleri daha iyi çözmelerini sağlıyor, daha analitik düşüncelerini sağlıyor...”*

4 öğretmen algoritmik düşünme konusunda genel olarak, öğrencilere bir işin nasıl yapılması gerektiği, işlem basamakları ve algoritmik düşünmenin öğretilmesi gerektiğinden bahsetmişlerdir. Aşağıdaki alıntıda Ö21 öğrencilere robotik araçlarla, programlarla öğrencilere kodlama öğretmeden önce algoritmik düşünmeyi öğretmek gerektiğinden bahsetmiştir.

Ö21: *“...Japonlar, Amerikanlar uzay aracını roketi geri geri indirme projeleri yapıyorlar. Esnek hareket edebilen robotik kollar yapıyorlar ve esnek hareket edebilen köpek yaptılar geçen. Bunun videolarını paylaşıyorlar. Bizse hazır robotları bir iki tane Scratch, Small basic kodlarıyla bir şeyler hareket ettirdik oo programlama yaptık. Programlama bu değil. Bizim ortaokul çağındaki çocuğa ilk yapmamız gereken şey algoritmik düşünmeyi sağlamaktır. Kod yazmayı öğretmek değil. Hazır Arduinio setler hazır robotik setlere bir şeyler bağlayarak usb cihazını bağlayarak oradan iki hazır kodu Scracth'ı yükleyerek yani robotik ve kodlama olmaz. Kodlama kesinlikle bu değildir kodlamadan önce o çocuğun bir kere algoritmik düşünmesi lazım...”*

3 öğretmen kodlama eğitimini öğrencilerin yaratıcılıklarını geliştirmesi açısından değerlendirmiştir. Aşağıdaki alıntı yaratıcılık alt kategorisine örnek olarak verilebilir.

Ö19: “...Öğrencilerin yaratıcılığını geliştiriyor. Özellikle böyle ilgi çeken programlar olduğu zaman başarılı bir şekilde yürüttükleri bir çalışma oluyor... Mesela benim bi internette okumuştum Minecraft ile programlama dilleri öğretimi ile ilgili bir çalışma vardı. Benim mesela çok derslere ilgisiz bir talebem vardı ama minecraft'ta inanılmaz tasarımlar yapıyor bina tasarımları hatta bana getirdi. Öğretmenim dedi çok böyle hali hazırda dersler ile ilgili olmayan bir öğrenci ama bilgisayarın başında öyle değişik şeyler tasarlamış ki bu mesela onun yaratıcı olduğunu gösterir. Mesela o şekilde programlarla programlama dilleri öğretmeye çalışılsa bence güzel çalışmalar ortaya çıkar diye düşünüyorum benim görüşüm...”

Mantıksal ve/veya matematiksel düşünme alt kategorisinden bahseden 3 öğretmen, kodlama eğitiminin öğrencilerde mantıksal düşünmeyi sağladığı bir diğer yandan hem mantıksal hem de matematiksel düşünmeyi sağladığı yönünde görüş bildirmişlerdir. Aşağıdaki alıntı mantıksal ve/veya matematiksel düşünme alt kategorisine örnek olarak verilebilir.

Ö8: “...mantıksal ve aritmetik düşünmeyi sağlayıcı ve tamamlayıcı bir şekilde gösterilmesi gerekiyor diye düşünüyorum...”

Kodlama eğitimi ile öğrencilere nasıl düşünceleri gerektiğini öğretip, onların düşünce yapılarını geliştirerek verileni aynı şekilde yapmak yerine, zenginleştirerek yapabilmelerine olanak sağlamak şeklinde düşüncelerini dile getiren 3 öğretmenin görüşleri, düşünme becerisini geliştirme alt kategorisinde toplanmıştır. Aşağıdaki alıntı düşünme becerisini geliştirme alt kategorisine örnek olarak verilebilir.

Ö10: “Kodlama eğitimi bence bi defa günümüzde olmazsa olmaz derslerden birisidir. Hele ki bizim dersimiz olacaksa, bilişim teknolojileri dersi olacaksa bunun içinde mutlaka yer almalıdır ve fazlası ile yer almalıdır. Çünkü çocuklarda hem düşünce yapısını geliştirme açısından hem de farklı bir bakış açısı olaylara sunması açısından bence çok faydalıdır kodlama eğitimi. ...Yani öğrencilerden farklı şeyler çıkabiliyor. Mesela eskiden anlattığımız konularda bişey istiyoruz öğrenci onu yapıyordu basmakalıp mesela bi tablo mesela wordde tablo excelde işte ne

oluşturuyorduk ders notlarını oluşturuyordu sadece onu yapıyordu ama bunda öğrenciler farklı şeyleri yapabiliyor. Kendi özel yeteneklerine göre farklı şeyler geliştirebiliyor mesela tasarımları değiştirebiliyor hayal güçlerini kullanarak olaylara daha da derinlik kazandırabiliyor o açıdan faydalı.”

Pratik problem çözme alt kategorisinden bahseden 2 öğretmen kodlama eğitimi ile pratik düşünme becerisi kazandıklarını böylelikle de problem çözerken problemi en kısa yoldan çözme veya farklı çözüm yolları geliştirebilme yetilerinden bahsetmişlerdir. Aşağıda pratik problem çözme alt kategorisinin oluşturulmasında kullanılan öğretmen yanıtı sunulmuştur.

Ö14: *“Kodlama eğitimi öğrencilerin önce düşünce yapısını değiştirmekle alakalı, çok faydalı bir eğitim. Yani öğrencide bir kere algoritma zekâsı geliyor, farklı bakış açıları geliyor, bir problem anında farklı çözüm yolları geliştirebiliyor. Bununla alakalı yani eğer sadece bilgisayar dışına çıkacaksa öğrencide günlük yaşamda çok fazla artıları var. Öğrencinin işte dediğim gibi bakış açıları değişiyor, çözüm yolları ile alakalı gideceği yollar farklılaşıyor, daha pratik bir düşünce yapısına sahip olabiliyor...”*

Bu başlıktaki ikinci kategori olan *Yazılım geliştiren bireyler yetiştirmek*'ten bahseden 6 öğretmenden 4'ü, genel anlamda bilişim çağı için bireyler yetiştirmeyi vurgulayan açıklamalarda bulunmuşlardır. Bu öğretmenler, günümüzde birçok alanda teknolojinin önderlik ettiği, ülkemizin de bu önderlik yarışında güzel bir yer edinebilmesi adına ülke olarak kullandığımız uygulamaları, programları dışarıdan almak yerine bunların üretimini yapacak bireyler yetiştirmemiz gerektiğini dile getirmişlerdir. Diğer öğretmenler ise dünyada artık her şey yazılım ile bağlantılı olmaya başladığından dolayı kodlama eğitimi ile bu alanda yetenekli öğrencilerin fark edilip yetiştirilmesi gerektiğini dile getirmişlerdir. Aşağıda yazılım geliştiren bireyler yetiştirmek kategorisinin oluşturulmasında kullanılan bir öğretmenin yanıtı sunulmuştur.

Ö7: *“...kodlama yöntemi aslında gerçekten iii işe yarar bir şey hani eee çünkü sonuçta biz bir uygulama bile yapamıyoruz sonuçta hep böyle dışarıdan alıyoruz*

genellikle bilgisayar programlarını falan. Keşke biz kendimiz yapsak daha iyi olur en azından bence şimdiki çocuklar alt yapısı oluşuyor bence iyi oldu yani bilgisayar derslerinde kodlama dersi olması çok iyi oldu en azından bir başlangıç yapmış olduk diyebilirim.”

Bu başlıktaki son kategori olan *Duyuş* başlığını oluşturacak nitelikte görüş bildiren 2 öğretmenden biri kodlama eğitimini öğrencilere oyun yaparak öğrenme imkânı sağladığı için severek ve mutlu olarak öğrenme gerçekleştirmeleri sebebiyle güdüleyici olarak görmektedir. Diğer öğretmen ise öğrencilerin kendine özgü nitelik taşıyan şeyler yapabilmelerine olanak sağladığından, öğrencilerde başarıya duygusu sağlamak şeklinde nitelendirmiştir. Aşağıdaki alıntı *duyuş* kategorisine örnek olarak verilebilir.

Ö6: *“Bir de çok seviyorlar oyunla öğrenmeyi hemen zaten hep oyun oynuyorsunuz. Kendi oyununuz kendiniz yapmanız sizde nasıl bir his uyandırır dediğimizde direk çocukların gözleri parlıyor zaten biz mi oyun yapcaz ama hemen söyledikleri GTA yapacak mıyız? Bu şekilde onları ileri düzeyde yapabilirsiniz ama mantığını aldığımız zaman eğer ilerde bu işle uğraşmak isterseniz yapabilirsiniz diyorum. Çok seviniyorlar...”*

Diğer kategorisinde ise kodlama eğitimini tanımlama ile ilgili olmayan ve az frekanslı öğretmen görüşleri yer almaktadır. Toplamda 4 öğretmenin görüşlerinin yer aldığı bu kategoride, kodlama eğitimi hakkında 3 öğretmen kodlama eğitimini etkileyen faktörlere değinmiştir. Bu faktörler arasında müfredat, öğrencilerin bilgisayar kullanmayı bildikleri algısı, bilişim teknolojileri ve yazılım dersi öğretmenlerine bakış açısı, eğitim sisteminin farklılıkları göz ardı etmesi yer almaktadır. Öğretmenlerden biri bilgisayarca düşünmeden bahsetmiş insan beyninin çalışma sisteminden örnek vererek açıklamaya çalışmıştır. Bir diğer öğretmen ise kodlama eğitimini bireylerin problem çözme stratejisi kazanmalarına yardımcı olması açısından değerlendirmiştir. Aşağıda diğer kategorisinin oluşturulmasında kullanılan öğretmen yanıtlarından bir örnek sunulmuştur.

Ö21: “*Milli eğitimlerde maalesef bilgisayar öğretmenlerine teknik servis olarak bakıldığı için hani programlama çok... hani havada kalacak bir şey bizim ülkemiz için. Nedeni şudur: Ben kendi adıma konuşacak olursam, benim bir yıldaki ders yükümlülüğüm 30-35 haftadan şey yaparsak 8 ile çarptığımız zaman yaklaşık 300-400 saattir. Ve ben bunun yarısına bile derse giremeyen biriyim. Çünkü ilçeyle ilgili birçok teknik problem oluyor ve teknoloji çağındayız ve bunların giderilmesi için bilgisayar öğretmenini illa yardım olmak zorunda. Ben bunların çoğuyla bunlarla ilgilenirken derslere yetişemezken programlamaya çok vakit ayırmak... kalmıyor bile vakit. Artı Bolu’ daki veya il merkezlerindeki öğrenci profili ve çoğunun elinde bilgisayar var ailelerin birçoğu okumuş ve çocuklar belli bir bilgisayar hazırbulunuşluluğu var.”*

4.2. Kodlamanın Amacı

Öğretmenlerin “*Neden kodlama eğitimi? Kodlama eğitimi öğrencilere neden verilmelidir? Sonda: Sizce kodlama eğitiminin temel amacı ne olmalıdır?*” sorusuna verdikleri yanıtlardan *iş yaşamı kaynaklı, düşünme becerileri, üretici olma, öğretim kaynaklı* ve *diğer* kategorileri ortaya çıkmıştır. Öğretmenler kodlamanın amacını en çok iş yaşamından kaynaklı sebepler bağlamında düşünmüşlerdir.

Kodlamanın amacı hakkında *iş yaşamı kaynaklı* kategorisini oluşturacak nitelikte görüşlerini bildiren 14 öğretmenin ifadeleri *istihdam, yazılımcı yetiştirmek ve bilişim çağı için bireyler yetiştirmek* alt kategorilerinde bir araya getirilmiştir. Kodlamanın amacını sağladığı iş olanakları sayesinde kaliteli yaşam sunması yönünden düşünülen 2 öğretmenin görüşleri istihdam alt kategorisinde toplanmıştır. Aşağıdaki alıntı istihdam alt kategorisine örnek olarak verilebilir.

Ö3: “*Hem de nedir hepimizin amacı iyi bir iş sahibi olmak veya nedir ilerde maddi durumunun yüksek olması ya da kaliteli bir yaşam. Hepimizin amaçları budur. Çocukların kazanmasını istediğimiz amaç ta budur. O yüzden de kodlamada bence iş var mı var. Ama biz bunları ne yazık ki ülke olarak geliştiremediğimizden ülkemizde yok. Ama Amerika’da ki bir mühendis çok rahat şartlarda mesela Google’in ofislerini*

Facebook'un ofislerini görüyoruz hatta öğrencilere gösteriyorum bayılıyorlar yani bu şartlarda çalışabilmek için bence kodlama eğitimini bilmek lazım.”

Yazılım geliştiren bireyler yetiştirip teknolojik araç üretimi sağlayarak yerli yazılımlar ile ülkenin gelişmesine ve kalkınmasına katkı sağlamak gibi ifadelerden bahseden 6 öğretmenin görüşleri yazılımcı yetiştirmek alt kategorisinde toplanmıştır. Aşağıdaki alıntı yazılımcı yetiştirmek alt kategorisine örnek olarak verilebilir.

Ö20: *“Kodlamanın önünün açık olduğunu düşünüyorum. Dediğim gibi bütün teknolojiler artık yazılım üzerinden programcılık üzerinden geliştiği için insanların yeni teknolojik araçlar üretebilmesini sağlar. Bunu zaten görüyoruz şu anda ülkemize de artık tamamen yerli teknolojik araçlara dönüş var. Bunu bir fırsat olarak görmeliyiz. Bunu öğrencilerimize şimdiki okullarımızda okuyan öğrencilerimize ilkökul, ortaokul, lise onlara bunu aktarabilirsek onlarla ilerde bizim ülkemiz için daha iyi ve yerli ve milli yazılımları, teknolojik araçları geliştirebileceklerdir.”*

Bulduğumuz yüzyılda var olmak adına, üretici bir toplum olmak için yazılım bilen insanlara ihtiyaç duyulduğu hatta sadece yazılım ile sınırlı kalınmaması gerektiği, bireylerin buldukları sektörde karşılaştıkları yazılımsal donanımsal sorunların sebebinin ne olduğunu bilecek veya çözüm bulabilecek seviyede olmaları gerektiğinden bahseden 6 öğretmenin ifadeleri bilişim çağı için bireyler yetiştirmek alt kategorisinde toplanmıştır. Aşağıdaki alıntı bilişim çağı için bireyler yetiştirmek alt kategorisine örnek olarak verilebilir.

Ö5: *“Yani üretici bir toplum yetiştirme adına zaten gerekli bir durum yani her şey bilişim teknolojileri üzerine dönmekte artık çiftçilikten tutunda mimarlık mühendisliğe kadar bu konuda kendi kullanacakları araçları geliştirmek veya kendi problemlerine yazılımsal anlamda çözebilmek hatta donanımsal anlamda da çözebilmek amaçlı insanlarda temelinin olması gerekiyor. Veya neler yapabileceğini bilmesi gerekiyor aslında veya neyi nereden bulabileceğini bilmesi gerekiyor...”*

Kodlamanın amacı hakkında *düşünme becerileri* başlığını oluşturacak nitelikte görüş bildiren 13 öğretmen kodlamanın amacını, kodlama eğitimi kategorisinde olduğu gibi öğrencilerin bilişsel becerilerini geliştirmesi açısından düşünmüşlerdir. Bu konuda öğretmenlerin görüşleri *çözüm şemaları*, *günlük yaşam problemleri*, *analitik düşünmeyi geliştirmek* ve *yaratıcılığı geliştirme* alt kategorilerinde toplanmıştır. 2 öğretmen, öğrencilerin bir problem ile karşılaştığında farklı çözüm yolları üretebilmeleri veya problem çözümü için farklı alternatiflerin de farkında olması sonucunda çözüm aşamasında en uygun çözüm yolunu seçebilme yetisine sahip olmalarından bahsetmiştir. Bu öğretmenlerin görüşleri *çözüm şemaları* alt kategorisinde toplanmıştır. Aşağıda *çözüm şemaları* alt kategorisinin oluşturulmasında kullanılan öğretmen yanıtları sunulmuştur.

Ö5: “...günlük hayatta karşılaştığı problemlere ilişkin çözüm yolları üretmek, bir çözüm değil birden fazla çözüm yolu seçebilmek, bulabilmek ve bunlardan hangisini kullanabileceğini seçebilmek adına öğrenciye kazanımlar sağlayacaktır mantıklı düşünme adına...”

Kodlamanın amacına yönelik düşünceleri *günlük yaşam problemleri* alt kategorisinde bir araya getirilen 4 öğretmen, kodlama eğitimi ile öğrencilerin algoritmik düşünmesi, problemlere algoritmik bakış açısı ile yaklaşmaları ve bunu *günlük yaşam problemlerinde* de kullanabilmeleri, kodlamanın çok yönlü düşünme sağladığı bu sayede de öğrencilerin *günlük yaşamlarındaki işlerini* algoritmik bir şekilde aşamalı olarak düzenli hale getirebilecekleri ve *günlük yaşamlarında* daha planlı ve programlı çalışma adına kazanımlar sağladığından bahsetmişlerdir. Aşağıdaki alıntı *günlük yaşam problemleri* alt kategorisine örnek olarak verilebilir.

Ö14: “Burada dediğim gibi öğrencileri tamamen bir yazılımcı olarak değil, öğrenci kodlama eğitimini aldıktan sonra *günlük hayatta da kullansın istiyorum*. Yani şimdi kodlama eğitimine başlamadan önce bir kere algoritma yapısını veriyorsun bir kere çocuğa yani bir problem var bu problemi nasıl çözebiliriz? Diye bir işlem basamakları istiyorsunuz. Bunu *günlük hayata çevirelim*. *Günlük hayatta çocuk bir problemle karşılaştığında bunu entegre edebilir oraya*. Dolayısıyla öğrencide bu şeyi,

uu bu hissiyatı uyandırmanın peşindeyiz biz. Bazıları tabii içinden bunu ilerletip yazılımcı olacak, bazıları bilgisayar mühendisi olacak. Her alanda kullanılmalı öğrenci bunu. Bu öğrencinin zekâsıyla, beyin yapısıyla alakalı bir şey ben öyle düşünüyorum kodlama eğitimini.”

Kodlamanın amacına yönelik düşünceleri analitik düşünmeyi geliştirmek alt kategorisinde bir araya getirilen 4 öğretmen ve yaratıcılığı geliştirme alt kategorisinde bir araya getirilen 3 öğretmen bulunmaktadır. Bu alt kategorilerde yer alan öğretmenler analitik düşünme ve yaratıcılık kavramlarından bahsederken kısa cümleler kurmuşlardır ve öğretmenlerin analitik düşünme ve yaratıcılıktan tam olarak ne kast ettikleri anlaşılmamıştır. Aşağıda analitik düşünmeyi geliştirmek alt kategorisi için örnek alıntı sunulmuştur.

Ö3: *“Kodlama neden verilmelidir çocukların analitik düşünmesini geliştirmek amacı ile.”*

Yaratıcılık alt kategorisinde de benzer bir durum söz konusudur. Öğretmenlerin cevapları kısadır. Aşağıdaki alıntı yaratıcılık alt kategorisine örnek olarak verilebilir.

Ö19: *“Yani yaratıcılıklarını geliştireceğini düşünüyorum ben.”*

Kodlamanın amacına yönelik olarak *öğretim kaynaklı* kategorisini oluşturacak nitelikte görüş bildiren 7 öğretmen bulunmaktadır. Bu öğretmenlerin görüşleri *otantik öğrenme* ve *diğer derslere entegrasyon* olarak ikiye ayrılmaktadır. Otantik öğrenmeden bahseden 5 öğretmen, öğrencilerin öğrendiği bilgileri önceki bilgi birikimlerine göre şekillendirerek günlük hayatta, günlük hayat problemlerini çözerken kullanabilmeleri yönünde ifadelerde bulunmuşlardır. Aşağıdaki alıntı otantik öğrenme alt kategorisine örnek olarak verilebilir.

Ö3: *“ilk başta bir sevdirmek lazım u nerde kullanacağına bağlı çocuklarda şuradan başlamak lazım ileriki hayatında bunu nerde kullanacak. En başta sordukları soru bu oluyor. Hani biz bu kodlamayı öğrendikte nerde kullancaz. Bi öğretmensiniz şu*

anda bir Fatih projesi var burda mesela nedir kodlamayı bildiğin anda kendi dersinle ilgili bir tane proje geliştirebilir onu sunabilirsin veya bir tane animasyon oluşturup dersi zevkli hale getirebilirsiniz. E bir doktor oldun doktorlar şuanda çoğu gelişimi onlar kendileri yapıyor mesela oralarda da mesela yine bir bilgisayar kullanılması gerekiyor. Bunları anlatarak kodlama sürdürülebilir.”

Kodlama eğitiminin diğer derslere entegrasyonundan bahseden 2 öğretmen, sayısal dersler ile kodlama dersinin karşılıklı etkileşim içinde olduğu hatta sağladığı pratik düşünme sayesinde sözel derslere de olumlu etkisinin olduğunu dile getirmişlerdir. Ayrıca kodlama dersinde edinilen bilgilerle diğer derslerin işlenişine yardımcı olacak bir yazılım geliştirilebileceği açıklamalarında bulunmuşlardır. Aşağıdaki alıntı diğer derslere entegrasyon alt kategorisine örnek olarak verilebilir.

Ö2: *“...çocukların sadece bilgisayar dersi ve kodlama olarak değil mesela matematik derslerine de katkısı olduğunu düşünüyorum çünkü biraz sayısal ağırlıklı olduğunu düşünüyorum. Ve matematiği iyi olan öğrencilerin bu derste daha başarılı olduğunu görüyorum ve bu derste tam tersi matematiğe de etki edeceğini düşünüyorum. Yani ne kadar analitik düşünürse diğer matematik gibi sayısal derslerde de başarılı olacağını düşünüyorum. Sözelde de aynı şekilde çünkü pratik fikir geliştirmesi gerekiyor. Hani aynı şekilde sözelde de bir anda aklına değişik fikirler gelmesi gerekecek. Dolayısıyla bu dersin diğer dersleri destekleyeceğini düşünüyorum...”*

Kodlamanın amacını üretici olmayı sağlamak açısından düşünen 6 öğretmenin görüşleri *üretici olma* kategorisi altında bir araya getirilmiştir. Öğretmenlerin görüşlerine bakıldığında genel anlamda, öğrencilerin çok hazırcı oldukları bu nedenle onları küçük yaşlarda üretici bireyler haline getirerek özgüven kazandırmaktan, sadece var olanı kullanmak yerine küçük yaşlarda elde ettikleri başarıma duygusu sayesinde yazılım geliştiren insanların kendileri gibi insanlar olduklarını fark etmelerini sağlamaktan ve üretici bir toplum haline gelerek ülke ekonomisine katkı sağlamaktan bahsettikleri görülmektedir. Aşağıdaki alıntı *üretici olma* kategorisine örnek olarak verilebilir.

Ö13: “Düşünce şekilleri derken yani dediğimiz gibi hani biz bugüne kadar hep böyle hazır programları kullandık. Birileri üretmiş yapmış oyunlar aynı şekilde biz var olanı tükettik artık üretime geçme zamanımız. Bizlerin işeyler üretmesi lazım. Çocukların işeyler üretip üretebildiğini fark etmesi lazım ki sonrası gelsin. Hep zaten bakıyoruz dünyanın en zengin şirketleri kimler yani bilgisayar sektörünün içinde olan şirketler, Bill Gates yani bunların hepsi bakıyoruz niye hep çünkü onlar programlamayı artık oturtmuşlar. Bunlar kazandı bunlar gidiyo. Ders anlatırken de diyorum ki hiç araştırıyor musunuz diyorum en zengin şirketler kimler ne yapıyor bunlar kim hayatlarından örnek verin diyorum bakıyorlar gerçekten diyorum ki sizin de öğrenmeniz lazım sizin de işeyler üretmeniz lazım, tüketmeyi bırakalım artık diyorum ya. Tüketme aşamasını bi geçelim diyorum...”

Diğer kategorisinde konu ile ilgisi olmayan ve az frekanslı öğretmen görüşleri yer almaktadır. Aşağıdaki alıntı diğer kategorisine örnek olarak verilebilir.

Ö11: “5’te zaten ne olduğunun farkında değil çocuk. 6. Sınıfa geçiyor işeylere meraklanmaya başlıyorlar tam hevesleniyorlar ama 6. Sınıftan sonra ders yok. Ders olmadıktan sonra zaten işey yapamıyosun yani seçmeli aldırımıyosun başka dersler var. Hafta sonu kursları her yerde yapılmıyor... belli bir işte her öğretmenin belli bir müfredata göre hareket etmesi gerekiyor yoksa bi dağınıklık olmadığı sürece bence başarılı olabilir, belli bir kaynaktan gidildiği sürece...”

