

Research Article/Araştırma Makalesi

The Opinions of Students about Scratch, Arduino and Python Using from Open Source Code Softwares

Sevil HANBAY TIRYAKI¹  Fatih BALAMAN^{*2} 

¹ Hatay Mustafa Kemal University, Institute of Science, Hatay, Turkey, sevilhanbay90@gmail.com

² Hatay Mustafa Kemal University, Faculty of Education, Hatay, Turkey, fatihbalaman2010@gmail.com


*Corresponding Author: fatihbalaman2010@gmail.com

Article Info

Received: 18 May 2021

Accepted: 27 August 2021

Keywords: Open sources, programming, free software

 10.18009/jcer.938706

Publication Language: Turkish

Abstract

The purpose of this study is to examine the opinions of secondary school students on Scratch, Arduino and Python from open source code softwares. The study group is made up of 9 students studying in a high school in Hatay on 2019-2020 education years and on 11th grade. To reveal the experiences and practices of the students on open source code softwares, phenomenologic design of qualitative researches was utilized. Content analysis on data obtained by using NVivo, the themes of "The advantages of open source softwares with respect to closed source softwares", "Comparison of Arduino, Python and Scratch" and "The Place of Using Software Nowadays" were reached. The results pointed that student had a certain level of awareness on open source code softwares. Students use Python, Scratch and Arduino as per their purposes and find learning these softwares necessary.



To cite this article: Hanbay-Tiryaki, S., & Balaman, F. (2021). Açık kaynak kodlu yazılımlardan scratch, arduino ve python kullanımı hakkında öğrenci görüşleri. *Journal of Computer and Education Research*, 9(18), 831-852. DOI: 10.18009/jcer.938706

Açık Kaynak Kodlu Yazılımlardan Scratch, Arduino ve Python Kullanımı Hakkında Öğrenci Görüşleri

Makale Bilgisi

Geliş: 18 Mayıs 2021

Kabul: 27 Ağustos 2021

Anahtar kelimeler: Açık kaynak, programlama, özgür yazılım

 10.18009/jcer.938706

Yayın Dili: Türkçe

Öz

Bu araştırmanın amacı ortaöğretim öğrencilerinin açık kaynak kodlu yazılımlardan Scratch, Arduino ve Python' a yönelik görüşlerini incelemektir. Araştırmanın çalışma grubu 2020-2021 eğitim-öğretim yılında, Hatay'da bir lisenin 11. sınıfında öğrenim gören 9 öğrenciden oluşmaktadır. Araştırma açık kaynak kodlu yazılımların kullanımları hakkında öğrencilerin deneyimlerini, yaşantıları, tecrübelerini ortaya çıkarmak için nitel araştırma desenlerinden olgubilim (fenomenoloji) desenindedir. Nvivo kullanılarak gerçekleştirilen içerik analizi sonunda "Açık kaynak yazılımların kapalı kodlulara göre avantajları", "Arduino, Scratch ve Python'ın karşılaştırılması", "Yazılım kullanmanın günümüzdeki yeri" temalarına ulaşılmıştır. Çalışmadan elde edilen sonuçlara göre öğrenciler açık kaynak kodlu yazılımlar hakkında belirli bir farkındalığa sahiptir. Öğrenciler amaçları doğrultusunda Scratch, Arduino ve Python yazılımlarını kullanmakta ve bunları öğrenmeyi gerekli bulmaktadır.

Summary

The Opinions of Students about Scratch, Arduino and Python Using from Open Source Code Softwares

Sevil HANBAY TİRYAKI ¹  Fatih BALAMAN ^{*2} 

¹ Hatay Mustafa Kemal University, Institute of Science, Hatay, Turkey, sevilhanbay90@gmail.com

² Hatay Mustafa Kemal University, Faculty of Education, Hatay, Turkey, fatihbalaman2010@gmail.com

*Corresponding Author: fatihbalaman2010@gmail.com

Introduction

Constantly improving and updating education curricula aim at equipping students with the abilities needed in 21st century. The abilities of 21st century appear in front of us as problem solving, questioning, algorithm design, computational thinking and computer-like thinking (Pinto & Escudeiro, 2014). Coding has an important place in the development of these skills. Having a more obvious impact on the improvement of the individual, coding education process may start from preschool ages and continue even during the adulthood with people from all ages and occupations trying to learn how to code from various platforms. Generally, starting from pre-school ages, the coding tools that make learning how to code easier (Korkmaz, 2016) minimize the learning difficulties (Malan & Leitner, 2007) and make the kids enjoy coding by creating curiosity (Genç & Karakuş, 2011) are utilized. Robotic coding tools that boost students' motivation of learning how to code (Karahoca, Karahoca & Uzunboylu, 2011) are also being used. Being used in more than 150 countries and having support for more than 60 languages, Scratch was developed for individuals at the age 8-16, though today people from all ages use Scratch to learn how to code (Scratch, 2021). In addition, Arduino is also one of the most frequently used robotic environments and it is a programming medium constituted by software applications and microcontroller cards. The literature illustrates that Arduino improves the mathematical thinking abilities of students and helps them acquire off-class experience (Lopez & Cuesta, 2016). Open source-code softwares are softwares whose source code are available to everyone thus people can observe, examine, change and develop these softwares (Steiniger & Hunter, 2013). One of the most significant steps taken in terms of an important feature of open source code softwares, which is shareability, was "GNU System" project in Massachusetts Technology Institute with

the leading of Richard Stallman in 1984. This project was based on increasing the sharing of software resources and distributing to codes such that they become available to be modified.

The purpose of this study is to examine the opinions of secondary school students on open source code softwares (Scratch, Arduino and Python). To this end, the answer to the question of “What are the opinions of secondary school students about open source code coding tools and programming languages?” was investigated. The conclusions of this study may serve as a feedback for experts preparing education curricula and teachers practicing those curricula. It is also thought that the suggestions developed based on the results of this study will contribute to continuous and more functional programming education. Therefore, more productive, problem-solving, creative thinking individuals that this age requires can be raised.

Method

The research was limited with open source code softwares of Arduino, Scratch and Python. The reason behind choosing these softwares was that they were present in the current curriculum of 10th grade Information Technologies lesson. The study group of the is made up of 9 students studying in a high school in Hatay on 2019-2020 education year and on 11th grade. 4 participants were female and 5 of them were male. To reveal the experiences and practices of the students on open source code softwares, phenomenologic model of qualitative researches was utilized. The semi-structured interview form made up 10 questions and developed by the researcher was used to gather the data. In order to determine whether the questions in the data /gathering tool served the purpose, expert opinion was referred to and content validity was established. After making some changes on interview questions based on expert opinions, the final form to be used was obtained.

Results

Interview text was verified by sharing with the participants and participant confirmation was obtained. Coder reliability was also calculated. Miles and Huberman (1994) suggested that coder reliability should be at least 70%. According to the formula, coder reliability was calculated as 81.13. It is safe to say that coder reliability is provided with this result. In the analysis of the data and visualization of the findings to better convey them to the reader, NVivo qualitative data analysis software was utilized. Hence, the research reliability and validity was reinforced.

Discussion and Conclusion

As a result of the content analysis on data obtained, the themes of “The advantages of open source softwares with respect to closed source softwares”, “Comparison of Arduino, Python and Scratch” and “The Place of Using Software Nowadays” were reached. The results pointed that student had a certain level of awareness on open source code softwares. Students use Python, Scratch and Arduino as per their purposes and find learning these softwares necessary. Students also think of learning programming as necessary due to their career goals, thinking of it as a must of the current age, and arguing that it would make them gain the skills such as creative thinking, fast and correct decision-making and being planned. The results illustrate that student found open source code nature of Scratch, Arduino and Python to be advantageous as it allows them to see what is happening in the back-end, free to use, and easily accessible. It was realized that students agree that programming languages were beneficial for them both in academic and daily life. Most of the students have spent efforts of developing software and product using these tools outside of the school time. Students expressed that they obtained products such as lamp turning on by clapping, distance calculator, radar, metal detector, and body mass index calculator. Students also stated that they obtained products on TUBITAK projects using these programming languages.

