

Makale Geçmişi / Article History

Alındı/Received: 04.01.2023

Düzeltilme Alındı/Received in revised form: 23.04.2023

Kabul edildi/Accepted: 28.06.2023

**ERKEN YAŞTA PROGRAMLAMA ÖĞRETİMİNDE METAFOR KULLANIMI:
METAFORLARLA PROGRAMLAMA ÖĞRENİYORUM ETKİNLİKLERİ***

Nur Banu Alagöz¹ , Hatice Gökçe Bilgiç²

Öz

Son yıllarda eğitim dünyasında programlama öğretimine yönelik artan ilgi sonucunda programlama öğretiminde kullanılacak etkili yöntemler üzerine çalışmaların önemi artmaktadır. Bu çalışmada erken yaşta programlama öğretiminde metafor kullanımının programlama öğretim sürecine etkisinin gözlemlenmesi amaçlanmaktadır. Çalışmada karma araştırma desenlerinden olan yakınsayan paralel araştırma deseni kullanılmıştır. Nicel veriler Akademik Başarı Testi, Etkinlik Algısı Ölçeği ve Bilgisayarca Düşünme Ölçeği ile toplanmıştır. Nitel veriler yapılandırılmış görüşmeler sonucu elde edilmiştir. Çalışma grubu Orta Karadeniz bölgesindeki bir ilde bulunan Milli Eğitim Bakanlığına bağlı özel bir okulda öğrenimine devam eden 50 ortaokul öğrencisinden oluşmaktadır. Akademik başarı testi ve Bilgisayarca Düşünme ölçeğine ait ön-test ve son-test puanlarına göre programlama öğretiminde metafor kullanımı ile öğrencilerin akademik başarısında ve bilgisayarlı düşünme becerilerinde anlamlı bir değişim gözlemlenmiştir. Etkinlik Algısı Ölçeği her hafta gerçekleştirilen “Metaforlarla Programlama Öğreniyorum” etkinlikleri sonrasında uygulanmıştır. Etkinlik Algısı Ölçeğinden elde edilen sonuçlara göre öğrencilerin katıldıkları etkinlikleri eğlenceli ve ilgi çekici buldukları görülmüştür. Görüşmelerden elde edilen verilere göre programlama öğretiminde metafor kullanımının akılda kalıcılığı arttırma, anlaşılabilirlik ve eğlence gibi katkıları öne çıkmıştır. Ayrıca öğrencilerin başka derslerde de metafor uygulamasını deneyimlemek istedikleri görülmüştür. Böylece özellikle erken yaşta programlama öğretiminde metafor kullanımının alternatif programlama öğretimi yaklaşımları arasında değerlendirilebileceği söylenebilir. Metaforlarla programlama öğretiminin bilişim teknolojileri öğretmenleri tarafından özellikle erken yaşta programlama öğretiminde temel programlama kavramlarının bilgisayarsız olarak öğretilebileceği etkinlikler çerçevesinde değerlendirilebileceği söylenebilir.

Anahtar Kelimeler: programlama öğretimi; bilgisayarlı düşünme; metafor.

* Bu çalışma ikinci yazarın danışmanlığında birinci yazarın yüksek lisans tez çalışmasından üretilmiştir.

¹ Lisansüstü Öğrenci, Milli Eğitim Bakanlığı, nurbanualagoz@gmail.com, orcid.org/0000-0003-3587-5870

² Doç. Dr., Ondokuz Mayıs Üniversitesi, hgokcebilgic@gmail.com, orcid.org/0000-0002-3925-2497

Nur Banu Alagöz, Hatice Gökçe Bilgiç

Yasal İzinler: Ondokuz Mayıs Üniversitesi Sosyal ve Beşeri Bilimler Etik Kurulu, Tarih: 11.10.2019, Sayı: 2019-296.

METAPHORS IN THE EARLY AGE PROGRAMMING TEACHING: LEARNING PROGRAMMING WITH METAPHORS

Abstract

In recent years, the increasing popularity of the programming teaching resulted with the needs of alternative methods of programming teaching. The aim of the study is to examine the effects of metaphors in the process of early age programming teaching. For this purpose, a converging parallel design, which was a mixed research method, was used in the study. Quantitative data of the study was collected through Academic Achievement Test, Activity Perception Scale and Computational Thinking Scale. Qualitative data was collected through structured interviews with students. The data were collected from 50 secondary school students. As a result of the study, it was observed that using metaphors in the early age programming teaching resulted with significant increase on the academic achievement scores and computational thinking scores of the students. According to the results obtained from the Activity Perception Scale, it was seen that students thought that activities was funny and interesting. Moreover, based on the data obtained from the interviews, learning with metaphors were more memorable, clear, enjoyable, and funny. Students also mentioned that they expected to experience learning with metaphors in other subject-area courses. Thus, IT teachers might use metaphors into the cs unplugged activities to teach basic concepts of programming.

Keywords: programming teaching; metaphors; computational thinking.

Legal Permissions: Ethics Committee of Social Sciences and Humanities of Ondokuz Mayıs University Date: 11.10.2019, Number: 2019-296.

Summary

Developing technologies especially digitalization affects the needs of the labour market in the world. Labour market needs different sets of skills in the individuals since digital systems undertake some of the human jobs. Individuals need to develop new century skills (European Schoolnet, 2015). World Economic Forum (2020) published the prominent skills for 2025. The first five of these skills are as following: (1) analytical thinking and innovation, (2) active learning and learning strategies, (3) complex-problem solving, (4) critical thinking and analysis, and (5) creativity, originality, and initiative.

Computational thinking is defined as a systematic way of problem solving. Computational thinking is also defined as thinking skills to use problem solving skills in any other disciplines (Bundy, 2007). It is also mentioned that computational thinking skill is not just a skill that computer scientists should have, but it is important that all individuals should have this skill (Wing, 2008). Furthermore, computational involves rationalization, reasoning, logic, algorithm, induction, deduction, and discovery (Terzidis, 2006). In the literature, the importance of programming teaching is emphasized to lead students gaining new century higher order thinking skills such as algorithmic thinking, problem solving, critical thinking and computational thinking (Jancheski, 2017; Kandemir, 2018; Kert, 2018; Yildiz-Durak & Güyer, 2018). In the World, programming teaching is beyond the vocational education, it is served as a structured education field to contribute the thinking and production skills of the individuals

(Kert, 2018). Thus, in the 21st century coding and programming skills are essential skills for individuals beyond technical skills who work in computer science field (European Commission, 2014). On the other hand, although programming teaching is popular and increasing in today's World, programming courses are mainly hard for new learners (Çetin, 2013). Programming languages and teaching methods that used in programming teaching were make harder to learn the nature and basics of programming (Saygıner & Tüzün, 2017a). It is also harder in the early age programming teaching. Thus, different application and methods of teaching was developed to eliminate the difficulties of syntax rules like block-based environments. Moreover, metaphors were powerful mental tool to use understanding the abstract and complicated facts (Şaban et al., 2006). It is mentioned that relating the learned facts with real-life examples increased the motivation and success of students (Yıldırım & Konur, 2014). As a result, this study aims to benefit from the power of metaphors to transfer the abstract concepts of programming in the early age programming teaching process.

The aim of the study is to examine the effects of metaphors in the process of early age programming teaching. The effects of metaphors to early age programming teaching was studied based on the variance on academic achievement, computational thinking skills and opinions about activity perception of the students. To accomplish the aim of the study, the following questions will be addressed:

1. Does the use of metaphors in the early age programming teaching make a significant difference on the academic achievement of the students?
2. Does the use of metaphors in the early age programming teaching make a significant difference on the computational thinking skills of the students?
3. What are the students' opinions about metaphors in the early age programming teaching process?

A converging parallel design, which was a mixed research method, was used in the study. Converging parallel design was used to examine more deeply (Creswell & Clark, 2011). In this design, qualitative and quantitative data collection process are conducted in the same period. Qualitative and quantitative data are analyzed separately and interpreted together. Quantitative data of the study was collected through Academic Achievement Test, Activity Perception Scale (Kasalak, 2017) and Computational Thinking Scale (Korkmaz et al., 2015). Academic Achievement Test was developed by the researcher. Qualitative data was collected through structured interviews with students. The data were collected from 50 secondary school students. Convenient sampling was used in the sample selection of the study.

The pre-test and post-test data results obtained from Academic Achievement Test and Computational Thinking Scale were analyzed with paired-samples t-test. In the study, repeated measures from the sample group before and after the experimental application was carried to observe whether a significant different occurs between the pre-test and post-test results. To analyze the data obtained from Activity Perception Scale item-based mean and standard deviation results were used. Moreover, structured interview form has both closed-ended and open-ended questions. Thus, descriptive data analysis (percentage and frequency) and content analysis were used together to analyze the interview data.

As a result of the study, it was observed that using metaphors in the early age programming resulted with significant increase on the academic achievement scores of the students. Pérez-Marín et al. (2018) also stated that using metaphors to teach basic concepts

of programming to children resulted with success. In another study, Pérez-Marín et al. (2020) used metaphors to support developing the basic Information Technology (IT) skills and the programming skills of the primary school students. They also mentioned the success of metaphors to develop programming skills of the students.

It was also observed that using metaphors in the early age programming resulted with significant increase on the computational thinking scores of the students. In the literature, it was mentioned that programming or coding positively affects the development of computational thinking skills of the students (Bers et al., 2014; Brennan & Resnick, 2012). Pérez-Marín et al. (2020) stated that teaching programming with metaphors caused with increase of computational thinking skills. Therefore, the positive effects of using metaphors in the programming teaching can be related with the development of computational thinking skills.

According to the results obtained from the Activity Perception Scale, it was seen that students thought that activities was funny and interesting. Moreover, based on the data obtained from the interviews, the students thought that learning with metaphors were more memorable, clear, enjoyable and funny. Finally, the students mentioned that they wanted to learn with metaphors in other subject-matter courses.

Giriş

Gelişen teknolojilerin birçok alanda olduğu gibi iş gücü dünyasında da etkileri yoğun bir şekilde görülmektedir. Özellikle bazı alanlarda makinelerin insan gücünün yerini alması sonucunda iş dünyasının iş gücü ihtiyaçları da değişmektedir. Bu değişim bireyler için yeni becerilerin geliştirilmesini öne çıkarmaktadır (European Schoolnet, 2015). World Economic Forum (2020) tarafından yayınlanan raporda 2025 yılı için öne çıkan beceriler sunulmaktadır. Bu becerilerin ilk beşi sırasıyla şu şekildedir: (1) analitik düşünme ve yenilik, (2) aktif öğrenme ve öğrenme stratejileri, (3) karmaşık problem çözümü, (4) eleştirel düşünme ve analiz, (5) yaratıcılık, özgünlük ve girişimcilik. World Economic Forum tarafından 2016 yılında hazırlanan geleceğin meslekleri ve becerileri raporunda ise şirketlerin iş hayatı ile ilişkili temel becerilere yönelik değişen talepleri arasında problem çözümü %36 ile en üst seviyede karşımıza çıkmaktadır. Bu sonuçlar yeni nesilde akademik ve teknolojik becerilerin ötesinde analitik düşünme, problem çözme ve yaratıcılık gibi üst düzey düşünme becerilerinin önemini göstermektedir.