4.3. Herkes İçin Kodlama

Öğretmenlerin “Kodlama eğitiminin bütün öğrencilere zorunlu olarak verilmesi konusunda ne düşünüyorsunuz?” sorusuna verdikleri yanıtlar sonucunda zorunlu olmalı, seçmeli olmalı, zorunlu ve seçmeli olmalı kategorileri ortaya çıkmıştır. Bu başlık altında öğretmenler tarafından en çok kodlama eğitiminin zorunlu olması gerektiği çeşitli sebepler ile vurgulanmıştır.

Kodlama eğitiminin bütün öğrencilere zorunlu olarak verilmesi konusunda görüş belirten 16 öğretmen *zorunlu olmalı* kategorisinde verdikleri yanıtlar ile *zihinsel becerileri geliştirdiği için, hazır kullanmak değil üretmek, teknoloji çağının gerekliliği, ekonomik sebepler, diğer ve neden belirtilmemiş* alt kategorilerini oluşturmuşlardır. Öğretmenlerden 3'ü kodlama eğitiminin yaratıcı düşünme, sistemli düşünme ve problem çözme gibi zihinsel becerileri geliştirdiğini düşünerek zorunlu olması gerektiğini ifade etmişlerdir. Bu öğretmenlerin görüşleri zihinsel becerileri geliştirdiği için alt kategorisi altında bir araya getirilmiştir. Aşağıdaki alıntıda Ö1 kodlama eğitiminin yaratıcı düşünme ve problem çözme gibi becerileri geliştirdiği için zorunlu olarak verilmesi gerektiğini dile getirmiştir.

Ö1: *“Evet zorunlu yani senle böyle konuştuktan sonra u nasıl deyim düşünce ufkumuz biraz genişlettikten sonra özellikle de yaratıcı düşünme ve problem çözme yeteneklerini geliştirme açısından kodlama dersi zorunlu olmalı.”*

Kodlama eğitiminin zorunlu olarak verilmesi konusunda 5 öğretmen, bireylerin bilgisayarı sadece oyun aracı ve hazır paket programlardan ibaret olarak görmemeleri, var olanı kullanmak yerine üretime geçmeleri ve ülke olarak yerli yazılımlar geliştirerek üretici bir toplum haline gelmemiz gerektiğinden bahsederek, hazır kullanmak değil üretmek için alt kategorisinde bir araya gelmişlerdir. Aşağıdaki alıntıda Ö3 üretici bir toplum olmak, yerli yazılım geliştirmek adına kodlama eğitiminin zorunlu olması gerektiğini açıklamıştır.

Ö3: *“...üretmek için. Bizim en büyük sıkıntımız şu anda kendi milletimiz devletimiz olarak sıkıntımız şu, tamam teknolojiyi çok güzel kullanıyoruz bazen de şunu zannediyoruz yani bir Facebook'u kullanmak teknolojiyi kullandık zannediyoruz. Ama gelgelelim arka planda ürettiğimiz ne var? Vestel diyor ki Türkiye'nin diyor yerli telefonunu ürettim. Ama chip'ini üretebildin mi bence üretmedin %80'e %20'lik kısmı var esas para eden kısmı da orası...”*

Teknoloji çağının bir gerekliliği olarak kodlama eğitiminin zorunlu olarak verilmesi gerektiğini dile getiren 4 öğretmen genel olarak alışveriş siteleri gibi günlük

hayatın her alanında teknoloji yer aldığından dolayı kodlamanın içinde olmamız gerektiğinden ve bulunduğumuz çağa ayak uydurmak adına kodlama eğitiminin çocukların düşünme becerilerini geliştirmelerine katkı sağladığından bahsetmişlerdir. Aşağıdaki alıntıda Ö8 teknoloji yaşamımızın her alanında olduğu için kodlamanın içinde olmalıyız diyerek kodlama eğitiminin zorunlu olması gerektiğini ifade etmiştir.

Ö8: “Yani 2000’lerden sonra Milenyum dönemindeyiz ve teknoloji çok hızlı bir şekilde geliyor ve hayatımızın her yerinde var şuan. İlk başta bilgisayar masaüstleriydi; laptoplar, minibookla, cep telefonu, android televizyonlar, android cep telefonları işte. Hayatımızın her yerinde olduğu için kodlamanın da içinde olmamız gerektiği...”

Kodlama eğitiminin zorunlu olarak verilmesi gerektiğini ekonomik sebeplerden dolayı düşünen 3 öğretmen yerli yazılım geliştiremediğimiz için ülke olarak yazılıma çok para harcadığımızı bu sebeple yazılım üreterek ülke kalkınmasına katkı sağlamamız gerektiğini ifade eden açıklamalarda bulunmuşlardır. Aşağıdaki alıntıda Ö20 yazılımın günümüzde önemli bir yere sahip olduğunu ve ciddi anlamda kazanç sağladığını ifade ederek kodlama eğitiminin zorunlu olarak verilmesi gerektiği yönünde düşüncelerini dile getirmiştir.

Ö20: “Dediğim gibi artık günümüz dünyası yazılımlarla ilerliyor. Bir oyunun bir uygulamanın maliyetini veya hani ücretini düşünürsek çok fazla yeri geliyor bazı ülkelerin gayri safi milli hasılasından bile daha fazla olduğunu söyleyebiliriz. Bundan dolayı zorunlu olmasını düşünüyorum.”

Diğer alt kategorisinde az frekanslı öğretmen görüşleri yer almaktadır. Öğretmenlerden bir tanesi kodlama eğitimini temel bir beceri olarak görmektedir bu yüzden de herkes tarafından bilinmesi gerektiğini düşünmektedir. Başka bir öğretmen ise çocukların varolan teknolojileri kullanabiliyor olmasının insanlarda yanlış bir algıya yol açtığı, kodlama eğitimini teknoloji kullanımı ile eşdeğer görmelerinden dolayı dersin seçmeli olması durumunda tercih edilmeyeceğini dile getirmiştir. Bir diğer öğretmen ise birçok Avrupa ve Asya ülkesinde olduğu gibi bizim ülkemizde de zorunlu

olarak verilmesi gerektiğini ifade etmiştir. Kodlama eğitiminin zorunlu olarak verilmesi gerektiğini düşünen 2 öğretmen ise bu düşüncelerine herhangi bir neden belirtmemiştir. Aşağıda Ö22'nin kodlama eğitiminin neden zorunlu olması gerektiğini açıklayan ifadeleri yer almaktadır.

Ö22: *“Zorunlu olmadığına eğer seçmeli ders olduğu zaman genelde bilişim teknolojilerine hep böyle arkaplanda kalıyor, işte hani çocuklar nasıl olsa biliyor şeklinde bakılıyor. İşte teknolojinin içine doğdu Facebook'a girebiliyor, Facebook'a girebiliyorsa bu çocuk süper bilgisayar kullanabiliyor ya da işte akıllı telefonu kullanabiliyor aslında şöyle birşey var Apple şirketi bu Iphone telefonları üretirken işte en aptal insanın bile kullanabileceği şekilde üretiyor. Yani o telefonu kullanmak aslında bir teknoloji katkı değil o sadece herkes kullanabilsin diye üretilmiş yani bu piyasası bu şekilde olsun diye işte insanlara daha fazla insana hitap edebilsin diye. Ama çocuklar işte bu algıyı değiştirebilmek için seçmeli olmaması kesinlikle zorunlu olması gerektiğini düşünüyorum. Seçmeli olsa nasıl olsa biliyor diye tercih edilmiyor.”*

Kodlama eğitiminin bütün öğrencilere seçmeli ders olarak verilmesi gerektiğini düşünen 3 öğretmen, kodlama eğitiminin ilgi duyan öğrencilere verilmesi gerektiğini belirtmişlerdir. Bu öğretmenlerin görüşleri *seçmeli* kategorisi altında bir araya getirilmiştir. Aşağıdaki alıntıda Ö4 kodlama eğitiminin ilgi duyan öğrencilere verilmesinin nedenini öğrenciler için daha yararlı olması ve ders verimliliği açısından ifade etmiştir.

Ö4: *“...seçmeli bir ders gibi olabilir bu istekli öğrencilerin olması daha mantıklı diye düşünüyorum. Çünkü her öğrencinin kapasitesi ve ilgisi aynı değil. Kimisi sanata ilgi duyuyor, kimisi işte bu şekilde bilişime o yüzden bilişime ilgi duyan öğrencilerin olması hem ders verimliliği açısından hem de öğrenci için daha yararlı olacağını düşünüyorum.”*

Kodlama eğitiminin hem zorunlu hem de seçmeli ders olarak verilmesi gerektiğini savunan 3 öğretmen kodlama eğitiminin temel olarak herkese verilmesi ardından ilgisi olan öğrencilere seçmeli ders olarak verilmesi gerektiğini belirten

açıklamalarda bulunmuşlardır. Bu öğretmenlerin görüşleri *zorunlu ve seçmeli olmalı* kategorisi altında bir araya getirilmiştir. Aşağıdaki alıntı *zorunlu ve seçmeli olmalı* kategorisine örnek olarak verilebilir.

Ö7: “... Ee eee çocuklar zaten severek geliyorlar da çok sıkılmıyorlar *zorunlu* olabilir ama *seçmeli* olarak da ayrı sadece kodlamayla ilgili ders verilebilir aslında hani seven öğrencilere o ders verilebilir...”

4.4. Kodlama Eğitimine Başlangıç Yaşı

Öğretmenlerin “*Kodlama eğitimine başlangıç yaşı hakkında ne düşünüyorsunuz*” sorusuna verdikleri yanıtlardan kodlama eğitimine başlangıç yaşı hakkında; *anasınıfı, ilkokul ve ortaokul* olmak üzere 3 kategori ortaya çıkmıştır. Kodlama eğitimine başlangıç yaşı konusundaki ifadeleri *ilkokul* başlığı altında bir araya getirilen 12 öğretmen çeşitli sebeplerle düşüncelerini dile getirmişlerdir. Bunlara bakacak olursak; ilkokul dönemindeki öğrencilerin algısı ve kodlamaya yönelik ilgisinin daha fazla olduğu, zihinsel ve bedensel gelişim düzeylerinin ilkokul döneminde uygun olduğu, kodlamanın bir dil öğrenmek gibi aşamalı olarak öğrenilmesi gerektiği bu nedenle ilkokulda eğitime başlanması gerektiği ve kodlama eğitimi ile öğrencilere kazandırılacak olan becerilerin diğer derslerde de olumlu etkisinin olacağından bahsederek kodlama eğitimine neden ilkokul düzeyinde başlanması gerektiğini açıklamışlardır. Aşağıdaki alıntı ilkokul kategorisine örnek olarak verilebilir.

Ö13: “*Daha erken ilkokul seviyesinde, okuma yazma öğrendikten sonra... Sayısal, yani şimdi okuma yazmayı neden öğrendikten sonra aslında hani görsel şeylerle de yapabiliyorlar ama en azından sayısal düşünce biraz daha geç geliyor, soyut düşünce. Soyut düşünce kazansınlar sonra ancak kodlamaya geçiş, soyut düşünemeyen çocuğa kodlama öğretmek somut düşünme döneminde çok zordur... Çocukların yani küçük yaşlarda başlarsak temelini oturturursak zaten sonradan kendisi gelişir yani kendi kendisi üzerine katabilir. Çok geç olduğunda bu mümkün değil, düşünme şekli ile alakalı.*”

Kodlama eğitimine başlangıç yaşı hakkındaki düşünceleri *anasınıfı* başlığı altında bir araya getirilen 8 öğretmen ağaç yaşken eğilir, üretkenlik, öğrenme kolay ve küçük yaşlarda da öğrenebilirler alt kategorileri ile kodlama eğitimine neden anasınıfında başlanması gerektiğini ifade etmişlerdir. Öğretmenlerin açıklamalarına bakıldığında, öğrencilerin okuma-yazma bilmesinin önemli olmadığı ve bu yaşlardaki çocukların uyarılara daha açık, algı düzeylerinin daha yüksek olduğu, öğrencilere küçük yaşlarda üretici olma bilinci kazandırmak gibi çeşitli sebepleri sunarak kodlama eğitimine anasınıfında başlama gerekçelerini dile getirmişlerdir. Aşağıdaki alıntı anasınıfı kategorisine örnek olarak verilebilir.

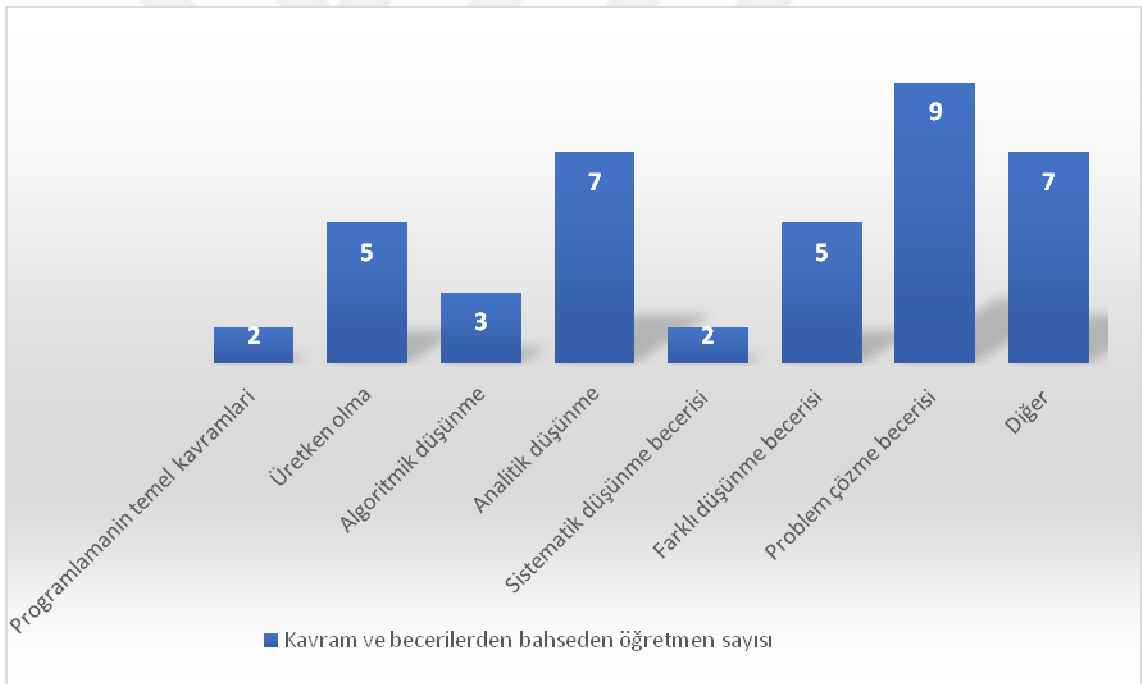
Ö20: *“Çocukların gelişim dönemlerine göre hani düşünürsek anasınıfındaki bir öğrencinin algı düzeyinin daha iyi olduğunu daha etkili olduğunu söyleyebiliriz. Bu yaştaki gruplara hani bu yaştaki çocuklar aldıkları eğitimi hiçbir zaman unutmayacaktır ve bu konuda daha iyi ilerleyeceklerdir. Bu eğitimi lisede veya üniversitede verdikten sonra biraz geç kalmış oluyoruz ama anaokulu, ilkokuldan itibaren bunu verirsek küçüklükten itibaren çocuklar yazılımı kodlamayı öğrendikten sonra kendi oyunlarını bile tasarlayabiliyorlar.”*

Kodlama eğitimine başlangıç yaşı hakkındaki düşünceleri *ortaokul* başlığı altında bir araya getirilen 2 öğretmenden birisi neden ortaokulda başlanması gerektiğini konusunda bir açıklama yapmamıştır. Diğer öğretmen ise ortaokul seviyesindeki öğrencilerin soyut düşünme becerisine sahip oldukları için daha iyi algılayabildiklerini bu nedenle de ortaokulda kodlama eğitimine başlanması gerektiğini ifade etmiştir. Aşağıdaki alıntıda öğretmenin bu konudaki görüşü yer almaktadır.

Ö11: *“...6. sınıftaki öğrenciler olabilir yani 11 yaş falan olabilir çünkü daha küçüklerde pek algılayamıyorlar. Çok soyut kalıyor yani 5. Sınıflarda normal e-posta adresi alırken bile zorluk yaşıyoruz işte forum sitelerini anlatırken bile zorluk yaşıyoruz. Ama 6. Sınıfta daha çok oturmuş oluyor daha çok iyi kavrayabiliyorlar. Yani 11 yaş olabilir.”*

4.5. Kodlama İçin Temel Kavram ve Beceriler

Öğretmenlerin “Sizce kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler nelerdir?” sorusuna verdikleri yanıtlar sonucunda, kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler ile ilgili görüşlerinden; *programlamanın temel kavramları, üretken olma, algoritmik düşünme becerisi, analitik düşünme becerisi, sistematik düşünme becerisi, farklı düşünme becerisi, problem çözme becerisi ve diğer* kategorileri ortaya çıkmıştır. Ortalama her öğretmen 2 kavram ve beceriden bahsetmiştir. Grafik 4.1’de ise her bir kavram ve becerinin kaç sayıda öğretmen tarafından bahsedildiği gösterilmiştir.



Grafik 4.1. Kodlama eğitimi ile kazandırılmak istenen temel kavram ve beceriler

Öncelikle ortaya çıkan kategorileri genel anlamda değerlendirecek olursak, öğretmenlerin bahsettikleri kavram ve beceriler hakkında gereken yanıtları vermelerine rağmen yaptıkları açıklamaların muğlak olduğu, net yanıtlar vermedikleri ve kafalarının karışık olduğu görülmektedir. Bu şekilde açıklamalarda bulunan öğretmen yanıtları aşağıdaki alıntı ile örneklendirilmiştir.

Ö16: “Kavram ve beceriler hakkında tam bi bilgim yok açıkçası o konuda... Yani... Kavram ve beceriler... Yani çocuğun mantıksal düşünmesini, sorunlarını çözme becerileri u... daha sonra ve ileriye görme yani ilerde bir harita oluşturabilmesi yani ilerde yaşayabileceği, ileri görüşlülüğüne artı olabilir... Farklı düşünme becerileri geliştiriyorlar... Başka şu an aklıma gelenler bunlar.”

Kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler ile ilgili düşüncelerini *problem çözme becerisi* olarak ifade eden 9 öğretmenden 4’ü problem çözme becerisinden bahsederken herhangi bir açıklamada bulunmamıştır. Diğer öğretmenler ise kodlama eğitimiyle öğrencilere problem çözme, karşılaşılan problemlere farklı çözüm yolları üretebilme, problemler karşısında çözüm odaklı düşünebilme ve problemleri kısa yoldan çözebilme becerisi kazandırılmasından bahsetmişlerdir. Aşağıdaki alıntı problem çözme becerisi kategorisine örnek olarak verilebilir.

Ö9: “Şöyle diyeyim. Ee bunu seviye seviye ayırmakta fayda var. Öncelikle mesela benim yıllık program... Hazırlamış olduğum yıllık programı açarsak benim kodlamaya dair ilk kazanımlarım “problem” kavramını tanımlar. Problem çözmeye odaklanır. Bir problem için farklı çözüm alternatifleri sunar. Aslında kodlama diyoruz ya hayatımızın içinde. Tamam kodlama bir dilden bahsediyor olabiliriz ama hayatı da ilişkilendirebiliriz. Çocuklara pratik düşünmeyi, sorunlara farklı çözüm yolları üretmeyi de öğretebilir bu ders aynı zamanda.”

Kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler ile ilgili düşüncelerini *analitik düşünme becerisi* kategorisini oluşturacak nitelikte dile getiren 7 öğretmenin analitik düşünme ile neyi kast ettiği tam olarak anlaşılammamaktadır. Aşağıdaki alıntı analitik düşünme becerisi kategorisine örnek olarak verilebilir.

Ö11: “yani bi analitik düşünme becerisi geliştirebilir. Çünkü ders siz de bilirsiniz ayrıntılı olduğu için kodlama işi düzen gerektirdiği için, çocuklara bu düzen ve analitik düşünme becerisini kazandırabilir.”

Kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler ile ilgili görüşlerini *üretken olma* kategorisini oluşturacak nitelikte dile getiren 5 öğretmenden 2'si kodlama eğitimi ile öğrencilerin üretme duygusunu geliştirmekten bahsetmiştir. Diğer öğretmenler ise öğrencilerin hayal dünyalarını genişleterek üretim odaklı düşüncelerini sağlamaktan, onlara üretici bireyler olabileceklerini göstermekten ve robotik çalışmalar gibi yapılan faaliyetler ile bu durumun sağlanabileceğinden bahsetmişlerdir. Aşağıdaki alıntı üretken olma kategorisine örnek olarak verilebilir.

Ö9: *“Kodlamayla birlikte bir şeyler başarabilmek, bazı projeleri hayata döndürebilmek. Ufak tefek projeleri hayata döndürebilmek olabilir. ...Şöyle diyeyim size, en önemlisi şu, bir öğrencinin aslında çok severek ilgileneceği konular bunlar öğrencileri tanıyorum çünkü. Şimdi robotik çalışmalar var biliyorsunuz bir öğrencinin hem bir oyuncak geliştiriyor, bunun ondaki muazzam hissini düşünün, hem bir şeyler başarmış olmanın tatmin duygusunu öz güvenini düşünün, hem de gerçekten programlama namına bir şeyler öğrenmiş oluyorlar bu esnada. Yani imkânlar el verseydi, yani öğrenciler kodlama noktasında bambaşka yerlerde olabilirdi.”*

Kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler ile ilgili düşünceleri *farklı düşünme becerisi* kategorisini oluşturacak nitelikte görüşlerini bildiren 5 öğretmenden 2'si öğrencilerin farklı açılardan düşünme becerisi kazandıklarından bahsederken diğer öğretmenler farklı düşünme becerisinde yaratıcılığa vurgu yapmışlardır ve yaratıcı düşünmeyi farklı düşünme becerisi ile bağdaştırmışlardır. Öğrencilerin varolan üzerinden değil, yaratıcılıklarını kullanarak kendi yollarından gittiklerini düşündükleri görülmüştür. Aşağıda kodlama eğitimi ile öğrencilerin farklı düşünme böylelikle farklı çözümler bulabilmelerinin sağlandığını düşünen öğretmenin ve farklı düşünme becerisinde yaratıcılığı vurgulayan bir öğretmenin açıklamaları yer almaktadır.

Ö6: *“...farklı düşüncelerini sağlıyor evet farklı çözümler bulabiliyorlar onu da sağlıyor. Çünkü hani çocuklar dedim ya benim hiç düşünemediğim şeyleri bile düşünebiliyorlar. Hocam şöyle de olabilir diye söyleyebiliyorlar. Farklı düşünce farklı çözüm yolları bulabiliyorlar.”*

Kodlama eğitimi aracılığı ile öğrencilere kazandırılması gereken kavram ve beceriler ile ilgili düşüncelerini *algoritmik düşünme becerisi* kategorisini oluşturacak nitelikte görüş bildiren 3 öğretmenden 2'si bu kavramdan açıkça bahsetmemiştir. Diğer öğretmen ise öğrencilerin karşılaştıkları herhangi bir durum için kafasında yapacaklarını, izleyeceği yolu oluşturabilmesi, şemalaştırabilmesini sağlamaktan bahsetmiştir. Aşağıdaki alıntı algoritmik düşünme becerisi kategorisine örnek olarak verilebilir.

Ö5: “Şöyle bir şey mesela bilgisayar başında ihtiyaç duyduğu bir program konusunda kendi algoritmasını yazabilmesi kendi kafasında veya kullandığı programda, örneğin bir oyun oynuyor, oyunda topa mesela tekme atıyor topa tekme attığında karşılaşılabileceği durumlarla ilgili algoritmayı düşünebilmesi, kafasında oluşturabilmesi, şemalaştırabilmesi yani ben bunun kodunu yazsaydım şu şekilde yazardım diyebilmesi...”

Kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler ile ilgili düşüncelerini *sistemik düşünme becerisi* kategorisini oluşturacak nitelikte görüş bildiren 2 öğretmenin bu kavramdan bahsederken yaptıkları açıklamaların yetersiz olduğu görülmektedir. Aşağıdaki alıntı sistemik düşünme becerisi kategorisine örnek olarak verilebilir.

Ö6: “...Bu ders çocuklarda belki biraz daha şey u sistemli düşünmeyi falan sağlayabilir çocuklarda belki. Daha planlı yapabiliyorlar her şeyi. Bilgisayar konusunda yaptıkça...”