Giriş

Çağdaş birey ve toplum olmanın en önemli yolu eğitimidir. Bu doğrultuda geniş kitlelere ulaşma, organize olma, altyapı gibi konularda ülkemizde en kapsamlı ve profesyonel kurum Milli Eğitim Bakanlığıdır. Teknoloji alanındaki gelişmeleri de dikkate alarak Milli Eğitim Bakanlığı (MEB), öğrencilerin bilişim teknolojileri ve yazılım alanlarında uzman bireyler olarak yetiştirilmesi için çalışmalarını yürütmektedir (Demirer & Sak, 2016). Bu kapsamda atılan adımlar sonrasında programlama eğitiminin de içerisinde yer aldığı yeni öğretim programı 2013 yılından itibaren 5. ve 6. sınıflarda Bilişim Teknolojileri ve Yazılım dersi olarak okutulmaya başlanmıştır (MEB, 2018a). 2016 yılında ise güncellenen ortaöğretim kademesi öğretim programı, kapsamında yer verilen Bilgisayar Bilimi dersi ile ortaöğretim öğrencilerinin programlama, programlama dilleri, robot programlama, mobil programlama gibi alanlarda yetkin bireyler yetiştirilmesi hedeflenmiştir (MEB, 2018b).

Sürekli olarak geliştirilen, yenilenen öğretim programları öğrencileri 21. yüzyıl becerileriyle donatmayı hedeflemektedir. 21. yüzyıl becerileri problem çözebilme, yaratıcı düşünme, eleştirel düşünme, karar alabilme, işbirliği yapabilme, sorgulama yapabilme, algoritma tasarlayabilme, hesaplamalı düşünme ve bilgisayarca düşünebilme gibi beceriler olarak karşımıza çıkmaktadır (Pinto & Escudeiro, 2014). İçinde bulunduğumuz bilgi çağı, bireylerin hızla değişen yeniliklere ayak uydurabilecek biçimde ve 21. yüzyıl becerileri ile donatılarak yetiştirilmeleri için bilgisayar biliminin öğretilmesini ve kodlama eğitiminin verilmesini zorunlu görmüştür (Karabak & Güneş, 2013). Nitekim yapılan birçok çalışma da bunu doğrulamaktadır (Lye & Koh, 2014; Wakil, Khdir, Sabir, & Nawzad, 2019). Araştırmalara göre kodlama eğitimi ile bu çağın ihtiyaç duyduğu üretken, problem çözebilen ve yaratıcı düşünebilen bireyler yetiştirmek mümkündür. Nitekim Özyurt vd. (2019) yaptıkları çalışmada uluslararası dergilerde yayınlanmış 34 makaleyi incelemiş ve bu çalışmaların büyük çoğunluğunun sonuçlarına göre üniversite öncesi düzeyde kodlama öğrenmek problem çözme, eleştirel düşünme, yaratıcı düşünme, mantıksal düşünme ve algoritmik düşünme gibi üst düzey düşünme becerilerini olumlu etkilemektedir.

Bireyin gelişimine sağladığı katkının giderek daha da fark edildiği kodlama eğitimi, okul öncesi dönemden başlayıp yetişkinlikte dahi devam eden bir süreç olarak karşımıza çıkmaktadır. Hayat boyu öğrenme anlayışıyla her yaştan, her meslek grubundan insanlar kodlamayı farklı platformlar aracılığı ile öğrenmeye çalışmaktadır. Genellikle okul öncesi dönemden itibaren kodlama öğrenmeyi daha kolay kılan (Korkmaz, 2016), öğrenme

güçlüklerini en aza indirgeyen (Malan & Leitner, 2007) ve çocuklara kodlamayı sevdirecek bu konuda merak uyandıran (Genç & Karakuş, 2011) blok tabanlı kodlama araçları kullanılmaktadır. Aynı zamanda yine okul öncesi dönemden itibaren öğrencilerin kodlamayı öğrenmesi kodlama öğrenmeye yönelik motivasyonlarını artırmaktadır (Karahoca, Karahoca & Uzunboylu, 2011). Ortaöğretim kademesinde ise kodlama eğitimleri kodlama ve robotik kodlama kazanımları doğrultusunda blok tabanlı kodlama ve robotik kodlamaya ek olarak metin tabanlı programlama ile de yürütülmektedir.

Açık Kaynak Kodlu Yazılımlar

Açık kaynak kodlu yazılımlar (AKKY), kaynak kodu herkese açık olan böylelikle de insanların yazılımın kodlarını görmesine, incelemesine, değiştirmesine, geliştirmesine ve paylaşmasına imkân tanıyan yazılımlardır (Steiniger & Hunter, 2013). AKKY'nin en önemli özelliklerinden biri olan paylaşılabilir olması adına atılan en önemli adımlardan biri 1984 yılında Richard Stallman başkanlığında, Massachusetts Teknoloji Enstitüsü'nde "GNU System" projesidir. Bu proje ile yazılım kaynaklarının paylaşımının artırılması ve kodların dağıtılarak geliştirilmeye açık hale getirilmesidir. 1989 yılında ise AKKY'nin çerçevesini daha net çizebilmek için Genel Kamu Lisansı (GPL- General Public Licence) geliştirilmiştir. Buna göre GPL' e bağlı yazılımlar ticari olarak kullanılabilir, değiştirilebilir, dağıtılabılır ve yazılımın orijinalini bulundurmamak zorunludur (Yılmaz, 2017).

AKKY genellikle lisans maliyeti olmaması, güvenlik açığının daha az olması, daha yüksek performansla çalışması, firma bağımlılıklarının olmaması, gönüllü geliştiriciler sebebiyle sürekli güncel ve uzun ömürlü olması gibi sebeplerle tercih edilmektedir (Hwang, 2005). Eğitim alanında da AKKY bütün bu özelliklere ek olarak genellikle yazılımın tüm özelliklerine her an erişebilmesi, sürekli güncellenmesi ve yaygın bir destek ağına sahip olması sebepleriyle öğrenciler tarafından tercih edilmektedir (Malkoç, 2012; Sönmez, 2017). Nitekim bu araştırmada açık kaynak kodlu kodlama araçlarından ve programlama dillerinden üçüne yer verilmiştir: Scratch, Arduino ve Python. Bu yazılımların seçilmesinin bir başka sebebi ise 10. sınıfta Bilgisayar Bilimi dersi mevcut müfredatında bu yazılımların yer almasıdır.

Scratch

150'den fazla ülkede kullanılan ve 60'dan fazla dil desteğine sahip Scratch özellikle 8-16 yaş arasındaki bireyler için tasarlanmış olsa da günümüzde her yaşta birey kodlamayı öğrenmek için Scratch'ı kullanmaktadır. Massachusetts Teknoloji Enstitüsü'ne (MIT) bağlı

Lifelong Kindergarten grubunun bir projesi olan Scratch görselleştirilmiş bir blok tabanlı kodlama programıdır. Blok kod yapısı, eğlenceli ve kolay kullanılan bir ara yüze sahip olması, kod bloklarının anlaşılabilir olması (Resnick vd., 2009) çevrimiçi ve çevrimdışı kullanım-paylaşım imkânı sunması, çoklu ortam desteği olması diğer blok tabanlı kodlama araçlarında olduğu gibi Scratch'ın başlıca özelliklerindedir (Resnick vd., 2009). Ayrıca birbiriyle ilişkili olmayan kod bloklarını birleştirmeye müsaade etmemesi ile Scratch, en çok karşılaşılan söz dizimi hatalarını da engelleyebilmektedir (Karabak & Güneş, 2013). Scratch ile yapılan çalışmalar göstermektedir ki blok tabanlı kodlama öğrencilerin kodlamaya olan tutumlarını (Alp, 2019), motivasyonlarını, programlama, problem çözme ve diğer üst düzey düşünme becerilerini artırmaktadır (Özyurt, Özyurt & Aras, 2019).

Arduino

Arduino çok sık kullanılan robotik ortamlarından biri olup yazılım uygulamalarından ve mikro kontrolcü kartlardan oluşan bir programlama ortamıdır. İtalyan mühendisler tarafından geliştirilmiş olan Arduino, donanım ve yazılım işlemlerinin birlikte yürütüldüğü açık kaynaklı, fiziksel program geliştirme ortamıdır (Ocak & Efe, 2018). Öğrenciler bir programlama dilini öğrenirken programlama dilinin yapısı ve içerdiği kavramların soyutluğu sebebiyle zorlanabilmektedirler. Robotik ise tam da bu süreçte programlamanın soyutluğunu somutlaştırmak için yardımcı olmakta, soyut kavram ve becerileri öğretmekte kullanılmaktadır (Öztürk & Özdemir, 2020).