Bilgisayım içerisinde rasyonelleştirme, akıl yürütme, mantık, algoritma, tümevarım, tündengelim, dış değer bulma, keşif ve tahminleri içeren bir kavram olarak karşımıza çıkmaktadır (Terzidis, 2006). Bilgisayım düşünme ise sorun çözmenin sistematik bir yolu olarak öne çıkmaktadır. Başka bir tanımda günlük hayattaki problemleri çözmek ve çözümleri sistematik çözümler haline getirmek için gerekli olan zihinsel etkinlikler olarak tanımlanmaktadır (Yadav vd., 2016). Bir diğer tanımda ise gerçek hayat problemlerine çözüm üretebilmek için bilgisayarı kullanabilecek donanımlara hâkim olmak olarak tanımlanmaktadır (Korkmaz vd., 2017). Wing (2008) de bilgisayar düşünme becerisinin sadece bilgisayar bilimcilerinin sahip olması gereken bir beceri olmadığını, bu becerinin tüm bireylerde bulunmasının önemini ifade etmektedir. Uluslararası Eğitim Teknolojisi Topluluğu (ISTE, 2022a) öğrenenler, eğitimciler, liderler, tasarımcılar gibi farklı alanlardaki bireyler için bilgisayar düşünme becerisi yetkinliklerini yayınlamıştır. Bu çalışmada eğitimciler için bilgisayar düşünme becerilerinin sınıf içi uygulamalara entegrasyonunun önemi

vurgulanmaktadır. ISTE (2022a) tarafından hazırlanan yetkinliklerde öğrencilerin bilgisayar bilimleri veya bilgisayarlı düşünmeye yönelik prensipleri uygulayabilecekleri ya da keşfedebilecekleri öğrenme materyalleri ve etkinliklerin tasarlanması öne çıkmaktadır. Öğrenciler için sunulan standartlarda bilgisayarlı düşünür ifadesi kullanılmaktadır (ISTE, 2022b). Öğrencilerin problemleri anlamak ve çözmek için stratejiler ve çözüm yolları geliştiren ve test etmek için teknolojinin gücünden yararlanan bireyler olması önemle vurgulanmaktadır.

Eğitim dünyası ile gelişen ve iyileşen insan becerileri ekonomik başarının, bireysel refahın ve toplumsal uyumun anahtar etmenleri olarak ifade edilmektedir (World Economic Forum, 2020). Bu nedenle eğitim dünyası iş gücü dünyasının ihtiyaçları doğrultusunda kendisini yenilemeli ve yeni becerilerin bireylere kazandırılmasını desteklemelidir. Dolayısıyla iş gücü dünyasının ihtiyaçları doğrultusunda eğitim dünyasında da bu becerilerin geliştirilmesi konusunda çalışmalar yapılmaktadır. Programlama öğretiminin günümüz yeterliliklerinin bireylere kazandırılması için önemi vurgulanmaktadır (Kalelioğlu, 2015). Programlama öğretimi eğitim dünyasında özellikle analitik düşünme, algoritmik düşünme, problem çözme ve bilgisayarlı düşünme gibi üst düzey düşünme becerilerinin kazandırılması konusunda karşımıza çıkan önemli yaklaşımlardan biridir (Jancheski, 2017; Kandemir, 2018; Kert, 2018; Yıldız-Durak ve Güyer, 2018).

Programlama Öğretimi

Program kelimesi “bilgisayara bir işlemi yaptırmak için yazılan komutlar dizisi” olarak tanımlanmaktadır (TDK, 2020). Programlama kavramı ise içinde birçok aşamayı barındıran ve kişinin yoğun bir bilişsel beceri kullanarak “karmaşık bir problem çözme süreci” olarak tanımlanmaktadır (Kert ve Uğraş, 2009). Programlama süreçlerinde kullanılan problem-çözme sistematiği, toplumsal hayatın diğer alanlarında kullanılması öngörülen problem-çözme süreçleri ile ortak noktalara sahiptir. Bilgisayar programının hazırlanabilmesi için yerine getirilmesi gereken temel aşamalar problemin tanımı, çözüm yolunun belirlenmesi, programın kodlanması, programın yorumlanması-derlenmesi ve programdaki hataların belirlenerek giderilmesidir (Kesici ve Kocabaş, 2007). Bilgisayarlara problemi çözerken hangi verileri nasıl işleyeceğine dair komutlar program tarafından sağlanır. Bu doğrultuda bakıldığında programı yazan kişinin problemin çözüm yolunu bilmesi gereklidir. Dolayısıyla programlama öğretimi sürecinde kazandırılacak bu becerilerin bireylerin günlük hayatında veya iş gücü dünyasında ihtiyacı olacak yeni yüzyıl becerilerini de destekleyeceği söylenebilir. Özellikle erken yaşta programlama öğretiminin günlük yaşam becerileri kazandırma, bilgi işleme becerilerini geliştirme ve problem çözme becerilerini destekleme gibi konularda önemi karşımıza çıkmaktadır. Temel programlama becerilerinin çağın bir gerekliliği olduğu belirtilirken, her bireyin kodlama becerilerine ihtiyacının olmadığı fakat “bir programcı gibi düşünmeyi öğrenmenin” birçok disiplinde yararlı olacağı da vurgulanmaktadır (Shein, 2014). Programlama dijital dünyada evrensel bir dil ve temel bir okuryazarlık becerisi olarak adlandırılmaktadır (Jancheski, 2017). Ayrıca bireylerin yoğun bir programlama deneyimi yaşadıklarında mantıksal düşünme becerilerinin geliştiği ileri sürülmektedir (Dillashaw ve Bell, 1985). Ayrıca günümüzde programlama öğretimi, üretmenin yeni bir yolu olarak ifade edilmektedir (Sayın ve Seferoğlu, 2016). Dünyada programlama öğretimi teknik becerilere odaklanan mesleki eğitimin ötesinde bireylerin düşünme ve üretim gibi yeteneklerine katkı sağlamak amacıyla yapılandırılmış bir eğitim alanı olarak vurgulanmaktadır (Kert, 2018). Bu nedenle dijitalleşen dünyada artık kodlama ve programlama becerileri sadece bu alanlarda çalışan bireylerin sahip olması gereken teknik becerilerin ötesinde her bireyde bulunması gereken temel 21. yüzyıl becerileri olarak karşımıza çıkmaktadır (European Commission, 2014).

Programlama kavramının bu kadar popüler olmasına ve günümüzde bu alanda verilen eğitimlerin artmasına rağmen alanyazına bakıldığında programlama derslerinin yeni öğrenen bireyler için genellikle zor bir ders olarak ifade edildiği görülmektedir (Çetin, 2013). Alanyazında programlama öğretimi farklı üst düzey düşünme becerileri gerektiren karmaşık ve zorlayıcı bir süreç olarak ifade edilmektedir (Yıldız-Durak, 2018; Winslow, 1996) Ayrıca kodlama eğitiminin kişiler için kolay olarak algılanmayan etkinlikler içerdiği (Koorse vd., 2015) için programlama performansının düştüğü ifade edilmektedir. Gomes ve Mendes (2007), kodlama eğitiminde yaşanan zorlukları “öğretme metotları, çalışma metotları, öğrencilerin beceri ve davranışları, programlamanın doğası ve psikolojik etkenler” olmak üzere 5 kategoride ele almıştır. Tobar vd. (2011) programlama dilini ilk defa öğrenenlerin bu süreci karmaşık ve zor olarak tanımladıklarını ifade etmişlerdir. Programlama dili öğretiminin farklı bir yaklaşım gerektiren zor bir süreç olduğu vurgulanmaktadır (Gomes ve Mendes, 2007). Programlama öğretiminde kullanılan programlama dilleri veya öğretim yöntemleri öğrencilerin programlamanın doğasını ve temel kavramlarını öğreniminde zorluklara neden olmaktadır (Saygıner ve Tüzün, 2017a). Özellikle erken yaşta programlama öğretiminde programlamanın doğasına ait temel kavramların aktarımında zorluklar artmaktadır. Çocuklar programlama dillerinin en büyük zorlukları arasında yer alan söz dizim kuralları arasında kaldıklarında programlama öğretimi sıkıcı ve zorlayıcı bir hal almaktadır (Jancheski, 2017). Bu yüzden özellikle erken yaşta programlama öğretiminde blok tabanlı ortamlar gibi söz dizim kurallarının zorluklarını ortadan kaldıran farklı uygulamalar geliştirilmiştir. Öte yandan, yapılan çalışmalar incelendiğinde öğrenilen kavram, bilgi ve olayların günlük hayatla ilişkilendirilmesinin öğrencilerin başarı ve motivasyonunu olumlu yönde etkilediğine dair görüşler yer almaktadır (Yıldırım & Konur, 2014).

Eğitimde Metafor Kullanımı

Metaforlar somut olgulardan yararlanarak soyut olguları anlamada ve anlatmada kullanılabilecek kuvvetli bir araç olarak değerlendirilmektedir (Heywood, 2002; Yob, 2003). Muran ve DiGiuseppe (1990) metaforu yeni bir anlam oluşturma süreci olarak tanımlamaktadır. Amundson (1988) metaforu karışık seviyedeki birden çok olgunun daha basit seviyeye indirgenmesi olarak tanımlamaktadır. Strong (1989) ise metaforu bilgiyi anlatmanın dolaylı bir yolu olarak ifade etmektedir.

Alanyazında metaforların eğitimde kullanımına dair örnek çalışmalar yer almaktadır. Pérez-Marín vd. (2018) çocuklara programlamayı öğretmek amacıyla öğretmen ve araştırmacılar için metaforlara dayalı bir metodoloji sunmuş ve öğrencilerin metaforları yararlı bulmaya yönelik görüşlerini incelemişlerdir. Araştırma sonucunda metafor kullanımının öğrencilere programlama kavramlarını öğretmek konusunda başarılı sonuçlar verdiği vurgulanmaktadır. Pérez-Marín vd. (2020) ayrıca metaforlarla öğrencilere bilişim teknolojileri alanında eğitim vermişler ve bu yöntemle bilgisayarca düşünmenin geliştirilip geliştirilemediğini incelemişlerdir. Bu çalışma ile ilköğretim öğrencilerinde bilişim teknolojileri becerilerinin geliştirilmesi ve bilgisayar programlama kavramlarının öğretilmesi için metafor kullanımının faydalı olduğu sonucuna ulaşılmıştır.

Tompkins ve Lawley (2002) metaforları, bir kavramı başka kavramla tanımlama, benzetme yönüyle tasvir etme ve başka açılardan görme olarak ifade etmektedir. Metaforlar ayrıca soyut ve karmaşık olguları anlamada kullanılabilecek güçlü zihinsel bir araç olarak sunulmaktadır (Saban vd., 2006). Dolayısıyla soyut kavramların günlük hayatla ilişkisinin kurulması için benzer kavramlardan yararlanılması etkili bir yöntem olarak düşünülebilir. Gomes ve Mendes (2017) programlama eğitiminin kişiselleştirilememesini, programlama

yapısının soyut yapısını ve öğrencilerin motivasyonlarının düşük olmasını programlama eğitiminde yaşanan zorlukların nedeni olarak ifade etmişlerdir. Bu amaçla bu çalışmada özellikle erken yaşta programlama öğretiminde programlama öğretimi içerisindeki sabit, değişken, bellek, algoritma, döngü ve koşul konuları gibi soyut kavramların öğrencilere aktarımında metaforların gücünden yararlanılması amaçlanmaktadır. Öğrencilerin programlama ile ilgili soyut kavramları somutlaştırmasında yardımcı olacak metaforlar geliştirilerek programlama öğretim sürecinde bu metaforlardan yararlanılan “Metaforlarla Programlama Öğreniyorum” etkinlikleri gerçekleştirilmiştir. Gerçekleştirilen bu etkinlikler ile programlama öğretiminde öğrenme sürecinin günlük hayatla ilişkilendirilmesi hedeflenmiştir. Böylece programlama ile ilişkili soyut kavramların öğreniminin özellikle erken yaştaki çocuklar için hem eğlenceli hem de daha kalıcı olması sağlanabilecektir.

Araştırmanın Amacı ve Araştırma Soruları

Bu çalışmanın amacı programlama öğretiminde metafor kullanımının erken yaşta programlama öğretim sürecine etkisinin gözlemlenmesidir. Metafor kullanımının programlama öğretimine etkisi öğrencilerin akademik başarısındaki anlamlı değişim, bilgisayarlı düşünme becerilerindeki anlamlı değişim ve etkinlik algılarına ait görüşleri üzerinden incelenmiştir.