Kodlama eğitimi aracılığıyla öğrencilere kazandırılması gereken kavram ve beceriler ile ilgili temel kavramlar, döngüler, koşul ifadeleri vb. öğretilmesi gerektiğini ifade eden ve öğrenilen bu temel becerilerin diğer derslerde de faydalı olabileceğini belirten 2 öğretmenin görüşleri ile *programlamanın temel kavramları* kategorisi oluşturulmuştur. Aşağıdaki alıntı programlamanın temel kavramları kategorisine örnek olarak verilebilir.

Ö1: “ ...bir programın nasıl yazılması gerekiyor nasıl u örnekle gidelim aslında u koşul cümleleri ya da işte nasıl diyelim eğerli kodları yazabilme yani bir nevi kod yazmadaki o birkaç nüansı verdikten sonra gerisini çocuk getirecek zaten yani bir dili kullanmayı öğrendikten sonra başka dilleri daha kolay zaten sende biliyorsun hani yeterki bir dilin mantığını aslında dilin mantığını değil program nasıl yazılıyor nasıl ilerliyor işte döngüler mesela nasıl döngü kuruluyor nasıl eğer koşulları kuruluyor...”

Kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler ile ilgili öğretmen yanıtlarında konu ile ilgisi olmayan ve az frekanslı öğretmen yanıtları ile diğer kategorisi oluşturulmuştur. Aşağıdaki alıntı diğer kategorisine örnek olarak verilebilir.

Ö2: “Yani bir an da sorunca bilemedim. Şey işte dediğim gibi çocuklar yaratıcılıklarını geliştirmeleri gerekiyor çünkü dediğim gibi o kadar şey ki mesela yazı yazarken bile “Kırmızıyla mı yazalım siyahla mı?” “İstedğiniz renkle yazın” diyorum çocuk onu benim söylememi bekliyor. O kadar kalıplaşmış her şey hani o kalıpların daha dışına çıkılması gerektiğini düşünüyorum. Ona kendisi karar vermeli, işte karar verme yetenekleri yok şu an çocukların. Her şeyi sizden bekliyorlar ve ben öyle olmasını istemiyorum. Mesela bir programda bile bir şey yaptıracağsın işte “Böyle mi yapayım?” “Nasıl istersen öyle yap” diyorum. Çünkü o senin programın, senin hayalin... Ama o kadar alışmışlar ki şartlanmışlar öğretmen ne derse onu yapmalıyız. Onun biraz yıkılması gerektiğini düşünüyorum yani o konuda.”

4.6. Programlama için Yazılım Donanım Gereksinimi

Öğretmenlerin “Kodlama eğitimi ile öğrencilere kazandırılması gereken bu kavram ve beceriler [Bir önceki sorudaki kavram ve beceriler kastedilmektedir] için kodlama ile ilgili hangi donanım ve yazılımları kullanırsınız? Sonda: Neden bu yazılım ve donanımları kullanmayı tercih edersiniz?” sorusuna verdikleri yanıtlar sonucunda kodlama eğitimi için yazılım donanım gereksinimi olarak görsel programlama araçları, mobil uygulamalar, 3B yazılımlar, robot yazılımlar, metin tabanlı ortamlar ve B³

etkinlikleri başlıkları oluşturulmuştur. Öğretmenler en çok görsel programlama araçları başlığı altında toplanmıştır.

Görsel programlama araçları başlığında en çok tercih edilen programlama aracı olan blok tabanlı programlama araçlarından bahseden öğretmenlerin yanıtları doğrultusunda 22 öğretmen Scratch'tan, 15 öğretmen Code.org'tan, 5 öğretmen Hacker.can'dan ve diğer başlığı altında 4 öğretmen Alice, Blockly Game, Ozoblockly.com vb. programlama araçlarından bahsetmişlerdir. Öğretmenlerin bahsettiği Scratch, Code.org ve Hacker.can'ı kullanma sebepleri olarak duyuş, öğrenci seviyesine uygun, popüler olması, düşük zemin, öğretmene kolaylık sağlaması, yazılımsal özellikler ve diğer kategorileri oluşturulmuştur.

9 öğretmen bahsettikleri görsel programlama araçlarını tercih etme sebepleri olarak; kodlamanın öğrencilere oyun tabanlı öğrenme ortamında görseller, ses efektleri, oyun karakterleri gibi ilgilerini çeken bir ortamda sunulmasını eğlenceli bulduklarını dile getirerek *duyuş* kategorisini oluşturmuşlardır. Aşağıdaki alıntıda Ö19 programın sunmuş olduğu görsellik, ses efektleri, tasarım yapabilme gibi özelliklerinin öğrencilerin ilgisini çektiğinden ve bireysel olarak birşeyleri yapabilmenin onlara vermiş olduğu mutluluktan bahsetmiştir.

Ö19: “...Daha böyle renkli, öğrencilerin ilgisini çeken, sesler falan ekli olan, seslendiriyor, konuşuyor, rengini kıyafetlerinin rengini değiştirebiliyor. Kendi kendine birtakım şeyler tasarlayıp hareket ettirdiği zaman yani mutlu oluyor, ilgisini çektiği için yoksa... klasik menüsü olan hiç resim olmayan görsel içerikleri olmayan bi program olsa hiç ilgisini çekmez. Oturupta uğraşmaz yani. İlkokul ya da ortaokul seviyesindeki bi öğrenci. Yapmak istemez ancak çok hani böyle bilgisayar başında aşırı vakit geçiren bişeyler tasarlamayı seven ileri düzey bir öğrenci olacak ki ancak öyle geçer hani yoksa normal bi öğrenci pek ilgisini çekmez bence.”

Bahsedilen görsel programlama araçlarını tercih etme nedeni olarak *öğrenci seviyesine uygun olmasından* bahseden 5 öğretmen bulunmaktadır. Öğretmenlerin ifadelerine bakacak olursak; öğrenci seviyesine uygun bir düzeyde kodlamayı anlatan, daha anlaşılır programlar olduğundan, öğrenci seviyesine göre gruplandırılmış

eğitimleri içerdiğinden genel anlamda öğrencilere daha yakın programlar olduklarından bahsetmişlerdir. Aşağıdaki örnekte Ö22, öğrencilerin anlayabileceği düzeyde seviyelerine uygun bir şekilde tasarlanmış, oyunlaştırılmış bir ortamda kodlama öğretimi sunan programlar olduğundan bahsetmiştir.

Ö22: *“Çünkü bunlar öğrencinin anlayabileceği şekilde tasarlanmış ve çok basite indirilmiş oyunlar. Aslında oyun gibi görüyor çocuklar bunu ama oyun değil yani aslında temelinde arka planda çocuk bir algoritmayı oluşturmayı öğrenmiş oluyor. Bir kod yazmış oluyor. Basit kodlarla hani basit bir ara yüzle kendi programını kendi hazırlamış oluyor. Oyun hazırlıyorlar animasyon tasarlıyorlar bu şekilde. Öğrenci seviyesine uygun olduğu için.”*

Bahsedilen programlama araçlarını tercih etme nedeni olarak *popüler olmasından* bahseden 5 öğretmen açıklamalarında; popüler programlar olması, şu anda kullanılıyor olması, dünyanın her yerinde tercih edilen bir program olması gibi ifadeler yer almaktadır. Aşağıdaki alıntıda Ö19 dünyanın birçok yerinde Scratch'ın tercih ediliyor olmasından, bilişim platformları, öğretmenlerin paylaşımları, EBA eğitim portalı gibi yerlerde Scratch üzerinden kodlama öğretiminin tercih edilmesinden bahsetmiştir.

Ö19: *“Ben şu an Scratch kullanıyorum. Zaten bildiğim kadarı ile dünyanın birçok ülkesinde o kullanılıyor. Şu an 2.0'ı çıktı uı başka programlarda var ama ben onu tercih ettim genelde öğretmenlerin hepsi internetten baktım genelde Scratch kullanıyorlar... bilgisayar bilişim teknolojisi öğretmenlerinin paylaşımlarının bulunduğu forumlar var uı EBA'ya biraz baktım kodlama eğitimi ile ilgili EBA'da Scratch ile ilgili anlatım yapmış hatta sayfa açmışlar Boğaziçi üniversitesinin öğrencilerinin yaptığı çalışmaları falan paylaşmışlar EBA'da da öyle olunca ben Scratch ile eğitim verdim.”*

Düşük zemin kategorisini oluşturacak nitelikte görüşlerini bildiren 5 öğretmen bahsedilen programlama araçlarını tercih etme nedeni olarak öğrencilerin kolaylıkla öğrenebilmesine olanak sağladığını düşünmüşlerdir. Bu kategoride öğretmenler bahsettikleri programlama araçlarının öğrencilere sunduğu puzzle gibi kodları bir araya

getirebilme olanağı ile kullanma kolaylığı sağladığı, sunduğu görsellik sayesinde kodlamaya dair birşeyler yapmanın daha kolay olduğu ve zorlanmadan, seviyelerine uygun düzeyde yapabilecekleri uygulamalar sunan programlar olduklarından bahsetmişlerdir. Aşağıdaki alıntıda Ö16 öğrencilere puzzle mantığı ile rahat bir kodlama yapabilme olanağı sunmalarından bahsetmiştir.

Ö16: *“Çocuklar daha rahat anlıyorlar ve daha rahat bir şekilde yapboz parçalarını birleştirir gibi bunları yerleştirerek daha rahat kodlama yapabiliyorlar bu da bu yaptıkça zaten kodlama becerilerini daha da artırıyor...”*

Bahsedilen programlama araçlarını tercih etme nedeni olarak *öğretmene kolaylık sağlaması*'ndan bahseden 3 öğretmen, hazır programlar üzerinden daha eğitici ve öğretici olduğunu ve öğretmene kolaylık sağladığını dile getirmişlerdir. Aşağıdaki alıntıda Ö8 hazır İnternet tabanlı programlar ile daha rahat ve öğretici olduğundan bahsetmiştir.

Ö8: *“Neden bu yazılım ve donanımları kullanmayı tercih ediyorum. Bu yazılım ve donanımlarla biz bilgisayar öğretmenlerinin işi çok rahatlıyor var olmuş hazır internet tabanlı bir program üzerinden daha eğitici ve öğretici olabiliyoruz...”*

Yazılımsal özellikler kategorisini oluşturacak nitelikte görüşlerini bildiren 3 öğretmen bahsedilen programlama araçlarını tercih etme nedeni olarak ücretsiz olması, erişim kolaylığı sağladığı için öğrencilerin evlerinde de ulaşma imkânı bulmaları, küçük boyutlu olduğu için fazla yer kaplamaması ve Türkçe kullanım desteği sayesinde öğrencilerin anlayabilmesi ve kolaylıkla kullanılması açısından fayda sağladığından bahsetmişlerdir. Aşağıdaki alıntıda Ö12 programlama araçlarını tercih etme sebebi olarak ücretsiz olması, erişim kolaylığı sağlaması ve fazla yer kaplamamasından bahsetmiştir.

Ö12: *“Ücretsiz, ulaşması kolay, çok fazla yer kaplamıyor, sistem olarak düşük bilgisayarlarda da çalışabiliyor bu sebeple yani.”*

Diğer kategorisinde ise az frekanslı öğretmen görüşleri yer almaktadır. Bu kategori altında bulunan 6 öğretmenin bahsettikleri programlama araçları hakkındaki ifadelerine bakıldığında; öğrencilere öğrendikleri basit bilgiler ile karmaşık şeyler yapabilme imkânı sunduğu (yüksek tavan), öğrencilerin etkinlikleri kolayca yapabilmesi sonucu mutlu oldukları ve motivasyonlarında artış olduğu, öğrencilere sunduğu görsel, işitsel öğrenme ortamları sayesinde kalıcı öğrenme sağladığı, dünyada geliştirilen birçok yazılımların, uygulamaların, web sitelerinin geliştirilmesi bu programlarla yapıldığı için ve meslektaş tavsiyesi sonucunda tercih ettiklerini ifade etmişlerdir. Aşağıdaki alıntıda Ö14 öğrencilerin edindikleri basit bilgiler ile çizgi film, oyunlar gibi karmaşık şeyleri yapılabileceğinden bahsetmiştir.

Ö14: *“Gerçekten burada şurada kaç hafta oldu Scratch’e başlayalı bir buçuk ay falan oldu. Çocuklar çizgi film yazmaya başladılar. Ya yapıyorlar ve hoşlarına gidiyor ve şu oluyor izledikleri çizgi filmlere “Aaa bu böyle miymiş!” falan şeyler olmaya başladı. Bende hayatım boyunca Scratch dersi veren bir adam değilim. Bende şaşırmağa başlıyorum “Aaa hakikaten böyle mi oluyormuş!” falan diye. Bir de oyun yazmaya başladılar. İşte şey oyunları, mesela geçen bir öğrenci yapmış: Labirent. İşte bir labirent çizmiş, labirente bir top koymuş yön tuşlarıyla topu en sola getiriyor. Orada bir skor veriyor falan sonra sağa sola çarptığında falan yeniden başlıyor. Öyle değişik oyunlar yapmışlar. Bunlar çok basit ama güzel şeyler diye düşünüyorum.”*

Kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler için robotların kullanılmasından bahseden 9 öğretmenin düşünceleri ile *robot yazılımlar* başlığı oluşturulmuştur. Bu öğretmenlerden 3’ü Arduino’dan, Maestro education, Makeblock, Dash & Dot robot gibi uygulamalardan birer öğretmen bahsetmiştir ve bazıları da sadece robotik yazılım diyerek uygulama ismi vermemiştir. Öğretmenler bu yazılımları kullanma nedeni olarak ise; öğrencilerin geliştirdikleri yazılımlar ile robotu hareket ettirme ve bunu nasıl gerçekleştirdiklerini öğrenmelerinin ve yapılan programlamanın sonucunu somut olarak görme imkânının ilgilerini çekmesi, meslektaş tavsiyesi, öğrenci seviyesine uygun olması ve bir şeyler üretebileceklerinin farkına varmalarını sağlamasından bahsetmişlerdir. Aşağıdaki alıntıda Ö5 öğrencilerin

yaptıklarını doğrudan görebileceği ve Scratch ile birlikte kullanma imkânı sunduğu için tercih edilebileceğinden bahsetmiştir.

Ö7: “Yaş aralığı bence iyi hani çocukların çok çabuk anlayabileceği bir şey bu Arduinio. Arduinio da en azından o basit Scratch’la Arduinio’yu kullanabiliyorsun o yönden seçerdim ya hani eee yaptığım programlamanın da sonucunu direkt o Arduinio sayesinde görebiliyorsunuz.”

3B yazılımlar başlığını oluşturacak nitelikte düşüncelerini belirten 5 öğretmen kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler için 3B yazılımların kullanılabilirliğinden bahsetmiştir. Öğretmenlerden 3’ü Tinkercad, 2’si Sketchup ve 1 tanesi Coreldraw’ı örnek olarak vermişlerdir. 3B yazılımlardan bahsedenler tercih etme nedeni olarak; öğrencilerin anlayabileceği şekilde tasarlanmış, eğlenceli, görsel olarak zenginleştirilmiş, oyun tabanlı bir ortamda öğrencilere kendi programlarını yazma oluşturma imkânı sunması, erişim kolaylığı sağlaması, ücretsiz ve küçük boyutlu olmalarından bahsetmişlerdir. Aşağıdaki alıntıda Ö3 derse görsellik katmak istediği zamanlarda öğrenci seviyesine uygun olan ve bilgisayarlarda rahatlıkla çalışabilecek donanımsal ve yazılımsal özellikleri barındırdığını düşündüğü Skecthup programını kullanmayı tercih ettiğini ifade etmiştir.

Ö3: “Tabi kodlama yaparken de sadece kodlama değil de biraz görsellik gerekiyor bu görsellik kısımlarını yapması içinde CorelDraw ile Sketchup denilen programlar var 3 boyutlu onları gösteriyorum genellikle bunlarla dersi işliyorum. Esas nedenim şu ne yazık ki bizim bilgisayarlarımız biraz kötü. Hani olabildiğince küçük program olması gerekiyor kaldırabilmesi için bilgisayarlarımızın donanım bakımından en iyi kaldırabilecek özelliklere baktığımızda Skecthup mesela 3 boyutlu çizebiliyoruz hem bilgisayarlarda kasma yapmıyor. Hem de öğrencilerin seviyesine yönelik...”

Metin tabanlı ortamlar başlığını oluşturacak nitelikte düşüncelerini belirten 3 öğretmen kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler için Visual Basic, PHP, Asp gibi metin tabanlı ortamların kullanılabilirliğinden

bahsetmiştir. Bu ortamlardan bahseden öğretmenler herhangi bir gerekçe sunmamışlardır.

Mobil uygulamalar başlığını oluşturacak nitelikte düşüncelerini belirten 2 öğretmen kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler için Android studio gibi mobil uygulama platformlarının kullanılabilceğinden bahsetmiştir. Aşağıdaki alıntıda Ö11 mobil uygulamaların en popüler yazılım ürünleri olduğunu ve öğrencilere vakit geçirdikleri telefon, tabletlerde uygulamalar oluşturmanın, ilgi alanlarına yönelik şeyler yapmanın onlara cazip geldiğini düşündüğünden dolayı tercih edeceğinden bahsetmiştir.

Ö11: “...ama belki bu mobil uygulama yazılımlarından hani ileri seviyede düşünebilirim onları çok beğeniyorum çünkü en popüler şeylerdi onlar artık mobil yazılım ürünleri... Mobil uygulamalarda genelde çocuklar hep tablet telefonlardı yani kendileri vakit geçirdiği için yani oynadıkları şeyleri kendileri üretmek isteyebilir çünkü daha çok ilgilerini çeker diye düşünüyorum o yüzden yani onları düşündüm, seçtim. Çünkü normal bir yazılım yazmaya kalksa işte ne biliyim masaüstü uygulaması yapmaya kalksa o kadar ilgisini çekmeyecektir. Kendi uğraştığı şeyleri yapmak daha cazip geliyor çocuklara.”

Son olarak B³ etkinlikleri başlığını oluşturacak nitelikte düşüncelerini belirten bir öğretmen, kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler için legoların kullanılabilceğinden bahsetmiştir. Bu başlıktan bahseden yalnızca bir öğretmen bulunduğu halde tutmamızın sebebi alanyazında sıklıkla geçen önemli başlıklardan birtanesi olmasıdır. Aşağıdaki alıntıda B³ etkinliklerinden bahseden öğretmenin görüşü sunulmuştur.

Ö21: “İlkokul ortaokulda yani şuan Avrupa’da neler kullanılıyor tam bilmiyorum da bizim ülkemizde bununla ilgili yazılmış birkaç kitap var ama daha çok kolejdaki öğrencilerin alabileceği kaynaklar falan olduğu için ben şimdi açık söyleyeyim ben tamamı tüm eşitlik olarak baktığım için tüm ülkeye şöyle söyleyeyim tamamen minimum malzemeyle eğitimcilerin tamamen sorular şeklinde analitik

düşünmeye algoritmik düşünmeye yönlendirebileceği bir yapılar olsun isterim ben. Mesela en basit en ucuzu nelerdir, Legolar.”

4.7. Öğretmenlerin Bilgisayarsız Bilgisayar Bilimi Hakkındaki Düşünceleri

Bilgisayarsız Bilgisayar Bilimi (B^3) etkinlikleri hakkında öğretmenlerin düşüncelerini ortaya çıkarmak amacı ile onlara *“Bu yazılım ve donanımları kullanmadan kodlama eğitimdeki kavram ve becerileri kazandırma hakkında ne düşünüyorsunuz?”* sorusu yöneltilmiştir. Öğretmenlerin soruya verdikleri yanıtlar sonucunda B^3 'ten bahsedenler, B^3 'ten bahsetmeyenler, informal B^3 , olumsuz bakanlar ve olumlu bakanlar kategorileri ortaya çıkmıştır.

B^3 'ten bahsedenler kategorisini oluşturacak nitelikte açıklamalarda bulunan 3 öğretmen katıldıkları eğitimler, atölye çalışmaları gibi faaliyetler sonucunda yazılım donanım kullanmadan nasıl B^3 etkinlikleri yapılacağından haberdar olduklarını ifade etmişlerdir. B^3 etkinlikleri olarak, oyun kartları hazırlama, kâğıt üzerinde çeşitli zekâ oyunları, define avı gibi oyunlar, code.org benzeri bilgisayarlı yapılabilecek uygulamalar ve birçok etkinliğin yapılabileceğinden haberdar olduklarını söylemişlerdir. Ancak öğretmenlerin tamamı bu etkinliklerin faydalı olacağını düşünmediklerini, öğrencilerin B^3 etkinliklerini uygularken sıkılacağı, aktivitelerin uygulanmasının zor ve yetersiz olacağı, öğrencilerin yazılım ve donanımlarla etkileşimde olmaları gerektiği gibi nedenlerle belirtmişlerdir. Aşağıdaki alıntı B^3 'ten bahsedenler kategorisine örnek olarak verilebilir.

Ö14: *“Ya bununla alakalı bir kaç defa eğitime gittim işte o İstanbul'a gittiğim Yıldız Teknik Üniversitesindeki eğitim bu donanım ve yazılımlar olmadan çocuklarda kodlama eğitimi nasıl olur? Birçok etkinlik yapıyorsun ama ben bunların çok faydalı olacağını düşünmüyorum. Çünkü öğrenci sıkılıyor. Beklenti farklı yani ben öyle düşünüyorum. Öğrenci sıkılıyor ya yani o tarz, öğrenciyi bile bırakın biz o etkinlikleri yaparken biz bile sıkıyoruz. Bizim sıkıldığımız bir şeyi öğrenciye veremezsin, zorlayamazsın. Çocuk buraya geliyor, 2 saat ders işliyor burada yeri geldiği zaman*

buradan sıkılmadan çıkacak bir tanesi ‘Of’ dedim mi gerçekten moralim bozuluyor. Yani niye sıkılıyor bu çocuk? Ne yaptık da sıkıldı? Bu tabii benim fikrim. Çünkü sıkılmayacakları o kadar çok şey var ki bu sınıfta, buna rağmen çocuk sıkılıyorsa demek ki yaptığımız etkinlikler, verdiğiniz şeylerle alakalı sorun var diye düşünüyorum. Ben o donanımlar olmadan kodlama eğitimlerinin öğrencide kazandıracacağı bir şey olduğunu düşünmüyorum. Belki anasınıfta öğrencileri, ilkokul birinci sınıf öğrencilerine biraz daha sembolik olarak o tarz şeyler verilebilir ama ortaokuldaki öğrencilere bu donanımlar ve yazılımlar olmadan kodlama ile alakalı bir şey verilmez. Bunların hepsi benim fikrim.”

Yazılım ve donanımları kullanmadan kodlama eğitimindeki kavram ve becerileri kazandıramayacaklarını düşünen 15 öğretmenin görüşü ile B³'ten bahsetmeyenler kategorisi oluşturulmuştur. Bu kategoride yer alan öğretmenlerden 3'ü açıkça bilgisayarsız bir ortamda kodlama eğitiminin nasıl yapılacağı hakkında bilgi sahibi olmadıklarını çünkü var olan hazır programlardan yararlandıklarını ifade ederken, 9 öğretmen yazılım donanım olmadan kodlama eğitiminin gerçekleştirilmesinin mümkün olmadığını çünkü anlatılanların soyut kalacağı ve öğrenilen bilgilerin kalıcı olmayacağından, 3 öğretmen algoritma ve akış diyagramları öğretiminden ileri gidilemediğinden bahsetmiştir. Öğretmenlerin açıklamalarına baktığımız zaman B³ etkinlikleri ile kodlama eğitimi gerçekleştirileceğinden haberdar olmadıkları ortaya çıkmaktadır. Aşağıdaki alıntı Ö9, uygulama yazılımları olmadan kavram ve becerilerin nasıl öğretilene dair bir yanıt bulamadığını, algoritma ve akış diyagramları öğretiminden ötesine geçeceğini düşünmediğini ifade etmiştir.

Ö9: *“Şu an benim yaptığım şey o aslında kodlama, bu yazılım ve donanımları kullanmadan dediği gibi akış diyagramları ve algoritmalarla öteye geçilebileceğini düşünmüyorum hani çünkü bu sonuçta bu biraz daha uygulayarak, yaşayarak öğrenilebilecek konular. Ben en azından bulunduğumuz imkânlarda işte çocuklar bir üst eğitim öğretim kurumuna geçtiğinde en azından işin bir noktasının farkına varmış biliyor olmaları da benim için bir kar diye düşünüyorum. Yani ben düşündüm hani nasıl yapılabilir o imkânlar olmadan ben ona bir cevap bulamadım açıkçası. Hadi ben*

tahtada gösteririm. Tahtaya bir kodlama programı kurup ben öğrencilerime gösteririm ama sizce bu ne kadar kalıcı olur?”