Robotik kodlama öğrencilerin yaratıcılıklarının ve araştırma tutumlarının gelişmesine katkı sağladığı gibi problem çözme, soyut düşünme, eleştirel düşünme ve işbirlikli çalışma gibi üst düzey becerilerinin de gelişmesini sağlamaktadır (Lye & Koh, 2014; Sarı & Kırındı, 2019). Ayrıca robotik kodlama sayesinde öğrencilerin programlamayı daha kolay ve hızlı bir biçimde öğrenmesi ve motivasyonlarının artması söz konusudur (Resinovic, 2015). Bunlara ilaveten literatürde Arduino ile yapılan çalışmalar göstermektedir ki Arduino öğrencilerin matematiksel düşünme becerilerini geliştirmekte ve ders dışı deneyimler edinmelerine (Lopez & Cuesta, 2016) yardımcı olabilmektedir.

Python

Metin tabanlı programlama dilleri, kaynak kodlarının sadece metin komutları ve bu komutlara ilişkin söz dizimi kuralları ile oluşan programlama dilleridir (Mihci, 2014). Metin tabanlı programlama dillerinin öğrenilmesi problem çözme, soyut düşünme, programlama ve bilgisayarca düşünme gibi üst düzey becerilerin gelişimine katkıda bulunmakta ve

öğrencilerin gelecekteki kariyerleri için avantaj sağlamaktadır (Kandemir, 2018). Blok tabanlı kodlamadan sonra metin tabanlı programlama dilleri öğretilirken öğrencilerin söz dizimine çok dikkat etmedikleri, derleme sırasında hatalarla karşılaştıkları ve bu ve bezeri durumların öğrencilerin özgüvenini düşürdüğü anlaşılmıştır (Powers, Ecott & Hirsfield, 2007). Bu yüzden metin tabanlı programlamaya geçişte uygun bir programlama dili seçimi önem arz etmektedir. Bu durumda Python iyi bir seçenek olabilir (Kandemir, 2018).

Guido van Rossum tarafından geliştirilen ve 1991’de ilk sürümü ortaya çıkan Python genel amaçlı, özgür ve açık kaynak kodlu bir programlama dilidir. Python programlama dili bazı avantajları sebebiyle eğitimde kullanmak için oldukça uygundur. Bu avantajlar şunlardır:

- Özgür açık kaynak kodlu olması sebebiyle öğrenciler istedikleri zaman Python kütüphanesini ve geliştirme araçlarını kullanabilmektedir (Malkoç, 2012).
- Kolay öğrenilebilir bir dil olması sebebiyle istenilen algoritmalara, kodlara odaklanmak daha kolaydır (Rossum, 1999).
- Python, yorumlanan bir dil olması sebebiyle yeni bir program yazılırken hızlıca deneme yapılmasına olanak sağlar (Malkoç, 2012).
- Öğrenme kaynaklarının ve standart kütüphanesinin genişliği öğrencilere akademik çalışmalarında ve diğer işlerinde kolaylık sağlamaktadır (Dubois, 2007).

Tüm bu sayılan sebeplerden dolayı blok tabanlı kodlamadan sonra programlama dili öğretilirken Python’ı kullanmak uygun bir seçenek olabilir (Kandemir, 2018).

Araştırmanın Amacı ve Önemi

Her ne kadar literatürde kodlama ve programlama kavramları birbirinin yerine kullanılıyor olsa da kodlama ve programlama arasında farklar vardır. Genel olarak programlama test etme, hata ayıklama gibi çok daha geniş bir beceri topluluğunu ifade etmekte (Bell, 2018) olan daha karmaşık (Lye & Koh, 2014) bir süreçtir. Soyut yapısı ve karmaşık sözdizimi gibi nedenlerle programlama eğitiminde yaşanan bazı zorluklar (Gomes & Mendes, 2007) vardır. Bu nedenle eğlenceli ve daha kolay anlaşılabilir olan blok tabanlı ve robotik kodlama araçları kodlama eğitimine başlarken daha çok tercih edilmektedir (Çatlak, Tekdal & Baz, 2015). Böylelikle öğrenenler metin tabanlı programlama dillerinin getirdiği dezavantajlardan kaynaklı zorluklardan büyük ölçüde kurtulmuş olmaktadır. Bu araçlar, kavramları somutlaştırarak kodları ezberlemeyi gerektirmeyen ve kodları unutmayı önleyen ilgi çekici ortamlar sunmaktadır (Saygıner & Tüzün, 2017). Böylece yazılım eğitimine

öğrenciler için daha eğlenceli, daha ilgi çekici ve daha kolay anlaşılabilir programlama dilleri ile başlanır. Bu sayede öğrencilerin programlamayı sıkıcı ve zor bir süreç olarak algılayıp olumsuz tutum geliştirmeleri ve programlama öğrenmeden uzaklaşmaları önlenmiş olunur (Genç & Karakuş, 2011).

Öğrencilerin programlama deneyimleri sonrasında yaşadıkları duygular, düşünceler, tutumlar önemlidir. Nitekim Özyurt ve Özyurt (2015) yaptıkları çalışmada öğrencilerin programlamaya karşı tutum ve programlama öz-yeterlikleri arasında pozitif yönlü ve orta düzeyde bir ilişkinin olduğu sonucuna ulaşmışlardır. Zira hayal kırıklığı, dikkat dağınıklığı, tekrar görev almak istememe, çözüm üretmekten vazgeçme, programlamanın zor olduğunu ve anlayamayacağını düşünme gibi olumsuz tutumlar nedeniyle öğrenciler programlama öğrenmekten uzaklaşabilmektedir (Beck, Emery & Greenberg, 2005). Programlama eğitimine başlarken son yıllarda her ne kadar öğrencilerin programlama öğrenmeye yönelik motivasyonlarını artırmak için blok tabanlı kodlama araçları kullanılsa da (Kandemir, 2018) öğrenciler metin tabanlı programlama dillerini öğrenmeye başladığında zorlanmaktadır (Powers, Ecott & Hirsfield, 2007). Öğrencilerin metin tabanlı programlama dillerini öğrenirken yaşadığı güçlükler cesaretlerini kırmaktadır (Powers vd., 2007). Kodlama eğitiminden başlayıp programlamaya kadar uzanan bu süreçte öğrencilerin programlama öğrenmeye yönelik motivasyonlarını artırmak ve öğrenmelerini teşvik edebilmek açısından, kodlama araçlarını ve programlama dillerini ortaöğretim öğrencilerinin bakış açısıyla irdelemek açısından bu araştırmanın önemli olacağı düşünülmüştür.

Literatür incelendiğinde programlama eğitimiyle ilgili yapılan çalışmaların birçoğunun blok tabanlı kodlama araçlarını (özellikle Scratch) kapsadığı ve bu çalışmaların büyük çoğunluğunun ilköğretim düzeyindeki öğrencilerle yürütüldüğü görülmektedir. Literatürde farklı kodlama araçlarının karşılaştırılmasına yer verilse de bu karşılaştırmaların genellikle blok tabanlı kodlama araçlarıyla sınırlı kaldığı görülmektedir (Eryılmaz & Deniz, 2019; Özyurt, Özyurt & Aras, 2019). Yapılan çalışmaların bir kısmında ise programlama eğitime yönelik öğretmenlerin, ortaokul ve lisans öğrencilerinin görüşlerine başvurulduğu görülmektedir (Eryılmaz & Deniz, 2019). Bu çalışmada ise ortaöğretim öğrencilerinin açık kaynak kodlu kodlama araçlarına ve programlama dillerine yönelik görüşlerini incelemek amaçlanmıştır. Araştırma bağlamında "Ortaöğretim öğrencilerinin açık kaynak kodlu kodlama araçlarına ve programlama dillerine yönelik görüşleri nelerdir?" sorusu cevaplanmaya çalışılmıştır. Bu bağlamda bu çalışmada öncelikle programlama eğitiminin

başlangıcından itibaren kullanılan blok tabanlı kodlama araçları ve robotik kodlama araçları ile programlama eğitiminin daha ileri aşamasında kullanılan metin tabanlı bir programlama dilinin ortaöğretim öğrencilerinin görüşlerine başvurularak karşılaştırılmasına yer verilmiştir. Gerek kodlama ve robotik araçları ile programlama dilini karşılaştırması bakımından gerekse de ortaöğretim öğrencileriyle yürütülmesi bakımından bu araştırmanın literatüre katkı sağlayacağı düşünülmektedir. Bunlara ilaveten bu araştırmanın sonuçları öğretim programını oluşturan uzmanlara ve programın uygulayıcıları öğretmenlere bir dönüt niteliğinde olabilir, öğretmenlerin öğrencilerine rehberlik etmesine yardımcı olabilir. Ayrıca araştırma sonuçları programlama eğitiminin daha sürekli ve işlevsel olmasına, açık kaynak kodlu yazılımlar hakkında farkındalık oluşturmaya katkı sağlayabilir. Araştırma için Arduino, Scratch ve Python yazılımların seçilmesinin sebebi, 10. sınıfta bilişim teknolojileri dersi mevcut müfredatında bu yazılımların gösteriliyor olmasıdır.