Bu amaçla “Metafor kullanımı erken yaşta programlama öğretimine etki etmekte midir?” araştırma sorusunun aşağıdaki alt araştırma soruları çerçevesinde cevaplanması hedeflenmektedir:

1. Programlama öğretiminde metafor kullanımı öğrencilerin akademik başarılarında anlamlı bir fark oluşturmada mıdır?
2. Programlama öğretiminde metafor kullanımı öğrencilerin bilgisayarlı düşünme becerilerinde anlamlı bir fark oluşturmada mıdır?
3. “Metaforlarla Programlama Öğreniyorum” etkinliklerine katılan öğrencilerin metaforlarla programlama öğretim sürecine ait görüşleri nelerdir?

Yöntem

Araştırma Deseni

Bu çalışmada nitel ve nicel araştırma yöntemlerinin birlikte uygulandığı karma yöntem desenlerinden açılımlı sıralı desen kullanılmıştır. Karma araştırma yönteminin seçilme nedeni nitel ve nicel araştırma yöntemlerini birlikte kullanarak tek bir yöntemden kaynaklanabilecek olan eksiklikleri veya hataları ortadan kaldırmaya çalışmaktır. Ayrıca karma araştırma yöntemi seçilerek çalışma sonuçlarının farklı verilerle desteklenmesi amaçlanmıştır. Karma araştırma yöntemlerinden olan açılımlı sıralı desende nicel bir aşama ile başlayan çalışma ilişkili sonuçları daha derin açıklamak için nitel bir aşama ile devam eder (Creswell ve Clark, 2011). Böylece nitel araştırma aşaması nicel verilerden elde edilen ilişkileri açıklamak için kullanılır. Araştırmanın nicel kısmında tek grulu ön-test ve son-test deseni kullanılmıştır. Tek grulu ön-test ve son-test deseninde bir gruba bağımsız değişkenler uygulanır ve deney öncesinde ve deney sonrasında ölçme yapılır. Bu ölçmeler sonucunda ön-test ve son-test arasındaki fark bağımsız değişkenin bağımlı değişken üzerindeki etkisini gösterir (Cohen, Manion, & Morrison, 2002). Bu çalışmada da Akademik Başarı Testi ve Bilgisayarca Düşünme Ölçeği ön test ve son test olarak uygulanmıştır. Etkinlik Algısı Ölçeği ise her hafta gerçekleştirilen etkinlikten sonra öğrencilere uygulanmıştır. Araştırmanın devamında da

sürece dair toplanan nicel verileri desteklemek amacı ile öğrenci görüşmeleri gerçekleştirilmiştir.

Bu araştırma kapsamında gerçekleştirilen veri toplama süreçleri, Ondokuz Mayıs Üniversitesi Sosyal ve Beşeri Bilimler Etik Kurulunun 11.10.2019 tarihli ve 2019-296 sayılı belgesi ile onaylanmıştır.

Çalışma Grubu

Çalışma grubu Orta Karadeniz bölgesindeki bir ilde bulunan Milli Eğitim Bakanlığına bağlı özel bir okulda öğrenimine devam eden 5, 6, 7 ve 8. sınıf öğrencilerinden oluşmaktadır. Araştırmacının çalıştığı ve araştırmacı tarafından erişilebilir olan bir okul içerisindeki 50 ortaokul öğrencisinden oluşan çalışma grubu ile çalışma yürütülmüştür. Katılımcıların %38'i 5. sınıf, %20'si 6. sınıf, %20'si 7. sınıf ve %22'si 8. sınıf öğrencilerinden oluşmaktadır. Ayrıca katılımcıların %56'sı erkek ve %44'ü kız öğrencilerdir (Bkz. Tablo 1).

Tablo 1. Çalışma grubuna ait bilgiler

| | Grup | f | % |
|-----------------|----------|----|-----|
| Sınıf | 5. Sınıf | 19 | 38 |
| | 6. Sınıf | 10 | 20 |
| | 7. Sınıf | 10 | 20 |
| | 8. Sınıf | 11 | 22 |
| | Toplam | 50 | 100 |
| Cinsiyet | Erkek | 28 | 56 |
| | Kız | 22 | 44 |
| | Toplam | 50 | 100 |

Veri Toplama Araçları

Araştırmanın nicel verileri Bilgisayarca Düşünme Ölçeği (Korkmaz, Çakır ve Özden, 2015), Etkinlik Algısı Ölçeği (Kasalak, 2017) ve araştırmacı tarafından geliştirilmiş Akademik Başarı Testi ile toplanmıştır. Araştırmanın nitel verileri ise araştırmacı tarafından geliştirilmiş Yapılandırılmış Görüşme Formu ile toplanmıştır.

Bilgisayarca Düşünme Ölçeği (BDÖ)

Korkmaz vd. (2015) tarafından geliştirilmiş ve ortaokul öğrenci seviyesine göre uyarlanmış Bilgisayarca Düşünme Ölçeği öğrencilerin bilgisayarlı düşünme becerilerine yönelik ön-test ve son-test puanlarını elde etmek için kullanılmıştır. Ölçek beş faktör ve 22 maddeden oluşmaktadır. Bu faktörler yaratıcılık, algoritmik düşünme, işbirliklilik, eleştirel düşünme ve problem çözmedir. Ölçeğe doğrulayıcı faktör analizi yapılmış 29 madde olan ölçek uyarlanarak 22 maddeye düşmüştür. Ölçeğin madde korelasyonları hesaplanmış ve korelasyon değerleri 0,655 ile 0,862 arasında değiştiği tespit edilmiştir. Bilgisayarca Düşünme Ölçeğinin güvenilirlik katsayısı 0,809 ve alt boyutlar için değerleri 0.640 ile 0.867 arasında değişmektedir. Ayrıca her faktörün amacına anlamlı yönde hizmet ettiği ve maddelerin istenen düzeyde ayırt edici olduğu belirlenmiştir. Yapılan analizler sonucunda ölçeğin hem geçerli hem de güvenilir olduğu sonucuna ulaşılmıştır (Korkmaz vd., 2015). Kullanılacak ölçek belirlendikten sonra ölçek sahibinden gerekli izinler alınarak kullanılmıştır.

Akademik Başarı Testi (ABT)

Akademik Başarı Testi seçilen müfredat konularına ve planlanan beş haftalık eğitim sürecine uyumlu olarak araştırmacı tarafından hazırlanmıştır. Geliştirilen ABT için Milli Eğitim Bakanlığı'na (MEB) bağlı okullarda çalışan üç Bilişim Teknolojileri öğretmeni ve üniversitelerde eğitim fakültesinde bulunan Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümünde görev yapan 2 öğretim elemanı olmak üzere beş ilgili alan uzmanının görüşüne başvurulmuştur. Alınan görüşler doğrultusunda akademik başarı testi üzerinde tekrar çalışılarak nihai soruları içeren akademik başarı testi oluşturulmuştur. Akademik Başarı Testi 1 açık uçlu algoritma tasarlama sorusu, 13 kapalı uçlu olmak üzere toplam 14 sorudan oluşmaktadır. Bu sorulardan 2'si bellek, 2'si koşul, 4'ü algoritma, 4'ü sabit ve değişken, 2'si soru döngü konularına ait kazanımlarla ilişkilendirilmiştir.

Etkinlik Algısı Ölçeği

Etkinlik algısı ölçeği Deci vd. (1994) tarafından geliştirilmiş olup ölçeğin Türkçe uyarlaması Kasalak (2017) tarafından yapılmıştır. Etkinlik algısı ölçeği öğrenci yaşantılarını tespit etmek amacıyla hazırlanmıştır. Ölçeğin Türkçe uyarlaması yapılırken kültürel ve eğitim sistemindeki farklılıklar düşünülerek 25 maddelik ölçeğin 11 maddesinin alınması kararlaştırılmıştır. Kullanılacak ölçek belirlendikten sonra ölçek sahibinden gerekli izinler alınarak kullanılmıştır. Ölçeğin uyarlaması sonucunda ölçek toplam puanından değil madde bazlı puanlardan yararlanılabileceği belirtildiği için bu çalışmada da öğrencilerin metafor etkinliklerine dair yaşantılarını değerlendirmek amaçlı madde bazlı puanlardan yararlanılmıştır.

Yapılandırılmış Görüşme Soruları

Çalışmanın nitel verileri araştırmacı tarafından geliştirilen Yapılandırılmış Görüşme formu ile toplanmıştır. Araştırmacı tarafından hazırlanan yapılandırılmış görüşme formunun iç ve dış geçerliğini sağlamak için MEB'e bağlı okullarda çalışan bir Bilişim Teknolojileri öğretmeni ve üniversitelerde eğitim fakültesinde bulunan Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümünde görev yapan bir öğretim elemanı olmak üzere iki alan uzmanının görüşüne başvurulmuştur. Alınan uzman görüşleri doğrultusunda yapılandırılmış görüşme soruları üzerinde tekrar çalışılarak nihai soruları içeren yapılandırılmış görüşme formu oluşturulmuştur. Görüşme soruları "Algoritma konusunu öğrenirken kullanılan örnekler/benzetmeler konuyu anlamana yardımcı oldu mu? Bu konuda aklında kalan bir benzetme ya da örneği söyleyebilir misin? Algoritma konusunu sen bana anlatsan nasıl bir benzetme ile konuyu açıklarsın?" şeklinde farklı konu ve kazanımlar için oluşturulmuştur. Araştırmada her hafta metaforla programlama eğitiminden sonra öğrencilere uygulanan etkinlik algısı ölçeğine dair bulgular, öğrencilerle yapılan görüşmelerden elde edilen bulgularla birlikte değerlendirilmiştir.

Veri Toplama Süreci

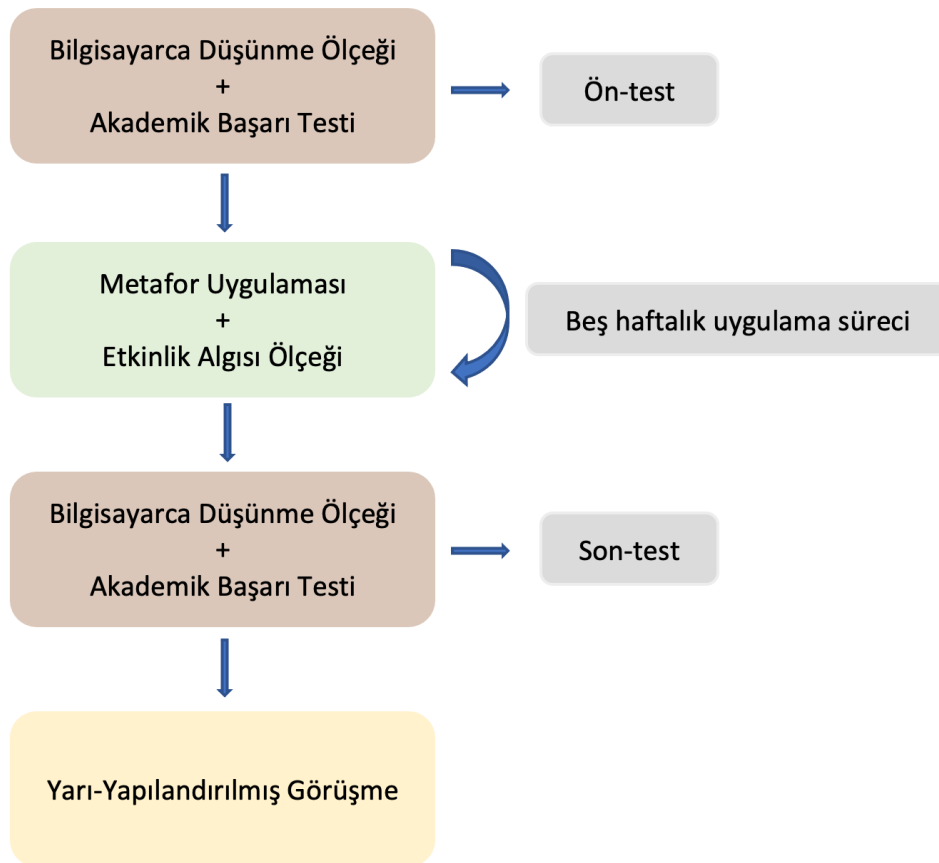
Öncelikle araştırmacı tarafından MEB müfredatı çerçevesinde metaforlarla programlama öğretimine uyumlu konular ve konular içerisinden kazanımlar seçilmiştir. Daha sonra araştırmacı tarafından belirlenen beş haftalık müfredat konularına yönelik kazanımlar için metaforlar ve metaforların dahil edildiği etkinlikler geliştirilmiştir. Geliştirilen metaforlar için MEB'e bağlı okullarda çalışan üç Bilişim Teknolojileri öğretmeni ve üniversitelerde eğitim fakültesinde bulunan Bilgisayar ve Öğretim Teknolojileri Eğitimi bölümünde görev yapan bir öğretim üyesi olmak üzere dört ilgili alan uzmanının görüşüne başvurulmuştur. Alınan görüşler

analiz edildiğinde, dört alan uzmanının beş haftalık metaforların ilgili konuların öğretiminde etkili olduklarını düşündükleri, belirttikleri eksik ya da hatalı metaforun olmadığını söyledikleri görülmüştür. Ayrıca bir alan uzmanı algoritma ile ilgili bir metafor önerisinde bulunmuş ve gerekli çalışmalar yapılarak örnek çalışmaya eklenmiştir. Alınan görüşler doğrultusunda geliştirilen metaforlar üzerinde tekrar çalışılarak nihai metaforlar ve bu metaforları içeren etkinlik rehberleri hazırlanmıştır. Geliştirilen metaforlara ve 5 haftalık etkinliklere ait örnekler Tablo 2’de sunulmuştur.