Yazılım ve donanımları kullanmadan kodlama eğitimindeki kavram ve becerilerin kazandırılabilirliğinden bahseden ancak bu konuda bir eğitim almadan çevrelerinden bir şekilde haberdar olmuş ya da farkında olmadan deneyimlemiş olan 4 öğretmenin görüşü ile *informal B³* kategorisi oluşturulmuştur. Bu öğretmenlerin görüşlerine baktığımız zaman kodlama eğitimindeki kavram ve becerilerin öğretiminin yazılım ve donanımları kullanmadan oyunlarla yapılabileceğine ilişkin, öğrencilere yönerge hazırlayıp saklı olan birşeyin buldurulması, kare boyama, algoritma ile ilgili yön buldurma gibi örnekler verdikleri görülmüştür. Öğretmenlerden biri bilgisayarsız ortamda birtakım etkinliklerin yapıldığından haberdar olduğunu, genel anlamda bu öğretimin nasıl gerçekleştirileceğine dair bilgisinin olmadığından MEB'in bu konuda yardımcı olmasını istediğini belirtmiştir. Aşağıdaki alıntıda Ö16, kodlama öğretiminde bilgisayara olmasa da olur mantığı ile bakmaktadır ve bu konuda çeşitli uygulamaların yapılabileceğini ancak uygulayamadığını sadece kare boyama şeklinde kâğıt üzerinde bir etkinlik yaptığını ifade etmiştir.

Ö16: *“Yani çeşitli algoritma ve şeyler var bildiğim kadarıyla ama çok uygulayamadım açıkçası. Mantıklı düşüncelerini sağlayabilecekleri çıktılarla hani yazılım elle yazarak kâğıt üzerinde onların da faydalı olduğunu düşünüyorum. Hatta bir tane yapmıştık. Kareler şeklinde kodlar yazmayı. Yani basit anlamda çocuk nereleri doldurması gerektiğini ve hangi yoldan gitmesi gerektiğini kısa yoldan onu çözmeye çalışıyor. Bu tür işte kare boyama şeklinde bi etkinlik yapmıştık. Mesela u yine zekâ oyunları da oynattığımız kare karalamaca da bu mantığa giriyor. Yani o şekilde de yapılabilir. Bilgisayar illa bilgisayar olmasına gerek yok.”*

Yazılım ve donanımları kullanmadan kodlama eğitimindeki kavram ve becerilerin öğretime karşı olumsuz bir bakış açısına sahip olan 15 öğretmenin açıklamaları ile *olumsuz bakanlar* kategorisi oluşturulmuştur. Bu konuda öğretmenlerin düşünceleri *verim düşük olur* ve *uygulama ve/veya görsellik gerekliliği* alt kategorilerine ayrılmıştır. Verim düşük olur alt kategorisini oluşturan öğretmenlerin

görüşlerine bakıldığında yazılım donanım kullanmadan yapılan kodlama eğitiminin etkili, faydalı, yeterli ve kalıcı olmayacağından, öğrencilere sıkıcı geleceğinden, bu şekilde kodlama öğretimi gerçekleştirilmesinin de öğrenciler tarafından öğrenilmesinin de zor olacağından bu nedenle öğrencilerin yazılım ve donanım ile etkileşimde olmaları gerektiğinden bahsettikleri görülmüştür. Aşağıdaki alıntı verim düşük olur alt kategorisine örnek olarak verilebilir.

Ö18: “Yapılabilir mi bence çok yapılmaya çalışsa da çok başarılı olunamaz bence bu konuda. Çünkü öğrenciyi bşeye yönlendirirken ilk önce çalışacağı ortamı veya bilgisayarı düzenlemek ve psikolojik olarak onu hazır hale getirmek önemlidir. Bu yazılımları kullanmazsak örneğin Scratch programını kullanmazsak bunu biz yazılım kullanmadan nasıl yapabiliriz bilmiyorum ben bence hiç mümkün değildir yani donanım olmadan hani elimizdeki imkânlarla bi şekilde ortamı ayarlayabiliriz ama yazılımlar olmadan yapamayız yani hiçbir şekilde.”

Uygulama ve/veya görsellik gerekliliği alt kategorisini oluşturan öğretmenlerin görüşlerine bakıldığında ise yazılım donanım kullanmadan yapılan kodlama eğitiminin zor olacağından, uygulama yaparak tam öğrenmenin gerçekleştiği aksi takdirde öğrenilenlerin havada kalacağından, öğrenilenler soyut kavramlar olduğundan öğrencilerin bu kavramları somutlaştırarak anlaması için yaparak yaşayarak öğrenmeleri gerektiğinden bu nedenlerle de yazılım ve donanımlara ihtiyaç olduğundan bahsettikleri görülmüştür. Aşağıdaki alıntıda Ö4 görsellik olmadan ve uygulama yapılmadan öğrenilenlerin havada kalacağı düşünülmektedir.

Ö4: “Bilişim görsel bir şey bunlar olmadan çocuk bu uygulamaları yapmadan bu çözümlenmeyi yapabileceğini zannetmiyorum. Bu sadece havada kalır. Ne yapabilirim ben bunu slayt hazırlayıp o şekilde mi gösteriyim hayır, resmini çekip mi göstereyim hayır, yazılarla yazdırarak mı anlatıyım olmaz. Bunu mecbur uygulamasının olması gerekiyor. Görsel ve uygulayarak yapmasının daha kalıcı olacağını düşünüyorum. Hem görececek hem uygulayacak.”

Yazılım ve donanımları kullanmadan kodlama eğitimindeki kavram ve becerilerin öğretimi hakkında B³ eğitimi almamalarına rağmen olumlu bir bakış açısına

sahip olan 4 öğretmenin görüşü ile *olumlu bakanlar* kategorisi oluşturulmuştur. Öğretmenlerin ifadelerine baktığımızda bilgisayar kodlama eğitimi ortamında gerekli görmediklerini kare boyama gibi çeşitli aktivitelerin yapılabileceğini ancak genel anlamda yazılım donanım kullanmadan nelerin yapılacağı konusunda bilgilerinin olmadığını ve eğlence tabanlı bir eğitimin yararlı olacağını belirtmişlerdir. Aşağıdaki alıntı olumlu bakanlar kategorisine örnek olarak verilebilir.

Ö21: “Çok yararlı olur. Çocuklara bu tür eğlenceli aktiviteler daha kalıcı öğrenmeler sağlar. Çocuğa yazı yazdırmadan sınavı yapmadan tamamen eğlence tabanlı verilen bu şekildeki bir algoritma ve analitik düşünme eğitiminin çok yararlı olacağını düşünüyorum. Bu konuda herhangi bir eğitim almadım veya herhangi bir araştırma yapmadım. Birçok aktivite çeşidi olduğunu okumuştum ama neler uygulanıyor ve uygulanabilir konusunda bilgim yok.”

4.8. Öğretmen Eksiklikleri

Öğretmenlerin “*Bilişim teknolojileri ve yazılım dersi öğretmenlerinin sizin gözünüzde öğrencilerin kodlama becerilerini geliştirme konusunda eksik olduğu noktalar var mıdır?*” sorusuna verdikleri yanıtlar sonucunda *alan bilgisi eksikliği*, *üniversitedeki pedagojik alan bilgisi eksikliği* ve *diğer* kategorileri oluşturulmuştur. Öğretmenlerin bu başlıkta en çok altında toplandığı kategori *alan bilgisi eksikliği* kategorisidir.

Öğrencilerin kodlama becerilerini geliştirme konusunda öğretmenlerin eksik olduğu noktalar hakkında *alan bilgisi eksikliği* başlığını oluşturacak nitelikte görüş bildiren 12 öğretmen genel anlamda öğretmenlerin kodlama/programlama hakkında bilgi ve beceri konusunda eksik olabileceklerinden bahsetmişlerdir. Bu konuda öğretmenler; *klasik diller/kodlama genel* ve *öğrenciler için güncel programlama ortamları bilgisi eksikliği* alt kategorilerinde görüş bildirmişlerdir.

Öğrencilerin kodlama becerilerini geliştirme konusunda öğretmenlerin eksik olduğu noktalar hakkındaki düşünceleri ile *öğrenciler için güncel programlama*

ortamları bilgisi eksikliği alt kategorisi oluşturulan 7 öğretmen, lisans döneminde öğrenilen kodlama bilgisinin ortaöğretim öğrencilerine öğretilen kodlama eğitiminden farklı olduğu, meslekte ilerlemiş öğretmenlerin güncel eğitim platformlarını bilmiyor olabileceği ve sürekli gelişen değişen bir bölüm olduğundan dolayı öğretmenlerin daima kendilerini yenilemesi gerektiği ancak öğretmenlerin bu konuda yetersiz kalabildiğinden bahsetmişlerdir. Aşağıdaki alıntı öğrenciler için güncel programlama ortamları bilgisi eksikliği alt kategorisine örnek olarak verilebilir.

Ö3: *“Tabiki de var. Şöyle deyim u benim mezun olduğum okulda bana öğretilen programlar şunlar; delphi, Pascal, PHP ama gelgelelim buraya geldiğinde delphi’yi Pascal’ı veya PHP’yi anlatamazsın ki bizim hocalarımızın hepsi mühendislik bölümü mezunu olduğu için bizden hep mühendislikle ilgili şeyler istediler. Siteden mesela bi mağaza programı istediler ama ben o mağaza programı burada işime yaramaz o yüzden eksiklerimiz biraz oradan kaynaklanıyor... elimizden geldiğince kendimizi geliştirmeye çalışıyoruz. Skecthup’tır Scratch’tır. Genellikle birazcık daha üniversitedeki gibi değil de ortaokuldaki öğrencilerin seviyesine inmeye çalışmakta biraz sıkıntılarımız var.”*

Klasik diller / kodlama genel alt kategorisini oluşturacak nitelikte görüş bildiren 6 öğretmen, lisans eğitiminde alınan programlama dilleri eğitimi yetersiz kaldığı için kodlama konusunda yeterli bilgi sahibi olunmadığı, birçok öğretmenin lisans döneminde öğrendiği programlama / kodlama bilgisini unutmuş olabileceği, öğretmenler kendilerini geliştirmedikleri için kodlama eğitimi konusunda yetersiz kalabildikleri ve öğretmenlerin birçoğunun kodlama konusunda birçok yönden eksikliklerinin olabileceği bu nedenlerden dolayı da kodlama bilgisi konusunda öğretmenlerin eksik olabileceklerini düşünmektedirler. Aşağıdaki alıntı klasik diller / kodlama genel alt kategorisine örnek olarak verilebilir.

Ö1: *“Tabi yani özellikle ilk yılları olmayanlar benim gibi 5 yılın üstü olanlar körelmiştir biraz. Çünkü eğer ekstra bi iş yapmadıysa word, excel, powerpoint anlatıyoruz paso. Düşünsene şimdi 10 yıllık bilgisayar öğretmeni word excel powerpoint anlatıyor körelmiştir kesinlikle, biraz açıcak bakıcak biz u şimdi şöyle*

söyleyim mezun olduğum bi arkadaşla kendi çapımızda program falan da yazmaya çalışıyoruz bişeyler yapmaya çalışıyoruz. Bişeyler yapmaya çalışıyoruz. Onun verdiği bişeyle biraz bişeyler kaldı ama bu işe de hani bu tarz programlar yazmaya da yeni yeni yazmaya başladık arkadaşla. Dedik bak köreliyoruz bişeyler yapalım falan işte Flash oyunu falan dedim ya bunları yapmazsak ki yapmayanları biliyorum onlar körelmiştir onların şeye ihtiyacı var kesinlikle sililmeye ihtiyacı var bakıcaklar biraz tekrar, ha eğitim eğitime çok gerek yok internet hani diyorum ya temel var bizde internetin başına geçecek birkaç video seyredecek gerekirse birkaç kod bakıcak entegre olur ama ihtiyaç var.”

Öğrencilerin kodlama becerilerini geliştirme konusunda öğretmenlerin eksik olduğu noktalar hakkında *üniversitelerdeki pedagojik alan bilgisi eksikliği* başlığını oluşturacak nitelikte düşüncelerini dile getiren 7 öğretmen genel anlamda lisans döneminde aldıkları dersler içerisinde kodlama bilgisinin öğrencilere nasıl aktarılacağı ile ilgili eğitim almadıklarını, bu nedenle ortaokul düzeyindeki öğrencilere kodlama eğitimi verme noktasında sıkıntı yaşadıklarını dile getirmişlerdir. Aşağıdaki alıntıda bu konuda görüş bildiren öğretmenlerden Ö5 kodlama bilgisini öğrencilere öğretebilecek bilgi düzeyine sahip olmadığını çünkü kodlama bilmenin ve öğretmenin farklı şeyler olduğunu, lisans döneminde alınan eğitimin bu konuda yetersiz kaldığını ifade etmiştir.

Ö5: *“Ben kendimi örnek veriyim. Ben bu konuda eksik hissediyorum kendimi yani müthiş eksikim hatta şöyle söyleyim kodlama eğitimi verebilecek bilgi düzeyim %10’dur belki öyle söyleyim size. Niye diyeceksiniz çünkü kodlama bilmek ayrı şey öğretmek ayrı şey bunu nasıl aktaracağınız konusunda benim üniversite yılarında aldığım eğitimi yeterli görmüyorum açıkçası. Şimdi belki yeni bilgisayar öğretmenleri bilişim BÖTE bölümlerine koyuyorlar bu dersi ama onlarınki de yeterli değil onlarda çok geriden takip ediyorlar. İşte şimdi biz tam böyle ateşin olduğu noktadayız şu an. Öğrencilerin beklentileri ile yetişen öğretmenlerin hazır bulunmuşlukları aynı düzeyde değil ...”*

Diğer kategorisinde ise konu ile ilgisi olmayan veya az frekanslı öğretmen görüşleri yer almaktadır. Bu kategori altında, öğrencilerin kodlama becerilerini

geliştirme konusunda öğretmenlerin eksik olduğu noktalar hakkında görüş bildirmeyen, eksik olduklarını söyleyip sebep belirtmeyen ve bilişim teknolojileri ve yazılım dersine ve öğretmenlerine verilen önemin artması gerektiği bunun beraberinde kaliteli öğretmen yetiştirileceğini ifade eden 3 öğretmenin görüşleri yer almaktadır. Aşağıdaki alıntı diğer alt kategorisine örnek olarak verilebilir.

“Ö18: Şöyle söyleyim ben eğitim fakültesi mezunuyum...şu anda biliyorsunuz ki bilişim öğretmenlerinin kadrosu çok az hani bekleyenimiz çok bu yüzden birçok öğretmenimiz öğretmen olmayı artık hayal olarak görüyor ve buradan aldığı eğitimi kullanamayacağını düşünüyor. Sadece 4 yıllık bir üniversiteden mezun olmuş olurum. Burdan sonra işte farklı iş alanlarına yönelirim diye düşünüyor. Bilişim öğretmenlerine önem verilip geleceğini artık gelecek kaygısından çok ders kaygısının veya öğrenme arzusunun arttırılması gerektiğini düşünüyorum. Böyle olursa hem bilişim öğretmenlerimiz kaliteli yetişmiş olur hem de kaliteli yetişen öğretmenlerimiz daha iyi öğrenci yetiştirmiş olur diye düşünüyorum.”

4.9. Öğretmen Eksiklikleri İçin Yapılması Gerekenler

Öğrencilerin kodlama becerilerini geliştirme konusunda öğretmenlerin eksik olduğu noktalar hakkında görüş bildiren öğretmenlere, “*Bu eksikliklerin giderilmesine yönelik olarak neler yapılmasını istersiniz?*” sorusu sorulduğunda verdikleri yanıtlar sonucunda *kişisel gelişim, üniversitelerdeki eksikliklerin giderilmesi, eğitimler verilmeli ve diğer alt kategorileri oluşturulmuştur.* Öğretmenlerin en çok altında toplandığı kategori eğitimler verilmeli kategorisidir.

Öğretmen eksikliklerinin giderilmesine yönelik olarak *eğitimler verilmeli* kategorisini oluşturacak nitelikte düşüncelerini ifade eden öğretmenlerin verdikleri yanıtlar sonucunda *seminer, çalıştay ve atölye gibi ve hizmetiçi eğitim* alt kategorileri oluşturulmuştur. Seminer, çalıştay ve atölye gibi eğitimler verilmesi gerektiğini düşünen 7 öğretmen farklı açılardan düşüncelerini dile getirmişlerdir. Bunlara bakacak olursak; internet ortamında edinilen bilginin çok etkili olmadığı ancak düzenlenen atölye

çalışmaları gibi etkinliklerin, farklı şehirlerdeki öğretmenlerin kodlama ile ilgili hangi uygulamaları yaptıklarını bilmelerini sağlaması ve öğretmenlerin özdeğerlendirme yaparak kendilerinin kodlama öğretiminde nerede olduklarını görmeleri açısından verimli olduğundan bahsetmişlerdir. Ayrıca bölümümüz güncel, sürekli gelişen ve değişen bir bölüm olduğu için öğretmenlerin bölümü geriden takip etmemek adına, kodlamaya dair bilmedikleri veya güncel programlama ortamları gibi konularda bilgi sahibi olabilme fırsatı bulabildiklerini bu ve benzeri eğitimlerin yaygınlaştırılabileceğini ifade etmişlerdir. Aşağıdaki alıntıda Ö2 çalıştay ve atölye çalışmaları gibi etkinliklerle öğretmenlere kodlamaya yönelik eğitimler verilebileceği, öğrencilere bu eğitimin nasıl aktarılacağı konusunda da seminerler düzenlenebileceğinden bahsetmiştir.

Ö2: “Eğitimler verilebilir ve bu eğitimleri şöyle düşünüyorum. Şimdi önce bize verilebilir bu eğitimler mesela ben gittim. Şimdi eğitimler genelde üniversitelerde veriliyor şimdi üniversitelerde verildiği zaman tamam ben anlıyorum, ben yapabiliyorum ama şimdi onu çocuk yapabilir mi? Mesela bunun da geri dönüşünün sağlanması gerekiyor diye düşünüyorum. Bize orada bir şey anlatıyorlar tamam biz yapabiliyoruz çünkü bizim belli bir bilgi birikimimiz var kapasitemiz var. Şimdi aynı şeyi çocuğa anlatmaya kalktığımızda yapamayabiliyor. Mesela bize Alice gösterdiler. Tamam çok güzel ben yaptım orada gösterilen etkinliği. Ben aynı şeyi burada çocuklara gösterdiğimde yani %80’i yapamayacak belki de %90’ı yapamayacak çünkü işte ne bileyim İngilizce kavramlar var, işte farklı matematiksel değerler var falan. Hani bu eğitimler sadece bize yönelik değil de çocuklara uygulandığında da nasıl geri dönüt alınıyor bunların incelenmesi gerektiğini düşünüyorum. Çünkü tamam biz eğitiliyoruz ama çocuklara bunu ne kadar verebiliyoruz, ne kadarını alabiliyor karşı taraf o da önemli onu düşünüyorum. Bir de belki üniversitelerde tamam dersler değişiyor ama bizim aldığımız eğitimde öyle kodlama durumu yoktu. En azından bu olmadığı için hani belli bir süreden sonra mezun veren yerlerde mesela benim veya benden daha önce göreve başlayan arkadaşlara bu tarz eğitimler verilebilir diye düşünüyorum.”

Öğretmen eksikliklerinin giderilmesine yönelik olarak hizmet içi eğitim verilmesi gerektiğini düşünen 10 öğretmen bulunmaktadır. Bu öğretmenlerden 4’ü doğrudan olarak hizmet içi eğitimlerin verilmesi gerektiğini söylerken, diğerleri hizmet

İçerik eğitimin hangi konularda ve neden verilmesi gerektiğine yönelik olarak açıklamalarda bulunmuşlardır. Öğretmenlerin bu konudaki görüşlerine bakacak olursak; öğretmenleri kodlama eğitimi hakkında bilgi sahibi yapmak, kodlama eğitimini öğrencilere nasıl aktaracağı, öğrenci düzeyine göre kodlama konusunda neler yapılabileceği, hangi yöntem ve teknikler kullanması gerektiği vb. konular hakkında bütün öğretmenlerin bilgi sahibi olabilmesi için hizmet içi eğitimlerin düzenlenmesi gerektiği bu konuda akademisyenlerin hizmet içi eğitim düzenleyebileceği veya bakanlık kararı ile her ilden birkaç öğretmene eğitim verildikten sonra o öğretmenlerin de buldukları ildeki diğer öğretmenlere hizmet içi eğitim vermesinin sağlanabileceğini ifade etmişlerdir. Aşağıdaki alıntıda Ö8 öğrencilere kodlama bilgisinin nasıl öğretileceği, öğrenci seviyesine göre kodlama eğitimi konusunda neler yapılabileceği konusunda öğretmenlere hizmet içi eğitim verilebileceğinden bahsetmiştir.

Ö8: “Bunların giderilmesine yönelik işte hizmet içi eğitimler planlanabilir. Dediğim gibi yükseköğretimde kodlama yazılım işte ya da kodlama nasıl öğretiriz çocuklara kodlamayı nasıl veririz. İşte çocuklarla birlikte 5. 6. Sınıf ortaokul düzeyi öğrencileri için yazılımı nasıl geliştirebiliriz diye herhangi bir eğitim almadığı için illaki bilişim öğretmenlerinin bu konuda eksiklikleri olabilir ve bunları kendi düşüncesiyle veya kendi araştırma yöntemiyle tamamladığı için bilgisayar öğretmenleri seviyeleri farklı olabilir bu seviyeleri eşit düzeye getirmemiz için bilişim teknoloji öğretmenlerine bir hizmet içi eğitim düzenlenebilir diye düşünüyorum o kadar.”

Öğretmen eksikliklerinin giderilmesine yönelik olarak üniversitelerdeki eksikliklerin giderilmesi başlığını oluşturacak nitelikte çeşitli nedenler sunarak düşüncelerini dile getiren 5 öğretmen bulunmaktadır. Öğretmenlerin açıklamalarına bakacak olursak; lisans eğitim programlarının mevcut hali yetersiz olduğu için düzenlenmesi gerektiği, öğrencilere kodlama eğitiminin nasıl aktarılması gerektiği ile ilgili eğitimin öncelikle üniversitelerde AR-GE çalışmalarının yapılması daha sonra üniversitelerde ders olarak verilmesi gerektiği, üniversitelerde alan mezunu akademisyenlerden oluşan uzman kadroların olması gerektiği, üniversite içi - üniversiteler arası uygulama yarışmaları gibi aktivitelerin olması ve üniversite son

sınıfta edinilen öğretmenlik deneyimini bütün lisans eğitimine yayarak öğretmenlik eğitiminin daha verimli olmasının sağlanması gerektiği şeklinde açıklamalarda bulunmuşlardır. Aşağıdaki alıntıda Ö9, öğretmen eksikliklerinin giderilmesine yönelik olarak öncelikle üniversite müfredatlarında kodlama eğitimi konusunda düzenlemeler yapılması gerektiğini ifade etmiştir.

Ö9: *“Öncelikle lisans programlarında işte bu dersin daha yani dediğiniz gibi düzenlenmesi gerekiyor yani bu en başta buradan başlamalı tabiki öğretmenin de kendini geliştirmesi orada okuyan lisans öğrencisinin de kendini geliştirmesi gerekiyor. Derste öğrenip bittiği kadar kalmamalı. Ama şu haliyle çok yetersiz.”*

Öğretmen eksikliklerinin giderilmesine yönelik olarak, öğretmenlere eğitim vermek yerine onların bireysel olarak kendilerini eksik oldukları noktalar konusunda değerlendirmeleri ve eksiklikleri konusunda bireysel olarak kendilerini geliştirmeleri gerektiğini ifade eden 3 öğretmenin görüşleri ile *kişisel gelişim* kategorisi oluşturulmuştur. Aşağıdaki alıntıda Ö19, EBA eğitim platformu gibi internet kaynakları üzerinden diğer öğretmenlerle birlikte ders içeriği, materyali ve etkinlikler konusunda paylaşım yoluyla öğretmenlerin bireysel olarak kendilerini geliştirmelerinden bahsetmiştir.