Yöntem

Araştırma Modeli

Araştırmada açık kaynak kodlu yazılımların kullanımları hakkında öğrencilerin deneyimlerini, yaşantılarını, tecrübelerini ortaya çıkarmak amaçlanmıştır. Bu sebeple araştırma olgubilim (fenomenoloji) modelindedir. Araştırmada açık kaynak kodlu yazılımlar Arduino, Scratch ve Python yazılımları ile sınırlandırılmıştır. Seggie ve Bayyurt (2017) bireylerin bir konuya, olguya yönelik yaşantılarını, bilgilerini, yaşantılarını açığa çıkaran türdeki çalışmaları olgubilim çalışmaları olarak nitelendirmiştir. Yıldırım ve Şimşek (2008) ise olgubilim araştırmalarını, bir konuda fikir sahibi olduğumuz fakat detaylı bilgimizin olmadığı konuları ayrıntılı olarak araştırıp bilgilenmeyi amaçlayan araştırmalar olarak nitelendirmiştir.

Çalışma Grubu

Çalışma grubunu 2020-2021 eğitim-öğretim yılında Hatay' da bir mesleki ve teknik anadolu lisesinin 11. sınıfında öğrenim gören 9 öğrenci oluşturmaktadır. Katılımcıların 4'ü kız, 5'i erkek bireydir. Çalışma grubu, 10. sınıfta bilişim teknolojileri dersinde Arduino, Scratch ve Python yazılımlarını gören öğrenciler arasından amaçlı çalışma gruplarından uygun durum çalışma grubu yöntemine göre seçilmiştir. Bu çalışma grubu türü ile çalışma grubuna daha kolay ulaşmak, veri toplama işlemini kolaylaştırmak amaçlanır (Sönmez & Alacapınar, 2013).

Geçerlik – Güvenirlik

Nitel arařtırmalarda iç geçerlik inandırıcılık ile saęlanabilir. İnandırıcılık için farklı yollar önerilmekle birlikte uzman görüşüne başvurma, katılımcı teyidi, derinlemesine veri toplama bu yöntemler arasındandır (Yıldırım & Şimşek, 2008). Veri toplama aracında soruların amaca hizmet edip etmediğini tespit etmek için uzman görüşü alınarak kapsam geçerliği saęlanmıştır. Görüşme sonunda elde edilen nitel verilerden arařtırmada kullanılacak şekilde görüşme metinleri, ilgili katılımcılarla paylaşarak doğrulanmış, katılımcı teyidi alınmıştır.

Güvenirlik kapsamında Miles ve Huberman' ın (1994) kodlayıcı güvenilirliği formülü (güvenirlik = görüş birliği / (görüş birliği + görüş ayrılığı)×100) dikkate alınarak iç güvenilirliği saęlamak amaçlanmıştır. Bu sebeple görüşme verileri arařtırmayı yürüten iki arařtırmacı tarafından ayrı ayrı kodlanmış ve kodlamalar karşılaştırılarak güvenilirlik değeri elde edilmiştir. Miles ve Huberman (1994) kodlayıcı güvenilirliğinin en az %70 olmasını önermektedir. Formüle göre kodlayıcı güvenilirliği %81.13 olarak hesaplanmıştır. Sonuca göre kodlayıcı güvenilirliğinin saęlandığı söylenebilir.

Kuş (2008) nitel çalışmalarda yazılım kullanımının daha şeffaf, objektif çalışmalar ortaya çıkardığını, verilerin analizinde yansız olmayı daha mümkün kıldığını dolayısıyla yazılım kullanımının geçerliğe ve güvenilirliğe katkıda bulunduğunu ifade etmiştir. Arařtırmada verilerin çözümlenmesinde, bulguların görselleştirilerek okuyucuya daha anlaşılabilir şekilde sunulmasında NVivo kullanılmıştır.

Veri Toplama Aracı

Verilerin toplanmasında arařtırmacılar tarafından geliştirilen yarı yapılandırılmış görüşme formu kullanılmıştır. Seggie ve Bayyurt (2017) yarı yapılandırılmış görüşmede konu başlıklarının önceden belli olduğunu ve bütün katılımcılara yaklaşık aynı sorular sorulduğunu ifade etmektedir. Veri toplama aracı 10 sorudan oluşmaktadır. Görüşme formu başlangıçta 13 soruluk taslak olarak hazırlanmıştır. Veri toplama aracının kapsam geçerliği için uzman görüşü alınmıştır. Görüşme formu taslağı 2'si öğretim üyesi, 2'si bilişim teknolojileri öğretmeni olmak üzere 4 alan uzmanına gönderilmiştir. Form ile birlikte alan uzmanlarının görüşlerini yazabileceğı değerlendirme formu da gönderilmiştir. Deęerlendirme formunda her bir görüşme sorusu için uygunluk seçenekleri (uygun/uygun deęil/düzeltilmeli) ve açıklama alanı bulunmaktadır. Uzman görüşlerinden gelen geri bildirimler doğrultusunda 1 soru formdan çıkarılmış, 2 soru dięer sorularla birleştirilmiş, 4

soruda da ifade bakımından düzenleme yapılmıştır. Bu aşamada görüşme formu 10 sorudan oluşan bir yapıya kavuşmuştur. Görüşme formundaki soruların açıklığı, anlaşılabilirliği ve öğrenci düzeyine uygunluğu bakımından değerlendirilmesi amacıyla form, katılımcılar ile aynı özellikleri taşıyan 2 öğrenciye gönderilmiş ve bahsedilen özellikler bakımından değerlendirmeleri incelenmiştir. Öğrenciler form üzerinde herhangi düzenlemeye ihtiyaç olmadığı yönünde görüş bildirmişlerdir.

Verilerin Analizi

Veriler sanal ortamda yazılı olarak toplanmıştır. Yüz yüze görüşmenin yapılmamasında içinde bulunduğumuz pandemi ve öğrencilerin tercihi etkili olmuştur. Seggie ve Bayyurt (2017) görüşme verilerinin yüz yüze toplanabilmesinin yanı sıra uzaktan internet veya telefon gibi iletişim araçlarıyla da toplanabileceğini ifade etmektedirler. Görüşme verileri içerik analizi yöntemi ile analiz edilmiştir. İçerik analizi yapılırken NVivo yazılımı kullanılmıştır. Kodların, temaların ve alt temaların oluşturulmasında, kelime bulutu görselinin oluşturulmasında NVivo' dan yararlanılmıştır. Maxwell (2018) verilerin depolanması, kodlamalar, sınıflandırmalar, sorgulamalar gibi niteliklerinden dolayı nitel çalışmalarda yazılım kullanmanın önemine vurgu yapmış, günümüzde bu yazılımların çok geliştiğini ifade ederek nitel veri analizi yazılımları arasında en yaygın olanın NVivo olduğunu iddia etmiştir. İçerik analizi sonunda temalara ait ifadeler görselleştirilerek frekans değeri yazılmış, doğrudan alıntılara da sıkça yer verilmiştir. Araştırmada 9 katılımcının her biri random olarak numaralandırılmış ve metin içinde katılımcılar Öğr.1, Öğr.2 şeklinde anılmıştır.