Tablo 2. Seçilen Bilişim Teknolojileri (BT) kazanımları ve ilgili metafor rehberi

| Hafta | İlişkili BT ve Yazılım dersi kazanımı | Metaforlar |
|-------|--|---|
| 1 | Sabit ve Değişken konusu ile ilgili kazanımlar: <ul style="list-style-type: none">● Problem çözmek için gerekli sabit kavramını açıklar.● Problem çözmek için gerekli değişken kavramını açıklar.● Belirli bir problem durumundaki sabit ve değişkenleri ayırt eder. | Sıcaklık, uzunluk, tabak, sınıf, yumurta, kek, ayran ve sayı metaforları kullanılmıştır. |
| 2 | Bellek konusu ile ilgili kazanımlar: <ul style="list-style-type: none">● Bellek kavramını açıklar.● Değişkenlerin bellekte tutulduğunu bilir. | Pazar, koli ve beyin metaforları kullanılmıştır. |
| 3 | Döngü konusu ile ilgili kazanımlar: <ul style="list-style-type: none">● Döngü yapısının işlevini açıklar.● Problem çözme sürecinde döngü yapısını kullanarak algoritma tasarlar.● Problem çözme sürecinde doğru döngü yapısına sahip algoritmayı seçer. | Işık, ayran, yumurta, kalem ve gözlerim bağlı metaforları kullanılmıştır. |
| 4 | Algoritma konusu ile ilgili kazanımlar: <ul style="list-style-type: none">● Algoritma kavramını açıklar.● Belirli bir problem için algoritma tasarlar.● Farklı algoritmaları inceleyerek doğru çözümü seçer. | Su şişesi, şeker, ayran, fotokopi makinası, masa hazırlama, el mikseri, beton mikseri ve çamaşır makinası metaforları kullanılmıştır. |
| 5 | Koşul konusu ile ilgili kazanımlar: <ul style="list-style-type: none">● Koşul yapısını açıklar.● Problem çözme sürecinde doğru koşul yapısına uygun algoritma tasarlar. | Seçilen konunun kazanımlarına uygun olarak hazırlanan kıyafet, 3 taş, içecek ve kutu metaforları kullanılmıştır. |

Uygulama süreci beş hafta olarak planlanmıştır. İlk hafta öğrencilere BDÖ ve ABT uygulanarak ön-test sonuçları elde edilmiştir. Beş hafta boyunca her hafta ilgili kazanıma yönelik planlanan metafor etkinliği öğrencilerle beraber sınıf içerisinde uygulanmıştır. Her hafta gerçekleştirilen etkinlikten sonra öğrencilere Etkinlik Algısı Ölçeği uygulanmıştır. Planlanan beş haftalık eğitimden sonra ise BDÖ ve ABT öğrencilere tekrar uygulanmış ve son-test sonuçları elde edilmiştir. Beş haftalık uygulama sonrasında ayrıca araştırmacı tarafından geliştirilmiş yapılandırılmış görüşme formu ile öğrencilerle görüşmeler gerçekleştirilmiştir. Yapılandırılmış görüşme formu tüm öğrencilere uygulanmamış olup, öğrenciler sınıflarına göre gruplara ayrılmıştır. Gruplar içerisinde akademik başarı testi son-test puanlarına göre alt ve üst grup oluşturulmuştur. Görüşme gerçekleştirilen 32 öğrenci toplam çalışma grubunun %64'ünü oluşturmaktadır. Yapılandırılmış görüşme formunun tüm öğrencilere uygulanmama sebebi öğrencilerin yoğun bir ders döneminde olmaları ve öğrencilerle görüşme yapmak için kullanılacak zamanın kısıtlı olmasıdır. Görüşme sırasında herhangi bir kayıt cihazı kullanılmamış olup öğrenci cevapları araştırmacı tarafından not edilerek görüşme kayıtları oluşturulmuştur. Uygulama ve veri toplama süreci özet olarak ayrıca Şekil 1'de gösterilmiştir.



Şekil 1. Uygulama ve veri toplama süreci

Verilerin Analizi

Verilerin analizinde SPSS 24 programı kullanılmıştır. Ön-test ve son-test olarak uygulanan Bilgisayarca Düşünme Ölçeği ve Akademik Başarı Testi sonuçları bağımlı gruplar t-testi (paired-samples t-test) ile analiz edilmiştir. İlişkili ölçümler aynı örneklem grubuna ait tekrarlı ölçümler veya eşleştirilmiş örneklemelerden elde edilen ölçümler için kullanılabilir (Büyüköztürk, 2012). Bu araştırmada aynı denek grubuna ait bir deneysel

işlemin öncesi ve sonrasındaki ölçümler arasında anlamlı bir farkın olup olmadığını gözlemlemek amacıyla tekrarlı ölçümler gerçekleştirilmiştir.

Etkinlik Algısı Ölçeğinin analizinde ise madde bazında ortalama ve standart sapma değerleri hesaplanmıştır. Yapılandırılmış görüşme formundaki evet/hayır şeklinde oluşan kapalı uçlu soruların cevaplarının analizinde betimsel istatistikler (yüzde ve frekans) kullanılmıştır. Yapılandırılmış görüşme formundaki açık uçlu sorulardan elde edilen nitel verilerin analizinde ise içerik analizi yönteminden yararlanılmıştır. Öncelikle betimsel analiz kullanılarak veriler özetlenerek yorumlanmıştır. Sonrasında betimsel analizde fark edilemeyen kavram ve temaların keşfedilmesi amacıyla içerik analizi uygulanmıştır (Yıldırım ve Şimşek, 2011). Bu süreç 4 aşamada gerçekleşmiştir: (1) verilerin kodlanması, (2) temaların bulunması, (3) kodların ve temaların düzenlenmesi, (4) bulguların tanımlanması ve yorumlanmasıdır (Yıldırım ve Şimşek, 2011). Yapılandırılmış Görüşme Formunda elde edilen veriler Etkinlik Algısı Ölçeğinde elde edilen sonuçları yorumlamak ve desteklemek amacıyla birlikte değerlendirilmiştir. Yapılandırılmış görüşme formuna ait alıntılar sunulurken öğrenciler öğrenci 1 (Ö1), öğrenci 2 (Ö2), öğrenci 3 (Ö3) gibi kısaltmalarla temsil edilmiştir.

Araştırmanın Geçerlik ve Güvenirliği

Araştırma sürecinde araştırmacı, veri toplama süreçlerini kendisi yönetmiş ve her hafta aynı araştırmacı tarafından veriler toplanmıştır. Araştırma verileri katılımcıların kendi sınıf ortamında toplanmıştır. Araştırma sürecinde kullanılan veri toplama araçlarından araştırmacı tarafından geliştirilen Akademik başarı testi için hem alandan bilişim teknolojileri öğretmenleri hem de akademisyen görüşleri alınarak nihai hali oluşturulmuştur. Araştırma sürecinde kullanılan diğer veri toplama araçları geçerlik ve güvenilirlik çalışmaları yayınlanmış araçlar arasından seçilmiştir.

Nitel araştırmalarda geçerlik ve güvenirliliğin sağlanması için tekrar edilebilir nitelikte olması ve uygulama sürecinin sistematik bir planda aktarılması önemlidir (Patton, 1999). Bu çalışmada veri toplama ve analiz süreci detaylı bir şekilde aktarılmıştır. Ayrıca görüşmelerden doğrudan alıntılar paylaşılmıştır. Son olarak tüm bu süreçte araştırmacı, ortak bir çalışma yürüterek verilerin analizinde birden fazla kişinin görüşünü alarak ilerlemiştir.

Bulgular

Araştırmanın bulguları alt araştırma soruları ile uyumlu olarak sunulmuştur.

Programlama Öğretiminde Metafor Kullanımına göre Öğrencilerin Akademik Başarılarına İlişkin Bulgular

Akademik Başarı Testi öğrencilerin seçilen kazanımlara ait ön bilgilerini tespit etmek amacıyla öncelikle ön-test olarak uygulanmıştır. Beş haftalık metaforlarla programlama öğretiminden sonra öğrencilere tekrar akademik başarı testi uygulanmış olup öğrencilerin ön-test ve son-test puanları arasındaki farkın anlamlı olup olmadığı gözlemlenmiştir. İlk olarak testin normalliğini kontrol etmek için normallik testleri yapılmıştır. Katılımcı sayısı 50'den fazla olduğu için ($n > 50$) normallik Kolmogorov-Smirnov Testi ile değerlendirilmiştir (Field, 2011). Kolmogorov-Smirnov testinin sonuçları Tablo 3'te gösterilmiştir.

Tablo 3. Akademik başarı testine ait normallik test sonuçları

| FarkPuan | Normallik Testleri | | | | | |
|----------|---------------------------------|----|-----------|--------------|----|-----------|
| | Kolmogorov-Smirnov ^a | | | Shapiro-Wilk | | |
| | İstatistik | sd | Anamlılık | İstatistik | sd | Anamlılık |
| | ,095 | 50 | ,200 | ,967 | 50 | ,179 |

Tablo 3'te Kolmogorov-Smirnov testi sonuçlarına göre $p > .05$ olduğundan veri setinin normal dağılım gösterdiği görülmüştür. Devamında "Öğrencilerin akademik başarı testi son-test ortalama puanları ön-test ortalama puanlarından anlamlı bir şekilde yüksek midir?" sorusu bağımlı gruplar t-testi sonuçlarına göre değerlendirilmiştir. Öğrencilerin, metaforlarla programlama öğretimi etkinlikleri öncesi ve sonrasındaki akademik başarı testi sonuçlarının karşılaştırıldığı bağımlı gruplar t-testi sonuçları Tablo 4'te sunulmuştur.

Tablo 4. Öğrencilerin akademik başarı testine ait ön test-son test puanlarının bağımlı gruplar t-testi sonuçları ve etki büyüklüğü (r) değeri

| | Testler | N | \bar{X} | ss | sd | t | p | r |
|------------------------------|---------|----|-----------|-------|-------|-------|------|-----|
| Akademik Başarı Testi | ÖnTest | 50 | 33.30 | 15.67 | | | | |
| | SonTest | 50 | 83.14 | 11.65 | 13.53 | 26.03 | .000 | 0.9 |

Tablo 4'te görüldüğü gibi akademik başarı testine ait bağımlı gruplar t-Testi sonuçlarına bakıldığında öğrencilerin son-test ortalamalarının ($\bar{X}=83.14$) ön-test ortalamalarından ($\bar{X}=33.30$) yüksek olduğu tespit edilmiştir. Bu farkın da istatistiksel olarak anlamlı olduğu ($t=26.03$, $p=.000$) görülmüştür. Etki büyüklüğü 0 ile 1 arasında bir değer almaktadır. Bu çalışmada akademik başarı testinin etki büyüklüğü $r=0.9$ olarak hesaplanmış olup 0.5'in üzerinde olduğu için etki büyüklüğünün büyük olduğu gözlemlenmiştir (Field, 2011).

Programlama Öğretiminde Metafor Kullanımına göre Öğrencilerin Bilgisayarca Düşünme Becerilerine İlişkin Bulgular

"Bilgisayarca Düşünme Ölçeği" öğrencilere ön-test ve son-test olarak uygulanmıştır. Uygulanan ön-test ile öğrencilerin bilgisayarlı düşünme becerilerine göre var olan durum tespitinin yapılması hedeflenmiştir. Beş haftalık metaforlarla programlama öğretiminden sonra öğrencilere tekrar Bilgisayarca Düşünme Ölçeği uygulanmış olup öğrencilerin ön-test ve son-test puanları arasındaki farkın anlamlı olup olmadığı gözlemlenmiştir.

İlk olarak testin normalliğini kontrol etmek için normallik testleri yapılmıştır. Katılımcı sayısı 50'den fazla olduğu için ($n > 50$) normallik Kolmogorov-Smirnov Testi ile değerlendirilmiştir (Field, 2011). Kolmogorov-Smirnov testinin sonuçları Tablo 5'te gösterilmiştir.

Tablo 5. Bilgisayarca düşünme ölçeğine ait normallik test sonuçları

| FarkPuan | Normallik Testleri | | | | | |
|----------|---------------------------------|----|-----------|--------------|----|-----------|
| | Kolmogorov-Smirnov ^a | | | Shapiro-Wilk | | |
| | İstatistik | sd | Anamlılık | İstatistik | sd | Anamlılık |
| | ,095 | 50 | ,200* | ,963 | 50 | ,120 |

Tablo 5'te Kolmogorov-Smirnov testi sonuçlarına göre $p > .05$ olduğundan veri setinin normal dağılım gösterdiği görülmüştür. Devamında "Öğrencilerin bilgisayarca düşünme ölçeğinden elde ettiği son-test ortalama puanları ön-test ortalama puanlarından anlamlı bir şekilde yüksek midir?" sorusu bağımlı gruplar t-testi sonuçlarına göre değerlendirilmiştir. Öğrencilerin, metaforlarla programlama öğretimi etkinlikleri öncesi ve sonrasındaki bilgisayarca düşünme ölçeği sonuçlarının karşılaştırıldığı bağımlı gruplar t-testi sonuçları Tablo 6'da sunulmuştur.

Tablo 6. Öğrencilerin bilgisayarca düşünme ölçeğine ait ön test-son test puanlarının bağımlı gruplar t-testi sonuçları ve etki büyüklüğü (r) değeri

| | Testler | N | \bar{X} | ss | sd | t | p | r |
|------------------------------------|----------------|----|-----------|------|------|------|------|------|
| Bilgisayarca Düşünme Ölçeği | ÖnTest | 50 | 72.62 | 9.60 | | | | |
| | SonTest | 50 | 80.36 | 5.86 | 8.38 | 6.52 | .000 | 0.68 |

Tablo 6'da görüldüğü gibi Bilgisayarca Düşünme Ölçeğine ait bağımlı gruplar t-Testi sonuçlarına göre son test puan ortalamasının ($\bar{X} = 80.36$) ön test puanı ortalamasına ($\bar{X} = 72.62$) göre daha yüksek olduğu tespit edilmiştir. Bu farkın da istatistiksel olarak anlamlı ($t = 6.52$, $p < 0.05$) olduğu sonucuna ulaşılmıştır. Bu çalışmada Bilgisayarca düşünme ölçeğinin etki büyüklüğü $r = 0.68$ olarak hesaplanmış olup 0.5'in üzerinde olduğu için etki büyüklüğünün büyük olduğu gözlemlenmiştir (Field, 2011).

Programlama Öğretiminde Metafor Kullanımına İlişkin Öğrenci Görüşleri

Beş hafta boyunca metaforlarla ders anlatımı sürecinde, her hafta seçilen konuya dair metaforlar öğrenciler ile uygulanmış olup ders sonunda öğrencilere Etkinlik Algısı Ölçeği uygulanmıştır. Öğrencilerin etkinlik algısı ölçeğinden aldıkları puanların ortalama ve standart sapma değerleri hesaplanmıştır. 11 maddeden oluşan ölçek maddeleri 4 faktör etrafında gruplandırılarak açıklanmıştır. Bunlar: (1) Deneyimleri eğlenceli bulma, (2) Etkinliklerin kişisel gelişime katkısı, (3) Etkinlikleri yapmaya ilişkin isteklilik ve (4) Etkinlikleri ilgi çekici bulmadır (Kasalak, 2017).

Metaforlarla Programlama Öğrenme Etkinliklerini Eğlenceli Bulmaya İlişkin Etkinlik Algısı

Öğrencilerin metaforlarla programlama öğrenme etkinliklerini eğlenceli bulmaya ilişkin algılarına yönelik hazırlanan maddelere verdikleri puanların ortalama ve standart sapma değerleri Tablo 7'de gösterilmiştir.

Tablo 7. Metaforlarla programlama öğrenme etkinliklerini eğlenceli bulmaya ilişkin maddelerin ortalama ve standart sapma değerleri

| | 1.Bu etkinliği yapmak eğlenceliydi. | | 3.Bu etkinliği yaparken çok eğlendim. | | 6.Bence bu sıkıcı bir etkinlikti. | |
|--------------------|-------------------------------------|------|---------------------------------------|------|-----------------------------------|------|
| | \bar{X} | SS | \bar{X} | SS | \bar{X} | SS |
| 1. Hafta Etkinliği | 4.72 | 0.53 | 4.70 | 0.58 | 1.10 | 0.36 |
| 2. Hafta Etkinliği | 4.80 | 0.40 | 4.68 | 0.51 | 1.28 | 0.60 |
| 3. Hafta Etkinliği | 4.86 | 0.40 | 4.80 | 0.40 | 1.38 | 1.00 |
| 4. Hafta Etkinliği | 4.82 | 0.38 | 4.78 | 0.46 | 1.14 | 0.45 |
| 5. Hafta Etkinliği | 4.92 | 0.27 | 4.80 | 0.40 | 1.20 | 0.57 |

Tablo 7 incelendiğinde 5 haftaya yönelik “Bu etkinliği yapmak çok eğlenceliydi” maddesine verilen ortalama puanların en düşük 4.72, en yüksek 4.92 olduğu, “Bu etkinliği yaparken çok eğlendim” maddesinden alınan en düşük ortalama puanın 4.68, en yüksek ortalama puanın ise 4.80 olduğu görülmektedir. “Bence bu çok sıkıcı bir etkinlikti.” maddesine verilen ortalama puanların ise 1.10 ile 1.38 arasında değişmekte olduğu görülmektedir. Elde edilen sonuçlara göre öğrencilerin metafor etkinliklerini eğlenceli bulmaya ilişkin etkinlik algı düzeylerinin oldukça yüksek, sıkıcı bulma algı düzeylerinin ise oldukça düşük olduğunu göstermektedir.

Metaforlarla Programlama Öğrenme Etkinliklerinin Kişisel Gelişime Etkisine İlişkin Etkinlik Algısı

Öğrencilerin metaforlarla programlama öğrenme etkinliklerinin kişisel gelişime etkisine ilişkin algılarına yönelik hazırlanan maddelere verdikleri puanların ortalama ve standart sapma değerleri Tablo 8’de gösterilmiştir.

Tablo 8. Metaforlarla Programlama Etkinliklerinin Kişisel Gelişime Etkisine İlişkin Etkinlik Algısı Sorularına Ait Ortalama ve Standart Sapma Değerleri

| | 2. Bu etkinliğin benim gelişimim için önemli olduğuna inanıyorum. | | 4. Bence bu gerçekten önemli bir etkinlikti. | | 7. Bu etkinliği faydalı olduğunu düşündüğüm için tekrar yapmak isterim. | | 8. Bu etkinliği yapmanın benim için faydalı olabileceğine inanıyorum. | | 9. Bu etkinliğin okulda daha iyi olmama yardımcı olabileceğine inanıyorum. | | 11. Bu etkinliğin bana kattığı bazı şeyler olduğu için tekrar yapmak isterim. | |
|--------------------|---|------|--|------|---|------|---|------|--|------|---|------|
| | \bar{X} | SS | \bar{X} | SS | \bar{X} | SS | \bar{X} | SS | \bar{X} | SS | \bar{X} | SS |
| 1. Hafta Etkinliği | 4.40 | 0.57 | 4.32 | 0.51 | 4.30 | 0.58 | 4.46 | 0.54 | 4.38 | 0.63 | 4.44 | 0.57 |
| 2. Hafta Etkinliği | 4.62 | 0.53 | 4.46 | 0.64 | 4.48 | 0.57 | 4.56 | 0.61 | 4.46 | 0.61 | 4.56 | 0.54 |
| 3. Hafta Etkinliği | 4.52 | 0.67 | 4.44 | 0.67 | 4.50 | 0.86 | 4.54 | 0.57 | 4.42 | 0.67 | 4.68 | 0.62 |
| 4. Hafta Etkinliği | 4.76 | 0.43 | 4.70 | 0.54 | 4.68 | 0.55 | 4.72 | 0.53 | 4.64 | 0.63 | 4.68 | 0.71 |
| 5. Hafta Etkinliği | 4.72 | 0.45 | 4.66 | 0.55 | 4.72 | 0.49 | 4.70 | 0.58 | 4.80 | 0.40 | 4.82 | 0.48 |

Tablo 8 incelendiğinde “Bu etkinliğin benim gelişimim için önemli olduğuna inanıyorum.” maddesine verilen ortalama puanların en düşük 4.40, en yüksek 4.76 olduğu ve “Bence bu gerçekten önemli bir etkinlikti.” maddesine verilen ortalama puanların en düşük 4.32, en yüksek 4.70 olduğu görülmektedir. “Bu etkinliği faydalı olduğunu düşündüğüm için tekrar yapmak isterim.” maddesine verilen ortalama puanların en düşük 4.30, en yüksek 4.72 olduğu ve “Bu etkinliği yapmanın benim için faydalı olabileceğine inanıyorum.” maddesine verilen ortalama puanların en düşük 4.46, en yüksek 4.72 olduğu görülmektedir. “Bu etkinliğin okulda daha iyi olmama yardımcı olabileceğine inanıyorum.” maddesine verilen ortalama puanların en düşük 4.38, en yüksek 4.80 olduğu ve “Bu etkinliğin bana kattığı bazı şeyler olduğu için tekrar yapmak isterim.” maddesine verilen ortalama puanların en düşük 4.44, en yüksek 4.82 olduğu görülmektedir. 5 haftaya ait metaforlarla programlama öğrenme etkinliklerinin öğrencilerin kişisel gelişimlerine ilişkin maddelere verdikleri puan ortalamalarının 4.30 ile 4.82 aralığında olduğu görülmektedir. Elde edilen sonuçlara bakılarak öğrencilerin metaforlarla programlama öğrenmeye yönelik algılarında bu etkinliklerin kişisel gelişimlerine katkısının olduğunu düşündükleri ifade edilebilir. Özellikle madde puanları incelendiğinde öğrencilerin yaptığı etkinliklerin onlara kattığı bazı şeyler olduğuna inandıkları için aynı etkinliği tekrar yapmak istedikleri görülmektedir. Ayrıca öğrencilerin katıldıkları etkinliklerin onların okulda daha iyi olmalarına katkı sağladığını düşündükleri de görülmektedir.

Metaforlarla Programlama Öğrenme Etkinliğini Yapmaya İlişkin İsteklilik ve Etkinliği İlgili Çekici Bulma

Öğrencilerin metaforlarla programlama öğrenme etkinliklerini yapmaya ilişkin isteklilik (gönüllü olmaya ilişkin) algısına ve etkinliği ilgi çekici bulmaya ilişkin algısına yönelik maddelere verdikleri puanların ortalama ve standart sapma değerleri Tablo 9’da gösterilmiştir.

Tablo 9. Metaforlarla programlama öğrenme etkinliklerini yapma isteğine ve etkinliği ilgi çekici bulmaya ilişkin maddelere ait ortalama ve standart sapma değerleri

| | 5. Bu etkinliği yapmak istediğim için yaptım. | | 10. Bunun çok ilgi çekici bir etkinlik olduğunu düşündüm. | |
|--------------------|---|------|---|------|
| | \bar{X} | SS | \bar{X} | SS |
| 1. Hafta Etkinliği | 4.56 | 0.57 | 4.32 | 0.58 |
| 2. Hafta Etkinliği | 4.68 | 0.51 | 4.36 | 0.66 |
| 3. Hafta Etkinliği | 4.56 | 0.88 | 4.62 | 0.66 |
| 4. Hafta Etkinliği | 4.86 | 0.40 | 4.76 | 0.55 |
| 5. Hafta Etkinliği | 4.88 | 0.32 | 4.78 | 0.46 |

Tablo 9 incelendiğinde “Bu etkinliği yapmak istediğim için yaptım.” maddesine verilen ortalama puanların en düşük 4.56, en yüksek 4.88 olduğu görülürken “Bunun çok ilgi çekici bir etkinlik olduğunu düşündüm.” maddesine verilen ortalama puanların da en düşük 4.32, en yüksek 4.78 olduğu görülmektedir. Elde edilen sonuçlara bakıldığında öğrencilerin yapılan etkinlikleri yapmak istedikleri için yaptıkları, etkinlikleri yapma konusunda olumsuz bir algıya sahip olmadıkları ve öğrencilerin etkinlikleri ilgi çekici buldukları söylenebilir.

Beş haftalık uygulama sonrasında ayrıca araştırmacı tarafından geliştirilmiş yapılandırılmış görüşme formu ile öğrencilerle görüşmeler gerçekleştirilmiştir. Görüşmelerde öncelikle öğrencilere 5 haftalık uygulama sürecinde anlatılan her konu için geliştirilen etkinlik ve metaforların konuyu anlamaya yardımcı olma durumu sorulmuştur. Öğrencilere yöneltilen “... konusunu öğrenirken kullanılan örnekler konuyu anlamaya yardımcı oldu mu?” sorusuna verilen cevaplar incelendiğinde öğrencilerin %100’ünün (f=32) 5 haftalık her uygulama/konu için olumlu yanıt verdiği görülmektedir. Böylece araştırmacı tarafından hazırlanan örneklerin öğrencilere konuyu aktarma konusunda etkili olduğu sonucuna ulaşılabılır.

Ayrıca öğrencilere 5 haftalık uygulama sürecinde her hafta o haftaki etkinliklerde kullanılan metaforlardan veya örneklerden akıllarında kalanlar sorulmuştur. Öğrencilere yöneltilen “Bu konuda aklında kalan bir benzetme ya da örneği söyleyebilir misin?” sorusuna verilen cevaplar incelendiğinde sabit-değişken konusunda öğrencilerin %93.75’inin (f=30), bellek konusunda öğrencilerin %100’ünün (f=32), algoritma konusuna öğrencilerin %100’ünün (f=32), döngü konusuna öğrencilerin %100’ünün (f=32) ve koşul konusuna ise öğrencilerin %96.9’unun (f=31) derste kullanılan metaforlardan en az bir tanesini hatırlayarak örneklendirdiği görülmektedir. Bir öğrencinin ise koşul konusunda kullanılan metaforlara dair bir örnek sunamadığı görülmüştür. Öğrencilerin örneklendirdikleri metaforlar Tablo 10’da sunulmuştur.

Tablo 10. Öğrenciler tarafından hatırlanan metaforlar/örnekler

| Konu | Metafor hatırlama durumu | f | % | Hatırlanan Metaforlar |
|-----------------------|----------------------------------|----|-------|--|
| Sabit-Değişken | Metafor sunan öğrenci sayısı | 30 | 93.75 | Kek yapma (13) Yumurta (11) Tabak (6) |
| | Metafor sunamayan öğrenci sayısı | 2 | 6.25 | |
| Bellek | Metafor sunan öğrenci sayısı | 32 | 100 | Beyin (14) Koli (13) Pazar poşeti (5) |
| | Metafor sunamayan öğrenci sayısı | 0 | 0 | |
| Algoritma | Metafor sunan öğrenci sayısı | 32 | 100 | Origami (27) Ayran (5) |
| | Metafor sunamayan öğrenci sayısı | 0 | 0 | |
| Döngü | Metafor sunan öğrenci sayısı | 32 | 100 | Fotokopi Makinesi (16) Şeker dağıtma (8) Çatal-kaşık (2) El mikseri (6) |
| | Metafor sunamayan öğrenci sayısı | 0 | 0 | |
| Koşul | Metafor sunan öğrenci sayısı | 31 | 96.9 | Kıyafet seçimi (14) Üçtaş oyunu (12) Kapak (5) |
| | Metafor sunamayan öğrenci sayısı | 1 | 3.1 | |
| Toplam | | 32 | 100 | |

Derste kullanılan metaforları örneklendirebilen öğrencilerin cevapları incelendiğinde en çok verilen örneklerin sabit-değişken konusunda kek yapma (f=13), bellek konusunda beyin (f=14), algoritma konusunda origami (f=27), döngü konusunda fotokopi makinası (f=16) ve koşul konusunda kıyafet seçimi (f=14) olduğu tespit edilmiştir. Öğrencilerin “5 haftadır kullandığımız metaforlar/benzetmeler/örnekler veya etkinlikler içerisinde sence en etkili ya da güzel olanı hangisiydi?” sorusuna ise dört farklı örnek sundukları görülmüştür. Öğrencilerin 23’ü origami (%71.9), 5’i fotokopi makinası (%15.7), 2’si üçtaş oyunu (%6.2) ve 2’si de beyin (%6.2) örneklerini vermişlerdir. Öğrencilerin etkinlikleri beğenmesinde eğlence, oyun, daha önceden oynadığı oyunlarla ilişkilendirme ve somutlaştırma öne çıkmaktadır. Bazı öğrenciler etkili ya da güzel olarak tanımladıkları etkinlikler için aşağıdaki ifadeleri kullanmıştır:

“Bence origamiydi çünkü sınıfta hem konuyu anladık hem eğlendik hem de dersten sonra oynayacağımız bir oyuncağımız oldu.” (Ö8)

“Fotokopi makinasıydı. 10 sayfa adınızı yazın dediğinizde nasıl yazacağımı o kadar çok diye düşündüm ve sonra fotokopi makinası örneğini verdiğiniz için aklımda kaldı.” (Ö9)

“Köyde hep oynardık üçtaş oyunu. Siz örneği verince aklımda kaldı.” (Ö10)

Görüşmelerde ayrıca öğrencilerin programlama öğretiminde metafor kullanımına dair görüşleri incelenmiştir. Öğrencilere bu amaçla yöneltilen “Programlama/kodlama konularını metaforlarla yani çeşitli benzetmelerle öğrenmenin konuyu anlamana yardımcı olduğunu düşünüyor musun?” sorusuna öğrencilerin tamamının (f=32, %100) evet yanıtı verdiği görülmektedir. Öğrencilere ayrıca derste kullanılan etkinlikler ve örneklerin konuyu hangi noktalarda daha rahat anlamalarına yardımcı olduğu sorulmuştur. Öğrencilerin bu soruya verdikleri cevaplar 7 kategori çerçevesinde açıklanmıştır. Bu kategoriler: (1) Akılda kalıcılığı artırma, (2) Anlaşılabilirlik, (3) Eğlence, (4) Tekrar kolaylığı, (5) Somutlaştırma, (6) İlgı çekicilik ve (7) Anlatımı zenginleştirmedir. Öğrencilerden bazıları metafor etkinliklerinin öğrenme süreçlerini nasıl desteklediğini aşağıdaki ifadelerle açıklamıştır:

“Annem kek yaparken ona yardım ettim o sırada dersi tekrar etmiş gibi oldum.” (Ö6)

“Konuları böyle anlatınca verdiğiniz örnekler aklımda daha kolay kaldı.” (Ö4)

“Sınıfta yaptığımız etkinlikler çok eğlenceliydi mesela ben algoritma konusunu origami ile kolay anladım” (Ö5)

Öğrencilerin uygulama sonrasında da programlama öğretimine benzer etkinliklerle devam etmek istedikleri görülmüştür. Öğrencilerin 31’i (%96.9) benzer etkinliklerle derse devam etmek istediklerini ifade etmişlerdir. Benzer etkinliklerle derse devam etmek istemeyen bir öğrenci de metaforlarla gerçekleştirilen etkinliklerde bilgisayar açılmadığı için doğrudan bilgisayarla olan etkinlikleri tercih ettiğini aşağıdaki şekilde belirtmiştir:

“Konuları böyle işlediğimizde bilgisayar açamıyoruz ben bilgisayarlı etkinlikleri seviyorum” (Ö7)

Ayrıca görüşmeler sonucunda öğrencilerin başka derslerde de metaforların uygulanmasını istedikleri görülmüştür. Öğrencilerin tamamı (f=32) başka derslerde metaforlarla etkinlikler düzenlenmesi istediğini belirtmiştir. Öğrencilerden 22’si matematik (%68.8), 7’si fen bilimleri (%21.9) ve 3’ü sosyal bilgiler (%9.3) dersinde benzer etkinlikleri kullanmak istediklerini ifade etmiştir.

Sonuç olarak hem etkinlik algısı ölçeğinden elde edilen sonuçlar hem de yapılandırılmış görüşmelerden elde edilen sonuçlar doğrultusunda öğrencilerin metafor etkinliklerini eğlenceli bulduğu görülmektedir. Yapılandırılmış görüşmelerden elde edilen sonuçlarda öğrencilerin 31’i (%96.87) programlama eğitimine benzer etkinliklerle devam etmek istediklerini belirtirken bu öğrencilerden 21’i (%65.62) de etkinliklerin eğlenceli olmasını, etkinlikleri yaparken mutlu olduklarını ve etkinliklerin çok zevkli değişik olması nedeniyle zamanın nasıl geçtiğini anlamadıklarını vurgulamıştır. Öte yandan öğrencilerin tamamı (f=32, %100) programlama konularını metaforlarla öğrenmenin konuyu anlamalarına yardımcı olduğunu belirtmiştir. Benzer şekilde etkinlik algısı ölçeğinde “Bu etkinliği yapmak çok eğlenceliydi.” maddesine verilen ortalama puanların en düşük 4.72, en yüksek 4.92 olduğu, “Bu etkinliği yaparken çok eğlendim.” maddesinden alınan en düşük ortalama puanın 4.68, en yüksek ortalama puanın ise 4.80 olduğu görülmektedir (Bkz. Tablo 7). Ayrıca yapılandırılmış görüşmelerde öğrencilerin tamamının (f=32, %100) programlama/kodlama konularını metaforlarla öğrenmenin konuyu anlamalarına yardımcı olduğunu düşündükleri görülmektedir. Benzer şekilde öğrencilerin etkinlik algısı ölçeğinde “Bu etkinliğin benim gelişimim için önemli olduğuna inanıyorum.” maddesine verilen ortalama puanların en düşük 4.40, en yüksek 4.76 olduğu ve “Bu etkinliği faydalı olduğumu düşündüğüm için tekrar yapmak isterim.” maddesine verilen ortalama puanların en düşük 4.30, en yüksek 4.72 olduğu

görülmektedir. Ayrıca “Bu etkinliği yapmanın benim için faydalı olabileceğine inanıyorum.” maddesine verilen ortalama puanların en düşük 4.46, en yüksek 4.72 olduğu görülmektedir (Bkz. Tablo 8). Son olarak öğrencilerin etkinliği yapmaya ilişkin isteklilik durumu ve etkinliği ilgi çekici bulmak noktasındaki görüşleri incelenmiştir. Yapılandırılmış görüşmeden elde edilen sonuçlarda da öğrencilerin tamamının (f=32, %100) programlama eğitimi süresince benzer etkinliklerle devam etmek istedikleri görülmektedir. Ayrıca öğrencilerin tamamının (f=32, %100) başka derslerde de benzer etkinlikleri kullanmak istedikleri görülmektedir. Buna ek olarak öğrencilerin 18’i metaforlarla programlama öğreniyorum etkinlikleri ile katıldıkları derslerin diğer derslerden farklı şeklinde işlendiğini ve bunun ilgilerini çektiğini belirtmiştir. Etkinlik algısı ölçeğinde de “Bu etkinliği yapmak istediğim için yaptım.” maddesine verilen ortalama puanların en düşük 4.56, en yüksek 4.88 olduğu görülürken “Bunun çok ilgi çekici bir etkinlik olduğunu düşündüm.” maddesine verilen ortalama puanların da en düşük 4.32, en yüksek 4.78 olduğu görülmektedir.

Sonuç ve Tartışma

Bu çalışmanın amacı programlama öğretiminde metafor kullanımının erken yaşta programlama öğretim sürecine etkisinin gözlemlenmesidir. Metafor kullanımının programlama öğretimine etkisi öğrencilerin akademik başarısındaki anlamlı değişim, bilgisayarlı düşünme becerilerindeki anlamlı değişim ve etkinlik algılarına ait görüşleri üzerinden incelenmiştir.

Çalışmanın verilerine göre erken yaşta programlama öğretiminde metafor kullanımı ile öğrencilerin akademik başarı puan ortalamasında anlamlı ve etki büyüklüğü yüksek bir değişim ortaya çıkmıştır. Benzer şekilde Pérez-Marín vd. (2018) çalışmalarında programlama kavramlarının öğretiminde metafor kullanımının başarılı sonuçlar verdiğini vurgulamıştır. Bir diğer çalışma da Pérez-Marín vd. (2020) ilköğretim öğrencilerinin bilişim teknolojileri becerilerini geliştirmek ve bilgisayar programlama kavramlarını kavramasını desteklemek için metafor kullanımı sonucunda benzer şekilde başarılı sonuçlar elde etmiştir. Esgil ve Gündüz’ün (2019) yaptıkları çalışma sonucunda görsel ve somut örneklerle gerçekleştirilen programlama öğretim sürecinde öğrencilerin derse katılım düzeylerinin arttığını belirtmiştir. Öte yandan, Küçük ve Şişman (2016) ise yaptıkları çalışma sonucunda robotik kodlamanın öğrencilerin motivasyonlarını arttırdığını ancak derse karşı tutumlarını etkilemediği sonucuna ulaşmışlardır. Oysaki metaforlarla programlama öğretimi etkinlikleri sonrasında öğrencilerin eğlenerek öğrendikleri, motivasyonlarının arttığı ve derslerini benzer etkinliklerle işlemek istedikleri görülmektedir.

Bu çalışmada ayrıca öğrencilere metaforlarla programlama öğreniyorum etkinlikleri öncesi ve sonrasında bilgisayarlı düşünme ölçeği uygulanmıştır. Bu ölçekten elde edilen veriler incelendiğinde ise öğrencilerin ölçek ortalama puanlarında anlamlı ve etki büyüklüğü yüksek bir değişim gözlemlenmiştir. Böylece programlama öğretiminde metafor kullanımının öğrencilerin bilgisayarlı düşünme becerisine olumlu yönde bir etkisinin olduğu söylenebilir. Oluk ve Korkmaz (2016) çalışmasında programlama becerisi ile bilgisayarlı düşünme becerisi arasında ilişki olduğunu vurgulamıştır. Portelance (2015) da çalışmasında erken yaşta programlama öğretiminin bireylerin bilgisayarlı düşünme becerilerini geliştirmede faydalı olduğunu ifade etmiştir. Brennan ve Resnick (2012) çalışmalarında kodlama alanında yapılan çalışmaların öğrencilerin bilgisayarlı düşünme becerilerinde gelişim gösterdiğini belirtmişlerdir. Bers vd. (2014) çalışmalarında, programlamanın bilgisayarlı düşünme becerileri üzerindeki etkisini incelemiş ve programlamanın bilgisayarlı düşünme becerilerini

geliştirdiği sonucuna ulaşmışlardır. Pérez-Marín vd. (2020) çalışmalarında metaforlar ile programlama öğretiminin öğrencilerin bilgisayarlı düşünme becerilerinde artışa neden olduğunu vurgulamıştır. Böylece programlama öğretiminde metafor kullanımının olumlu katkıları bilgisayarlı düşünme becerilerinin geliştirilmesi ile ilişkilendirilebilir. Öte yandan, Uslu vd. (2018) ortaokul seviyesindeki öğrencilerin bilgi-işlemsel düşünme becerilerini geliştirmek amacıyla bir çalışma gerçekleştirmişlerdir. Bu çalışma sonucunda ise görsel programlama etkinliklerinin öğrencilerin bilgi-işlemsel düşünme becerilerinde anlamlı bir artışa sebep olmadığı belirtilmiştir. Kalelioğlu ve Gülbahar (2014) da Scratch ile uyguladıkları programlama öğretim yaklaşımının öğrencilerin problem çözme becerileri üzerindeki etkisini araştırmışlardır. Çalışmanın sonucunda öğrencilerin problem çözme becerilerinde anlamlı farklılık olmadığı sonucuna ulaşmışlardır. Bunlarla birlikte Lai ve Yang (2011) tarafından yapılan çalışmada da programlama öğretimi için kullandıkları görsel programlama yaklaşımı ile öğrencilerin mantıksal düşünme becerileri üzerinde anlamlı bir değişim olmadığını belirtmişlerdir. İncelenen çalışma sonuçları ve bu çalışmadan elde edilen bulgular neticesinde farklı yöntemlere karşın metaforlarla programlama etkinliklerinin bilgisayarlı düşünme becerilerini geliştirmede kullanılabilir bir yöntem olduğu söylenebilir.

Öğrencilerin katıldıkları “Metaforlarla Programlama Öğreniyorum” etkinliklerine yönelik eğlence, yararlı bulma ve isteklilik gibi algıları Etkinlik Algısı Ölçeğinden elde edilen veriler ile değerlendirilmiştir. Elde edilen sonuçlara göre öğrencilerin metafor etkinliklerini eğlenceli bulmaya ilişkin etkinlik algı düzeylerinin oldukça yüksek, sıkıcı bulma algı düzeylerinin ise oldukça düşük olduğu görülmektedir. Elde edilen sonuçlara bakılarak öğrencilerin metaforlarla programlama öğrenmeye yönelik algılarında bu etkinliklerin kişisel gelişimlerine katkısının olduğunu düşündükleri ifade edilebilir. Ayrıca öğrencilerin yapılan etkinlikleri yapmak istedikleri için yaptıkları, etkinlikleri yapma konusunda olumsuz bir algıya sahip olmadıkları ve öğrencilerin etkinlikleri ilgi çekici buldukları söylenebilir.

Öğrencilerle yapılan görüşme sonuçlarına göre ise araştırmacı tarafından hazırlanan örneklerin öğrencilere konuyu aktarma konusunda başarılı olduğu görülmüştür. Töremen ve Döş (2009) çalışmasında metaforları bir konuyu anlatmada sözcüklerin yetersiz kalması durumunda kullanılabilir çok faydalı bir teknik olarak tanımlamaktadır. Ocak ve Gündüz (2006) ise çalışmasında metaforların soyut konuyu daha somut konular ile ifade etmeyi desteklediğine ve anlaşılması zor olan soyut olan konunun anlaşılmasına imkân sağladığına vurgu yapmıştır. Araştırma bulguları soyut olan programlama kavramlarının öğreniminde metafor kullanımının öğrencilere konu aktarımı noktasında etkili bir yöntem olduğunu destekler niteliktedir. Ayrıca araştırmacı tarafından öğrencilere uygulanan metafor/benzetmelerin öğrencilerin aklında kaldığı ve öğrencilerinin çoğunun uygun bir metafor/benzetme ile konuyu açıklayabildiği sonucuna ulaşılmıştır. Arslan ve Bayrakçı (2006) çalışmasında metaforların sadece yeni konunun öğretilmesi için kullanılacak bir teknik olmadığını vurgulamıştır. Çalışmada metaforların yeni öğrenilen konunun akılda daha iyi kalması ve konunun zihinde daha iyi yer edebilmesi için kullanılan bir araç olduğu vurgulanırken ayrıca metaforlar aracılığıyla öğrencilerin mevcut şemalarının üzerine taşıyarak önceki bilgileri ile bağlantı kurduğunu ifade etmişlerdir. Bu çalışmada da araştırma bulguları bu bilgileri destekler niteliktedir. Bunlara ek olarak, öğrencilerin programlama/kodlama konularını metaforlarla öğrenmenin konuyu anlamalarına yardımcı olduğunu düşündükleri, öğrencilerin klasik yöntemler yerine metaforlarla ders işlenmesinden memnun oldukları ve öğrencilerin benzer derslerde de metafor etkinliklerini kullanmak istedikleri sonucuna

ulaşmıştır. Osborn (1997) da çalışmasında metaforların sınıf korkusunu ve isteksizliğini ortadan kaldırdığını vurgulamıştır.

Öneriler

Programlama öğretiminin 21. yüzyılda bireylerden istenen problem-çözme, algoritmik düşünme, analitik düşünme gibi üst düzey düşünme becerilerinin desteklenmesi noktasındaki önemi eğitim dünyasında öne çıkmaktadır. Bu konuda programlama öğretiminin yaygınlaştırılması ve özellikle teknolojik yetersizlikler sebebiyle sıkıntılar yaşayan bilişim teknolojileri ve yazılım öğretmenlerine alternatif örnek etkinlikler sunulması önemlidir. Aynı zamanda programlama öğretiminde dersin güçlüklerini azaltacak ve öğrencilerin ders içeriklerini ve hedef kazanımlara ulaşmasındaki isteklerini arttıracak yöntemler kullanılması da önemlidir. Bu çalışmada da özellikle programlama öğretiminde güçlük yaşanan soyut kavramların öğretiminde metaforların gücünden yararlanılması hedeflenmiştir. Çalışmanın sonuçlarına göre metaforlarla programlama öğretimi özellikle erken yaşta programlama öğretiminde temel bilgisayar kavramlarının bilgisayarsız olarak öğretilebileceği etkinlikler çerçevesinde ele alınarak geliştirilebilir. Bu bağlamda programlama öğretiminde kullanılabilecek metaforlara yönelik bir çalışma gerçekleştirilerek Bilişim Teknolojileri öğretmenlerine kılavuzluk edecek şekilde farklı yaş gruplarına hitap eden metaforları içeren kaynaklar geliştirilebilir.

Ayrıca öğrenme sürecinde öğrenen motivasyonu önemlidir. Bu çalışmada metaforlarla programlama öğretimi öğrenenlerin eğlenerek öğrenmesini de desteklemiştir. Bunlara ek olarak, öğrencilerin hem programlama öğretiminde hem de diğer branş derslerinde metaforlarla öğrenme deneyimini yaşamak istedikleri görülmüştür. Programlama öğretiminde de disiplinlerarası yaklaşımı benimseyen bir bakış açısı temel alınarak öğrencilerin farklı derslerde öğrendiği kavramlarla ilişkilendirebilecekleri metaforlar geliştirilebilir. Bu metaforların da programlama öğretim sürecinde kullanımı desteklenebilir. Ayrıca diğer branş öğretmenleri de işbirliği gerçekleştirilerek bu metaforları destekleyen bir yaklaşımın bu derslerde de kullanımı teşvik edilebilir.

Kaynakça

- Akkoyunlu, B., & Tuğrul, B. (2002). Okulöncesi çocukların ev yaşantısındaki teknolojik etkileşimlerinin bilgisayar okuryazarlığı becerileri üzerindeki etkisi. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 23(23), 12-21.
- Amundson, N. E. (1988). The use of metaphor and drawings in case conceptualization. *Journal of Counseling & Development*, 66(8), 391-393.
<https://doi.org/10.1002/j.1556-6676.1988.tb00895.x>
- Arabacıoğlu, T., Bülbül, H. İ., & Filiz, A. (2007). Bilgisayar programlama öğretiminde yeni bir yaklaşım. IX. *Akademik Bilişim Konferansı Bildirileri Kitabı*, 193-197.
- Arslan, M. M., & Bayrakçı, M. (2006). Metaforik düşünme ve öğrenme yaklaşımının eğitim-öğretim açısından incelenmesi. *Milli Eğitim*, 35(171), 100-108.
- Atman Uslu, N. Mumcu, F. & Eğin, F. (2018). Görsel programlama etkinliklerinin ortaokul öğrencilerinin bilgi-işlemsel düşünme becerilerine etkisi. *Ege Eğitim Teknolojileri Dergisi*, 2 (1), 19-31

- Bers, M., Flannery, L., Kazakoff, E., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157. <https://doi.org/10.1016/j.compedu.2013.10.020>
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*. Vancouver, 13-17 April 2012. <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>
- Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2).
- Büyüköztürk, Ş. (2012). Sosyal bilimler için veri analizi el kitabı (17. Baskı) Ankara: Pegem Akademi Yayıncılık.
- Creswell, J. W. & Clark, V.L.P. (2011). Designing and conducting mixed method research. Sage Publications, Inc.
- Çetin, İ. (2013). Visualization: A tool for enhancing students' concept images of basic object-oriented concepts. *Computer Science Education*, 23(1), 1-23. Doi: [10.1080/08993408.2012.760903](https://doi.org/10.1080/08993408.2012.760903)
- Deci, E. L., Eghrari, H., Patrick, B. C., & Leone, D. (1994). Facilitating internalization: The self-determination theory perspective. *Journal of Personality*, 62, 119-142.
- Dillashaw, F., & Bell, S. (1985). Learning outcomes of computer programming instruction for middle-grades students: A pilot study. *Proceedings of the 58th Annual Meeting of the National Association for Research in Science Technology*.
- Esgil, M. & Gündüz, Ş. (2019). Kodlama etkinliklerinin öğrencilerin bilgisayara yönelik tutum ve bilişim dersine duyuşsal katılımları üzerine etkisi. *Ahmet Keleşođlu Eğitim Fakültesi Dergisi*, 1 (2), 162-174.
- European Commission (2014). Coding - the 21st century skill. European Commission. [Çevrim-içi: <https://ec.europa.eu/digital-single-market/coding-21st-century-skill>, Erişim tarihi: 17.07.2019.]
- European Schoolnet (2015). The e-skills manifesto. http://eskillsjobsspain.com/wp-content/uploads/2015/12/e-Skills_Mnifesto2015.pdf adresinden 15 Haziran 2019 tarihinde alınmıştır.
- Field, A. (2011). *Discovering statistics using SPSS*. 3th Edition, Sage, Thousand Oaks, CA.
- Gomes, A., & Mendes, A. J. (2007). *Learning to program difficulties and solutions*. *International conference on Engineering Education*. Coimbra, Portugal. <http://icee2007.dei.uc.pt/proceedings/papers/411.pdf>
- Heywood, D. (2002). The place of analogies in science education. *Cambridge Journal of Education*, 32(2), 233-247.
- ISTE. (2022a). ISTE computational thinking competencies. <https://www.iste.org/standards/iste-standards-for-computational-thinking> adresinden 7 Mart 2022 tarihinde alınmıştır.
- ISTE. (2022b). Explore the ISTE student standards. <https://www.iste.org/standards/iste-standards-for-students> adresinden 7 Mart 2022 tarihinde alınmıştır.

- Jancheski, M. (2017). Improving teaching and learning computer programming in schools through educational software. *Olympiads in Informatics*, 11, 55-75.
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, 200-210.
- Kalelioğlu, F., & 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.
- Kandemir, C. M. (2018). *Metin tabanlı programlama*. İçinde Gülbahar, Y., Karal, H. (Eds). Kuramdan Uygulamaya Programlama Öğretimi, Bölüm 10, 297-336.
- Kasalak, İ. (2017). *Robotik kodlama etkinliklerinin ortaokul öğrencilerinin kodlamaya ilişkin özyeterlik algılarına etkisi ve etkinliklere ilişkin öğrenci yaşantıları*. Yayımlanmamış Yüksek Lisans Tezi, Hacettepe Üniversitesi, Ankara, Türkiye.
- Kert, S. B., & Uğraş, T. (2009). Programlama eğitiminde sadelik ve eğlence: Scratch örneği. *1st International Congress of Educational Research*.
- Kert, S.B. (2018). Bilgisayar bilimi eğitime giriş. İçinde Y. Gülbahar (Ed.), *Bilgi İşlemsel Düşünmeden Programlamaya*. Bölüm 1, 1-22. Ankara: Pegem Akademi.
- Kesici, T., & Kocabaş, Z. (2007). *Bilgisayar 2 ders kitabı* (2. Baskı).
- Koorsse, M., Cilliers, C., & Calitz, A. (2015). Programming assistance tools to support the learning of IT programming in South African secondary schools. *Computers & Education*, 82, 162-178.
- Korkmaz, Ö., Çakır, R., & Özden, M. (2015). Bilgisayarca düşünme beceri düzeyleri ölçeğinin (bdbd) ortaokul düzeyine uyarlanması. *Gazi Eğitim Bilimleri Dergisi*, 143-162.
- Korkmaz, Ö., Çakır, R., Özden, Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior* 72, 558-569.
- Lai, A.F. & Yang, S.M. (2011). The learning effect of visualized programming learning on 6th graders' problem solving and logical reasoning abilities. In *2011 International Conference on Electrical Control Engineering* (pp. 6940-6944).
- Muran, J. C., & DiGiuseppe, R. A. (1990). Towards a cognitive formulation of metaphor use in psychotherapy. *Clinical Psychology Review*, 10(1), 69-85.
- Ocak, G., & Gündüz, M. (2006). Eğitim fakültesini yeni kazanan öğretmen adaylarının öğretmenlik mesleğine giriş dersini almadan önce ve aldıktan sonra öğretmenlik mesleği hakkındaki metaforlarının karşılaştırılması. *Afyon Kocatepe Üniversitesi Sosyal Bilimler Dergisi*, 8(2), 293-309.
- Oluk, A., ve Korkmaz, Ö. (2016). Comparing students' scratch skills with their computational thinking skills in terms of different variables. *International Journal of Modern Education and Computer Science*, 11, 1-7.
- Osborn, M. (1997). The Play of Metaphors, *Education*, Fall, 118(1), 1-4.
- Patton, M. Q. (1999). Enhancing the quality and credibility of qualitative analysis. *HSR: Health Services Research*, 34 (5), 1189-1208.
<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1089059/pdf/hsresearch00022-0112.pdf>

- Pérez-Marín, D., Hijón-Neira, R., & Martín-Lope, M. (2018). A Methodology proposal based on metaphors to teach programming to children. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 13(1), 46-53. Doi: 10.1109/RITA.2018.2809944
- Pérez-Marín, D., Hijón-Neira, R., Babelo, A., & Pizarro, C. (2020). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children?. *Computers in Human Behavior*, 105, 1-10. <https://doi.org/10.1016/j.chb.2018.12.027>
- Portelance, D.J. (2015). *Code and tell: An exploration of peer interviews and computational thinking with ScratchJr in the early childhood classroom*. Yayınlanmamış yüksek lisans tezi, Tufts Üniversitesi, Massachusetts.
- Saban, A., Koçbeker, B. N., & Saban, A. (2006). An investigation of the concept of teacher among prospective teachers through metaphor analysis. *Educational Sciences: Theory & Practice*, 6(2), 509-522.
- Saygıner, Ş. ve Tüzün, H. (2017a). Programlama eğitiminde yaşanan zorluklar ve çözüm önerileri. *11th International Computer Education and Instructional Technologies Symposium*, 24-26 Mayıs 2017, İnönü Üniversitesi, Malatya.
- Sayın, Z., & Seferoğlu, S. S. (2016). Yeni bir 21. yüzyıl becerisi olarak kodlama eğitimi ve kodlamanın eğitim politikaları üzerine etkisi. *Akademik Bilişim 2016* (s. 1-5). Aydın: Adnan Menderes Üniversitesi.
- Shein, E. (2014). Should everybody learn to code? *Communications of the ACM*, 57(2), 16-18. doi:10.1145/2557447
- Strong, T. (1989). Metaphors and client change in counselling. *International Journal for the Advancement of Counselling*, 12(3), 203-213.
- TDK. (2020, 03 20). Türk Dil Kurumu. <http://www.tdk.gov.tr> adresinden alınmıştır.
- Terzidis, K. (2006). *Algorithmic Architecture*. Oxford: Elsevier Ltd.
- Tobar, C. M., Adán-Coello, J. M.; de Faria, E. S. J., de Menezes, W. S., & de Freitas, R. L. (2011). Forming groups for collaborative learning of introductory computer programming based on students' programming skills and learning styles. *International Journal of Information and Communication Technology Education*, 7(4), 34-46. <https://doi.org/10.4018/jicte.2011100104>
- Tompkins, P. ve Lawley, J. (2002). The magic of metaphor. *The Caroline Myss Newsletter*, March 2002. <https://www.cleanlanguage.co.uk/articles/articles/21/1/The-Magic-of-Metaphor/Page1.html> adresinden 5 Ekim 2019 tarihinde alınmıştır.
- Töremen, F., & Döş, İ. (2009). İlköğretim öğretmenlerinin müfettişlik kavramına ilişkin metaforik algıları. *Kuram ve Uygulamada Eğitim Bilimleri*, 9(4), 1973-2012.
- Uslu, N. A., Mumcu, F., & Eğin, F. (2018). Görsel programlama etkinliklerinin ortaokul öğrencilerinin bilgi-işlemsel düşünme becerilerine etkisi. *Ege Eğitim Teknolojileri Dergisi*, 2(1), 19-31.
- Wing, J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society*, 366, 3717-3725. doi:10.1098/rsta.2008.0118

- Winslow, L. E. (1996). Programming pedagogy psychological overview. *ACM Sigcse Bulletin*, 28(3), 17-22.
- World Economic Forum (2016). *The future of jobs: Employment, skills and workforce strategy for the forth industrial revolution*.
http://www3.weforum.org/docs/WEF_Future_of_Jobs.pdf adresinden 19 Mart 2019 tarihinde erişilmiştir.
- World Economic Forum (2020). *The future of jobs report 2020*.
www3.weforum.org/docs/WEF_Future_of_Jobs_2020.pdf adresinden 19 Mart 2022 tarihinde erişilmiştir.
- Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: pedagogical approaches to embedding 21st century problem solving in k-12 classrooms. *Tech Trends*, 565-568, doi: 10.1007/s11528-016-0087-7
- Yıldırım, A. & Şimşek, H. (2011). *Sosyal bilimlerde nitel araştırma yöntemleri* (8. Baskı). Ankara: Seçkin Yayıncılık.
- Yıldırım, N., & Konur, K. B. (2014). Fen bilgisi öğretmen adaylarının kimya kavramlarını günlük hayatla ilişkilendirebilmelerine yönelik gelişimsel bir araştırma. *The Journal of Academic Social Science Studies*(30), 305-323.
- Yıldız-Durak, H. (2018). Flipped learning readiness in teaching programming in middle schools: Modelling its relation to various variables. *Journal of Computer Assisted Learning*, 34(6), 939-959. Doi: <https://doi.org/10.1111/jcal.12302>
- Yıldız-Durak, H., & Güyer, T. (2018). Design and development of an instructional program for teaching programming processes to gifted students using scratch. In J. Cannaday (Ed.) *Curriculum development for gifted education programs* (pp. 61-99). Hershey: IGI Global.
- Yob, I. M. (2003). Thinking constructively with metaphors. *Studies in Philosophy and Education*, 22, 127-138.