Ö19: *“Bu eksikliklerin giderilmesine yönelik ben genelde öğretmenlerin paylaşımlarına bakıyorum kendim takip ediyorum internetten. EBA çok işime yarıyor. EBA’da öğretmenler çok fazla programlama dilleri ile ilgili içerikler paylaşıyorlar. İnternette çok paylaşım oluyor. Üniversite öğrencileri eğitim fakültelerindeki üniversite öğrencileri proje çalışmaları yapıyor onları paylaşıyorlar internetten ben onları indiriyorum bakıyorum. Kodlama çalışmaları neler yaptırmışlar ben neler yaptırabilirim, ama mesela bazı program kodlamaları oluyor mesela onları ben öğrencilerime çok ileri seviye olduğu için onlara.. bide seviyeye de bakmak gerekiyor... Evet zaten öyle bilişim teknolojileri öğretmenleri kendilerini kendileri yetiştiriyor yani. İçerikleri paylaşıyoruz, etkinlikleri paylaşıyoruz, ben paylaşıyorum onlar paylaşıyor karşılıklı.. bu şekilde paylaşım yoluyla hani aslında ders materyalleri de bu şekilde oluyor bizim dersimizde. Hani normalde elimde mesela benim bi öğretim programım*

var çizelge halinde, temel şeyler yazıyor işte temel beceriler kazanımlar onları kazandırmak için materyalleri her bilişim teknolojileri öğretmeni kendi öğrencilerinin seviyesine göre kendisi hazırlıyor.”

Diğer kategorisinde yer alan 1 öğretmen, öğrencilerin kodlama becerilerini geliştirme konusunda öğretmenlerin eksik olduğu noktalar hakkında neler yapılabileceğine dair görüş belirtmemiştir.



V. BÖLÜM

5. Tartışma, Sonuç ve Öneriler

Bu bölümde ilk olarak araştırma sonucunda elde edilen bulguların temelinde ulaşılan sonuçlar, tartışma ve sonuç başlığı altında değerlendirilmiştir. Ardından sonuç ve yorumlardan yola çıkarak bundan sonra yapılacak olan çalışmalara ve alanyazına katkı sağlayacağı düşünülen önerilerde bulunulmuştur.

5.1. Tartışma ve Sonuç

Bu çalışma bilişim teknolojileri ve yazılım dersi öğretmenlerinin kodlama eğitimine yönelik düşünce ve görüşlerini ortaya çıkarmayı amaçlamaktadır. Bu amaca yönelik olarak nitel bir yaklaşımla yarı-yapılandırılmış görüşme formu kullanılarak bilişim teknolojileri ve yazılım dersi öğretmenlerinden veri toplanmıştır. Öğretmenlerle yapılan görüşmeler sonucunda elde edilen bulgularda öncelikle kodlama eğitiminin öğretmenler için ne ifade ettiğine bakılmıştır. Bu sonuçlar kodlama eğitimi, kodlamanın amacı, herkes için kodlama, kodlama eğitimine başlangıç yaşı, kodlama için temel kavram ve beceriler, programlama için yazılım donanım gereksinimi, öğretmenlerin bilgisayarsız bilgisayar bilimi hakkındaki görüşleri ve öğretmen eksiklikleri ve bu eksikliklerin giderilmesi için yapılması gerekenler başlıkları altında incelenmiştir.

5.1.1. Kodlama eğitimi

Kodlama eğitimi kategorisinde öğretmenlerin kodlama eğitimi ile ilgili genel görüşleri bulunmaktadır. Öğretmenlerin kodlama ile ilgili görüşleri incelendiğinde

öğretmenlerin ağırlıklı olarak düşünme becerilerinden bahsettiği görülmektedir. Bu becerilerin algoritmik düşünme, yaratıcılık, mantıksal ve/veya matematiksel düşünme ve düşünme becerisini geliştirme, pratik problem çözme ve çözüm şemaları başlıklarından oluştuğu gözlemlenmiştir. Kodlama eğitimi, öğrencilerin yalnızca bilişim okuryazarı olmalarını sağlamak veya kodlama yapabilen bireyler olmalarını sağlamakla sınırlı değildir. Kodlama dersleri ile öğrencilerde bilgi işlemsel düşünme becerileri de geliştirilmelidir (Brennan ve Resnick, 2012). Wing (2006), bilgi işlemsel düşünmenin herkes tarafından temel bir beceri olarak öğrenilmesi gerektiğini ifade etmiştir. Bilgi işlemsel düşünmenin tanımı hakkında bir görüş birliği yoktur, fakat tanımında bulunması konusunda ortak görüş sağlanan birtakım kavramlar vardır. Bunlar; düşünme süreci, problem çözme, yaratıcılık, algoritmik düşünme, ayrıştırma, soyutlama, genelleme ve değerlendirmedir (Aho, 2011; Çakır, Korkmaz ve Özden, 2017; Çetin ve Toluk Uçar, 2017; Kalelioğlu, Gülbahar ve Kukul, 2016; Selby ve Wollard, 2013; Yecan, Özçınar ve Tanyeri, 2017; Wing, 2006). Öğretmenlerin kodlama eğitimi ile ilgili görüşleri, alanyazında kodlama eğitimine yönelik ortaya konulan temel kavramların kapsamı ve içeriği ile karşılaştırıldığında, bu görüşlerin sınırlı olduğu tespit edilmiştir. Öğretmenler tarafından ayrıştırma, soyutlama, genelleme ve değerlendirme gibi bilgi işlemsel düşünmenin temel bileşenlerinden hiç bahsedilmemiştir. Buna ek olarak, her ne kadar görüşmelerde öğretmenlere bu fırsat tanınmış olsa da öğretmenlerin fikir sahibi olduğu başlıkların da alanyazını tam olarak kapsadığını söylemek güçtür. Bu durumun nedeni araştırma katılımcılarının da bahsettiği gibi öğretmenlerin lisans döneminde kodlama konusunda yeterli bilgiye sahip olamadıkları ve kendilerini geliştirme konusunda yetersiz kaldıklarından olabilir.

Problem çözme ile ilgili alanyazın incelendiğinde temel olarak problem çözmede bağlamın etkisi, problem çözme stratejileri, problem çözme şemaları, problem çözme için gerekli bilgi ve beceriler, problem çözmede üst bilişin etkisi ve problem türleri gibi alanların çalışıldığı görülmüştür (Flavell, 1979; Jonassen, 2000; Mayer, 1992; Newell ve Simon, 1972; Toluk, Uçar ve Altun, 2006). Buna karşın öğretmenlerin problem çözme ile ilgili söyledikleri dikkate alındığında çözüm şemaları ve pratik problem çözme başlıklarından bahsedildiği görülmüştür. Bu başlıklar incelendiğinde genel olarak öğretmenlerin kısa yoldan problem çözmek, farklı çözüm yolları üretmek

ve çözüm üretmek için sahip olunması gerekli şemalar hakkında görüş bildirdikleri gözlenmiştir. Problem çözme alanyazını ile öğretmenlerin problem çözme üzerine söyledikleri karşılaştırıldığında öğretmenlerin görüşlerinin derinlemesine olmadığı ve alanyazında bulunan birçok başlığı içermediği görülmektedir. Öğretmenlerin kodlama eğitimi ile ilgili problem çözme becerisinden bahsederken, problem çözmenin alanyazında bahsedilen temel alanlarının birçoğunu bilmemeleri, onların alan bilgisi konusunda eksikliklerinin olmasından kaynaklanabilir.

Algoritmik düşünme alanyazını incelendiğinde algoritmik düşünmenin bir problemi çözerken ihtiyaç duyulan adımları oluşturabilme, uygulayabilme ve değerlendirme becerilerinden oluştuğu, bir bilgi işleme birimi tarafından bir problemin çözümü için üretilen metottan bahsedildiği, problemleri çözme veya belirli durumları anlamlandırma yolu olarak farklı tanımlamalarla ifade edildiği görülmektedir (Csizmadia ve diğerleri, 2015, s.7; Sedgewick ve Wayne, 2011). Öğretmenler algoritmik düşünmeyi, bir işin nasıl yapılacağı ve bu süreçteki işlem basamaklarının bilinmesi olarak tanımlamaktadırlar. Algoritmik düşünme alanyazını ile öğretmenlerin algoritmik düşünme ile ilgili söyledikleri karşılaştırıldığında öğretmenlerin görüşlerinin derinlemesine olmadığı görülmüştür.

Yaratıcılık alanyazını incelendiğinde yaratıcılığın yeni şeyler düşünmek veya yeni yollarla düşünmek, fikirler arasındaki bağlantıları etkin bir şekilde ortaya çıkarmak, çalışmalara yeni ve alışılmadık dışında bir bakış açısı getirmek gibi tanımlamaların yapıldığı görülmektedir (Ercan, 2017; Lau, 2011; Yecan, Özçınar ve Tanyeri, 2017). Aynı zamanda yaratıcılığın yenilikçi yaratıcılık, yenilenebilir yaratıcılık, ifadede yaratıcılık, üretken yaratıcılık ve keşfedici yaratıcılık gibi alt türlerinin olduğundan bahsedilmiştir (Baker, Rudd ve Pomeroy, 2001). Yaratıcılık ile ilgili alanyazın ile öğretmenlerin ifadeleri karşılaştırıldığında öğretmenlerin, kodlama eğitiminin öğrencilerin yaratıcılıklarını ortaya çıkaracağından bahsederken yaratıcılığın içeriğini ve kapsamını tam olarak ortaya koymadıkları görülmüştür.

Kodlama eğitimi kategorisindeki ikincil başlık ise yazılım geliştiren bireyler yetiştirmek olmuştur. Çalışmanın bulgularında da belirtildiği gibi kodlama eğitiminin

öğretmenler için ifade ettiği anlamlara bakıldığında kodlama eğitimi aracılığıyla bu alanda yetenekli olan öğrencilerin fark edilip yetiştirilmesi imkânı sağlandığından, günümüzde birçok alana teknolojinin önderlik ettiği bu nedenle kullandığımız uygulamaları, programları üretecek bireyler yetiştirerek ülke olarak bu önderlik yarışında Türkiye'nin de güzel bir yer edinebileceğinden ve bilişim çağı için bireyler yetiştirilmesi gerektiğinden bahsettikleri görülmüştür. Kodlama eğitimi alanyazını incelendiğinde, son zamanlarda tüm süreçler teknoloji ile zenginleştirilmeye çalışıldığı için bireylerden teknolojiyi kullanma ve teknoloji ile üretme anlamında farklı yeterliliklere sahip olmalarının beklenildiği, bulut bilişim, yapay zeka, robot programlama, artırılmış ve sanal gerçeklik gibi pek çok mesleğin geleceğe yön vereceği bu nedenle teknoloji bilmenin yetmeyeceği bilgisayar bilimine hâkim olunması gerektiği, önümüzdeki yıllarda mesleklerin büyük çoğunluğunun bilgisayar bilimi bilen kişileri gerektireceği ve bunların da istihdamı arttıracığı yönünde ifadeler yer almaktadır (Bidwell, 2013; Dünya Ekonomi Forumu, 2016; Gülbahar, 2017; Horizon, 2017). Bunların yanında Türkiye'nin donanımı ithal eden konumdan yazılım sektöründe söz sahibi olan ülkeler arasında yer almasını sağlayacak politika ve planlardan da bahsedilmektedir (Gülbahar, 2017). Sonuç olarak öğretmenlerin yanıtlarından yapılan çıkarım ile alanyazın karşılaştırıldığında öğretmenlerin programlama eğitiminin toplumsal ve ekonomik motivasyonundan kapsamlı bir şekilde bahsettiği gözlemlenmiştir.

5.1.2. Kodlamanın amacı

Kodlamanın amacından bahseden öğretmenlerin ifadelerine bakıldığında “Kodlama eğitimi nedir?” sorusuna verdikleri yanıtları daha detaylı bir şekilde kodlama eğitiminin amacı olarak ifade ettikleri görülmüştür. Fakat kodlamanın amacından bahsederken öğretmenlerin kodlama eğitimi kategorisine göre daha yüksek oranda sosyal ve ekonomik faktörlerden bahsetmektedirler. Kodlama eğitimi kategorisinde yazılım geliştiren bireyler yetiştirmek başlığı altında 6 cevap bulunurken, kodlamanın amacı kategorisinde üretici olma ve iş yaşamı başlıklarında toplamda 20 cevap bulunmaktadır. Buradan yola çıkılarak bu çalışmaya katılan öğretmenlerin genel görüş

olarak kodlama eğitimine baktığında ağırlıklı olarak bilişsel bir bakış açısına sahipken kodlamanın amacı söz konusu olduğunda hem bilişsel hem de sosyal ve ekonomik faktörleri göz önünde bulundurduğu söylenebilir.

Öğretmenlerin birçoğu, kodlama eğitimine yönelik görüşlerinde olduğu gibi kodlamanın amacından bahsederken de kodlamanın amacını ekonomik ve sosyal yönden alanyazında bahsedilen başlıklara paralel açıklamalarda bulunmasına rağmen bilişsel açıdan değerlendirirken alanyazına göre hem sınırlı açıklamalarda buldukları hem de bazı kavramlardan bahsederken bahsettikleri kavramın kapsam ve içeriği konusunda eksiklikleri olduğu görülmüştür. Öğretmenlerin ifadelerine bakıldığında kodlamanın amacını, alanyazına paralel olarak, hemen hemen her alanda yazılım var olduğundan bulunduğumuz yüzyılda üretici bir toplum olabilmek adına bilgisayar bilimine hâkim bireyler yetiştirmek; yazılım geliştiren bireyler yetiştirerek teknolojik araç üretimi ile ülke kalkınmasına ve gelişmesine katkı sağlamak; alanın sağladığı iş olanakları sayesinde kaliteli yaşam sunarak istihdam sağlamak; kodlama eğitimi aracılığıyla var olanı tüketmek yerine üretici olabilmek adına öğrencileri küçük yaşlarda üretici bireyler olma bilinciyle yetiştirmekten ve üretici bir toplum haline gelerek ülke ekonomisine katkı sağlamak ifadeleri ile belirttikleri görülmüştür.

Sosyal ve ekonomik görüşlerinin aksine öğretmenlerin bilişsel ve öğretimsel açıdan ortaya koyduğu görüşler daha sınırlıdır. Örnek vermek gerekirse, öğretmenlerin yaratıcılık, analitik düşünme, çözüm şemaları, günlük yaşam problemleri başlıklarından bahsettikleri görülmüştür. Kodlama eğitimi kategorisinde çözüm şemalarının problem çözmenin üzerinde çalışıldığı alanlardan biri olduğundan ve yaratıcılıkla ilgili alanyazındaki farklı tanımlamalardan bahsetmiştik. Analitik düşünme ile ilgili alanyazına bakıldığında analitik düşünemenin bilgiye dayalı problem çözme ve karar vermeyi yargılama, karşılaştırma, analiz etme ve değerlendirmeyi içerdiğinden aynı zamanda analitik düşünme ile problem çözme ve karar verme durumlarında zihinsel süreçlerin bilinçli yönlendirildiğinden bahsedilmektedir (Grigorenko, Jarvin ve Sternberg, 2002; Sternberg ve Grogorenko, 2004; Tok ve Sevinç, 2010). Öğretmenler kodlamanın amacını öğrencilerde analitik düşünmeyi ve yaratıcılığı geliştirmek şeklinde ifade etmişlerdir. Ancak öğretmenlerin bu becerilerin kapsam ve içeriğinin ne olduğu

veya kodlamanın bu becerilerin gelişmesine nasıl veya neden katkı sağlayacağını belirtmedikleri görülmektedir.

Problem çözüme alanyazınına bakıldığında problemlerin yapı, içerik ve soyutluk bakımından farklılık gösterdiği görülmektedir (Çetin ve Toluk Uçar, 2017; Jonassen, 2000). Yapısına göre problemler iyi ve kötü yapılandırılmış problemler olarak iki türde incelenmektedir. Kötü yapılandırılmış problemler önceden tahmin edemeyeceğimiz, günlük yaşamda karşımıza çıkan, problemin nedeninin belli olmadığı, problem çözücünün çözüm aşamasında hangi bilgiyi kullanarak nasıl bir yol izleyeceğine karar vermesi gereken problem türleridir (Toluk Uçar ve Altun, 2006). Öğretmenlerin görüşleri incelendiğinde öğrencilerin kodlama eğitiminin kazandırdığı algoritmik düşünme becerisi sayesinde günlük yaşam problemlerine algoritmik bakış açısı ile yaklaştıkları ve kodlama eğitiminin sağladığı çok yönlü düşünme becerisi ile de öğrencilerin günlük yaşam problemlerinde aşamalı olarak algoritmik bir düzende çözümler bulabileceklerinden bahsettikleri görülmüştür. Problem çözüme alanyazını tam olarak öğretmenlerin bu iddialarını desteklememektedir. Bir alanda öğrenilen bilgiyi başka bir alan aktarmak (transfer) oldukça zordur ve dikkatlice ele alınması gereken bir süreçtir (Perkins ve Salomon, 2012). Bu durum kodlama eğitimi için de geçerlidir. Kodlama dersleri bağlamında öğrenilen kavramların başka alanlara aktarılması otomatik olarak gerçekleşmez ve ciddi bir çabayı gerektirir (Salomon ve Perkins, 1987).

5.1.3. Herkes için kodlama

Bu kategori altında öğretmenlerin kodlama eğitiminin zorunlu olarak verilip verilmemesi konusundaki görüşleri incelenmiştir. Bu çalışmaya katılan öğretmenlerin ezici bir çoğunluğu kodlamanın zorunlu olarak verilmesi gerektiğini düşünürken küçük bir azınlık kodlamanın seçmeli olarak verilmesi gerektiğini iddia etmiştir. Bilgi işlemsel düşünmenin öğrencilere kazandırdığı temel beceriler, bireyler için oluşturduğu iş imkânları, ülkelere olan ekonomik katkısı, teknolojik vatandaşlığın gereği olarak bilgi işlemsel düşünme gibi faktörlerden dolayı birçok Avrupa ülkesi kodlama derslerini zorunlu ya da seçmeli ders olarak anaokulu ve/veya ilköğretim ve/veya ortaöğretimde

müfredatlarına eklemişlerdir veya eklemeyi düşünmektedirler (Bocconi ve diğerleri, 2016).

Öğretmenlerin önemli bir çoğunluğu kodlama eğitiminin öğrencilere zorunlu olarak verilmesi gerektiğini ifade etmiştir. Bulgular kısmında da belirtildiği gibi bu öğretmenler tarafından bireylerin kodlama eğitimi ile yerli yazılımlar geliştirilmesine katkı sağlayacak üreticiler olmalarını sağlamak ve hazır tüketen değil üreten bir toplum haline gelmeyi hedeflemek, böylelikle ülke olarak kendi yazılımlarımızı üretilip ülke kalkınmasına katkı sağlamaktan (Kafai, Burke ve Resnick 2014) bahsedilmektedir. Aynı zamanda hayatımızın her alanında teknolojinin var olduğu günümüze ayak uydurabilmek adına bireylerin düşünme becerilerinin gelişimine katkı sağladığından dolayı kodlama eğitiminin zorunlu olması gerektiğinden bahsetmektedirler. Ayrıca öğretmenler kodlama dersi ile öğrencilerde yaratıcı düşünme, sistemli düşünme ve problem çözme becerisi gibi zihinsel becerilerin geliştirilmesine katkı sağlandığı ve temel bir beceri olarak görüldüğü için herkesin bilmesi gerektiğini düşündüklerinden kodlama eğitiminin zorunlu olarak verilmesini gerekli görüyorlar. Avrupa Birliği Komisyonu tarafından 2016 yılında yayınlanan “Zorunlu Eğitimde Bilgi İşlemsel Düşünme Geliştirme: Politika ve Uygulama Önerileri” raporuna göre müfredatlarında güncelleme çalışmaları yapan ülkelerin amacı; öncelikli olarak öğrencilerin farklı düşünme becerileri kazanmalarını, gerçek yaşam problemlerini farklı açılardan çözmelerini ve teknoloji kullanarak kendilerini ifade etmelerini sağlamaktır. Bunun yanında öğrencileri gelecek mesleklere hazırlamak için bilgi işlemsel düşünme becerisi kazandırmak, bilgi ve iletişim teknolojileri alanında istihdamı arttırmak ve ekonomik büyümeye katkı sağlamaktır (Gülbahar, 2017). Yayınlanan rapora göre ülkelerin müfredatlarında değişikliğe gitme nedenleri ile görüşme yapılan öğretmenlerin kodlama eğitiminin zorunlu olarak verilmesini isteme nedenleri arasında öğrencilerin zihinsel becerilerine katkı sağlama, teknoloji çağının bir gerekliliği olarak ve ekonomik sebepler yönünden tutarlılık olduğu gözlemlenmiştir.

Öğretmenlerin küçük bir kısmı kodlama eğitiminin seçmeli olması gerektiğini veya temel kodlama derslerinin verilmesinin ardından seçmeli olarak farklı alanlarda kodlama derslerinin verilebileceğini ifade etmişlerdi. Kodlama eğitiminin seçmeli ders

olarak müfredatta yer almasını dile getiren öğretmenlerin ifadelerine bakıldığında kodlamanın farklı bir dersin altında verilebileceğinden ya da STEAM yaklaşımı gibi disiplinler arası bir yaklaşımla öğretilebileceğinden bahsetmemişlerdir. STEAM (FeTeMM) fen, teknoloji, mühendislik ve matematik alanlarına ait bilgi ve becerileri kapsamaktadır (Cotabish, Dailey, Robinson ve Hughes, 2013). Öğretmenlerden bazılarının, kodlama eğitimini öğrenmek isteyen ve bu alana ilgi duyan öğrencilere verilmesinin doğru olacağını ifade ettikleri görülmüştür. Öğretmenlerin bir kısmı ise kodlama eğitiminin öncelikle temel bir eğitim olarak herkese verilmesi gerektiğinden ardından ilgi duyan öğrencilere verilmesinin daha doğru olacağından bahsetmektedirler. Kodlama eğitimini müfredatlarına ekleyen ülkelere bakıldığında bunlardan bazıları bilgisayar bilimini seçmeli ders olarak, bazı ülkeler ise İsrail gibi diğer derslerin altında kodlama eğitimi vermektedir. Bunun yanında, STEAM gibi disiplinler arası yaklaşımla, farklı alanlarla birlikte bilgi, beceri ve deneyimlerin bütünleştirilmesiyle de verilebilmektedir (Yıldız, 2017). Sonuç olarak öğretmenlerin de, ülkelerde görülen farklı yaklaşımlarda olduğu gibi, değişik sebeplere bağlı olarak kodlama eğitiminin zorunlu ve/veya seçmeli olarak verilmesi gerektiğini savundukları görülmektedir.

5.1.4. Kodlama eğitimine başlangıç yaşı

Bilgisayar bilimi eğitimi üniversite ve yüksek lisans düzeyinde kurulmuş bir disiplindir. Ancak teknolojinin hayatımızla iç içe olduğu günümüzde A-12 düzeyindeki okullarda da bilgisayar bilimi eğitimi yerini almaya başlamıştır (Bell, Andrea ve Lambert, 2010; Hubwieser, 2012; Sayın, 2017). Bilgisayar bilimi eğitimi konusunda ciddi adımlar atan ülkelere olan Estonya ilköğretim, ortaokul ve ortaöğretim olmak üzere tüm kademelerde, İngiltere 5-6, 7-11 ve 11-14 yaş aralığı olmak üzere 3 aşamada, ABD her eyalette farklı bir eğitim öğretim programı ile bilgisayar bilimi eğitimi sunmaktadır. Türkiye’de ise bilgisayar bilimi eğitimi ilköğretim (5, 6, 7 ve 8. sınıflar) ve ortaöğretim (9 ve 10. sınıflar) düzeyinde verilmektedir (Sayın, 2017; Tebliğler Dergisi, 2016). Öğrencilerin zihinsel ve fiziksel gelişimleri dikkate alınarak hazırlanan öğretim programları ile öğrencilerin hangi seviyede neyi, nasıl öğrenecekleri belirlenebilir (Sayın, 2017).