Bulgular

11.sınıfta öğrenim gören katılımcıların bir önceki yıl öğrendikleri açık kaynak yazılımlarını halen kullandıkları, Python, Arduino ve Scratch dışında da farklı açık kaynak yazılım kullananların olduğu anlaşılmıştır. Ayrıca her bir katılımcının en az bir yazılım kullandığı, birden fazla yazılım kullanan katılımcının da olduğu anlaşılmıştır. Buna göre 1 kişi Linux, 6 kişi Python, 5 kişi Ardunio, 3 kişi Scratch, 1 kişi Pidgin, 1 kişi Gimp kullanmaktadır. Kullanılan yazılımlara göre en fazla Phyton, Arduino ve Scratch kullanıcılarının olduğu görülmektedir. Bu yazılımlar lise son sınıfta okutulan ders kapsamında müfredatta yer alan yazılımlardır. Buna göre öğrencilerin açık kaynak yazılım kullanımı tercihlerinde aldıkları eğitimin etkisi olabilir.

İçerik analizi sonunda kodlamalar yapılmış, kodlamalarla temalara ulaşılmış ve temalarla ilişkili alt temalar oluşturulmuştur. Buna göre “Açık kaynak yazılımların kapalı kodlulara göre avantajları”, “Arduino, Scratch ve Phyton' ın karşılaştırılması”, “Yazılım kullanmanın günümüzdeki yeri” temalarına ulaşılmıştır. İlgili temalar NVivo ile modellenmiş, alt temalar ise frekansları ile birlikte verilmiştir.

Tablo 1. “Açık kaynak yazılımların kapalı kodlulara göre avantajları” teması ve alt temaları

<i>Açık kaynak yazılımların kapalı kodlulara göre avantajları</i>	f
Açık kaynak yazılımın gelişime açık olması	7
Kullanıcılar arası etkileşimi artırması	5
Kullanıcıyı geliştirmesi	4
Değişiklik yapılması özgürlüğü	4

Açık kaynak yazılımların kapalı kodlulara göre bazı avantajları olduğu anlaşılmıştır. Bir yazılımın kodlarının erişilebilir olması o yazılımı kullananların ilgisini çekmekte, kodları geliştirmeye gönüllü olanlar yazılımın daha ileri özelliklere sahip olmasında etkili olabilirler. Gönüllü kullanıcıların açık kaynak yazılımların geliştirilmesine katkıda bulunmaları hem kullanıcıların güvenlik açısından yazılımdan emin olmalarını hem de yazılımın yeteneklerini, niteliğini artırmaktadır. Açık kaynak yazılımlara kullanıcıların katkı sağlaması, kullanıcıların bir birleri ile etkileşimlerini, iletişimlerini artırmaktadır. Yardımlaşarak kullanıcıların yazılımlarda tespit ettikleri eksiklikleri tamamlamaları aynı zamanda bireylerin yazılım bilgilerinin gelişmesine de katkı sağlamaktadır. Yazılım kullanılırken arka planında olup bitenlerden haberdar olmak, çalışan kodları görmek kullanıcılara özgürlük hissi uyandırabilmektedir. Aksi durumda kapalı kodlar kullanıcılarda çeşitli şüphelerin oluşmasına neden olabilmektedir. Bu açıdan açık kaynak yazılımları özgür yazılımlar, kullanıcıları ise özgür yazılım kullanıcıları olarak nitelendirilmektedir.

“Açık kaynak yazılımların kapalı kodlulara göre avantajları” temasına ilişkin katılımcılar şu görüşlere yer vermişlerdir:

Öğr.1; “Programların nasıl çalıştığını görebilmenin öğrenimde kolaylık ve özgürlük bakımından ticari programlama dillerine göre her zaman bir adım önde olduğunu düşünüyorum.”

Öğr.6; “Açık kaynak kodlu programlama dilinde istediğiniz projeye göre değişiklik yapabilirsiniz. Ticari programlama kodunda ise hak sahibinin izin verdiği kadar yapabilirsiniz.”

Öğr.6; “Açık kaynak kodlu programlama dili herkese açıktır, ücretsizdir ve ayrıca yaratıcılığa bağlı her türlü modifikasyonlar yapılabilir.”

Öğr.4; “Açık kaynak kodlu yazılımlar, programcı topluluğunun gelişimini sağlar. Bu sayede eğitime, yaratıcılığa ve ilham almaya yardımcı olur. Açık kaynak kodlu yazılımlarda örneğin bir açık bulunduğunda, onunla ilgilenen çok fazla kişi olduğundan bu açık daha çabuk kapatılabilir. Kapalı kaynak kodlu yazılımlarda ise uzun bir süre beklemeniz gerekebilir.”

Tablo 2. “Arduino, Scratch ve Phyton' in karşılaştırılması” teması ve alt temaları

Arduino, Scratch ve Phyton' in karşılaştırılması	f
Bu yazılımları kullanmak keyfidir.	6
Benzer ve farklı yönleri vardır.	5
Bu yazılımları kullanmak önemlidir.	4
Öğrenilmeleri kolaydır.	4

10.sınıfta Bilişim Teknolojileri dersinde Arduino, Scratch ve Phyton yazılımlarını öğrenen ve kullanan katılımcılar görüşmelerde genellikle bu 3 yazılımı çeşitli yönleriyle karşılaştırmışlardır. Benzerliklerini ve farklılıklarını çeşitli yönleriyle ortaya koyan kullanıcılar genellikle bu yazılımların öğrenilmesinin kolay olduğunu ifade etmişlerdir. Farklı özellikleri olan ve farklı amaçlar için kullanılan bu yazılımların önemli işlevleri vardır ve yazılımla ilgilenenler için birçok ihtiyaca cevap verebilecek niteliktedirler. Bu bağlamda katılımcılar bu yazılımları kullanmanın ve öğrenmenin önemli olduğunu vurgulamışlar ve birçoğu kullanırken keyif aldıklarını ifade etmişlerdir. Görüşmelerde mezuniyet sonrasında meslek hayatlarında yazılımcı olmak isteyenler de olmuştur. Öğrenmenin ve kullanmanın önemli olduğu vurgulanan bu yazılımları öğrencilerin iş hayatlarında da kullanmaya istekli oldukları anlaşılmıştır.

“Arduino, Scratch ve Phyton' in karşılaştırılması” temasına ilişkin katılımcılar şu görüşlere yer vermişlerdir:

Öğr.2; “Şimdiye kadar uğraşmış olduğum programlar içinde eğlenceli olması ve Phytona nazaran daha kolay olması sebebiyle Scratch'i çok sevmiştim. Kullandığım ilk program olan Scratch'da çok kez animasyonlar tasarladım. Fakat daha profesyonel hissettirmesi sebebiyle Phyton ile uğraşmaktan da çok zevk almış, dahil olduğum okul projelerinde ve sınıf ortamı etkinliklerinde kullanmıştım.”

Öğr.1; “Benzerlik açısından Arduino ve Python'da kodları kendimiz yazdığımızdan benzer olduklarını söyleyebilirim. Bu durumda Scratch'in bloklarla daha çok oyun tarzı olması bakımından diğer 2 araca göre farklılık gösterdiğini söyleyebilirim. Arayüz ve menülerin kullanılabilirliğine gelince Scratch'in kesinlikle aralarında en kullanışlı olduğunu düşünüyorum.”

Öğr.2; “Scratch programında kodların puzzle şeklinde hazırlanması öğrenmeyi kolaylaştırıyor ve görsel açıdan daha akılda kalıcı olmasını sağlıyor. Her ne kadar kodlarda renklendirmeye gidilmiş olsa da Phyton programının bu konuda daha zorlayıcı olduğunu düşünüyorum. Bunu anlamanın en kolay yolu programı kullandığımda sık sık hata vermesi ve problemin sebebini bulmanın çok zor olması.”

Öğr.4; "Python ve Arduino daha rahat öğrendiğim kodlama araçlarıdır."

Öğr.5; "Arayüzü ve menü olarak en kullanışlısı Scratch. Öğrenme kolaylığı olarak en basiti Scratch. Python ve Arduinoda kodları öğrenmek daha zor."

Öğr.9; "Arayüzün ve menülerin kullanılabilirliği bakımından örneğin Scratch da hazır kod verildiği için daha kullanışlı ve kod yazmak daha kısa. Python'da ise kodlar elle yazıldığı için kod yazmak daha uzun sürer. Öğrenme kolaylığı açısından hazır kodlu yazılımları öğrenmek daha kolaydır. Scratch da hazır kodlar var. Python da ise doğru-yanlış belirten renkler var. Bu nedenle yanlış yaptığımızda yanlış görüp düzeltebiliriz ama diğerlerinde bunun gibi özellikler yoktur."

Öğr.2; "Öğrenme kolaylığı ve eğlenceli olması, elimizdeki sınırlı teknolojik araçlarda kullanımının daha kolay olması sebebiyle Scratch'i keşfettiğimde çok sevdim."

Öğr.3; "En çok Python kullanmaktan zevk aldım çünkü bize tüm kolaylıklar sağlanmış."

Öğr.6; "Ben öğrenmekten keyif aldığım hem Python hem Scratch. Çünkü Python'un işlemsel ve yazı yönünü çok sevdim. Scratch'in ise daha çok görsel ve animasyon yönünü sevdim."

Öğr.8; "Python'ı basit olduğu, direkt olarak sonuç alındığı için daha fazla beğendim."

Öğr.2; "Profesyonel ve daha geniş bir uğraş alanı içinde olan Python'ı diğerlerine göre daha önemli buluyorum."

Öğr.5; "Kodlama araçlarından benim için en önemlisi Scratch. Çünkü arayüzü kodlamaya yeni başlayan biri için çok ideal. Bundan dolayı kodlamaya Scratch ile başlanmalı."

Öğr.5; "Aslında 3 yazılım da birbirinden önemli fakat Python biraz daha geniş alanlarda kullanılabilir bence. Bu yüzden Python daha önemli."

Tablo 3. "Yazılım kullanmanın günümüzdeki yeri" teması ve alt temaları

Yazılım kullanmanın günümüzdeki yeri	f
Günümüzde kodlamagereklidir.	4
Yazılım öğrenmek önemlidir.	3
Yaşamda bir ürün elde edilebilir.	3

Dijital çağ olarak da adlandırılan günümüz çağında bütün gereksinimlerimiz sanal ortamda karşılanabilmektedir. İnsan ile etkileşim dijital ara yüz ile sağlanırken aslında arka planda çalışan kodlar, yazılımlar kullanıcının tamamen bilgisi dışındadır ve asıl işlevleri yerine getiren de bu kodlardır. Görüşülen öğrenciler de yazılımların günümüzde ne denli önemli olduğuna vurgu yapmışlar ve yazılım öğrenmenin gerekliliğini ifade etmişlerdir. Görüşmelerde sadece bir ürün ortaya koyarak eksiklikleri giderici işlevlerinin yanında yazılım öğrenme ve kullanmanın günlük yaşamdaki problemlere çözüm olabileceğinden bahsetmişlerdir. Öğrenciler kodlamada problemin bilgisayar dili ile nasıl çözülebileceğine ilişkin kodlar üretirler, sistematik hareket etmeyi öğrenirler ve bunu davranışa dönüştürdüklerinde günlük yaşantılarına yansıtırlarsa yaşamda karşılaştıkları problemlerin üstesinden gelmede öğrendiklerini kullanabilirler.

“Yazılım kullanmanın günümüzdeki yeri” temasına ilişkin katılımcılar şu görüşlere yer vermişlerdir:

Öğr.5; “Tek başıma alkış ile yanan lamba, mesafe ölçer, radar, metal dedektörü - mayın dedektörü gibi uygulamalar yaptım.”

Öğr.1; “Bir kişinin vücut kitle endeksini hesaplayıp kategorilere ayıran uygulama yaptım ve bunu gerçek hayatta kullandım. Bu, bende inanılmaz bir iz bırakmıştı.”

Öğr.5; “Arduino ile hayal ettiğim herşeyi ürün olarak yapabiliyorum.”

Öğr.1; “Kodlamada programlama dili yalnızca araçtır. İnsan önce kafasında düşüncelerle kodlar ve sonrasında bu düşünceleri bilgisayara aktarır. Yani kodlama bizi analitik düşünmeye sürükler. Kısacası kodlama insanı daha yaratıcı yapar. Problem çözme yeteneğimizi geliştirir ve en önemlisi düşünmeden yaşayanların çoğunlukta olduğu bu dönemde insanın düşünmesine ve problemleri kendi kendine çözmesine zemin hazırlar.”

Öğr.3; “Günümüzde kodlamayı öğrenmek çok önemli. Çünkü dünyamız gün geçtikçe her gün yeni bilgiler eklenerek gelişen bir dünya oluyor. Bu durum, gelişen dünyada kodlamayı öğrenmeyi gerekli kılıyor. Çünkü yaşamımızdaki her şey artık manuel değil de kodlama ve yazılım.”

Öğr.4; “Teknoloji gelişmekte ve teknolojiye ayak uydurmak, teknolojiye katkı sağlamak için bu yazılımları bilmek gerekli”

Öğr.6; “Kodlama geleceğin dünyası için en önemli yeteneklerdendir. Neredeyse her yerde program ve kodlama var artık. En önemli gerekçelerden biri de iş bulma açısından avantajdır, ileriye dönük planlar için. Yani kısacası hayatımızın her yerinde kodlama ve yazılım var. Bu yüzden her insanın en azından bir kodlama dilini bilmesi gerekir.”

Öğr.7; “Kodlama öğrenmek bence gerekli çünkü insanın her konuda yaratıcılığını arttırıyor, planlayarak bir işi yapmayı öğretiyor ve hızlı ve doğru kararlar vermemizi kolaylaştırıyor.”

Öğr.1; “Günlük yaşama dair uygulamalar gerçekleştirdim. Okulumuzun bilişim teknolojileri öğretmeni ile Tübitak fuarına Arduino ve Scratch kullanarak birden fazla ürün çıkarıp gören herkesin takdirini kazanmayı başardık.”

Öğr.7; “İlerde yazılım mühendisi olmak istediğim için akademik açıdan yazılımların bana katkısı olacağına inanıyorum. Yaratıcılığımı ve düşünme tarzımı geliştirdiği için de günlük hayatıma katkısı olduğunu düşünüyorum.”

Kelime Bulutu

Görüşmelerde yoğun olarak ifade edilen terimlerin belirlenmesi, daha çok hangi kavramlardan bahsedildiğinin anlaşılması için kelime bulutu oluşturulabilmektedir. Araştırmada görüşme esnasında en fazla kullanılan kelimelerin açığa çıkarılması, en fazla değinilen ifadelerin tespit edilmesi için NVivo ile kelime bulutu oluşturulmuştur. Kelime bulutuna göre yazı boyutu büyük olan kelimeler en sık tekrar edilen kelimeler olarak göze çarpmaktadır. Buna göre görüşme boyunca en fazla kodlama, Phyton, Sctrach, kaynak, programlama gibi kelimelerin geçtiği anlaşılmıştır.



Şekil 1. Nvivo ile elde edilen kelime bulutu

Tartışma, Sonuç ve Öneriler

Açık kaynak yazılımların avantajları öğrenciler için ilgi çekicidir. Öğrenciler bu yazılımlardan okulda öğrendikleri arasında benzerlikleri, farklılıkları ve tercihlerini ifade etmişler, günümüzde yazılım kullanmayı bilmenin önemini vurgulamışlardır. Öğrencilerden çok az bir kısmının (2 öğrenci) açık kaynak kodlu yazılımlara örnek olarak lisanslı yazılımları örnek vermesi, GPL lisansının özellikleri ile ilgili yorum yapamaması veya GPL lisansına ait olmayan özellikleri ifade etmesi öğrencilerin açık kaynak kodlu yazılımlar ve GPL lisansı ile ilgili bilgilerinde eksiklik ve kavram karmaşası olduğunu göstermektedir. Bununla birlikte genel olarak öğrenciler açık kaynak kodlu yazılımların daha şeffaf ve güvenilir olduğunu düşünmekte ve bu yazılımların kodlarının görülebilme, değiştirilebilme ve geliştirilebilme özellikleri üzerinde durmuşlardır. GPL lisansı için ise öğrenciler genel olarak öğrenme alanında bu lisansla lisanslanan yazılımların kendileri için kolaylık ve özgürlük sağladığı, değiştirilebilirliği ve gelişime daha açık olduğu üzerinde durmuşlardır. Sonuç olarak öğrencilerin açık kaynak kodlu yazılımlar konusunda belirli bir farkındalığı vardır.