Çalışmanın katılımcıları tarafından kodlama eğitimine başlangıç yaşı olarak farklı zaman dilimlerinden bahsedilmektedir. Çalışmanın katılımcılarının çoğunluğuna göre kodlama eğitimi ilkokul seviyesinden itibaren verilmelidir. Öğretmenler fikirlerini temellendirirken farklı sebepleri vurgulamıştır. Çalışmanın katılımcılarının bir kısmı ilkokul düzeyindeki öğrencilerin kodlama eğitimi için zihinsel ve bedensel gelişim düzeylerinin daha uygun olduğunu düşünerek bu seviyede başlanması gerektiğini savunmuştur. Zihinsel ve bedensel gelişim çalışmanın katılımcıları sunduğu sebeplerinden sadece biridir. Çalışmanın katılımcılarının bir kısmı öğrencilerin kodlamaya olan ilgilerinin daha fazla olduğu ve kodlama eğitiminin aşamalı bir şekilde öğrenilmesi gerektiği için ilkokulda öğretime başlanması gerektiğini söylemiştir. Bunlara ek olarak çalışmanın katılımcılarından bazıları kodlama eğitimi ile öğrencilerin kazanmış olacakları becerilerin diğer derslere de olumlu etkisinin olacağını düşünerek kodlama eğitimine ilkokul düzeyinde başlanılmasının uygun olacağından bahsedilmektedir. Sonuç olarak ilkokul seviyesinde kodlama öğretiminin başlamasını savunan araştırma katılımcıları öğrencilerin bu yaşlarda kodlama öğrenme yeterliliğinin olduğundan, bu yaşlarda verilen kodlama öğretimi sayesinde öğrencilerin ileriki seviyelerde daha avantajlı olacağından ve kodlama öğretiminin diğer derslerin öğretime avantaj sağlayacağından bahsetmektedirler. Alanyazın incelendiğinde Avusturya, Almanya, Bulgaristan gibi ülkelerin de ilkokul seviyesinde kodlama öğretime başlandığı görülmektedir. İlkokul seviyesinde kodlama eğitiminin verilmesini savunanların sebeplerine bakıldığında alanyazında, ilkokul seviyesinde öğrencilerin bilgisayar teknolojisinin sağladığı olanakların ve sınırlılıkların daha iyi farkına varmış olacakları, bu sayede öğrencilerin erken yaşlarda bilişim teknolojilerinin üretim amaçlı kullanımını öğrenmeleri, bilinçli karar verebilmeleri ve çevrim içi bilgilerin güvenilirliği konusunda fikir sahibi olmalarının sağlanabileceğinden bahsedilmiştir (Sayın, 2017). Bu sebepler araştırma katılımcılarının söyledikleri ile karşılaştırıldığında benzer sebeplerden ötürü kodlama dersinin ilkokul seviyesinde verilmesi gerektiğinin savunulduğu görülür.

Çalışmanın katılımcılardan bir kısmı kodlama eğitimine başlangıç yaşının anasınıfı düzeyindeki öğrenciler için uygun olacağından bahsetmiştir. Alanyazın incelendiğinde bilgisayar bilimi eğitimine başlangıç yaşının Massachusetts ve

Ontario’da 6 yaş olduğu görülmüştür. İngiltere okul öncesi bilgisayar bilimi eğitiminde 5-7 yaş aralığındaki çocuklar için algoritmalar, programlar, veri, internet, iletişim ve bilgisayar konularını öğretmeyi hedeflemiştir (Computing at School ve NAACE, 2014). Kodlama eğitimine başlangıç yaşının anasınıfı düzeyindeki çocuklar için uygun olduğundan ve bu yaşlardaki çocuklara kodlama eğitiminin verilmesi gerektiğinden bahseden çalışmanın katılımcıları, kodlama eğitimi verebilmek için okuma-yazma bilmenin önemli olmadığı, bu yaşlardaki öğrencilerin algı düzeylerinin yüksek olduğu bu nedenle öğrenmenin kolay olacağını ve üretici bireyler olma bilincinin küçük yaşlarda aşılması gerektiğini düşündüklerinden bahsetmişlerdir. Alanyazında anasınıfı düzeyinde kodlama eğitiminin verilmesini isteyenlerin gerekçelerine baktığımızda problem çözme becerisi, yaratıcı düşünme becerisi, döngü, koşul ve değişken gibi kavramların öğretiminin bu düzeydeki öğrencilere kazandırılabilceğini ifade ettikleri görülmektedir (Bell, Alexander, Freeman ve Grimley, 2009; Cortina, 2015). Alanyazında bahsedilenler çalışmanın katılımcılarının söyledikleri ile karşılaştırıldığında alanyazında anasınıfı düzeyinde öğrencilere kazandırılmak istenen beceriler için çalışmanın katılımcılarının bahsettikleri bu düzeydeki öğrencilerin algı düzeylerinin yüksek olması, okuma yazma bilmenin olmadığına birbirleri ile ilişkili ifadeler olduğu ancak alanyazında bahsedilen kazandırılmak istenen yeterliliklerden bahsetmedikleri görülmüştür.

Son olarak öğretmenlerin küçük bir kısmının kodlamaya başlangıç seviyesi olarak ortaokulu önerdiği görülmektedir. Kodlama eğitimine başlangıç için ortaokul düzeyindeki öğrencilerin uygun olduğundan bahseden araştırma katılımcıları kodlama konularının küçük yaşlardaki öğrenciler için soyut kaldığından bu nedenle soyut düşünme becerisine sahip olan ortaokul düzeyindeki öğrencilerin daha kolay algılayabileceklerinden ve öğrenebileceklerinden bahsedilmektedir. Çalışmanın katılımcılarının açıklamalarına bakıldığında bilgisayar biliminin çerçevesini oluşturan yeterliliklere göre hangi seviyede hangi konuların öğretilmesi gerektiğini düşünmeden kodlama eğitimine başlangıç yaşından bahsettikleri görülmektedir. Alanyazın incelendiğinde Macaristan, Malta ve İtalya gibi ülkelerde ortaokul seviyesinde kodlama öğretilmesine başlanıldığı görülmektedir. Alanyazında ortaokul seviyesinde kodlama eğitiminin verilmesini savunanların gerekçelerine bakıldığında ortaokul seviyesindeki

öğrencilerin de ilkokul seviyesindeki öğrenciler gibi bilgisayar teknolojisinin sağladığı olanakların ve sınırlılıkların daha iyi farkına varmış olacakları bu sayede öğrencilerin bilişim teknolojilerinin üretim amaçlı kullanımını öğrenmeleri, bilinçli karar verebilmeleri ve çevrim içi bilgilerin güvenilirliği konusunda fikir sahibi olmalarının sağlanabileceğinden bahsedilmektedir. Alanyazında ifade edilen gerekçeler ile çalışmanın katılımcılarının ortaokul seviyesinde kodlama öğretimi gerekçeleri karşılaştırıldığında, çalışmanın katılımcıları alanyazında bahsedilenlerden farklı olarak öğrencilerde soyut düşünme becerisine vurgu yaparak kodlama eğitiminin ortaokul düzeyinde olması gerektiğinden bahsetmektedir.

5.1.5. Kodlama için temel kavram ve beceriler

Bu kategori altında öğretmenlerin kodlama ile öğrencilere kazandırılması istenen kavram ve beceriler üzerinde durulmuştur. Çalışmanın katılımcılarının kodlama eğitimi ile öğrencilere kazandırmak istedikleri kavram ve beceriler ile ilgili açıklamaları genel anlamda alanyazına paralel olmakla birlikte katılımcıların yaptıkları açıklamalardaki temel kavramların tam olarak net olmadığı ve kapsam ve içerik açısından görece olarak yüzeysel olduğu görülmektedir.

Ülkelerin A-12 düzeyindeki öğretim programlarını kodlama eğitimi ile bütünleştirmek isteme gerekçeleri, öğrencilere kazandırmak istedikleri kavram, beceri ve yetenekler konusunda çeşitli farklılıklar ve benzerlikler göstermektedir (CSforAll, 2016; National Research Council, 1999; Sayın, 2017). Örneğin; Finlandiya mantıksal düşünme becerisi, problem çözme becerisi ve kodlama becerisini A-12 düzeyinde kazandırmayı isterken, Avusturya, İsrail, Polonya, Çek Cumhuriyeti ve Türkiye gibi ülkelerin mantıksal düşünme becerisi, problem çözme becerisi, kodlama becerileri ve diğer tüm yeterlilikleri kazandırmak amacıyla kodlama eğitimini öğretim programlarına ekledikleri görülmüştür (Balanskat ve Engelhardt, 2015; Tebliğler Dergisi, 2013; Tebliğler Dergisi, 2016).

Çalışmanın katılımcıları tarafından kodlama eğitimi ile öğrencilere kazandırmak istedikleri temel kavram ve beceriler konusunda problem çözme becerisi, algoritmik düşünme becerisi, analitik düşünme becerisi, farklı düşünme becerisi ve programlamanın temel kavramlarından bahsedilmektedir. Kodlama eğitimini öğretim programına eklerken öğrencilere problem çözme becerisi kazandırmayı amaçlayan ülkeler arasında Polonya, Fransa, Estonya, Bulgaristan, Finlandiya, Avusturya, Türkiye gibi ülkeler bulunmaktadır (Balanskat ve Engelhardt, 2015; Tebliğler Dergisi, 2013; Tebliğler Dergisi, 2016). Alanyazında problem çözme becerisi ile ilgili genel anlamda bilişsel işlemleri bir düzen içerisinde hedefe yöneltmek ve problem çözme sürecinde kişiden kişiye değişse de belirli aşamaların takip edilmesi gerektiğinden bahsedilmiştir (Anderson, 1980; D’Zurilla ve Goldfried, 1971; Morgan, 1999). Aynı zamanda kodlama eğitimi kategorisi altında problem çözenin hangi alanlar üzerinde çalıştığından da çalışmanın ikinci bölümünde bahsetmiştik. Öğretmenlerin problem çözme ile ilgili görüşlerine bakıldığında bir kısmı karşılaşılan problemlere farklı yollardan çözüm üretebilme, problemi kısa yoldan çözebilme gibi becerilerin kazandırılması gerektiğinden bahsederken diğer kısmı problem çözme becerisi ile ilgili herhangi bir açıklamada bulunmamıştır. Problem çözme alanyazını ile öğretmenlerin öğrencilere kavram ve beceri olarak problem çözme becerisi kazandırmak istemelerindeki açıklamaları karşılaştırıldığında görüşlerinin alanyazındaki birçok başlığı içermediği görülmektedir.

Kodlama eğitimini öğretim programına eklerken öğrencilere algoritmik düşünme becerisi kazandırmayı amaçlayan ülkelere İngiltere ve Türkiye örnek olarak verilebilir (Tebliğler Dergisi, 2016; National Curriculum, 2017). Alanyazında algoritmik düşünme becerisi ile ilgili genel anlamda problem çözme ya da durumları anlama yolunda ihtiyaç duyulan adımların tanımlanması, talimatlar, kurallar şeklinde tanımlamalar yapılmaktadır (Csizmadia ve diğerleri, 2015; Futschek, 2006, s. 160; Olsen, 2000). Kodlama eğitimi aracılığıyla öğrencilere kazandırılması gereken kavram ve beceri olarak algoritmik düşünme becerisinden bahseden öğretmenlerden yalnızca bir tanesi açıklamada bulunarak karşılaşılan bir durum karşısında izleyeceği yolu, adım adım yapacaklarını şemalaştırmak olarak algoritmik düşünme becerisini alanyazındaki tanımlamalara paralel bir şekilde açıklamıştır.

Kodlama eğitimini öğretim programına eklerken öğrencilere programlamanın temel kavramlarını öğretmeyi amaçlayan ülkeler arasında İsrail, Avusturya, Çek Cumhuriyeti ve Türkiye gibi ülkeler yer almaktadır. Programlamanın temel kavramları ile ilgili alanyazına bakıldığında sıralama, döngüler, koşul ifadeleri, matematiksel ve mantıksal işlemler ile algoritmalarından bahsedildiği görülmektedir (Gedik, Çetin ve Koca, 2017). Görüşme yapılan öğretmenlerin ise kodlama eğitimi aracılığıyla kazandırılması gereken kavram ve beceriler ile ilgili koşul ifadeleri, döngüler gibi programlamanın temel kavramlarının öğretilmesi gerektiğini ifade ederek alanyazında bahsedilenlerden kısıtlı olarak bahsettikleri görülmektedir.

İngiltere Eğitim Bakanlığı bilgi işleme dersinin amacı olarak analitik bir yaklaşımla sorunları çözmeyi öğretmekten bahsetmiştir (National Curriculum, 2017). Analitik düşünme becerisi ile ilgili alanyazına baktığımızda genel anlamda bir bütün analiz edilerek parçalara ayrıldığında parçaların kendi aralarında ve bütün ile ilişkisinin belirlenebilmesinden bahsedildiği görülmüştür (Elder ve Paul, 2007; Tsalapatas, Heidmann, Alimisi, Tsalapatas, Florou ve Houstis, 2011). Çalışmanın katılımcıları da kodlama eğitimi ile öğrencilere kazandırmak istedikleri beceri olarak analitik düşünme becerisinden bahsedilmektedir. Ancak katılımcılar açıklamalarında analitik düşünme becerisi ile neyi kast ettiklerini tam olarak belirtmemişlerdir.

Ülkelerin kodlama eğitimini A-12 düzeyindeki öğretim programları ile bütünleştirirken öğrencilere kazandırmak istedikleri kavram ve becerilerde yaratıcı projeler üretme, bilgi ve iletişim teknolojilerini yaratıcı kullanabilme gibi yaratıcı düşünmeyi içeren farklı yeterlilikleri ifade ettikleri görülmüştür (NAACE, 2014; Tebliğler Dergisi, 2016). Aynı zamanda yaratıcı düşünme becerisinin günümüz bireylerinin sahip olması gerektiği alanyazında birçok yazar tarafından desteklenmektedir (Koray vd., 2007; Seferoğlu ve Akbıyık, 2006; Tok ve Sevinç, 2010). Yaratıcı düşünme alanyazınına incelendiğinde çalışmalara farklı bir bakış açısı getirebilmek, yeni şeyler düşünmek veya önceden düşünülmüş olanı farklı yollardan düşünebilmek tanımlamaları yapıldığından çalışmanın ikinci bölümünde bahsetmiştik (Ercan, 2017; Lau, 2011; Yecan, Özçınar ve Tanyeri, 2017). Çalışmanın katılımcıları tarafından kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler

konusunda yaratıcı düşünme becerisinden farklı düşünme becerisi olarak bahsedildiği görülmüştür. Yaratıcı düşünme becerisi ile ilgili alanyazın ile çalışmanın katılımcılarının ifadeleri karşılaştırıldığında öğretmenlerin yaratıcılığı genel anlamda farklı düşünme becerisi olarak nitelendirdiği fakat alanyazındaki tanımlamalara göre kısıtlı kaldığı söylenebilir.

5.1.6. Programlama için yazılım donanım gereksinimi

Bilgisayar bilimi eğitiminde kullanılan programlama araçlarına bakıldığında çoğunlukla blok tabanlı uygulamalar, robot uygulamaları, metin tabanlı ortamlar ve bilgisayarsız bilgisayar bilimi öğretiminin tercih edildiği görülmektedir (Berry, 2014; Bers, Flannery, Kazakoff ve Sullivan, 2014; Nishida ve diğerleri, 2009; vanDam, 2000; Weinberg, 2013). Görsel programlama araçları ile ilgili alanyazın incelendiğinde kodlamaya yeni başlayanlar için Scratch, Alice, App Inventor gibi görsel araçlar arasında en çok tercih edilen blok tabanlı programlama araçlarının ilgi çekici ve eğlenceli ortamlar sunarak öğrenme kolaylığı sağladığı, deneyimli kullanıcıları cesaretlendirerek gelişmiş projeler oluşturmalarına olanak sağladığı, uygulama yaparken farklı yollardan ve farklı projeler oluşturma imkânı sağladığı ve bireyleri bir araya getirerek öğrenme toplulukları oluşturmaya olanak sağladığından bahsedilmektedir (Çetin ve Toluk Uçar, 2017; Schwartz, Stagner ve Morrison, 2006).

Çalışmanın katılımcılarının kodlama öğretimi için tercih ettikleri yazılım ve donanımlara bakıldığında blok tabanlı programlama araçlarının yaygın bir şekilde tercih edildiği görülmüştür. Çoğunlukla Scratch, Alice, Code.org ve Hacker.can gibi blok tabanlı programlama araçlarından bahseden katılımcıların bahsettikleri programlama araçlarını kullanma nedenlerine baktığımızda; oyun tabanlı öğrenme ortamında sunduğu imkânlardan dolayı öğrencilerin ilgilerini çektiği ve eğlenceli bulduklarını, öğrenci düzeylerine uygun bir şekilde kodlamayı anlatan programlama araçları olduğunu vurguladıkları görülmektedir. Aynı zamanda da sağladığı düşük zemin özelliği ile kolay ve güvenle uyum sağladıkları, yüksek tavan özelliği ile de karmaşık uygulamalar yapma imkânına sahip oldukları, öğretmenlerin programlama araçları sayesinde daha eğitici ve

öğretici oldukları, dünyanın her yerinde tercih edilen programlama araçları olması, kalıcı öğrenme sağlaması ve yazılımsal kolaylıklar ve avantajlar sağlamasından bahsedilmektedir.

Alanyazında bahsedilen görsel programlama araçlarının özellikleri ile katılımcıların görsel programlama araçlarını kullanma nedenleri karşılaştırıldığından katılımcıların alanyazında bahsedilen alçak zemin, yüksek tavan ve geniş duvarlara sahip yazılımlar olduğu, ilgi çekici ve eğlenceli ortamlar sunduğundan bahsedilmektedir. Aynı zamanda öğretmenlere sağladığı kolaylık, yazılımsal özelliklerinin sağladığı kolaylık ve avantajlar, kalıcı öğrenme sağlaması gibi özelliklere sahip olduğundan da bahsetmektedirler.

Çalışmanın katılımcılarından küçük bir kısmı robot programlamayı tercih ettiklerinden bahsetmiştir. Robot programlama ile ilgili alanyazına baktığımızda eğitsel robotların yaparak yaşayarak öğrenmeyi sağladığı, genel olarak gruplar halinde çalışma ortamı ile işbirlikçi öğrenmeyi desteklediği, çoklu bakış açısı sağladığı, eğlenceli ve ilgi çekici öğrenme ortamı sunduğu, soyut kavramların öğretiminde etkili bir araç olduğu, öğrencilerin yaratıcılıklarını, problem çözme ve kritik düşünme becerilerini geliştirmelerine yardımcı olduğundan bahsedilmektedir (Bers, 2008; Eguchi, 2014; Buss ve Gamboa, 2017; Üçgül, 2017). Ayrıca eğitsel robotlar ile uygulamalar yaparken öğrencilerin, mantıksal sorgulama, algoritmik düşünme, soyutlama, genelleme, ayırıştırma ve değerlendirme gibi (Catlin ve Wollard, 2014; Csizmadia ve diğerleri, 2015; Eguchi, 2016; Üçgül, 2017), birtakım bilgi işlemsel düşünme becerilerini kullandıklarından da bahsedilmektedir. Araştırmanın katılımcıları Arduino, Code.bug, Dash & Dot robot gibi eğitsel robot programlama araçlarından bahsederken öğrencilere yapılan programlamanın sonucunu somut olarak görebilme imkanı sağladığı aynı zamanda öğrencilerin de bir şeyler üretebileceklerini deneyimledikleri için kullanmayı tercih ettiklerini dile getirmişlerdir. Alanyazında eğitsel robot etkinliklerinin sunduğu olanaklara, öğrencilere kazandırdığı becerilere bakıldığında araştırma katılımcılarının robot programlamanın eğitim ortamlarında sunduğu imkânlar konusunda kısıtlı şeyler söylediği görülmektedir.

Çalışmanın katılımcılarından çok küçük bir kısmının metin tabanlı programları tercih ettikleri görülmektedir. Metin tabanlı programlama ortamları ile ilgili alanyazın incelendiğinde bu ortamların programlama mantığını bilmeyi ve algoritmik düşünme sürecine hâkim olmayı gerektirdiği bu nedenle de daha soyut bir süreci ifade ettiği görülmüştür (Kalelioğlu ve Keskinılıç, 2017; Kandemir, 2017). Erken yaşta programlama öğretimi için Logo, Small Basic ve Python gibi blok tabanlı programlama ortamlarına benzer, söz dizimi kurallarının basitliği ile kolaylık sağlayan metin tabanlı programlama ortamlarının kullanımının uygun olacağından bahsedilmiştir (Berry, 2014). Araştırma katılımcıları Visual Basic, PHP, Asp gibi metin tabanlı programlama ortamlarının öğrencilere programlama öğretiminde kullanılabilirliğinden bahsetmişler. Ancak programlama öğretiminde neden bu araçları kullanmak istedikleri hakkında yorumda bulunmamışlardır. Araştırma katılımcılarının bahsettikleri programlama araçlarına bakıldığında Logo, Small Basic ve Python gibi küçük yaşlardaki öğrenciler için daha uygun olabilecek metin tabanlı programlama ortamlarından bahsetmedikleri görülmüştür.

Bilgisayar bilimi öğretiminde kullanılan yaklaşımlardan bilgisayarsız bilgisayar bilimi etkinlikleri ile ilgili alanyazına bakıldığında B³ etkinliklerinin küçük yaşta bilgisayar bilimini öğrenenlere kavramsal süreçleri somutlaştırmada uygun bir seçenek olduğu ve teknolojik alt yapı sorunu yaşayanlar için fırsat eşitliği sunduğu söylenebilir (Kalelioğlu, 2017; Nishida ve diğerleri, 2009). Çalışmanın bulgularına bakıldığında araştırma katılımcılarının bilgisayarsız bilgisayar bilimi öğretimi konusunda çoğunlukla fikir sahibi olmadıkları görülmüştür. Yalnızca bir katılımcı B³ etkinliklerinden olan Legoların kodlama eğitimi ile öğrencilere kazandırılmak istenen kavram ve beceriler için kullanılabilirliğinden bahsetmiştir. Legoları kullanma nedeni olarak da kavram ve becerilerin öğretiminde, minimum malzeme kullanarak bütün ülkedeki öğrencilerin faydalanabileceği kaynaklar kullanılması gerektiğini ifade ederek, aslında alanyazında bahsedilen fırsat eşitliğinin sağlanması gerektiğinden bahsetmiştir.

Kodlama için yazılım donanım gereksinimi kategorisi altında öğretmenlerin kodlama eğitimi için kullanılması gereken donanım ve yazılım araçları hakkındaki görüşleri incelenmiştir. Bu araştırmanın katılımcıları arasında kodlama için kullanılması

gereken araçlar içerisinde birinci sırayı ezici bir çoğunlukla blok tabanlı görsel programlama ortamları almıştır. Bunu sırasıyla robotlar, 3B yazılımlar ve metin tabanlı ortamlar takip etmiştir. Görsel kodlama ortamlarının birinci sırada bulunması bu ortamların görece olarak ucuz olması, erişimin kolay olması, popüler olması, sağladığı kodlama olanakları göz önünde bulundurulduğunda doğal olarak algılanabilir. Fakat öğretmenlerin robotlar, metin tabanlı ortamlar veya bilgisayarsız bilgisayar biliminden çok sınırlı olarak bahsetmesi dikkate değer bir bulgudur. Bunların arasından bilgisayarsız bilgisayar biliminin öğretmenler tarafından bir donanım veya yazılım olarak algılanmadığı düşünülebilir fakat bir sonraki kategori olan öğretmenlerin bilgisayarsız bilgisayar bilimi hakkındaki görüşlerinden de anlaşılacağı üzere öğretmenlerin büyük bir çoğunluğu bilgisayarsız bilgisayar biliminden haberdar değildir. Benzer şekilde öğretmenlerin bir kısmının tercih etmedikleri için robotlar ve metin tabanlı ortamlardan bahsetmedikleri düşünülebilir. Fakat görüşmeler esnasında onlara “Başka ne gibi araçlar kullanırsınız?” gibi sorular vasıtasıyla fırsatlar verildiği halde robotlar ve metin tabanlı ortamlardan nadiren bahsetmeleri bu ihtimali zayıflatır. Ayrıca öğretmenlerin Logo, Small Basic ve Python gibi öğrenciler için uygun olabilecek metin tabanlı ortamlar yerine Visual Basic, PHP ve Asp gibi dillerden bahsetmesi onların eğitimsel olarak kullanılan metin tabanlı ortamlardan tam olarak haberdar olmadığına dair bir göstergedir. Sonuç itibarıyla bu çalışmaya katılan öğretmenlerin genel olarak görsel programlama ortamlarından ve kısmi olarak eğitimsel robotlardan haberdar olduğu söylenebilir. Fakat aynı şeyi bilgisayarsız bilgisayar bilimi ve metin tabanlı ortamlar için söylemek güçtür.

5.1.7. Öğretmenlerin bilgisayarsız bilgisayar bilimi hakkındaki görüşleri

Bilgisayar bilimi kavramları ile bilgi işlemsel düşünme kavram ve yaklaşımlarını bilgisayar desteği olmadan ve elektronik cihazlar kullanmadan öğretme şekline bilgisayarsız bilgisayar bilimi (B³) öğretim etkinlikleri denilmektedir (Kalelioğlu, 2017). B³ etkinlikleri öğrencilere fiziksel oyun etkinlikleri ya da farklı materyallerle, bilgisayar bilimi kavramları ile bilgi işlemsel düşünme kavram ve yaklaşımlarını, sınıf içinde veya farklı bir ortamda öğrenme fırsatı sağlayan etkinlikler

olarak bilinmektedir Eğlenceli bir öğrenme ortamı sağlayan B³ etkinlikleri öğrencilerin keşfederek öğrenmesine olanak sağlamaktadır.

Yazılım ve donanımları kullanmadan kodlama eğitimindeki kavram ve becerilerin öğretilmesi konusunda araştırmanın katılımcılarının düşünceleri bulgular kısmında B³'ten bahsedenler (3 öğretmen), B³'ten bahsetmeyenler (15 öğretmen), informal B³ (4 öğretmen), olumsuz bakanlar (15 öğretmen) ve olumlu bakanlar (4 öğretmen) başlıklarında bir araya gelmiştir. B³ etkinliklerinden bahseden öğretmenlerin bir kısmı aldıkları eğitimler, seminerler sonucunda bir kısmı ise herhangi bir eğitim almadan çevre faktörü ile bir şekilde B³ etkinliklerinden haberdar olduklarını dile getirmiştir. Bu öğretmenlerin B³ etkinliklerinden olan oyun kartları hazırlama, kâğıt üzerinde zekâ oyunları, kare boyama, algoritma ile ilgili yön buldurma, define avı ve bilgisayarsız code.org gibi etkinliklerin yapılabileceğinden bahsettikleri görülmüştür. Ancak öğretmenlerin büyük bir çoğunluğunun bilgisayarsız bilgisayar bilimi öğretiminden haberdar olmadıkları, herhangi bir yazılım ve donanım kullanmadan kodlama eğitimi ile ilgili kavram ve becerilerin kazandırılmasının mümkün olmayacağını, algoritma ve akış diyagramı öğretiminden ileri gidilemeyeceği ve genel olarak hazır programlar kullandıkları için bilgisayarsız bir ortamda kodlama eğitiminin nasıl yapılacağı hakkında bilgilerinin olmadığını dile getiren ifadelerinden anlaşılmaktadır.

Görüşme yapılan öğretmenlerden B³ etkinlikleri hakkında olumlu düşünceye sahip olanlar ifadelerinde, kodlama eğitim ortamında yazılım donanım kullanmadan yapılabilecek aktivitelerin olduğu ancak bu aktivitelerin neler olduğu konusunda yeterli bilgiye sahip olmadıklarından bahsedilmektedir. Öğretmenlerden B³ etkinlikleri konusunda olumsuz düşünceye sahip olanlar ise B³ etkinlikleri ile yapılacak olan eğitimin kalıcı, etkili ve faydalı olmayacağını, öğrencilerin sıkılacağını aynı zamanda kodlama eğitiminde uygulama ve görsellik olması gerektiğini ifade etmektedirler. Görüşme yapılan öğretmenlerden B³ etkinliklerine yönelik olumlu düşünceye sahip olanların alanyazında bahsedilen B³ etkinliklerinin öğrencilere sağladığı katılardan hiç bahsetmedikleri fakat B³ etkinlikleri hakkında bilgisi olan öğretmenlerin neden B³ etkinliklerinin faydalı olmayacağı hakkında düşüncelerini belirttikleri görülmektedir.

Alanyazın ile öğretmen görüşleri karşılaştırıldığında alanyazının genel itibariyle öğretmenlerin görüşlerinin aksi yönünde olduğu söylenebilir. Alanyazına göre bilgisayarlı bilgisayar bilimi etkinlikleri küçük yaşta öğrenenlerin programlama kavramlarının amacını ve önemini daha iyi anlamaları için fiziksel olarak bir problem çözümünde yer alma, çözümü alt problemlere ayırma, çözümü sıralı bir şekilde açıklayabilme, yönergeyi takip etme, değişken, döngü, koşul gibi kavramları bilgisayarı kullanmadan eğlenceli bir şekilde öğrenebilirler (Bell, Alexander, Freeman ve Grimley, 2008). B³ etkinliklerinin öğrencilere sağladığı olumlu katkılara bakıldığında B³ etkinlikleri ile grup içerisinde arkadaşları ile birlikte çalışmak, birlikte düşünerek çözümler üretmek, gözlem yoluyla ve deneyimleyerek öğrenmek ve yaratıcılıklarını geliştirmek gibi becerilere sahip olmaları sağlanmış olur (Bocconi ve diğerleri, 2016; Cortina, 2015). Ayrıca yapılan bazı çalışmalar B³ etkinliklerinin öğrencilerin hayal gücü ve düşünme becerilerini geliştirmede katkı sağladığı, bilgisayar bilimine olan tutumlarının olumlu yönde geliştiği, matematik ile ilgili kendilerine olan güvenin arttırmada yardımcı olduğu görülmüştür (Mano ve diğerleri, 2010; Nishida ve diğerleri, 2008; Taub ve diğerleri, 2009; Taub ve diğerleri, 2012; Thies ve Vahrenhold, 2013). Ayrıca B³ etkinlikleri teknolojik alt yapı sorunu yaşayan okullarımızdaki öğretmenlere, bilgisayar bilimi ve bilgi işlemsel düşünme kavram ve yaklaşımlarını öğretme fırsatı sunmaktadır (Kalelioğlu, 2017).

5.1.8. Öğretmen eksiklikleri ve eksikliklerin giderilmesi için yapılması gerekenler

Öğretmen yeterliliğini teknoloji, pedagoji ve alan bilgisi olarak üç boyutta incelendiğinde alan bilgisi boyutunda öğretmenlerin; dünya üzerinde bilgisayar biliminin rolü ve etkisi ile dijital aygıtları, ağları ve sistemleri anlama, veri gösterimi ve soyutlama yapabilme, algoritma tasarlama, geliştirme ve test etme konularına hâkim olmaları beklenilmektedir (ISTE, 2011). Pedagojik boyutta öğretmen yeterlilikleri ile ilgili alanyazına bakıldığında; öğretim yapılacak hedef kitlenin özelliklerini tanıma, öğretme öğrenme sürecinde proje temelli ve işbirliğine dayalı öğretim yöntem ve tekniklerini kullanma, uygulama ve değerlendirmede uygun stratejileri belirleme, kullanabilme ve geliştirebilme ve sınıf yönetimi konularından bahsedildiği görülmüştür

(ISTE, 2011; Koehler ve Mishra, 2008). Teknolojik boyutta öğretmen yeterliliği ile ilgili alanyazına bakıldığında ise; teknolojik yeniliklere açık olmak, yeni nesil teknolojileri bilme ve kullanma becerisine sahip olmak, yazılım ve donanımlar hakkında bilgi ve becerilere hâkim olmaktan bahsedilmektedir (Karataş, 2014; Koehler ve Mishra, 2008; Ozgun-Koca ve diğerleri, 2010).

Çalışmanın katılımcıları bilişim teknolojileri ve yazılım dersi öğretmenlerinin alan bilgisi ve pedagojik alan bilgisi yönünde eksikliklerinin olduğunu dile getirmişlerdir. Çalışmanın katılımcıları alan bilgisi konusunda öğretmen eksikliklerine farklı nedenlerden bahsedilmektedir. Öğretmenlerin kodlama / programlama konusunda lisans döneminde yeterli bilgi sahibi olamadıkları ya da öğrendikleri bilgileri zamanla unutmuş olabileceklerinden bahsedilmektedir. Aynı zamanda birçok öğretmenin kendisini geliştirmediği için de yetersiz kaldığı ve kodlama konusunda birçok yönden eksikliklerinin olabileceğini de dile getirmektedirler Buna ek olarak, çalışmanın katılımcılarının bir kısmı ortaöğretim öğrencilerine öğretilen kodlama ile lisans döneminde alınan kodlama bilgisinin farklı olduğundan ve öğretmenlerin bu konuda kendilerini sürekli yenilemeleri gerektiği halde kendilerini geliştirme noktasında da yetersiz kalabildiklerinden bahsedilmektedir. Son olarak araştırmanın katılımcıları lisans döneminde öğrendikleri kodlama bilgisini ortaöğretim öğrencilerine nasıl aktaracakları konusunda eğitim almadıklarından bu sebeple de sıkıntı yaşadıklarından bahsedilmektedir.

Birçok ülkede öğretmenlerin öğretmen eğitimi konusunda alan bilgisi, pedagojik ve teknolojik açılardan kendilerini geliştirmelerine yönelik hizmet içi eğitimler, dernek faaliyetleri, çevrimiçi öğretim, çalıştaylar, özel ve devlet kuruluşları, öğretmenler birliği ve özel girişimler gibi çeşitli yollarla olanak sağlanmaktadır (Balannskat ve Engelhardt, 2015; Bocconi ve diğerleri, 2016). Örneğin; Finlandiya’da öğretmenlere üniversite ve yerel yönetimler tarafından sağlanan hizmet içi eğitimler dışında yine üniversitelerin desteği ya da dernekler tarafından projeler yapılmaktadır. İrlanda’ya baktığımızda dönemlik veya yaz döneminde mesleki gelişim kurslarının ilk ve ortaöğretim öğretmenleri için sağlandığı görülmektedir. Portekiz ise proje, çalıştay veya etkinlikleri düzenlemektedir (Karadeniz, 2017). Çalışmanın katılımcılarından alan

bilgisi ve üniversitelerdeki pedagojik alan bilgisi konularında farklı sebeplerle eksikliklerin olduğunu dile getirenler bu eksikliklerin giderilmesine yönelik olarak birtakım önerilerde bulunmuşlardır. Öğretmen eksikliklerinin giderilmesi için neler yapılabileceğine dair çalışma katılımcılarının önerilerine bakacak olursak; öğretmenler için eğitimler düzenlemek yerine her öğretmenin bireysel olarak kendilerini eksik gördükleri yerler konusunda geliştirmeleri gerektiği ve üniversite eğitimi müfredatlarında ve akademisyen kadrosunda yapılacak değişiklikler ile alandan mezun olan öğretmen adaylarının kodlama becerisi ve öğrenci düzeyine göre anlatımı konusunda donanımlı yetişmesinin sağlanmasının gerektiğini ifade ettikleri görülmektedir. Bunun yanında seminer, çalıştay, atölye ve hizmet içi eğitimler verilmesi gerektiğinden bahseden öğretmenlerin önerilerine baktığımızda; yapılan atölye çalışmalarının öğretmenlerin kendileri hakkında öz değerlendirme yaparak kodlama öğretiminde nerede olduklarını fark etmelerini sağladığı için verimli olduğundan, aynı zamanda seminerler, çalıştaylar ve çevrim içi eğitimler ile de öğretmenlerin kodlama bilgisi ya da öğrenciye öğretimi esnasında yapılacak faaliyetler, öğretim yöntem ve teknikleri gibi konularda eksikliklerinin giderilebileceğinden bahsedilmektedir.

5.2. Öneriler

Bu bölümde yer alan, çalışmanın bulgularına ve yorumlara dayanılarak sunulan önerilerin alanyazına ve bundan sonra yapılacak olan çalışmalara katkı sağlayacağı düşünülmektedir.

1. Üniversitenin BÖTE bölümü öğretim programlarına Eğitsel Yazılımların Seçimi ve Değerlendirilmesi, Bilgisayar Eğitiminde Öğretim Metotları, Algoritma Tasarımı ve Geliştirme gibi çeşitli dersler eklenmiştir. Öğretim programlarına eklenen bu dersler ikinci ve dördüncü. yarıyıl gibi dönemlerde verilmektedir. Oysa öğretim programındaki bu dersler alan derslerinin ardından verildiğinde, yeni yetişecek olan öğretmen adaylarının tekno pedagojik alan bilgileri açısından daha etkili ve verimli olabilir.

2. Hâlihazırdaki öğretmenlerin ihtiyaç ve isteklerine yönelik hizmet içi eğitimler, çalıştaylar, atölye faaliyetleri ve seminerler düzenlenerek kodlama eğitimi ve öğretimi konusunda öğretmenlerin teknolojik, pedagojik ve alan bilgisi gibi yetersiz kaldıklarını düşündükleri alanlarda kendilerini geliştirmelerine katkı sağlanabilir.
3. Kodlama eğitimine küçük yaşlarda başlanıldığı ve kodlama eğitiminin STEAM gibi disiplinler arası yaklaşımla farklı alanlarla birlikte verildiği günümüzde her öğretmen bilinçli olmalıdır. Bu bağlamda kodlama eğitiminin, eğitim fakülteleri gibi öğretmen yetiştirme kurumlarında bütün alanlardaki öğretmen adaylarına yönelik seçmeli ders olarak öğretim programlarında yer verilmesi ya da atölyeler gibi eğitsel etkinlikler ile verilmesi faydalı olabilir.
4. Öğretmenlerin kodlama eğitimi ve pedagojik gelişimleri konusunda düzenli olarak kendilerini geliştirmeleri ve yenilemeleri alana hâkim olmalarını ve öğrenci düzeyine uygun öğretim yöntem ve tekniklerini kullanma, uygulama ve değerlendirme esnasında faydalı olabilir.
5. Bu çalışma ile öğretmenlerin kodlama eğitimine yönelik düşünce ve görüşleri ortaya çıkarılmaya çalışılmıştır. Yapılacak diğer çalışmalarda bu çalışma içerisinde sorgulanmayan, öğretmenlerin alan bilgisine ne kadar hâkim oldukları ve kendilerini alana yönelik olarak hangi konularda geliştirmek istediklerine yönelik çalışmalar başka araştırmacılar tarafından yapılabilir.

KAYNAKÇA

- Australian Curriculum Assessment and Reporting Authority. (2012). Australian Curriculum: English. Version 3.0 Sydney: Australian Curriculum Assessment and Reporting Authority. 5 Aralık 2018 tarihinde www.acara.edu.au sitesinden alınmıştır.
- Akbulut, A. Y., ve Looney, C. A. (2009). Improving IS student enrollments: Understanding the effects of IT sophistication in introductory IS courses. *Journal of Information Technology Education*, 8, 87-100.
- Akçay, T. (2009). *Perceptions of students and teachers about the use of a kid's programming language in computer courses*. Yayınlanmamış yüksek lisans tezi, Orta Doğu Teknik Üniversitesi, Ankara.
- Anderson, J. R. (1980). *Cognitive Psychology and It's Implications*, San Fransisco: Freeman.
- 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. DOI: 10.1145/1929887.1929905.
- Baker, M., Rudd, R., ve Pomeroy, C. (2001). Relationships between critical and creative thinking. *Journal of Southern Agricultural Education Research*, 51(1), 173-188.
- Balanskat, A., ve Engelhardt, K. (2015). All You Need is Code Gets Europeans Coding. 8 Ağustos 2018 tarihinde <http://www.eun.org/news/detail?articleId=655141> sitesinden alınmıştır.

- Başer, M. (2013). Bilgisayar programlamaya karşı tutum ölçeği geliştirme çalışması. *The Journal of Academic Social Science Studies*, 6(6), 199-215.
- Bell, T., Andreae, P., ve Lambert, L. (2010). Computer science in New Zealand high schools. *Proceedings of the Twelfth Australasian Conference on Computing Education*, 103 (ss. 15-22). Australian Computer Society, Inc.
- Bell, T. (2014). Ubiquity Symposium: The science in computer science: unplugging computer science to find the science. *Ubiquity*, 3. DOI: 10.1145/2590528.2590531.
- Bell, T., Alexander, J., Freeman, I. ve Grimley, M. (2008). Computer science without computers: new outreach methods from old tricks. In *Proceedings of the 21st Annual Conference of the National Advisory Committee on Computing Qualifications*.
- Bell, T., Alexander, J., Freeman, I., ve Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13(1), 20-29.
- Bers, M. U. (2008). Using robotic manipulatives to develop technological fluency in early childhood. On N. Saracho and B. Spodek (Ed.), *Contemporary Perspectives on Science Technology in Early Childhood Education* (105-225). Charlotte, NC: Information Age Publishing, In.
- Bers, M. U., Flannery, L., Kazakoff, E. R., ve Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157.
- Berry, M. (2014). Text based programming in primary schools? 6 Haziran 2018 tarihinde <http://milesberry.net/2014/02/text-based-programming-in-primary-schools/> sitesinden alınmıştır.

- 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, 1(25)*. Vancouver, Canada.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., ve Punie, Y. (2016). *Developing computational thinking in compulsory education*. European Commission, JRC Science for Policy Report.
- Bower, M., ve Falkner, K. (2015). Computational Thinking, the Notional Machine, Pre-service Teachers, and Research Opportunities. *Proceedings of the 17th Australasian Computing Education Conference (ACE 2015), 27(30)*.
- Bogdan, R. ve Biklen, S. K. (1992). *Qualitative Research for Education: an Introduction to Theory and Method*. (5. Baskı). Boston: Allyn and Bacon Inc.
- Buss, A., ve Gamboa, R. (2017). Teacher transformations in developing computational thinking: Gaming and robotics use in after-school settings. *Emerging research, practice, and policy on computational thinking* (189-203). Springer, Cham.
- Büyüköztürk, Ş., Çakmak, E. K., Akgün, Ö. E., Karadeniz, Ş. ve Demirel, F.(2011). *Bilimsel araştırma yöntemleri* (6. Baskı). Ankara: Pegem A Yayıncılık.
- Byrne, P., ve Lyons, G. (2001). The effect of student attributes on success in programming. *ACM SIGCSE Bulletin*. 33(3), 49-52. ACM. DOI: 10.1145/507758.377467.
- Catlin, D. (2016). 29 Effective Ways You Can Use Robots in the Classroom. *International Conference EduRobotics 2016* (ss. 135-148). Springer, Cham.

- Celaya, D. (2016). Teaching, Learning and Assessing. 13 Nisan 2018 tarihinde <https://djcelaya.wordpress.com/2016/04/28/teaching-learning-and-assessing/> sitesinden alınmıştır.
- Cetin, I. (2016). Preservice Teachers' Introduction to Computing: Exploring Utilization of Scratch. *Journal of Educational Computing Research*, 54(7), 997-1021.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., ve Woollard, J. (2015). Computational thinking-A guide for teachers. 1 Nisan 2018 tarihinde <https://community.computingschool.org.uk/resources/2324/single> adresinden alınmıştır.
- CSTA (Computer Science Teachers Association). (2011). CSTA K-12 computer science standards, Revised, 2011, CSTA. 17 Aralık 2017 tarihinde http://scratch.ttu.ee/failid/CSTA_K-12_CSS.pdf sitesinden alınmıştır.
- CSTA (Computer Science Teachers Association). (2016). K-12 Computer Science Standards. 1 Haziran 2017 tarihinde https://www.csteachers.org/page/CSTA_Standards sitesinden alınmıştır.
- Computing at School ve NAACE. (2014). Computing in the national curriculum: A guide for secondary teachers. 10 Ağustos 2018 tarihinde https://www.computingschool.org.uk/data/uploads/cas_secondary.pdf sitesinden alınmıştır.
- Cortina, T. J. (2015). Reaching a broader population of students through unplugged activities. *Communications of the ACM*, 58(3), 25-27.
- Cotabish, A., Dailey, D., Robinson, A., ve Hughes, G. (2013). The effects of a STEM intervention on elementary students' science knowledge and skills. *School Science and Mathematics*, 113(5), 215-226.

- Cox, S. (2008). *A conceptual analysis of technological pedagogical content knowledge*. Yayınlanmamış Doktora Tezi, Brigham Young University, Provo, UT.
- Creswell, J. W. ve Clark, V. L. P. (2007). *Designing and conducting mixed methods research*. Thousand Oaks, CA:Sage.
- CSforAll. (2016). Computer Science For All. 8 Ağustos 2018 tarihinde <https://goo.gl/MH1Ytx> sitesinden alınmıştır.
- Çetin, I. (2013). Visualization: A tool for enhancing students' concept images of basic object-oriented concepts. *Computer Science Education*, 23(1), 1–23.
- Çetin, İ. (2016). Preservice teachers' introduction to computing: exploring utilization of scratch. *Journal of Educational Computing Research*. 0(0), 1-25, DOI: 10.1177/0735633116642774.
- Çetin, İ., ve Andrews-Larson, C. (2016). Learning sorting algorithms through visualization construction. *Computer Science Education*. 26(1), 27-43, DOI: 10.1080/08993408.2016.1160664.
- Çetin, İ., ve Toluk Uçar, Z. (2017). Bilgi işlemsel düşünme tanımı ve kapsamı. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya* (41-78). Ankara: Pegem Akademi.
- Denner, J., Werner, L., ve Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, 58(1), 240–249.
- Demirci, S. (2007). Programlama Paradigmaları. 15 Nisan 2018 tarihinde <https://e-bergi.com/y/programlama-paradigmaları/> sitesinden alınmıştır.
- Denning, P. J. (2007). Computing is a natural science. *Communications of the ACM*, 50(7), 13-18.

- Denning, P. J. (2011). Ubiquity symposium: What have we said about computation?: closing statement. *Ubiquity*, 1-7.
- Du Boulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, 2(1), 57-73.
- D'Zurilla, T. J., ve Goldfried, M. R. (1971). Problem solving and behavior modification. *Journal of Abnormal Psychology*, 78(1), 107-126.
- Eckerdal, A., ve Thune, M. (2005). Novice Java programmers' conceptions of object and class, and variation theory. *ACM SIGCSE Bulletin* 37(3), 89-93. New York, NY: ACM Press. DOI: 10.1145/1151954.1067473.
- Eguchi, A. (2014). Learning experience through RoboCupJunior: Promoting STEM education and 21st century skills with robotics competition. *Society for Information Technology & Teacher Education International Conference* (ss. 87-93). Association for the Advancement of Computing in Education (AACE).
- Eguchi, A. (2016). Computational thinking with educational robotics. *Society for Information Technology & Teacher Education International Conference* (ss. 79-84). Association for the Advancement of Computing in Education (AACE).
- Elder, L., ve Paul, R. (2007). The thinker's guide to analytic thinking: How to take thinking apart and what to look for when you do: the elements of thinking and the standards they must meet. Dillon Beach, CA: Foundation Critical Thinking. 12 Ağustos 2018 tarihinde <http://www.criticalthinking.org/> sitesinden alınmıştır.
- Ersoy, H. (2016). Durum çalışması. M. Y. Özden ve L. Durdu (Ed.). *Nitel Araştırma Yöntemleri* (3-18). Ankara: Anı Yayıncılık.
- Ersoy, A., ve Yalçınoğlu, P. (Ed.). (2013). *Nitel Araştırmaya Giriş*. (5. Baskı). Ankara: Anı Yayıncılık.

- Feurzeig, W., ve Papert, S. A. (2011). Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments*, 19(5), 487-501.
- Fleury, A.E. (2000). Programming in java: Student constructed rules. *ACM SIGCSE Bulletin*, 32(1), 197-201. ACM.
- Frorer, P., Manes, M., ve Hazzan, O. (1997). Revealing the faces of abstraction. *International Journal of Computers for Mathematical Learning*, 2(3), 217-228.
- Fullan, M., ve Langworthy, M. (2014). *A rich seam: How new pedagogies find deep learning*. London: Pearson.
- Gedik, N., Çetin, M., ve Koca, C. (2017). Examining the Experiences of Preschoolers on Programming via Tablet Computers Okul Öncesi Çocuklarının Tabletlerle Programlama Deneyimleri. *Mediterranean Journal of Humanities*, 7(1), 193-203, DOI: 10.13114/MJH.2017.330
- Graham, C.R., Burgoyne, N., Pamela Cantrell, Smith L., Clair L.S. ve Harris R. (2009). TPACK Development in Science Teaching: Measuring the TPACK Confidence of Inservice Science Teachers. *Tech Trends*, 53 (5), 70-79.
- Grigorenko, E. L., Jarvin, L., ve Sternberg, R. J. (2002). School-based tests of the triarchic theory of intelligence: Three settings, three samples, three syllabi. *Contemporary educational psychology*, 27(2), 167-208.
- Göl, M. (2016). *Yönetim Bilimi Açısından Eğitim Örgütlerindeki Öğretmenlerin Teknolojik Pedagojik Alan Bilgilerinin Araştırılması*. Yayınlanmamış Yüksek Lisans Tezi. İstanbul Gelişim Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul.
- Guzdial, M. (2008). Education: Paving the way for computational thinking. *Communications of the ACM*, 51(8), 25-27.

- Guzdial, M. (2012). A nice definition of computational thinking, including risks and cyber-security. Computing Education Blog. 18 Aralık 2016 tarihinde <https://computinged.wordpress.com/2012/04/06/a-nice-definition-of-computational-thinking-including-risks-and-cyber-security/> sitesinden alınmıştır.
- Gülbahar, Y. (Ed.). (2017). *Bilgi İşlemsel Düşünmeden Programlamaya* (1. Baskı). Ankara: Pegem Akademi.
- Harris, J., Mishra, P., ve Koehler, M. (2009). Teachers' technological pedagogical content knowledge and learning activity types: Curriculum-based technology integration reframed. *Journal of Research on Technology in Education*, 41(4), 393-416.
- Huang, W., Deng, Z., ve Rongsheng, D. (2009). Programming courses teaching method for ability enhancement of computational thinking. *Computer Science and Information Technology-Spring Conference, 2009. IACSITSC'09. International Association of* (ss. 182-185). IEEE.
- Hubwieser, P. (2012). Computer Science Education in Secondary Schools--The Introduction of a New Compulsory Subject. *ACM Transactions on Computing Education (TOCE)*, 12(4), 16. DOI: 10.1145/2382564.2382568
- İmal, N. ve Eser, M. (2009). *Programlama dili öğrenmedeki zorluklar ve çözüm yaklaşımları*. (2009). Sözlü bildiri, Elektrik Elektronik Bilgisayar Biyomedikal Mühendislikleri Eğitimi IV. Ulusal Sempozyumu, Eskişehir Osmangazi Üniversitesi Kongre ve Kültür Merkezi, Eskişehir, Türkiye.
- Jonassen, D. H. (1995). Computers as cognitive tools: Learning with technology not from technology. *Journal of Computing in Higher Education*, 6(2), 40-73.
- Jonassen, D. H. (2000). *Computers as mindtools for schools: Engaging critical thinking*. Prentice Hall.

- Kafai, Y. B., Burke, Q. ve Resnick, M. (2014). Connected code: Why children need to learn programming. 19 Mart 2017 tarihinde <https://mitpress.mit.edu/connected-code> sitesinden alınmıştır.
- Kalelioğlu, F. ve Gülbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education*, 13(1), 33–50.
- Kalelioğlu, F. (2017). Bilgisayarsız bilgisayar bilimi (B³) öğretimi. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya* (183-206). Ankara: Pegem Akademi
- Kalelioğlu, F., ve Keskinılıç, F. (2017). Bilgisayar bilimi eğitimi için öğretim yöntemleri. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya* (155-182). Ankara: Pegem Akademi.
- Kandemir, C. M. (2017). Metin tabanlı programlama. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya* (267-294). Ankara: Pegem Akademi.
- Karadeniz, Ş. (2017). Bilişim için öğretmen eğitimi. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya* (341-357). Ankara: Pegem Akademi.
- Karataş, A. (2014). *Lise Öğretmenlerinin Fatih Projesi'ni Uygulamaya Yönelik Teknolojik Pedagojik Alan Bilgisi Yeterliliklerinin İncelenmesi: Adıyaman İli Örneği*. Yayımlanmamış Yüksek Lisans Tezi. Sakarya Üniversitesi, Eğitim Bilimleri Enstitüsü, Sakarya.
- Kelleher, C., Pausch, R., ve Kiesler, S. (2007). Storytelling alicemotivates middle school girls to learn computer programming. Proceedings of the SIGCHI conference on Human factors in computing systems (ss. 1455-1464). ACM.

- Kinnunen, P., ve Malmi, L. (2008). CS minors in a CS1 course. *Proceedings of the Fourth international Workshop on Computing Education Research*, 79-90. ACM.
- Koray, Ö., Köksal, M. S., Özdemir, M., ve Presley, A. İ. (2007). Yaratıcı ve eleştirel düşünme temelli fen laboratuvarı uygulamalarının akademik başarı ve bilimsel süreç becerileri üzerine etkisi. *İlköğretim Online*, 6(3).
- Korkmaz, Ö. (2016). The effect of scratch- and lego mindstorms Ev3-based programming activities on academic achievement, problem-solving skills and logical-mathematical thinking skills of students. *Malaysian Online Journal of Educational Sciences*, 4(3), 73-88.
- 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.
- Korkmaz, Ö., Çakır, R., Özden, M. Y., Oluk, A., ve Sarıoğlu, S. (2015). Bireylerin bilgisayarca düşünme becerilerinin farklı değişkenler açısından incelenmesi. *Ondokuz Mayıs Üniversitesi Eğitim Fakültesi Dergisi*, 34(2), 68-87.
- Koehler, M. J., ve Mishra, P. (2005). What happens when teachers design educational technology? The development of technological pedagogical content knowledge. *Journal of educational computing research*, 32(2), 131-152.
- Koehler, M. J., ve Mishra, P. (2008). Introducing TPCK. J. A. Colbert, K. E. Boyd, K. A. Clark, S. Guan, J. B. Harris, M. A. Kelly ve A. D. Thompson (Ed.), *Handbook of technological pedagogical content knowledge for educators* (ss. 1-29). New York: Routledge.

- Kölling, M. (2015). Lessons from the Design of Three Educational Programming Environments: Blue, BlueJ and Greenfoot. *International Journal of People-Oriented Programming (IJPOP)*, 4(1), 5-32.
- Kramer, J. (2007). Is abstraction the key to computing?. *Communications of the ACM*, 50(4), 36-42.
- Lahtinen, E., Ala-Mutka, K., ve Jarvinen, H. M. (2005). A study of the difficulties of novice programmers. *Acm Sigcse Bulletin*, 37(3), 14-18.
- Lamagna, E. A. (2015). Algorithmic thinking unplugged. *Journal of Computing Sciences in Colleges*, 30(6), 45-52.
- Lau, J. Y. (2011). *An introduction to critical thinking and creativity: Think more, think better*. New Jersey: John Wiley & Sons.
- MakeBlock mBot (2018). *MakeBlock mBot Kiti*. 20 Mart 2018 tarihinde https://articulo.mercadolibre.com.mx/MLM-551430728-kit-makeblock-mbot-robot-educativo-para-blue-kids-version-d-_JM sitesinden alınmıştır.
- Malec, J. (2001, Mart). *Some thoughts on robotics for education*. Sözlü bildiri, 2001 AAAI Spring Symposium on Robotics and Education, Department of Computer Science Lund University, Sweden.
- Mano, C., Allan, V., ve Cooley, D. (2010). Effective in-class activities for middle school outreach programs. *Frontiers in Education Conference (FIE), 2010 IEEE* (pp. F2E-1).
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., ve Settle, A. (2014). Computational thinking in K-9 education. *Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference* (ss. 1-29). ACM.

Marlowe, B.A., Page, M. (1998) *Creating and sustaining the constructivist classrooms*. Page Publications Limited, London, UK.

Miglino, O., Lund, H. H., ve Cardaci, M. (1999). Robotics as an Educational Tool. *Journal of Interactive Learning Research*, 10(1), 25-47. Charlottesville, VA: AACE.

Mishra, P., ve Koehler, M. J. (2008). Introducing technological pedagogical content knowledge. *In Annual Meeting of the American Educational Research Association*, 1-16.

Morgan, C. T. (1999). *Psikolojiye Giriş*, (Çev: Hüsnu Arıcı ve diğ.) , Ankara: Meteksan Yayınları.

National Research Council. (1999). *National Research Council Committee on Information Technology Literacy, Being Fluent with Information Technology*, National Academy Press, Washington.

National Research Council. (2011). Report of a workshop on the pedagogical aspects of computational thinking. National Academies Press. 14 Mayıs 2018 tarihinde <https://www.nap.edu/catalog/13170/report-of-a-workshop-on-the-pedagogical-aspects-of-computational-thinking> sitesinden alınmıştır.

National Research Council. (2010). Report of a Workshop on the Scope and Nature of Computational Thinking. 14 Mayıs 2018 tarihinde <https://www.nap.edu/catalog/12840/report-of-a-workshop-on-the-scope-and-nature-of-computational-thinking> sitesinden alınmıştır.

National Curriculum. (2017). National curriculum in England: computing programmes of study. Department for Education. 10 Ağustos 2018 tarihinde <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study> sitesinden alınmıştır.

- Niess, M. L. (2005). Preparing teachers to teach science and mathematics with technology: Developing a technology pedagogical content knowledge. *Teaching and Teacher Education*, 21(5), 509–523.
- Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T., ve Kuno, Y. (2008). A CS unplugged design pattern ISSEP 2008 Proceedings, LNCS 5090.
- Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T. and Kuno, Y. (2009). A CS unplugged design pattern. Lewandowski, G., and Wolfman, S. (Ed.), *Proceedings of the 40th SIGCSE technical symposium on Computer Science Education*, 41(1), 231-235. Chattanooga, Tennessee, USA: ACM, New York.
- Nourbakhsh, I. R., Hamner, E., Crowley, K., ve Wilkinson, K. (2004). Formal measures of learning in a secondary school mobile robotics course. *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, 2, 1831-1836.
- Olsen, F. (2000). Computer Scientist Says All Students Should Learn to Think 'Algorithmically'. The Chronicle of High Education. 31 Mart 2018 tarihinde https://link.springer.com/chapter/10.1007/11915355_15 sitesinden alınmıştır.
- Ozgun-Koca, S. A., Meagher, M., ve Edwards, M. T. (2010). Preservice Teachers' Emerging TPACK in a Technology-Rich Methods Class. *Mathematics Educator*, 19(2), 10-20.
- Ozobot (2018). *Ozobot*. 20 Mart 2018 tarihinde <https://www.zdnet.com/pictures/bots-for-tots-5-robots-that-help-kids-learn/5/> sitesinden alınmıştır.
- Özden, M. Y. ve Durdu, L. (2016). *Nitel Araştırma Yöntemleri* (1. Baskı). Ankara: Anı Yayıncılık.

- Öztürk, E., ve Horzum, M. B. (2011). Teknolojik pedagojik içerik bilgisi ölçeği'nin türkçeye uyarlaması. *Ahi Evran Üniversitesi Kırşehir Eğitim Fakültesi Dergisi*, 12(3), 255-278.
- Papert, S. (1993). *Mindstorms: Children, computers, and powerful ideas*. (2. Baskı). New York, NY: Basic Books.
- Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*, 1(1), 95-123.
- Patton, M. Q. (2002). *Qualitative research & evaluation methods*. (3. Baskı). CA: Sage Publications.
- Paul, A. M. (2015). Teaching computer science-without touching a computer. *The Education Digest*, 80(5), 23-26.
- Perkins, D. N., ve Salomon, G. (2012). Knowledge to go: A motivational and dispositional view of transfer. *Educational Psychologist*, 47(3), 248-258.
- Piaget, J. (1964). Cognitive development in children: Development and learning. *Journal of Research in Science Teaching*, 2(3), 176–186.
- Polat, A. (2015). Computational thinking nedir? 21 Haziran 2016 tarihinde <http://www.ahmetpolat.net/computational-thinking-nedir> sitesinden alınmıştır.
- Resnick, M. (2012). Let's teach kids to code. *TED Talks*. 7 Nisan 2018 tarihinde https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code sitesinden alınmıştır.
- Robins, A., Rountree, J., ve Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172.

- Robinson, R. (2005). Sports philanthropy: *An analysis of the charitable foundations of major league teams*. Yayınlanmamış Yüksek Lisans Tezi, University of San Francisco.
- Rogers, C., ve Portsmore, M. (2004). Bringing engineering to elementary school. *Journal of STEM Education: innovations and research*, 5(3), 17.
- Salomon, G., ve Perkins, D. N. (1987). Transfer of cognitive skills from programming: When and how?. *Journal of educational computing research*, 3(2), 149-169.
- Sayın, Z. (2017). Bilgisayar bilimi eğitim kapsamı. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya* (133-154). Ankara: Pegem Akademi.
- Schwartz, J., Stagner, J., ve Morrison, W. (2006). Kid's programming language (KPL). In *ACM SIGGRAPH 2006 Educators program* (s. 52). ACM. DOI: 10.1145/1179295.1179348.
- Schreier, M. (2012). *Qualitative content analysis in practice* (3. Baskı). London: Sage Publications.
- Sedgewick, R., ve Wayne, K. (2011). *Algorithms* (4. Baskı). Addison-Wesley Professional.
- Seferoğlu, S. S., ve AKBIYIK, C. (2006). Eleştirel düşünme ve öğretimi. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 30(30).
- Selby, C., ve Woollard, J. (2013). Computational thinking: the developing definition. 16 Kasım 2016 tarihinde https://eprints.soton.ac.uk/356481/1/Selby_Woollard_bg_soton_eprints.pdf sitesinden alınmıştır.
- Shulman, L. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher*, 15(2), 4-14.

- Sternberg, R. J., ve Grigorenko, E. L. (2004). Successful intelligence in the classroom. *Theory into practice*, 43(4), 274-280.
- 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.
- Taub, R., Armoni, M., ve Ben-Ari, M. (2012). CS unplugged and middle-school students' views, attitudes, and intentions regarding CS. *ACM Transactions on Computing Education (TOCE)*, 12(2), 8.
- Tebliğler Dergisi. (2013). 2669 Sayılı Tebliğler Dergisi. 10 Ağustos 2018 tarihinde <http://goo.gl/32FonN> sitesinden alınmıştır.
- Tebliğler Dergisi. (2016). 2709 Sayılı Tebliğler Dergisi. 10 Ağustos 2018 tarihinde <http://goo.gl/uhnxt6> sitesinden alınmıştır.
- The National Academies Press. (2010). Report of a Workshop on the Scope and Nature of Computational Thinking. 20 Aralık 2016 tarihinde <https://www.nap.edu/catalog/12840/report-of-a-workshop-on-the-scope-and-nature-of-computational-thinking> sitesinden alınmıştır.
- The National Academies Press. (2011). Report of a Workshop of Pedagogical Aspects of Computational Thinking. 20 Aralık 2016 tarihinde <https://www.nap.edu/catalog/13170/report-of-a-workshop-on-the-pedagogical-aspects-of-computational-thinking> sitesinden alınmıştır.
- Thies, R., ve Vahrenhold, J. (2013). On plugging unplugged into CS classes. *Proceeding of the 44th ACM technical symposium on Computer science education* (ss. 365-370).
- Tok, E., ve Sevinç, M. (2010). Düşünme becerileri eğitiminin eleştirel düşünme ve problem çözme becerilerine etkisi. *Pamukkale Üniversitesi Eğitim Fakültesi Dergisi*, 27(27), 67-82.

- Toluk Uçar, Z., ve Altun, A. (2006). Sorun çözme becerilerinin öğretimi. A. Şimşek (Ed.), *İçerik Türlerine Dayalı Öğretim* (101-131). Ankara: Nobel Yayın Dağıtım.
- Top, E. (2017). Düşünme becerilerinin önemi. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya* (23-39). Ankara: Pegem Akademi.
- Tok, E., ve Sevinç, M. (2010). Düşünme becerileri eğitiminin eleştirel düşünme ve problem çözme becerilerine etkisi. *Pamukkale Üniversitesi Eğitim Fakültesi Dergisi*, 27(27), 67-82.
- Tsalapatas, H., Heidmann, O., Alimisi, R., Tsalapatas, S., Florou, C., ve Houstis, E. (2011, June). *Visual programming towards the development of early analytical and critical thinking*. Sözlü bildiri, International Conference on Future of Education, Floransa, İtalya.
- Üçgül, M. (2017). Eğitsel robotlar ve bilgi işlemsel düşünme. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya* (295-317). Ankara: Pegem Akademi.
- vanDam, A. (2000). Exploratories: From Algorithm Animations and Interactive Illustrations to Explorable Microworlds. Keynote Adress. *SIGCSE Technical Symposium*, (8-12). Austin, Texas.
- Vujosevic – Janicic, M., ve Tomic, D. (2008). The role of programming paradigms in the first programming courses. *The Teaching of Mathematics*, (21), 63-83.
- Wilson, C., ve Guzdial, M. (2010). How to make progress in computing education. *Communications of the ACM*, 53(5), 35-37.
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35.

- Wing, J. M. (2007). Computational Thinking. 13 Aralık 2016 tarihinde http://www.cs.cmu.edu/afs/cs/usr/wing/www/Computational_Thinking.pdf sitesinden alınmıştır.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences*, 366(1881), 3717-3725.
- Wing, J. M. (2011). Research Notebook: Computational Thinking - What and Why? The Link. Pittsburgh, PA: Carneige Mellon.
- vanDam, A. (2000). Exploratories: From Algorithm Animations and Interactive Illustrations to Explorable Microworlds. Keynote Adress. *SIGCSE Technical Symposium*, (8-12). Austin, Texas.
- Weinberg, A. E. (2013). Computational Thinking: An Investigation of the Existing Scholarship and Research (Colorado State University). 7 Temmuz 2018 tarihinde <http://people.cs.vt.edu/~kafura/CS6604/Papers/CT-Existing-Scholarship-Research-Dissertation.pdf> sitesinden indirilmiştir.
- 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* (9. Baskı). Ankara: Seçkin Yayıncılık.
- Yıldız, B. (2017). Disiplinlerarası öğretim yaklaşımı: bilgi işlemsel düşünme ve FeTeMM. Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya* (319-339). Ankara: Pegem Akademi.
- 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.

- Zengin, M. (2016). İlkokul, ortaokul ve lise öğrencilerin disiplinlerarası eğitim & öğretiminde robotik sistemlerinin kullanımına yönelik görüşleri. *Üstün Yetenekliler Eğitimi ve Araştırmaları Dergisi (UYAD)*, 4(2), 48-70.
- Koehler, M. J., ve Mishra, P. (2009). What is technological pedagogical content knowledge? *Contemporary Issues in Technology and Teacher Education*, 9(1), 60-70.



EKLER

Ek 1

GÖRÜŞME FORMU

Merhaba ben Aslı Göncü Abant İzzet Baysal Üniversitesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi ABD'nda yüksek lisans yapıyorum. Bilgi-işlemsel düşünme (Computational thinking) hakkında bilişim teknolojileri öğretmenlerinin düşünceleri üzerine bir araştırma yapıyorum. Bu kapsamda sizinle kodlama eğitimi ile ilgili olarak konuşmak istiyorum. Bu görüşmede amacım, bilişim teknolojileri öğretmenlerinin bilgisayarlı düşünme ile ilgili bilgilerini ve düşüncelerini ortaya çıkarmaktır. Bu araştırma ile ortaya çıkacak olan sonuçların, eğitim sistemimizdeki bilgi ve iletişim teknolojileri entegrasyonu çalışmalarının daha sağlıklı bir şekilde gerçekleştirilebilmesi sürecinde faydalı olacağını ümit ediyorum. Bu nedenle sizin kodlama eğitimi ile ilgili düşüncelerinizi öğrenmek istiyorum.

Bana görüşme sürecinde söyleyeceğinizin tümü gizlidir. Bu bilgileri araştırmacıların dışında herhangi bir kimsenin görmesi mümkün değildir. Başlamadan önce bu söylediklerimle ilgili belirtmek istediğiniz bir düşünce ya da sormak istediğiniz bir soru var mı?

Görüşmeyi izin verirseniz kaydetmek istiyorum. Bunun sizce bir sakıncası var mı?

İzin verirseniz sorulara başlamak istiyorum.

GÖRÜŞME SORULARI

Cinsiyet: Kadın Erkek

1. Mezun olduğunuz lisans programı nedir?
2. A. Kaç yıldır öğretmenlik yapıyorsunuz?
B. Kaç yıldır formatörlük yapıyorsunuz?

3. Derslerinizde kaç yıldır kodlama / programlama öğretiyorsunuz?
4. Sizce kodlama eğitimi nedir? Size göre kodlama eğitimi ne ifade ediyor?
5. 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?
6. Neden kodlama eğitimi, kodlama eğitimi öğrencilere neden verilmelidir?
- Sonda: Sizce kodlama eğitiminin temel amacı ne olmalıdır?
7. Sizce kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler nelerdir?
8. Kodlama eğitimi ile öğrencilere kazandırılması gereken kavram ve beceriler için kodlama ile ilgili hangi yazılım ve donanımları kullanırsınız?
- Sonda: Neden bu yazılım ve donanımları kullanmayı tercih edersiniz?
9. Bu yazılım ve donanımları kullanmadan kodlama eğitimindeki kavram ve becerileri kazandırma hakkında ne düşünüyorsunuz?
- Yanıt Evet ise (verimsiz olur ama kazandırabilirim derse)*
- Sonda: Peki bunu nasıl yaparsınız?
- Sonda: Peki bunun için ne tür aktiviteler yaparsınız?
- Sonda: Hangi araçları kullanırsınız?
- Sonda: Derste hangi yöntem ve teknikleri kullanırsınız?
- Yanıt Hayır ise*
- Sonda: Neden kazandıramayacağınızı düşünüyorsunuz?
10. Bilişim teknolojileri ve yazılım dersi öğretmenlerinin, sizin gözünüzde, öğrencilerin kodlama becerilerini geliştirme konusunda eksik olduğu noktalar var mıdır?

Sonda: Bu eksikliklerin giderilmesine yönelik olarak neler yapılmasını istersiniz?



Ek 2 ETİK KURUL ONAY FORMU



Abant İzzet Baysal Üniversitesi
Sosyal Bilimlerde İnsan Araştırmaları Etik Kurulu

Aslı GÖNCÜ
Abant İzzet Baysal Üniversitesi
Eğitim Bilimleri Enstitüsü
Bilgisayar ve Eğitim Teknolojileri Eğitimi A. B. D.

Sayın Aslı GÖNCÜ,

"Bilişim Teknolojileri ve Yazılım Öğretmenlerinin Kodlama Eğitimine Yönelik Düşünceleri" konulu araştırmanız ile ilgili olarak Abant İzzet Baysal Üniversitesi Sosyal Bilimlerde İnsan Araştırmaları Etik Kuruluna yapmış olduğunuz başvuru (Protokol NO. 2016/146) Kurulumuzun 14.11.2016 tarihli ve 2016/06 toplantısında değerlendirilerek etik olarak uygun bulunmuştur. Bilgilerinize sunarız.


Prof. Dr. Hamit COŞKUN (Başkan)


Prof. Dr. Mehmet ERYİĞİT (Üye)


Doç. Dr. Altay EREN (Üye)


Doç. Dr. H. Birol YALÇIN (Üye)


Doç. Dr. Seval ALKOY (Üye)


Y. Doç. Dr. Abdullah DURAKOĞLU (Üye)


Av. Zuhal Demirci (Üye)

EK 3 İL MİLLİ EĞİTİM MÜDÜRLÜĞÜ ARAŞTIRMA İZİN BELGESİ

T.C.
BOLU VALİLİĞİ
İl Millî Eğitim Müdürlüğü

Sayı : 12050492-605-E.14453702
Konu : Araştıma İzni (Aslı ÖNCÜ)

22.12.2016

MÜDÜRLÜK MAKAMINA

İlgi: Abant İzzet Baysal Üniversitesi Eğitim Bilimleri Enstitüsü Müdürlüğünün 13/12/2016 tarih ve 15032 sayılı yazısı.

Abant İzzet Baysal Üniversitesi Eğitim Bilimleri Anabilim Dalı Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalı Yüksek Lisans programı öğrencisi Aslı ÖNCÜ'nün ilgi dilekçesinde; "Bilişim Teknolojileri ve Yazılım Öğretmenlerinin Kodlama Eğitimine Yönelik Düşünceleri" konulu araştırmasına veri sağlamak amacıyla İlimiz merkez ve ilçelerde görev yapan bilişim teknolojileri ve yazılım öğretmenlerine uygulama yapılması istenmektedir.

Söz konusu talep edilen araştırma, Bakanlığımız Yenilik ve Eğitim Teknolojileri Genel Müdürlüğünün 2012/13 sayılı Genelgesi gereğince yapılması hususunu olurlarınıza arz ederim.

Metin KANDIŞ
Şube Müdürü

OLUR
22.12.2016

İbrahim KUNUK
Millî Eğitim Müdür V.

Verilecek cevaplarda yazımızın tarih ve sayısının bildirilmesi
Millî Eğitim Müdürlüğü Tabaklar Mahallesi 14100 BOLU
e-posta: meslekiteknikegitim14@meb.gov.tr

İrtibat: Memur: Günay YILDIZ
Tel: 254 50 81 Faks : 254 51 30

Ek 4 TEZ BAŞLIĞINDA DEĞİŞİKLİK TUTANAĞI**TUTANAK**

Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalı yüksek lisans programı öğrencisi Aslı GÖNCÜ'nün 10.01.2019 tarihinde yapılan tez savunmasında "Bilişim Teknolojileri ve Yazılım Dersi Öğretmenlerinin Kodlama Eğitimine Yönelik Algıları" tez başlığının "Bilişim Teknolojileri Ve Yazılım Dersi Öğretmenlerinin Kodlama Eğitimi Hakkındaki Görüşleri" olarak değiştirilmesinin uygun olduğuna. (10.01.2019)

Doç. Dr. İbrahim ÇETİN
Tez Danışmanı



Prof. Dr. Süleyman Sadi SEFEROĞLU
Üye



Prof. Dr. Erkan TEKİNARSLAN
Üye



ÖZGEÇMİŞ

1993 yılında Ankara'da doğdu. İlk ve Orta öğrenimini Ankara'da tamamladı. 2010-2011 yılında İMKB Anadolu Teknik Lisesi, Bilgisayar Programcılığı bölümünü ikincilikle bitirdi. Lisans eğitimini 2011-2015 yılları arasında Abant İzzet Baysal Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümünde tamamlamıştır.

İletişim Adresleri

e-mail: goncuasl@gmail.com

Telefon: 05342074337