Araştırma sonuçlarına göre öğrenciler kariyer hedefleri açısından, içinde bulunduğumuz çağın bir gereği olduğunu düşünmeleri açısından ve kendilerine yaratıcı düşünme, problem çözme, hızlı ve doğru karar alma, planlı olma gibi beceriler kazandıracağını düşündükleri için programlama öğrenmenin gerekli olduğunu düşünmektedirler. Öğrenciler Scratch, Arduino ve Python kodlama araçlarının açık kaynak kodlu olmasının kodların arka planda nasıl çalıştığını görmelerine izin vermesi, ücretsiz kullanabilmeleri ve kolay şekilde ulaşabilmeleri açısından avantajlı bulmuşlardır. Scratch, Arduino ve Python benzerlik, farklılık, ara yüzün ve menülerin kullanılabilirliği, öğrenme kolaylığı ve ortaya konulan ürünler açısından karşılaştırılmışlardır.

Öğrenciler her programlama dilinin kendine özgü yapısı ve terimleri olduğunu buna karşın döngüler ve algoritma mantığı bakımından benzer olduklarını ifade etmişlerdir. Aynı zamanda bu programlama dillerinin hepsini kullanarak ortaya bir ürün çıkarmanın mümkün olduğunu, Arduinoda ise somut ürünler ortaya çıkarılabildiğini dile getirmişlerdir. Öğrencilere göre Scratch diğer yazılımlara kıyasla daha görseldir, ara yüzü ve menüleri daha kullanışlıdır. Kodların bloklar halinde hazır şekilde olması sebebiyle daha kolay öğrenildiğini düşünmektedirler. Ayrıca öğrencilerin Scratch'ı daha çok benimsedikleri, daha akılda kalıcı ve eğlenceli buldukları ve daha hızlı kullandıkları anlaşılmıştır.

Öğrenciler için Python'ın daha zorlayıcı olduğu, kodların hazır olmaması sebebiyle kod yazımının daha uzun sürdüğü, Gezgin, Özcan, Ergün, Köse ve Emir'in (2017) de çalışmalarında ulaştığı sonuç gibi Python'da daha söz dizimi ve derleme sebebiyle sık hata aldıkları ve bu hatanın kaynağını bulmakta güçlük çektikleri sonuçlarına ulaşılmıştır. Bununla birlikte öğrenciler Python'ı daha profesyonel bulduklarını, günlük yaşamda ve kariyer hedeflerinde daha çok işlerine yarayacaklarını düşündüklerini belirtmiştir.

Genel olarak öğrenciler Scratch'ı kullanmanın daha kolay olduğunu düşünse de Python'ı da kolay bulan öğrenciler vardır. Bu noktada öğrencilerin uygulamalardan beklentileri, kişisel özellikleri, programlamaya yönelik tutumları ve ön öğrenmeleri programlama dillerinin öğrenimini kolaylaştırıcı etkiye sahiptir. Nitekim Python'ın daha çok işine yarayacak olduğunu düşünen ve programlamaya ilgisi olan öğrenciler, Python'ın kolay öğrenildiğini düşünmektedir. Bununla birlikte öğrenciler daha profesyonel olması ve daha çok alanda kullanılabilirliği sebebiyle Python'ı daha önemli bir yazılım olarak değerlendirmişlerdir. Arduino'yu ise genel olarak öğrenciler karmaşık ve öğrenilmesi zor olarak ifade etmişlerdir.

Farklı sebeplerden veya yazılımın farklı özelliklerinden dolayı öğrencilerin tercihlerinin değişebildiği anlaşılmıştır. Daha görsel ve animasyon içerdiği, oyuna benzediği için Scratch'ı, daha işlevsel, elde edilen ürünlerin gerçek yaşamda daha kullanılabilir, yazılıma olan ilgilerini çektiği için Python'ı, somut ürünler elde edebildikleri için ise Arduino'yu tercih eden öğrenciler olmuştur. Öğrencilerin ürettikleri ürünler göz önünde bulundurulduğunda Karabak ve Güneş'in (2013) gerçek hayatta karşılaşılan problemlere uygulanabilirlik kriterine göre öğrencilere verilen programlama eğitiminin başarılı olduğu söylenebilir.

Öğrenciler programlama dillerinin akademik ve günlük yaşantılarında kendilerine katkıları olduğunu ve çağımızın gereği olarak en az bir programlama dilini öğrenmek gerektiğini ifade etmişlerdir. Bazı öğrenciler Ceylan ve Gündoğdu'nun (2018) da belirttiği gibi gelecekteki kariyer hedeflerine kendilerini hazırladığı için, bazı öğrenciler ise bu programlama dillerini kullanarak günlük yaşamda işlerini kolaylaştıracak ürünler ortaya çıkarabildikleri için kendilerine katkıları olduğunu düşünmektedirler. Bu sonuçlar Göksoy ve Yılmaz'ın (2018) çalışmasında elde ettiği programlamanın öğrencilerin akademik başarılarına, günlük yaşam becerilerine katkı sunduğu sonuçlarıyla örtüştüğü söylenebilir.

Görüşülen öğrencilerin çoğunluğu okul dışı süreçte de bu programlama dillerini kullanarak yazılım ve ürün geliştirme çalışmaları yapmışlardır. Bu denemelerde öğrenciler alkış ile yanan lamba, mesafe ölçer, radar, metal dedektörü, vücut kitle indeksini hesaplayan yazılım gibi ürünler ortaya çıkarmışlardır. Ayrıca TÜBİTAK projelerinde de bu programlama dilleri ile ürün ortaya koyanlar olmuştur.

Öğrencilerin bu programlama dillerinin öğretimi konusunda bilişim teknolojileri öğretmenlerinden, okul idaresinden ve daha üst mercilerden çeşitli beklentileri olmuştur. Bu beklentiler aynı zamanda araştırmanın önerilerine kılavuzluk etmesi bakımından önemli görülmektedir. Beklentilerin karşılanması durumunda daha etkili ve verimli çalışmalar yürütebileceklerine inanmaktadırlar. Öğrenciler beklentilerini şu şekilde ifade etmişlerdir:

Öğrencilere tablet veya bilgisayarlar dağıtılarak bu programlama dillerinin öğretimi desteklenmelidir.

Programlama öğretimine daha erken sınıf düzeylerinde başlamalıdır.

Eğitimde uygulamaya daha fazla vakit ayrılmalı.

Yetkili merciler ve okul yönetimi teknolojik imkânları artırmak için yardımcı olmalıdır. Okullara bilgisayar laboratuvarları açılmalıdır.

Araştırma bir mesleki ve teknik anadolu lisesinin 11. sınıfında öğrenim gören 9 öğrenci ile yazılım bakımından Arduino, Scratch ve Python yazılımları ile sınırlıdır.

Bu araştırmanın sonuçları temel alınarak geliştirilen öneriler şu şekildedir:

Öğrencilere programlama öğretiminde daha fazla uygulama yapma imkânının sağlanması için gerekli teknolojik araçlar yetkililer tarafından temin edilmeli ve okullarda kapanan bilgisayar laboratuvarları tekrar açılmalıdır. Öğretim programları düzenlenirken programlama konusunda açık kaynak kodlu yazılımlar dikkate alınarak teşvik edilmelidir.

Başka araştırmalarda farklı açık kaynak kodlu yazılımlar ile ve/veya farklı öğretim basamağındaki öğrenciler ile araştırma yürütülebilir.

Etik Kurul Belgesi

Etik Kurul Komisyon Adı: Hatay Mustafa Kemal Üniversitesi, Sosyal ve Beşeri Bilimler Bilimsel Araştırma ve Yayın Etiği Kurulu

Etik Kurul Belge Tarihi: 04/02/2021

Etik Kurul Belgesi Sayı ve Numara: E-21817443-050.99-11996

Yazar Katkı Beyanı

Sevil HANBAY TİRYAKİ: *Literatür taraması, veri toplama aracının geliştirilmesi, uzman görüşleri alma, verilerin toplanması, analizi, temaların ve alt temaların oluşturulması, kodlamaların yapılması, tartışma ve sonuç bölümleri yazımı.*

Fatih BALAMAN: *Literatür taraması, veri toplama aracının geliştirilmesi, uzman görüşleri alma, verilerin analizi, temaların ve alt temaların oluşturulması, kodlamaların yapılması, bulguların yazımı, tartışma ve sonuç bölümleri yazımı.*

Kaynaklar

- Alp, Y. (2019). *Blok tabanlı programlama öğretiminin ortaokul öğrencilerinin problem çözme becerisine ve bilgisayara yönelik tutumuna etkisi* (Yayınlanmamış yüksek lisans tezi). İnönü Üniversitesi Eğitim Bilimleri Enstitüsü, Malatya.
- Bell, T., Alexander, J., Freeman, I. & Grimley, M. (2008). Computer science without computers: New outreach methods from old tricks. Retrieved 09.04.2021 from <http://www.jasonalexander.kiwi/pdf/NACCO08.pdf>
- Ceylan, V. K. & Gündoğdu, K. (2018). Bir olgubilim çalışması: Kodlama eğitiminde neler yaşanıyor? *Eğitim Teknolojisi Kuram ve Uygulama*, 8(2), 1-34.
- Çatlak, Ş., Tekdal, M. & Baz, F. Ç. (2015). Scratch yazılımı ile programlama öğretiminin durumu: Bir doküman inceleme çalışması. *Journal of Instructional Technologies ve Teacher Education*, 4(3), 13-25.
- Demirer, V. & Sak, N. (2016). Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar, *Eğitimde Kuram ve Uygulama*, 12(3), 521-546.
- Dubois, P. F. (2007). Guest editor's introduction: Python: Batteries included. *Computing in Science & Engineering*, 9(3), 7-9.
- Genç, Z. & Karakuş, S. (2011, Eylül). *Tasarımla öğrenme: Eğitsel bilgisayar oyunları tasarımında Scratch kullanımı*. 5th International Computer & Instructional Technologies Symposium, Fırat University, Elazığ, Turkey.
- Gezgin, D. M., Özcan, S. N., Ergün, K., Köse, Ö. & Emir, N. (2017). *Bilgisayar programlama eğitiminde Scratch programı kullanımına ilişkin lise öğrencilerinin görüşleri*. Proceedings Book of 2nd International Scientific Researches Congress on Humanities and Social Sciences, 182-188, İstanbul.

- Gomes, A. & Mendes, A. J. (2007, January). *Learning to program difficulties and solutions*. International conference on Engineering Education, Coimbra, Portugal.
- Göksoy, S. & Yılmaz, İ. (2018). Bilişim teknolojileri öğretmenleri ve öğrencilerinin robotik ve kodlama dersine ilişkin görüşleri. *Düzce Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 8(1), 178-196.
- Hwang, S.-Y. (2005). Adopting open source and open standards in the public sector: Five deciding factors behind the movement, *Michigan Journal of Public Affairs*, 2, 1-19.
- Kandemir, C.M. (2018). Metin tabanlı programlama. Y. Gülbahar, H. Karal (Ed.). *Kuramdan uygulamaya programlama öğretimi içinde*, Ankara: PEGEM Akademi.
- Karabak, D. & Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 2(3), 163-169.
- Karahoca, D., Karahoca, A. & Uzunboylu, H. (2011). Robotics teaching in primary school education by project based learning for supporting science and technology courses. *Procedia Computer Science*, 3, 1425-1431.
- 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
- Kuş, E. (2009). *Nitel - nitel araştırma teknikleri*. Ankara: Anı Yayıncılık.
- López-Rodríguez, FM. & Cuesta, F. (2016). Andruino-A1: Low-cost educational mobile robot based on android and Arduino. *Journal of Intelligent & Robotic Systems*, 81, 63-76.
- Lye, S. Y. & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61.
- Malan, D. J. & Leitner, H. H. (2007). Scratch for budding computer scientists. *ACM Sigcse Bulletin*, 39(1), 223-227.
- Malkoç, B. (2012, Şubat). *Temel bilimler ve mühendislik eğitiminde programlama dili olarak Python*. XIV. Akademik Bilişim Konferansı Bildirileri, Uşak Üniversitesi, Uşak.
- Maxwell, J. (2018). *Nitel araştırma tasarımı - etkileşimli bir yaklaşım* (Çev: M.Çevikbaş). Ankara: Nobel Yayınları.
- Miles, M. B. & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. (2nd ed). Thousand Oaks, CA: Sage Publications.
- Milli Eğitim Bakanlığı (MEB), (2018a). Bilişim teknolojileri ve yazılım dersi öğretim programı (Ortaokul 5 ve 6. sınıflar) <https://mufredat.meb.gov.tr/ProgramDetay.aspx?PID=374 adresinden> 07 Ocak 2021 tarihinde alındı.
- Milli Eğitim Bakanlığı (MEB), (2018b). Ortaöğretim bilgisayar bilimi dersi öğretim programı <https://mufredat.meb.gov.tr/ProgramDetay.aspx?PID=335 adresinden> 07 Ocak 2021 tarihinde alındı.
- Ocak, M. A. & Efe, A., (2018). *Arduino ile kodlama ve mikro denetleyici uygulamaları*. Ankara: Anı Yayıncılık.
- Öztürk, F. & Özdemir, D. (2020). The effect of STEM education approach in science teaching: Photosynthesis experiment example. *Journal of Computer and Education Research*, 8(16), 821-841. DOI: 10.18009/jcer.698445
- Pinto, A. & Escudeiro, P. (2014, June). *The use of Scratch for the development of 21 st century learning skills in ICT*. In Information Systems and Technologies (CISTI), 9th Iberian Conference on, IEEE7, Barcelona.
- Powers, K., Ecott, S. & Hirshfield, L. M. (2007). *Through the looking glass: Teaching CS0 with Alice*. SIGCSE 2007: 38th SIGCSE Technical Symposium on Computer Science Education, 213-217, New York, USA, <https://doi.org/10.1145/1227310.1227386>

- Raadt, M. (2008). Teaching programming strategies explicitly to novice programmers. Retrieved 09.09.2020 from http://eprints.usq.edu.au/4827/2/de_Raadt_2008_whole.pdf
- Resinovic, B. (2015, October). *The use of Nao, a humanoid robot, in teaching computer programming*, The Proceedings of International Conference on Informatics in Schools: Situation, Evolution and Perspectives (ISSEP), Ljubljana, Slovenia.
- Resnick, M., Silverman, B., Y., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67.
- Rossum, G. (1999). Computer programming for everybody (revised proposal): A scouting expedition for the programmers of tomorrow. *Technical Report*, Corporation for National Research Initiatives.
- Sarı, U. & Kırındı, T. (2019). Using arduino in physics teaching: arduino-based physics experiment to study temperature dependence of electrical resistance. *Journal of Computer and Education Research*, 7 (14), 698-710. DOI: 10.18009/jcer.579362
- Saygıner, Ş. & Tüzün, H. (2017, Mayıs). *Programlama eğitiminde yaşanan zorluklar ve çözüm önerileri*. I. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu, İnönü Üniversitesi, Malatya.
- Seggie, N. & Bayyurt, Y. (2017). *Nitel araştırma yöntem, teknik, analiz ve yaklaşımlar*. Ankara: Anı Yayıncılık.
- Sönmez, V. & Alacapınar, F. (2013). *Örneklendirilmiş bilimsel araştırma yöntemleri*. Ankara: Anı Yayıncılık.
- Steiniger, S. & Hunter, A. J. (2013). The 2012 free and open source GIS software map-A guide to facilitate research, development, and adoption. *Computers, Environment and Urban Systems*, 39, 136-150.
- Sönmez, Y. (2017). *Açık kaynak programlar ile internet üzerinden öğrenci takip sistemi tasarımı ve performans analizi* (Yayımlanmamış Yüksek Lisans Tezi). İstanbul Kemerburgaz Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Wakil, K., Khdir, S., Sabir, L. & Nawzad, L. (2019). Student ability for learning computer programming languages in primary schools. *International e-Journal of Educational Studies (IEJES)*, 3 (6), 109-115. DOI: 10.31458/iej.591938
- Yıldırım, A. & Şimşek, H. (2008). *Sosyal bilimlerde nitel araştırma yöntemleri*. Ankara: Seçkin Yayınevi.
- Yılmaz, N. (2017). *Açık kaynak yazılımlarda bakım yapılabilirliği ve güvenilirliği ölçmek için iki boyutlu değerlendirme metodu* (Yayımlanmamış Yüksek Lisans Tezi). Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara.