Croatian Journal of Education
Vol.22; No.2/2020, pages: 563-595
Original research paper
Paper submitted: 30th January 2019
Paper accepted: 27rd March 2020
https://doi.org/10.15516/cje.v22i2.3560

A Framework for Studying Programming Teaching in Secondary Education

Ali Kürşat Erümit[1], Ali İhsan Benzer[2] and Güven Şahin[3]
[1]*Trabzon University*
[2]*Hatay Mustafa Kemal University*
[3]*Rize Recep Tayyip Erdoğan University*

Abstract

This study extensively reviews the literature on teaching programming to middle schoolers, with a qualitative content analysis method, and intends to put forth a research design framework for researchers that will guide them in the processes of planning and designing their research on teaching programming for middle school learners. For access to the relevant literature; the databases were searched by using the following keywords: "computer", "programming", and "middle school" together, limiting the findings to the articles published after 2000. As a result, an upward tendency was noted in studies about the teaching of programming at secondary level considering years, most of which are comprised of empirical ones. Also, the existing studies were mostly carried out with 6th graders predominantly employing data collection tools of questionnaires/scales and achievement tests. As for programming tools, Scratch was seen to be the most commonly used one. Although quite a few articles are investigating the context of the programming teaching lesson, some of the studies were found to use programming as a means of teaching mathematics, natural sciences, languages, writing skills, and social sciences. In conclusion, the present study is expected to pave the way for future research by highlighting the overall situation of programming teaching

Keywords*:* *coding; computer programming; content analysis; middle school; programming teaching*

Introduction

In the working life, the increased demand for experts in the field of informatics has pushed up the interest in computer sciences, which has been translated into the teaching of computer sciences courses at different levels of education (Qian & Lehman, 2016). Several researchers and studies emphasize the potentially significant benefits of inclusion of programming teaching in primary and middle school curricula (Saez-Lopez et al., 2016). The teaching of programming supports the development of several skills referred to as the skills for the 21st century, which are problem-solving (Calao et al., 2015; Fessakis et al., 2013), creative thinking (Gupta et al., 2012; Navarrete, 2013), algorithmic thinking (Hromkovic et al., 2016), reflective thinking (Kalelioğlu, 2015), critical thinking, and computational thinking skills. The augmenting effects of programming are not limited to the abovementioned skills as it also promotes learning of mathematics, natural sciences, social sciences, and languages as well (Baki & Özpınar, 2007; Baytak, Land, & Smith, 2011; Brown et al., 2008; Burke, 2012; Ferrer-Mico et al., 2012; Ke, 2014; Moreno et al., 2016; Navarrete, 2013; Saez-Lopez et al., 2016). For these reasons, it is seen that there has been an intense attempt to teach programming in schools, especially in recent years (Moreno-Leon et al., 2016). Most countries have already placed programming teaching, which is expected to bring considerable benefits, into their education curricula and have already put it into effect at various levels of schooling, starting from the primary (Han et al., 2016; Heinth et al., 2016; Mone, 2018).

Programming is the process of transferring problems to the computer environment through software after modelling them (Benzer & Erümit, 2017). Lye and Koh (2014) think that programming requires abstraction and analysing skills, beyond merely coding. The fact that programming as a process of generating solutions for problems entails several mental skills (Benzer & Erümit, 2017) causes students to face difficulty in programming learning teaching and perceive programming, too difficult (Aşkar & Davenport, 2009; Guzdial & Soloway, 2002; Lahtinen et al., 2005). Particularly during middle school years, when learners ascend from the concrete operational phase to the abstract operational phase, today's graphics-based visual programming tools are used to overcome the difficulties. Software products such as Alice, Scratch, Microsoft Small Basic, Toontalk, Stagecast Creator, and Code GameLab serve this purpose. These tools are preferred especially for enhancing algorithmic thinking skills of middle school students and facilitating the teaching of programming.

With the introduction of software that facilitates the teaching of programming, it can be easily noticed that academic studies in this area have become widespread across different age groups, courses, and objectives. Lately, the interest and curiosity of researchers about programming teaching, which plays an important role in the teaching of the 21st-century skills, have increased. At the same time, the number of studies dealing with programming education has increased significantly. As a result of these, research designs of studies have shown certain trends recently.

The purpose of the studies on programming teaching was mostly to improve the cognitive and/or affective characteristics of the participants. Studies have examined the effects of different pedagogical approaches and teaching environments for this purpose.

Shim, Kwon, & Lee (2017) and Tocháček, Lapeš, & Fuglík (2016) emphasized that robotic programming environments positively improve students' attitudes towards programming and help them learn to program. Meerbaum-Salant, Armoni & Ben-Ari (2013) and Oluk, Korkmaz & Oluk (2018) have determined that the Scratch environment is effective in teaching programming concepts. Vatansever & Baltacı-Göktalay (2018) concluded that programming with Scratch positively affected the development of the problem-solving skills of the participants. Werner & Denning (2009) and Hartl et al. (2015) found that pair programming positively affects students' motivation and problem-solving skills. Johnson (2017) and Baytak & Land (2011) stated that game-based programming teaching is effective in teaching programming concepts.

The increase in academic studies on the teaching of programming has led to the appearance of common points and differences in the planning of research designs. Still, distinctions remain in study aim, the programming tool used, the content, the data collection tools, and duration of research related to programming teaching.

This study aims to analyze the articles related to programming teaching for middle school pupils by using content analysis method and to propose a framework for the design of studies on programming teaching in middle schools. The ultimate goal of the study is to depict the overall look of the teaching of programming and guide future studies accordingly. The study seeks to answer the following questions.

1. What does the distribution of published articles on middle school students look like against certain criteria (year, research method, participants' grade, data collection tools, programming tool, content, and duration)?
2. What do published articles about middle school students have in common considering research design qualities?

Method

In this study, a review was conducted on articles published about teaching and using programming for middle school students. For data analysis, the content analysis method was used as required by the nature of the study, namely document review. Content analysis is a research method based on composing, classification, comparison, and theoretical interpretation of documents. This method was chosen to bring together the data concerned and to present them in a way readable and understandable to readers in this study. The articles were reviewed to find out the year, method, sample level, data collection tool, programming tool used, teaching time, content, aim, and results. Study data were collected by using an article review form and then analysed with descriptive statistical methods.

Population and Sample

The study population consists of the articles on teaching and using programming for middle school students published in refereed journals.

Collection and Analysis of Data

During the literature review, the keywords "computer" "programming", and "middle school" were used at one same time in the databases Google Scholar, ERIC, and Web of Science for articles published after 2000. The search was limited to study groups consisting of middle school students at grades 5 to 8, or students aged 10 to 15 since middle school covers different years in different countries.

The main parts of the articles such as title, abstract, and keywords were used as inclusion criteria in this study. In cases where decisive information could not be obtained from these parts, other sections of the articles were examined. A total of 62 articles that meet the criteria were obtained from the preliminary examination for further analysis and included in the study. The articles covered in our study were analysed and coded against the criteria of year, content, aim, and results by using the article review form. The selection and coding of the articles were carried out by two researchers. Cohen's kappa statistic was used to determine whether the reliability between the two coders was acceptable. Kappa coefficient was calculated as 0.92. According to this result, compatibility between coders is perfect (Landis & Koch, 1977). The results of the analysis are presented in tables and diagrams.

Results

This study was intended to reveal the distribution of the studies on programming teaching to find out the similarities and eventually devise a framework to be used in designing studies for programming teaching at the secondary level. As a result of the analysis of the articles by year, research method, sample group age, data collection tool, programming tool, content, duration, aim, and results, the distributions are displayed in charts. The examined articles are presented in Appendix 1. In light of these findings, common aspects of the studies related to design were elicited and a research design framework was formed. The framework is shown in Figure 11.

Analysis of Articles According to the Years

Distribution of the studies discussing the teaching of programming to middle school students was analysed and by years and the findings were given in Figure 1 below.

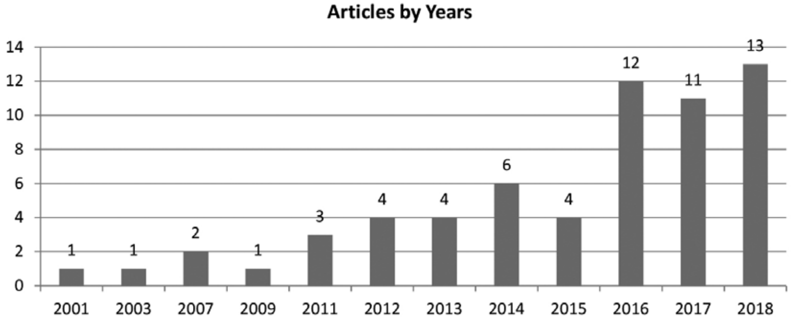

Figure 1. Distribution of Studies by Years

According to Figure 1, the number of studies related to programming teaching at the secondary level has increased across years, the highest number of studies being published in 2018.

Analysis of Articles According to the Research Methods

Distribution of the articles by the research methods was shown in Figure 2 below.

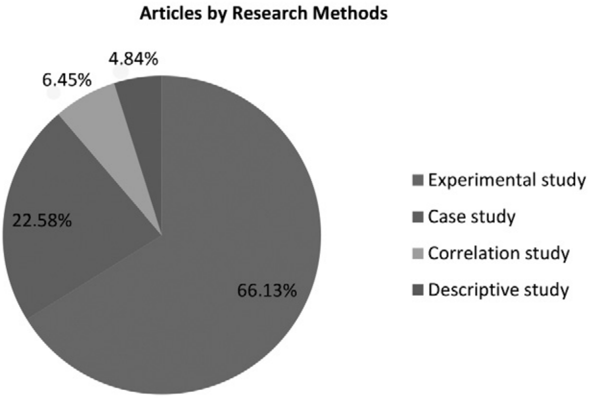

Figure 2. Distribution of Studies by Research Methods Used

Figure 2 demonstrates that the majority of the articles under consideration comprised of experimental studies. According to our analysis, a total of 62 articles include 41 (66 %) experimental studies, 14 case studies (22 %), 4 correlation studies (6 %), and 3 descriptive studies (4 %).

Analysis of Articles According to the Participants

Below is shown the distribution of the articles by study group age in Figure 3.

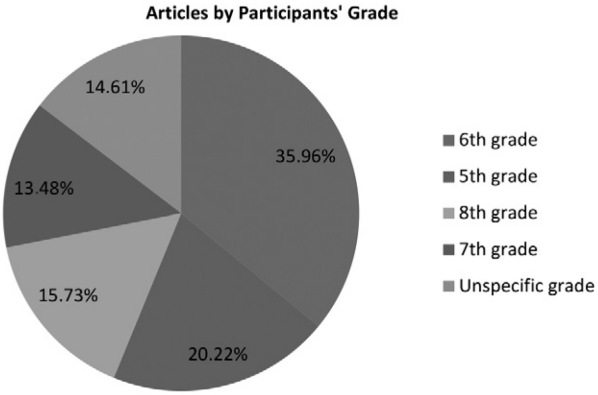

Figure 3. Distribution of Studies by Participants' Grade

Figure 3 shows that within secondary education, the 6th grade is the most frequently preferred grade level for researches on programming teaching (36 %), which is followed by the 5th grade (20 %) and 7th grade (13 %). However, some of the studies were found to have mixed classes covering more than one specific class level. Likewise, there were a large number of studies which indicate the range of age only, without the grade level (14 %).

Analysis of Articles According to the Data Collection Tools

Distribution of the investigated studies by data collection tools is shown in Figure 4.

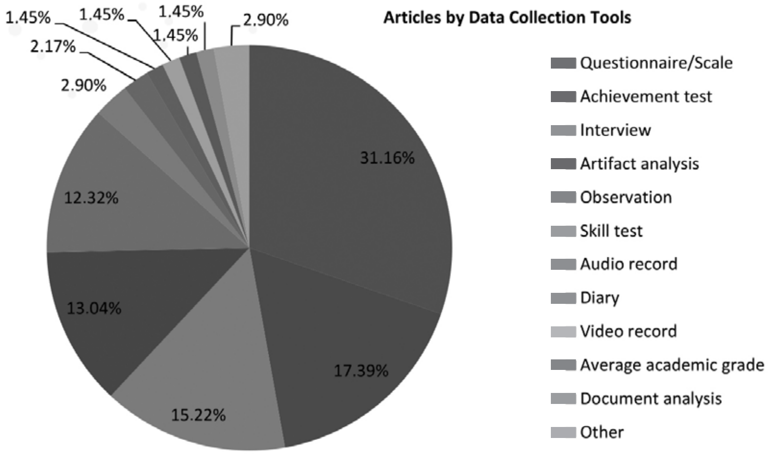

Figure 4. Distribution of Studies by Data Collection Tools Used

As seen in Figure 4, it was found out that the most commonly used data collection tools in studies investigating the teaching of programming at middle school level were questionnaires/scales (31 %), followed by achievement tests (17 %), and interviews (15 %). Besides, more than one data collection tool was used in most studies.

Analysis of Articles According to the Programming Tools

Figure 5 below displays the distribution of the studies in scrutiny by programming tools used.

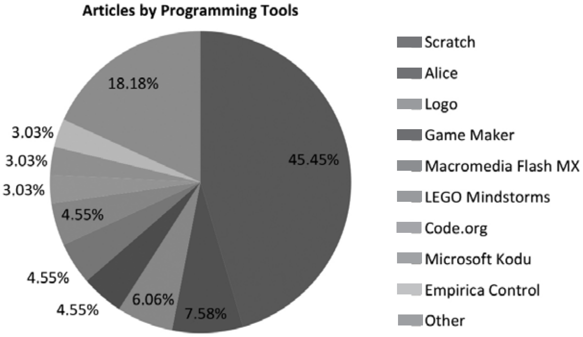

Figure 5. Distribution of Studies by Programming Tools Used

Figure 5 reveals that Scratch is the most popular programming tool in the studies discussing programming teaching at the secondary level (45 %). Scratch was followed by other tools as Alice (7 %) and Logo (6 %), respectively. Apart from these, some of the studies employed more than one programming tool. In the pie chart above, "Other" (18.18 %) contains the tools used in one specific study and once only.

Analysis of Articles According to the Contents

In Figure 6, the distribution of the sample studies by the contents is shown.

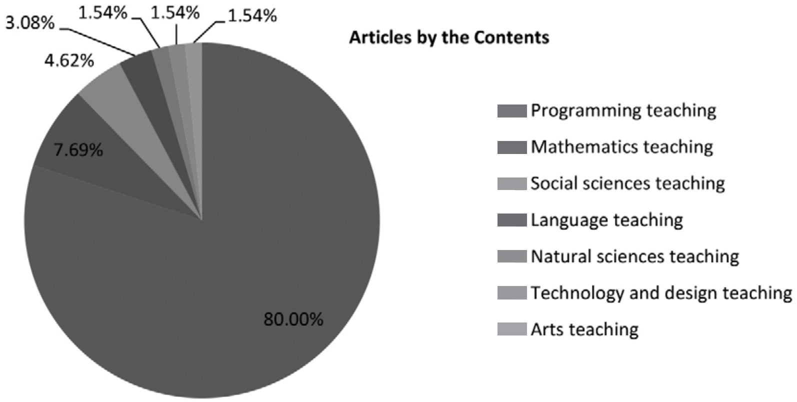

Figure 6. Distribution of Studies by the Contents

Figure 6 demonstrates that a substantial portion of the studies in our sample relates to programming instruction. However, some studies were seen to address more than one particular content.

Analysis of Articles According to the Experimental Duration

Distribution of the studies examined here by experimental duration is as shown in Figure 7.

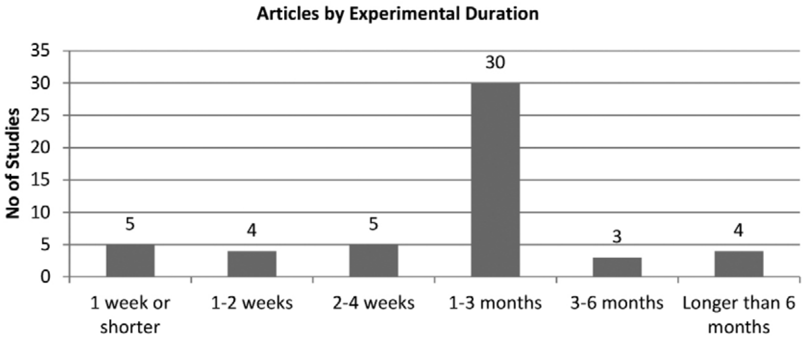

Figure 7. Distribution of Studies by Experimental Duration

As understood from Figure 7, the majority of the studies investigated here were implemented for 1 to 3 months. In addition, it was found out that the duration was the same, 1-3 months, in studies carried out to develop both cognitive and affective traits and those targeting cognitive traits only.

Analysis of Articles According to the Aims

It was found out that the overall goal of the educational studies reviewed here is to examine or develop cognitive or affective domains of participants. The aims of the studies here were divided into two groups as cognitive and affective studies, considering the aims and data collection tools used in respective studies. The results and distributions are shown in Figure 8.

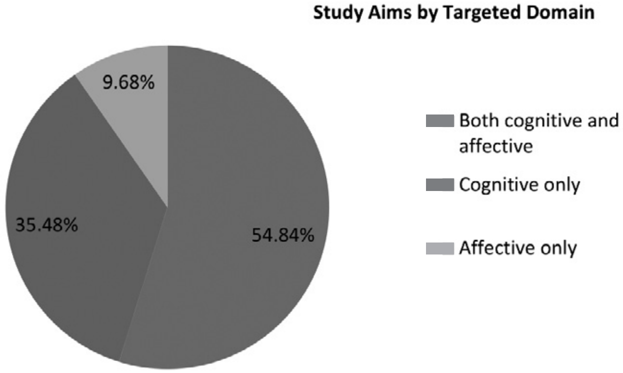

Figure 8. Distribution of Studies by Cognitive or Affective Domain in Target

In Figure 8, it is seen that 34 of a total of 62 studies intend to analyse or improve both cognitive and affective traits of participants. This is followed by 22 studies (35 %) aimed at examining or developing only cognitive traits and 6 examples (9 %) targeting to examine or develop only affective traits, respectively.

In particular, analysis of the studies discussing cognitive domain revealed that 40 of 56 studies deal with academic achievement and 21with cognitive skills. On the other hand, 35 of 40 studies studying the affective domain attitude were seen to focus on attitudes (interest, engagement, and student opinion), 4 on self-efficacy, and 2 deal with self-confidence.

From the perspective of data collection tools employed in cognitive studies, it was found out that achievement tests and artefact analysis were the most common means of measuring academic success. As regards to the measurement of cognitive skills (problem-solving, computational thinking), the most frequently applied tools can be listed as questionnaires/scales, skill tests, interviews, observations, and audio records. In the studies investigating affective traits, it was reported that questionnaires/scales, interviews, and observations were mostly employed to analyse variables such as attitude/interest.

Analysis of Articles According to the Result

Results from 51 studies related to various effects of programming teaching were evaluated from the perspective of cognitive and affective purposes. Distribution of the results from both cognitive and affective intervention studies is given in Figure 9 and the results from those with cognitive domain only are distributed in Figure 10. Lastly, a special chart is not provided for affective studies only because of the number of such studies as small as 2.

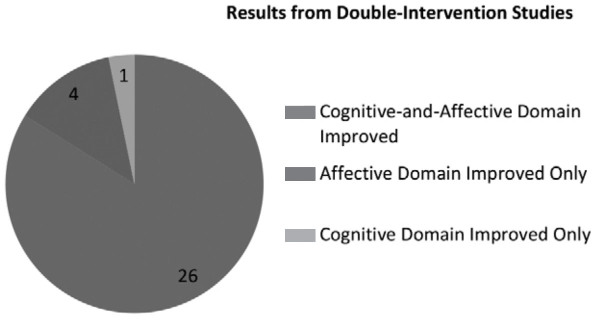

Figure 9. Distribution of Results from Cognitive-and-Affective Intervention Studies

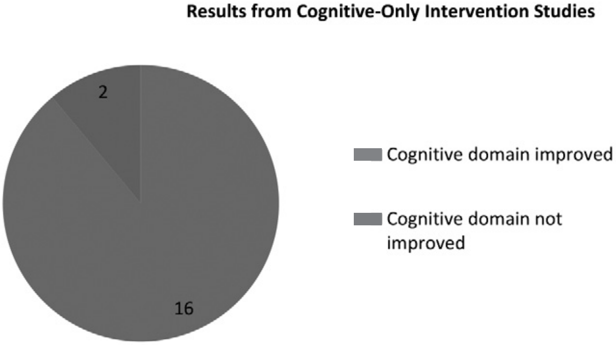

Figure 10. Distribution of Results from Cognitive-Only Intervention Studies

As Figure 9 and Figure 10 show, 31 intervention studies were carried out with the aim of developing both cognitive and affective traits of participants. In 26 of them, positive results were reported in both cognitive and affective traits of the students; while 4 studies could not reach any difference in cognitive traits of the students, still influencing affective traits positively. In the remaining 1 study, the intervention did not prove any difference in affective traits of the students, while making a positive effect on cognitive traits. Of all the studies, 18 aimed to improve only the cognitive abilities of the participants. As a result, a positive change was noted in cognitive traits of the participants in 16 of the studies. The rest of the two studies ended up with no cognitive improvement at all. As for the other domain, 2 studies were designed to develop only affective traits of the participants, both of which yielded expected results.

Discussion and Conclusion

In this research, the articles related to programming teaching to middle school students were examined with content analysis method according to year, research method, sample level, data collection tool, programming tool, content, experimental duration, study aim, and results. As a result, a research design framework was established for similar studies.

When studies on programming teaching for middle school students are viewed by years, it becomes noticeable that there has been a great increase in the number of studies in this matter especially in recent years, recording the peak number in 2018. This result seems in agreement with the findings of Benzer and Erümit (2017) reviewing postgraduate theses discussing the teaching of programming in the Turkish context, Şahiner and Kert (2016) reviewing studies related to computational thinking, and also Çatlak et al. (2015) investigating the examples which use Scratch. The studies inspecting effects of programming on cognitive processes are rooted in the 1960s when the programming tool LOGO was developed (Baki & Özpınar, 2007; Feurzeig, Papert, Bloom, Grant, & Solomon, 1969; Milner, 1973; Pea, 1983). In relation to the tendencies in academic studies, the huge increase in the number of studies on programming teaching in recent years seems to arise from the fact that the importance and necessity of programming teaching are appreciated for development of the 21st-century skills (Gunbatar & Karalar, 2018; Han et al., 2016; Shim et al., 2017; Soykan & Kanbul, 2018), most countries add programming-related courses to their primary, middle, and high school curricula (Han et al., 2016; Heintz et al., 2016; MoNE, 2018; Moreno-Leon et al., 2016), and programming tools have been specially designed for the teaching of programming at the secondary level (Kalelioğlu & Gülbahar, 2014; Korucu & Atıcı, 2018; Lye & Koh, 2014; Resnick et al., 2009).

The majority of the articles under review consist of experimental studies. The literature stresses that programming teaching has an important potential in developing the skills for the 21st century (Gunbatar & Karalar, 2018; Han et al., 2016; Kanbul & Uzunboylu, 2017; Papadakis, Kalogiannakis, & Zaranis, 2016; Shim et al., 2017; Soykan & Kanbul, 2018). This has led to an increase in the number of experimental studies examining the effects of programming instruction on the development of the 21st-century skills acquired by students.

An overview of the study samples in question demonstrates that the most preferred class level was the 6th grade and a lot of studies were carried out with students studying at different grade levels. This finding seems in accord with Benzer and Erümit (2017), which also noted that the 6th grade is the most often preferred class level among studies about programming teaching in middle schools. This class level covers a special period when children shift from concrete operations to abstract operations (Moreno-Leon et al., 2016). Good abstract thinking skills are regarded as crucial factors for success in the teaching of programming (Koppelman & van Dijk, 2010; Kramer & Hazzan, 2006). These reasons seem to stand as the main reasons for the researchers' choice of the 6th grade as the sample level.

As another criterion for inclusion in our study, the data collection tools were listed as questionnaires/scales with the highest frequency. Other most common instruments were found as achievement tests and interviews. This result appears compatible with findings from several previous studies including Şahiner and Kert (2016) investigating studies on computational thinking; Bacca et al. (2014) reviewing studies on augmented reality in instructional settings; Göktaş et al. (2012) screening works related to educational Technologies in Turkey, and Şimşek et al. (2009) reviewing master theses on educational technologies in Turkey. It is thought that questionnaires/scales are preferred with the highest frequency owing to the convenience of collecting data from larger groups easily and fast (Büyüköztürk, Kılıç-Çakmak, Akgün, Karadeniz, & Demirel, 2017). In addition, it was determined that more than one data collection tool was used in most studies. When it is necessary to collect both quantitative and qualitative data in studies, it may be necessary to use more than one data collection tool (Balcı, 2016).

Again, the studies reviewed here indicate Scratch as the most used programming tool. Nevertheless, some studies were found to use more than one programming tool. The result is consistent with the results of the study performed by Benzer and Erumit (2017). This might be accounted by the fact that text-based programming environments pose many difficulties to students (Özmen & Altun, 2014); while they encounter less trouble with visual-based programming environments, leading to smooth learning and understanding of programming (Wilson & Moffat, 2010). In most studies, it was reported that Scratch as a visual-based programming tool advances both cognitive and affective aspects of learning of programming by students (Brown et al., 2008; Burke, 2012; Deveci-Topal, Çoban-Budak, & Kolburan-Geçer, 2017; Ferrer-Mico et al., 2012; Meerbaum-Salant et al., 2013; Oluk et al., 2018). Yet, the way that different programming tools affect students' cognitive and affective traits remains a curious research topic.

In our review, it was seen that a significant part of the studies is related to programming teaching. In addition to this, studies are available which survey contributions of computer programming on language and social science teaching, especially mathematics teaching. This finding seems in conformity with Çatlak et al. (2015) and Şahiner and Kert (2016). Programming has the potentials to support teaching of not only computer sciences but also mathematics (Baki & Özpınar, 2007; Brown et al., 2008; Ferrer-Mico et al., 2012; Ke, 2014), natural and social sciences (Baytak et al., 2011; Navarrete, 2013; Saez-Lopez et al., 2016), and languages (Burke, 2012; Moreno-Leon et al., 2016). Since most of the research was done in computer science courses, it does not sound surprising to find the contents related to programming teaching.

In relation with duration, it can be said that studies on programming teaching at the secondary level mainly last for 1 to 3 months. This result is similar to findings reached by a set of preceding studies such as Benzer and Erümit (2017); Hazır-Bıkmazet al. (2013) investigating dissertations on educational curricula and teaching; Polat (2015) analysing studies on critical thinking skill; and Dirlikli et al. (2016) looking into master theses on collaborative learning in Turkey. This could be accounted for the fact that programming teaching is often restricted to one semester in curricula.

From the target domain point of view, it is seen that studies investigating cognitive traits (achievement and cognitive skills) outweigh those examining affective traits (attitude, perception, interest, motivation, opinion, self-confidence, self-efficacy). These results seem in conformity with Ulutaş and Ubuz (2008) researching the studies on mathematics education in Turkey and also Turhan-Türkkan and Arslan-Namlı (2018) giving insight into postgraduate Turkish theses regarding the use of software in the teaching of mathematics. More than that, the bulk of the studies investigated here address both cognitive and affective traits of participants. In our study, success was seen to be the most widely studied cognitive variable and attitude/interest as the most common affective variable, respectively. In broad terms, the objective of educational studies is to analyse or enhance cognitive or affective domains of participants. Bearing in mind that programming is promising for developing cognitive skills in the first place as a part of the 21st-century skills, it seems more than natural to investigate or develop cognitive traits in programming-related research. The emphasis in the literature on the existence of a positive relationship between the cognitive domain and the affective domains (Anastasiadou & Karakos, 2011; Hawi, 2010; Hongwarittorrn & Krairit, 2010; Korkmaz & Altun, 2013) is considered to justify the supremacy of the studies dealing with both domains.

As another criterion, the probing of the study results reveals that the majority could reach the desired objectives. Again, most of the examples were concluded with a positive effect on both the cognitive and the affective traits of the participants. The rate of success in the affective domains was found even higher than in the cognitive domains. In a similar study, Çatlak et al. (2015) reviewed the studies using Scratch as a programming tool. They ascertained that the entirety of the results was positive in connection with the affective domain and it was the case for a large portion of the cognitive domain. As another example, Lye & Koh (2014) screened studies which seek to enhance computational thinking skills through programming. They also reported positive results in most of the studies investigated. So we are at liberty to say that our result is compatible with Çatlak et al. (2015) and Lye & Koh (2014).

Lastly, the results of the studies of intervention let out the following statements with the highest incidence. Game design activities, binary programming methods, and robotic programming activities contribute to students' learning of concepts related to programming and improve their attitudes and motivation towards programming and thus their interest in the course. This finding is similar to the results recorded by Çatlak et al. (2015) on the impact of Scratch as a programming tool on the teaching process. However, it is possible to notice some contradictions concerning the results obtained from the studies discussing computational thinking and problem-solving skills as the prominent skills required in the 21st century. In some studies, the teaching of programming has a positive effect on computational thinking and problem solving skills (Akcaoglu, 2014; Brown et al., 2008; Chen et al., 2017; Vatansever & Baltacı-Göktalay, 2018; Werner & Denning, 2009; Yünkül, Durak, Çankaya, & Mısırlı, 2017); whereas some others mention

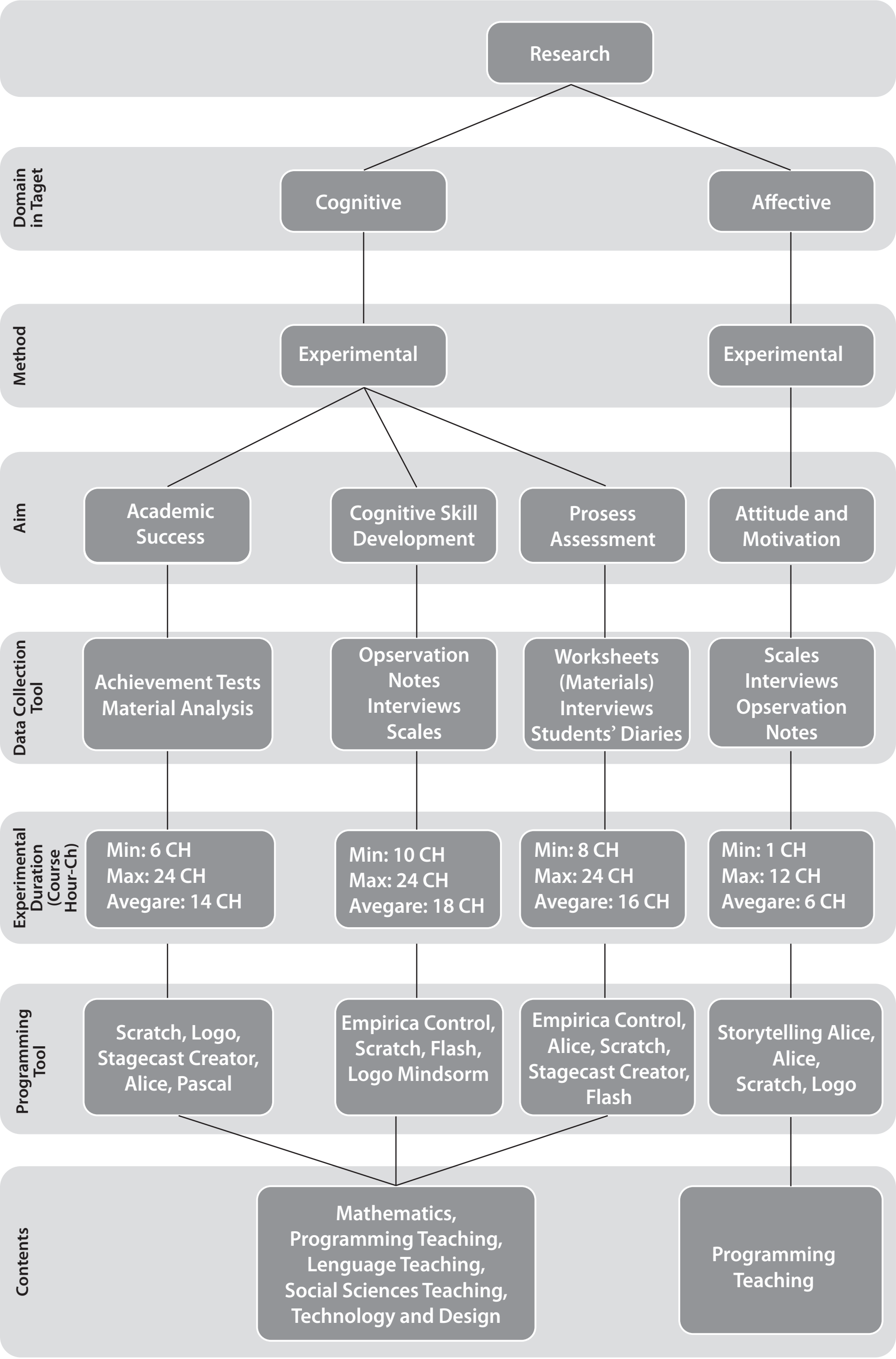
Research
Domain in Taget
Cognitive
Affective
Method
Experimental
Experimental
Aim
Academic Success
Cognitive Skill Development
Prosess Assessment
Attitude and Motivation
Data Collection Tool
Achievement Tests Material Analysis
Opservation Notes Interviews Scales
Worksheets (Materials) Interviews Students' Diaries
Scales Interviews Opservation Notes
Experimental Duration (Course Hour-Ch)
Min: 6 CH Max: 24 CH Avegare: 14 CH
Min: 10 CH Max: 24 CH Avegare: 18 CH
Min: 8 CH Max: 24 CH Avegare: 16 CH
Min: 1 CH Max: 12 CH Avegare: 6 CH
Programming Tool
Scratch, Logo, Stagecast Creator, Alice, Pascal
Empirica Control, Scratch, Flash, Logo Mindsorm
Empirica Control, Alice, Scratch, Stagecast Creator, Flash
Storytelling Alice, Alice, Scratch, Logo
Contents
Mathematics, Programming Teaching, Lenguage Teaching, Social Sciences Teaching, Technology and Design
Programming Teaching

Figure 11. Design Framework for Studies on Teaching Programming no benefit of the implementation on the said skills (Kalelioğlu & Gülbahar, 2014; Kukul & Gökçearslan, 2014; Oluk & Saltan, 2015). It would not be too daring to suggest that high-level skills such as problem-solving cannot be developed in a short time and hence long-term and multi-application studies should be carried out (Kalelioğlu & Gülbahar, 2014; Kukul & Gökçearslan, 2014; Oluk & Saltan, 2015).

In conclusion, although the studies examined here were planned with disparate programming tools, data collection tools, durations, contents, and aims, they exhibit similar aspects, which lays the foundations for a research framework appealing to researchers studying effects of programming teaching at the secondary level. The aspects of research design shared by the studies here were determined from the distributions of the studies and a design framework was created for programming teaching studies subsequently. The research design framework is modelled in Figure 11 below.

Figure 11 shows the design framework for studies on teaching programming in secondary education. It is believed that this framework will guide researchers in the processes of planning and designing their research on programming teaching. The model was drafted by identifying clusters of method, size, aim, data collection tool, teaching time, programming tool, and the contents of the course in the studies under scrutiny. It is seen that the studies on programming teaching employed diverse research methods depending on the respective aim in connection with the affective or the cognitive domain. Research methods vary depending on the design of the study. Another striking inference from the distribution of the researches is that in the whole of the studies, programming instruction was shaped under supervision of the students or researcher, instead of building them on a certain rationale or theoretical standard.

Although studies on teaching programming have been realized since the 1960s, it seems worth noting that the examples about programming teaching at the secondary level have densely attracted researchers' attention shortly and are expected to continue in the future. We think that the unveiling of similar and dissimilar aspects of the related studies is likely to guide and assist responsive researchers.

References

Akcaoglu, M. (2014). Learning problem-solving through making games at the game design and learning summer program. *Educational Technology Research and Development, 62*(5), 583-600. https://doi.org/10.1007/s11423-014-9347-4

Anastasiadou, S. D., & Karakos, A. S. (2011). The beliefs of electrical and computer engineering students' regarding computer programming. *The International Journal of Technology, Knowledge and Society, 7*(1), 37-51. https://doi.org/10.18848/1832-3669/CGP/v07i01/56170

Aşkar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for Java Programming among engineering students. *Turkish Online Journal of Educational Technology, 8*(1), 26-32.

Bacca, J., Baldiris, S., Fabregat, R., Graf, S., & Kinshuk. (2014). Augmented reality trends in education: A systematic review of research and applications. *Educational Technology & Society, 17*(4), 133-149.

Baki, A., & Özpınar, İ. (2007). Effects of LOGO assisted geometry teaching material on students' academic achievement and students' views on practice. *Cukurova University Faculty of Education Journal, 34*(3), 153 - 163.

Balcı, A. (2016). *Research methods, techniques and principles in social sciences (12th edition).* Ankara: Pegem Publishing.

Baytak, A., & Land, S. M. (2011). An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom. *Educational Technology Research and Development, 59*(6), 765-782. https://doi.org/10.1007/s11423-010-9184-z

Baytak, A., Land, S. M., & Smith, B. K. (2011). Children as educational computer game designers: An exploratory study. *Turkish Online Journal of Educational Technology, 10*(4), 84-92.

Benzer, A. İ., & Erümit, A. K. (2017). The analysis of the graduate theses related to programming instruction. *Journal of Instructional Technologies & Teacher Education, 6*(3), 99-110.

Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E., & Fontecchio, A. (2008). Computer aided instruction as a vehicle for problem solving: Scratch programming environment in the middle years classroom. Paper presented at 2008 Annual Conference & Exposition, Pittsburgh, Pennsylvania. Retrieved from https://peer.asee.org/3826.

Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *Journal of Media Literacy Education, 4*(2), 121-135.

Büyüköztürk, Ş., Kılıç-Çakmak, E., Akgün, Ö. E., Karadeniz, Ş., & Demirel, F. (2017). Scientific research methods. Ankara: Pegem Publishing.

Calao, L. A., Moreno-Leon, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with Scratch. In G. Conole, T. Klobučar, C. Rensing, J. Konert, & E. Lavoué (Eds.), *Design for Teaching and Learning in a Networked World* (pp. 17-27). Toledo: Springer International Publishing. https://doi.org/10.1007/978-3-319-24258-3_2

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education, 109*, 162-175. https://doi.org/10.1016/j.compedu.2017.03.001

Çatlak, Ş., Tekdal, M., & Baz, F. Ç. (2015). The status of teaching programming with Scratch: A document review work. *Journal of Instructional Technologies &Teacher Education, 4*(3), 13-25.

Deveci-Topal, A., Çoban-Budak, E., & Kolburan-Geçer, A. (2017). The effect of algorithm teaching on the problem-solving skills of deaf-hard hearing students. *Program: Electronic Library and Information Systems, 51*(4), 354-372. https://doi.org/10.1108/PROG-05-2017-0038

Dirlikli, E., Aydın, K., & Akgün, L. (2016). Cooperative learning in Turkey: A content analysis of theses. *Educational Sciences: Theory & Practice, 16*(4), 1251-1273.

Ferrer-Mico, T., Prats-Fernàndez, M. À., & Redo-Sanchez, A. (2012). Impact of Scratch programming on students' understanding of their own learning process. *Procedia-Social and Behavioral Sciences, 46*, 1219-1223. https://doi.org/10.1016/j.sbspro.2012.05.278

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education, 63*, 87-97. https://doi.org/10.1016/j.compedu.2012.11.016

Feurzeig, W., Papert, S., Bloom, M., Grant, R., & Solomon, C. (1969). Programming-languages as a conceptual framework for teaching mathematics. Final Report on the First Fifteen Months of the LOGO Project (ERIC Document Reproduction Service No. ED 038034).

Göktaş, Y., Küçük, S., Aydemir, M., Telli, E., Arpacık, Ö., Yıldırım, G., & Reisoğlu, İ. (2012). Educational technology research trends in Turkey: A content analysis of the 2000-2009 decade. *Educational Sciences: Theory & Practice, 12*(1), 191-196.

Gunbatar, M. S., & Karalar, H. (2018). Gender differences in middle school students' attitudes and self-efficacy perceptions towards mBlock programming. *European Journal of Educational Research, 7*(4), 925-933. https://doi.org/10.12973/eu-jer.7.4.925

Gupta, N., Tejovanth, N., & Murthy, P. (2012, January). Learning by creating: Interactive programming for Indian high schools. In 2012 IEEE International Conference on Technology Enhanced Education (ICTEE) (pp. 1-3). https://doi.org/10.1109/ICTEE.2012.6208643

Guzdial, M., & Soloway, E. (2002). Teaching the Nintendo generation to program. *Communications of the ACM, 45*(4), 17-21. https://doi.org/10.1145/505248.505261

Han, B., Bae, Y., & Park, J. (2016). The effect of mathematics achievement variables on Scratch programming activities of elementary school students. *International Journal of Software Engineering and Its Applications, 10*(12), 21-30. https://doi.org/10.14257/ijseia.2016.10.12.03

Hawi, N. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. *Computers & Education, 54*(4), 1127-1136. https://doi.org/10.1016/j.compedu.2009.10.020

Hartl, A. C., DeLay, D., Laursen, B., Denner, J., Werner, L., Campe, S., & Ortiz, E. (2015). Dyadic instruction for middle school students: Liking promotes learning. *Learning and Individual Differences, 44*, 33-39. https://doi.org/10.1016/j.lindif.2015.11.002

Hazır-Bıkmaz, F., Aksoy, E., Tatar, Ö., & Atak-Altınyüzük, C. A. (2013). The content analysis of PhD theses completed in the field of curriculum and instruction (1974-2009). *Education and Science, 38*(168), 288-303.

Heintz, F., Mannila , L., & Farnqvist, T. (2016, October). A review of models for introducing computational thinking, computer science and computing in K-12 education. In *Frontiers in Education Conference (FIE), 2016 IEEE* (pp. 1-9). https://doi.org/10.1109/FIE.2016.7757410

Hongwarittorrn, N., & Krairit, D. (2010, April). Effects of program visualization (jeliot3) on students' performance and attitudes towards java programming. *In the spring 8th International conference on Computing, Communication and Control Technologies* (pp. 6-9).

Hromkovic, J., Kohn, T., Komm, D., & Serafini, G. (2016). Examples of algorithmic thinking in programming education. *Olympiads in Informatics, 10*, 111-124. https://doi.org/10.15388/ioi.2016.08

Johnson, C. (2017). Learning basic programming concepts with game maker. *International Journal of Computer Science Education in Schools, 1*(2), 1-20. https://doi.org/10.21585/ijcses.v1i2.5

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.

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. *Computers in Human Behavior, 52*, 200-210. https://doi.org/10.1016/j.chb.2015.05.047

Kanbul, S., & Uzunboylu, H. (2017). Importance of coding education and robotic applications for achieving 21st-century skills in North Cyprus. *International Journal of Emerging Technologies in Learning (iJET), 12*(1), 130-140. https://doi.org/10.3991/ijet.v12i01.6097

Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education, 73*, 26-39. https://doi.org/10.1016/j.compedu.2013.12.010

Koppelman, H., & van Dijk, B. (2010, June). Teaching abstraction in introductory courses. In *Proceedings of the 15th Annual Conference on Innovation and Technology in Computer Science (ITiCSE'10)* (pp. 174-178). https://doi.org/10.1145/1822090.1822140

Korkmaz, Ö., & Altun, H. (2013). Engineering and CEIT student's attitude towards learning computer programming. *The Journal of Academic Social Science Studies International Journal of Social Science, 6*(2), 1169-1185. https://doi.org/10.9761/jasss_690

Korucu, A., & Atıcı, K. (2018). The determination of metacognitive awareness situations of secondary school students receiving programming education with Alice. *Journal of Learning and Teaching in Digital Age, 3*(1), 3-11.

Kramer, J., & Hazzan, O. (2006, May). The role of abstraction in software engineering. In *Proceedings of the 28th International Conference on Software Engineering (ICSE)* (pp. 1017-1018). https://doi.org/10.1145/1134285.1134481

Kukul, V., & Gökçearslan, Ş. (2014, September). Investigating the problem solving skills of students attended scratch programming course. In *Proceedings of the 8th International Computer & Instructional Technologies Symposium (ICITS)* (pp. 58-63).

Lahtinen, E., Ala-Mutka, K., & Jarvinen, H. M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin, 37*(3), 14-18. https://doi.org/10.1145/1151954.1067453

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics, 33*(1), 159-174. https://doi.org/10.2307/2529310

Lye, S. Y., & Koh, J. H. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51-61. https://doi.org/10.1016/j.chb.2014.09.012

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with Scratch. *Computer Science Education, 23*(3), 239-264. https://doi.org/10.1080/08993408.2013.832022

Milner, S. (1973). The effects of computer programming on performance in mathematics. Paper presented at the Annual meeting of the American Educational Research Association, New Orleans, Louisiana.

Ministry of National Education (MoNE). (2018). *The curriculum of the information technology and software course*. Retrieved from http://mufredat.meb.gov.tr/Dosyalar/2018124103559587-Bilişim Teknolojileri ve Yazılım 5-6. Sınıflar.pdf

Moreno-Leon, J., Robles, G., & Roman-Gonzalez, M. (2016). Code to learn: Where does it belong in the K-12 curriculum? *Journal of Information Technology Education: Research, 15*, 283-303. https://doi.org/10.28945/3521

Navarrete, C. C. (2013). Creative thinking in digital game design and development: A case study. *Computers & Education, 69*, 320-331. https://doi.org/10.1016/j.compedu.2013.07.025

Oluk, A., & Saltan, F. (2015). Effects of using the Scratch program in 6th grade information technologies courses on algorithm development and problem solving skills. *Participatory Educational Research, 2*(5), 10-20. https://doi.org/10.17275/per.15.spi.2.2

Oluk, A., Korkmaz, Ö., & Oluk, H. (2018). Effect of Scratch on 5th graders' algorithm development and computational thinking skills. *Turkish Journal of Computer and Mathematics Education, 9*(1), 54-71. https://doi.org/10.16949/turkbilmat.399588

Özmen, B., & Altun, A. (2014). Undergraduate students' experiences in programming: Difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry, 5*(3), 1-27. https://doi.org/10.17569/tojqi.20328

Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: A case study. International *Journal of Mobile Learning and Organisation, 10*(3), 187-202. https://doi.org/10.1504/IJMLO.2016.077867

Pea, R. D. (1983). *Logo programming and problem solving [Technical Report No. 12.].* Retrieved from https://files.eric.ed.gov/fulltext/ED319371.pdf

Polat, S. (2015). Content analysis of the studies in Turkey on the ability of critical thinking. *Educational Sciences: Theory & Practice, 15*(3), 659-670.

Qian, Y., & Lehman, J. D. (2016). Correlates of success in introductory programming: A study with middle school students. *Journal of Education and Learning, 5*(2), 73-83. https://doi.org/10.5539/jel.v5n2p73

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., . . . Kafai, Y. (2009). *Scratch: Programming for all. Communications of the ACM, 52*(11), 60-67. https://doi.org/10.1145/1592761.1592779

Saez-Lopez, J. M., Roman-Gonzalez, M., & Vazquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education, 97*, 129-141. https://doi.org/10.1016/j.compedu.2016.03.003

Shim, J., Kwon, D., & Lee, W. (2017). The effects of a robot game environment on computer programming education for elementary school students. *IEEE Transactions on Education, 60*(2), 164-172. https://doi.org/10.1109/TE.2016.2622227

Soykan, F., & Kanbul, S. (2018). Analysing K12 students' self-efficacy regarding coding education. TEM *Journal-Technology Education Management Informatics, 7*(1), 182-187.

Şahiner, A., & Kert, S. B. (2016). Examining studies related with the concept of computational thinking between the years of 2006-2015. *European Journal of Science and Technology, 5*(9), 38-43.

Şimşek, A., Özdamar, N., Uysal, Ö., Kobak, K., Berk, C., Kılıçer, T., & Çiğdem, H. (2009). Current trends in educational technology research in Turkey in the new millennium. *Educational Sciences: Theory & Practice, 9*(2), 961-966.

Tocháček, D., Lapeš, J., & Fuglík, V. (2016). Developing technological knowledge and programming skills of secondary schools students through the educational robotics projects. *Procedia-Social and Behavioral Sciences, 217*, 377-381. https://doi.org/10.1016/j.sbspro.2016.02.107

Turhan-Türkkan, B., & Arslan-Namlı, N. (2018). Examination of postgraduate dissertations regarding using computer software in mathematics teaching. *International Online Journal of Educational Sciences, 10*(4), 38-62. https://doi.org/10.15345/iojes.2018.04.003

Ulutaş, F., & Ubuz, B. (2008). Research and trends in mathematics education: 2000 to 2006. *Elementary Education Online, 7*(3), 614-626.

Vatansever, Ö., & Baltacı-Göktalay, Ş. (2018). How does teaching programming through Scratch affect problem-solving skills of 5th and 6th grade middle school students? *International Journal of Eurasia Social Sciences, 9*(3), 1778-1801.

Werner, L., & Denning, J. (2009). Pair programming in middle school: What does it look like? *Journal of Research on Technology in Education, 42*(1), 29-49. https://doi.org/10.1080/15391523.2009.10782540

Wilson, A., & Moffat, D. C. (2010, September). Evaluating Scratch to introduce younger schoolchildren to programming. In Proceedings of the 22nd Annual Psychology of Programming Interest Group.

Yünkül, E., Durak, G., Çankaya, S., & Mısırlı, Z. A. (2017). The effects of Scratch software on students' computational thinking skills. *Necatibey Faculty of Education Electronic Journal of Science and Mathematics Education, 11*(2), 502-517. https://doi.org/10.17522/balikesirnef.373424

Appendix 1 – The examined studies for content analysis

Akcaoglu, M. (2014). Learning problem-solving through making games at the game design and learning summer program. *Educational Technology Research and Development, 62*(5), 583-600. https://doi.org/10.1007/s11423-014-9347-4

Asad, K., Tibi, M., & Raiyn, J. (2016). Primary school pupils' attitudes toward learning programming through visual interactive environments. *World Journal of Education, 6*(5), 20-26. https://doi.org/10.5430/wje.v6n5p20

Atman-Uslu, N., Mumcu, F., & Eğin, F. (2018). The effect of visual programming activities on secondary school students' computational thinking skills. *Journal of Ege Education Technologies, 2*(1), 19-31.

Baki, A., & Özpınar, İ. (2007). Effects of LOGO assisted geometry teaching material on students' academic achievement and students' views on practice. *Cukurova University Faculty of Education Journal, 34*(3), 153 - 163.

Baytak, A., & Land, S. M. (2011). An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom. *Educational Technology Research and Development, 59*(6), 765-782. https://doi.org/10.1007/s11423-010-9184-z

Baytak, A., Land, S. M., & Smith, B. K. (2011). Children as educational computer game designers: An exploratory study. *Turkish Online Journal of Educational Technology, 10*(4), 84-92.

Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E., & Fontecchio, A. (2013). Computer aided instruction as a vehicle for problem solving: Scratch programming environment in the middle years classroom.

Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *Journal of Media Literacy Education, 4*(2), 121-135.

Çakır, N. A., Gass, A., Foster, A., & Lee, F. J. (2017). Development of a game-design workshop to promote young girls' interest towards computing through identity exploration. *Computers & Education, 108*, 115-130. https://doi.org/10.1016/j.compedu.2017.02.002

Çankaya, S., Durak, G., & Yünkül, E. (2017). Education on programming with robots: Examining students' experiences and views. *Turkish Online Journal of Qualitative Inquiry, 8*(4), 428-445. https://doi.org/10.17569/tojqi.343218

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education, 109*, 162-175. https://doi.org/10.1016/j.compedu.2017.03.001

DeLay, D., Hartl, A. C., Laursen, B., Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Learning from friends: Measuring influence in a dyadic computer instructional setting. *International Journal of Research & Method in Education, 37*(2), 190-205. https://doi.org/10.1080/1743727X.2013.784961

Denner, J., & Werner, L. (2007). Computer programming in middle school: How pairs respond to challenges. *Journal of Educational Computing Research, 37*(2), 131-150. https://doi.org/10.2190/12T6-41L2-6765-G3T2

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education, 58*(1), 240-249. https://doi.org/10.1016/j.compedu.2011.08.006

Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education, 46*(3), 277-296. https://doi.org/10.1080/15391523.2014.888272

Deveci-Topal, A., Çoban-Budak, E., & Kolburan-Geçer, A. (2017). The effect of algorithm teaching on the problem-solving skills of deaf-hard hearing students. *Program: Electronic Library and Information Systems, 51*(4), 354-372. https://doi.org/10.1108/PROG-05-2017-0038

Ferrer-Mico, T., Prats-Fernàndez, M. À., & Redo-Sanchez, A. (2012). Impact of Scratch programming on students' understanding of their own learning process. *Procedia-Social and Behavioral Sciences, 46*, 1219-1223. https://doi.org/10.1016/j.sbspro.2012.05.278

Fokides, E. (2017). Students learning to program by developing games: Results of a year-long project in primary school settings. *Journal of Information Technology Education: Research, 16*, 475-505. https://doi.org/10.28945/3893

Garcia, P. G. F., & De la Rosa, F. (2016). RoBlock-web app for programming learning. International *Journal of Emerging Technologies in Learning (iJET), 11*(12), 45-53. https://doi.org/10.3991/ijet.v11i12.6004

Gunbatar, M. S., & Karalar, H. (2018). Gender differences in middle school students' attitudes and self-efficacy perceptions towards mBlock programming. *European Journal of Educational Research, 7*(4), 925-933. https://doi.org/10.12973/eu-jer.7.4.925

Han, B., Bae, Y., & Park, J. (2016). The effect of mathematics achievement variables on Scratch programming activities of elementary school students. *International Journal of Software Engineering and Its Applications, 10*(12), 21-30. https://doi.org/10.14257/ijseia.2016.10.12.03

Hartl, A. C., DeLay, D., Laursen, B., Denner, J., Werner, L., Campe, S., & Ortiz, E. (2015). Dyadic instruction for middle school students: Liking promotes learning. *Learning and Individual Differences, 44*, 33-39. https://doi.org/10.1016/j.lindif.2015.11.002

Howard, A. M., Park, C. H., & Remy, S. (2012). Using haptic and auditory interaction tools to engage students with visual impairments in robot programming activities. *IEEE Transactions on Learning Technologies, 5*(1), 87-95. https://doi.org/10.1109/TLT.2011.28

Jenson, J., & Droumeva, M. (2016). Exploring Media Literacy and Computational Thinking: A Game Maker Curriculum Study. *Electronic Journal of e-Learning, 14*(2), 111-121.

Johnson, C. (2017). Learning basic programming concepts with game maker. *International Journal of Computer Science Education in Schools, 1*(2), 1-20. https://doi.org/10.21585/ijcses.v1i2.5

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.

Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education, 73*, 26-39. https://doi.org/10.1016/j.compedu.2013.12.010

Keçeci, G., Alan, B., & Zengin, F. K. (2016). Educational computer games assisted learning coding attitude scale: Validity and reliability study. *Education Sciences, 11*(4), 184-194. https://doi.org/10.12739/NWSA.2016.11.3.1C0661

Kim, S., Chung, K., & Yu, H. (2013). Enhancing digital fluency through a training program for creative problem solving using computer programming. *The Journal of Creative Behavior, 47*(3), 171-199. https://doi.org/10.1002/jocb.30

Kong, S. C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers & Education, 127*, 178-189. https://doi.org/10.1016/j.compedu.2018.08.026

Korucu, A., & Atıcı, K. (2018). The determination of metacognitive awareness situations of secondary school students receiving programming education with Alice. *Journal of Learning and Teaching in Digital Age, 3*(1), 3-11.

Kukul, V., Gökçearslan, Ş., & Günbatar, M. S. (2017). Computer programming self-efficacy scale (CPSES) for secondary school students: Development, validation and reliability. *Educational Technology Theory and Practice, 7*(1), 158-179. https://doi.org/10.17943/etku.288493

Lavonen, J. M., Meisalo, V. P., & Lattu, M. (2001). Problem solving with an icon oriented programming tool: A case study in technology education. *Journal of Technology Education, 12*(2), 21-34. https://doi.org/10.21061/jte.v12i2.a.2

Lavonen, J. M., Meisalo, V. P., Lattu, M., & Sutinen, E. (2003). Concretising the programming task: a case study in a secondary school. *Computers & Education, 40*(2), 115-135. https://doi.org/10.1016/S0360-1315(02)00101-X

Lewis, C. M. (2011). Is pair programming more effective than other forms of collaboration for young students? *Computer Science Education, 21*(2), 105-134. https://doi.org/10.1080/08993408.2011.579805

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with Scratch. *Computer Science Education, 23*(3), 239-264. https://doi.org/10.1080/08993408.2013.832022

Mladenović, M., Rosić, M., & Mladenović, S. (2016). Comparing elementary students' programming success based on programming environment. *I. J. Modern Education and Computer Science, 8*, 1-10. https://doi.org/10.5815/ijmecs.2016.08.01

Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED- Revista de Educación a Distancia, 46*, 1-23.

Moreno-Leon, J., Robles, G., & Roman-Gonzalez, M. (2016). Code to learn: Where does it belong in the K-12 curriculum? *Journal of Information Technology Education: Research, 15*, 283-303. https://doi.org/10.28945/3521

Mouza, C., Marzocchi, A., Pan, Y. C., & Pollock, L. (2016). Development, implementation, and outcomes of an equitable computer science after-school program: Findings from middle-school students. *Journal of Research on Technology in Education, 48*(2), 84-104. https://doi.org/10.1080/15391523.2016.1146561

Navarrete, C. C. (2013). Creative thinking in digital game design and development: A case study. *Computers & Education, 69*, 320-331. https://doi.org/10.1016/j.compedu.2013.07.025

Oluk, A., & Korkmaz, Ö. (2016). Comparing students' Scratch skills with their computational thinking skills in terms of different variables. *I. J. Modern Education and Computer Science, 8*(11), 1-7. https://doi.org/10.5815/ijmecs.2016.11.01

Oluk, A., & Saltan, F. (2015). Effects of using the Scratch program in 6th grade information technologies courses on algorithm development and problem solving skills. *Participatory Educational Research, 2*(5), 10-20. https://doi.org/10.17275/per.15.spi.2.2

Oluk, A., Korkmaz, Ö., & Oluk, H. (2018). Effect of Scratch on 5th graders' algorithm development and computational thinking skills. Turki*sh Journal of Computer and Mathematics Education, 9*(1), 54-71. https://doi.org/10.16949/turkbilmat.399588

Qian, Y., & Lehman, J. D. (2016). Correlates of success in ıntroductory programming: A study with middle school students. *Journal of Education and Learning, 5*(2), 73-83. https://doi.org/10.5539/jel.v5n2p73

Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018). Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal of Child-Computer Interaction, 18*, 47-58. https://doi.org/10.1016/j.ijcci.2018.06.004

Saez-Lopez, J. M., Roman-Gonzalez, M., & Vazquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. Computers & Education, 97, 129-141. https://doi.org/10.1016/j.compedu.2016.03.003

Saito, D., Washizaki, H., & Fukazawa, Y. (2017). Comparison of text-based and visual-based programming input methods for first-time learners. *Journal of Information Technology Education-Research, 16*(1), 209-226. https://doi.org/10.28945/3775

Salahli, M. A., Yildirim, E., Gasimzadeh, T., Alasgarova, F., & Guliyev, A. (2017). One mobile application for the development of programming skills of secondary school students. *Procedia Computer Science, 120*, 502-508. https://doi.org/10.1016/j.procs.2017.11.271

Shim, J., Kwon, D., & Lee, W. (2017). The effects of a robot game environment on computer programming education for elementary school students. *IEEE Transactions on Education, 60*(2), 164-172. https://doi.org/10.1109/TE.2016.2622227

Soykan, F., & Kanbul, S. (2018). Analysing K12 students' self-efficacy regarding coding education. *TEM Journal-Technology Education Management Informatics, 7*(1), 182-187.

Su, A. Y. S., Huang, C. S. J., Yang, S. J. H., Ding, T. J., & Hsieh, Y. Z. (2015). Effects of annotations and homework on learning achievement: An empirical study of Scratch programming pedagogy. *Journal of Educational Technology & Society, 18*(4), 331-343.

Tekerek, M., & Altan, T. (2014). The effect of Scratch environment on student's achievement in teaching algorithm. *World Journal on Educational Technology, 6*(2), 132-138.

Tocháček, D., Lapeš, J., & Fuglík, V. (2016). Developing technological knowledge and programming skills of secondary schools students through the educational robotics projects. *Procedia-Social and Behavioral Sciences, 217*, 377-381. https://doi.org/10.1016/j.sbspro.2016.02.107

Vatansever, Ö., & Baltacı-Göktalay, Ş. (2018). How does teaching programming through Scratch affect problem-solving skills of 5th and 6th grade middle school students? *International Journal of Eurasia Social Sciences, 9*(3), 1778-1801.

Werner, L., & Denning, J. (2009). Pair programming in middle school: What does it look like? *Journal of Research on Technology in Education, 42*(1), 29-49. https://doi.org/10.1080/15391523.2009.10782540

Witherspoon, E. B., Schunn, C. D., Higashi, R. M., & Shoop, R. (2018). Attending to structural programming features predicts differences in learning and motivation. *Journal of Computer Assisted Learning, 34*(2), 115-128. https://doi.org/10.1111/jcal.12219

Yıldız-Durak, H. (2018). Digital story design activities used for teaching programming effect on learning of programming concepts, programming self-efficacy, and participation and analysis of student experiences. *Journal of Computer Assisted Learning, 34*(6), 740-752. https://doi.org/10.1111/jcal.12281

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. https://doi.org/10.1111/jcal.12302

Yıldız-Durak, H. (2018). The effects of using different tools in programming teaching of secondary school students on engagement, computational thinking and reflective thinking skills for problem solving. *Technology, Knowledge and Learning*, 1-17. https://doi.org/10.1007/s10758-018-9391-y

Yüksel, S., & Gündoğdu, K. (2018). The effect of jigsaw technique in teaching Scratch program on attitude, achievement, and retention of knowledge. *Ege Journal of Education, 19*(1), 245-261. https://doi.org/10.12984/egeefd.340362

Yünkül, E., Durak, G., Çankaya, S., & Mısırlı, Z. A. (2017). The effects of Scratch software on students' computational thinking skills. *Necatibey Faculty of Education Electronic Journal of Science and Mathematics Education*, 11(2), 502-517. https://doi.org/10.17522/balikesirnef.373424

Ali Kürşat Erümit
Trabzon University
Söğütlü Mah. Adnan Kahveci Boulevard, 61335
Akçaabat-Istanbul, Trabzon, Turkey
kursaterumit@gmail.com

Ali İhsan Benzer
Hatay Mustafa Kemal University
Tayfur Sokmen Campus 31040 Antakya, Hatay, Turkey
aibenzer@gmail.com

Güven Şahin
Rize Recep Tayyip Erdoğan University
Fener Mahallesi 53100 Center, RizE, Turkey
guvennsahinn@gmail.com

Okvir za istraživanje nastave programiranja u srednjoškolskom obrazovanju

Sažetak

Ovo istraživanje naširoko razmatra literaturu o nastavi programiranja u srednjim školama pomoću metode kvalitativne analize sadržaja i pretpostavlja se da će istraživačima ponuditi okvir za dizajniranje istraživanja koji će ih voditi u procesima planiranja i projektiranja njihovih istraživanja o nastavi programiranja u srednjoj školi. Za potrebe pristupa relevantnoj literaturi, pretražili smo baze podataka pomoću ključnih riječi „računalo", „programiranje" i „srednja škola" uzetih zajedno, ograničavajući rezultate na članke objavljene nakon 2000. godine. Kao rezultat toga, primijetili smo porast trendova u istraživanjima o nastavi programiranja na srednjoškolskoj razini, uzimajući u obzir godine, od kojih je većina istraživanja empirijske naravi. Osim toga, postojeća su istraživanja uglavnom provedene s učenicima šestih razreda koji pretežno koriste alate za prikupljanje podataka u obliku upitnika/ljestvica i testova postignuća. Što se tiče alata za programiranje, Scratch je najčešće korišten. Iako postoji dosta članaka koji istražuju kontekst nastave programiranja, otkriveno je da neka istraživanja navode programiranje kao pomoćno sredstvo u učenju matematike, prirodnih znanosti, jezika, vještina pisanja i društvenih znanosti. Zaključno, valja napomenuti da se očekuje da će sadašnje istraživanje otvoriti put budućim istraživanjima, rasvjetljavajući cjelokupnu situaciju nastave programiranja.

Ključne riječi: *analiza sadržaja; kodiranje; poučavanje programiranja*; *računalno programiranje; srednja škola.*

Uvod

U radnom životu, povećana potražnja za informatičkim stručnjacima podigla je interes za računalnu znanost, što je rezultiralo i organiziranjem informatičkih tečajeva na različitim razinama obrazovanja (Qian i Lehman, 2016). Brojni istraživači i studije naglašavaju potencijalne značajne prednosti uključivanja nastave programiranja u osnovne i srednje škole (Saez-Lopez i sur., 2016). Učenje programiranja potiče razvoj nekoliko vještina koje nazivamo vještinama za 21. stoljeće, što uključuje rješavanje problema (Calao i sur., 2015; Fessakis i sur., 2013), kreativno razmišljanje (Gupta i sur., 2012; Navarrete, 2013), algoritamsko razmišljanje (Hromkovic i sur., 2016), refleksivno razmišljanje (Kalelioğlu, 2015), kritičko razmišljanje i vještine računalnoga razmišljanja. Ovi blagodatni učinci programiranja nisu ograničeni na gore navedene vještine jer pridonose i učenju matematike, prirodnih znanosti, društvenih znanosti i jezika (Baki i Özpınar, 2007; Baytak i sur., 2011; Brown i sur., 2008; Burke, 2012; Ferrer-Mico i sur., 2012; Ke, 2014; Moreno-Leon i sur., 2016; Navarrete, 2013; Saez-Lopez i sur., 2016). Iz tih se razloga vidi da je došlo do intenzivnih pokušaja poučavanja programiranja u školama, osobito posljednjih godina (Moreno-Leon i sur., 2016). Većina zemalja već je uvrstila nastavu programiranja (za koju se očekuje da će donijeti značajne prednosti) u svoje nastavne programe te su je uključile u različite razine školovanja počevši od osnovne škole (Han i sur., 2016; Heintz i sur., 2016; MoNE, 2018).

Programiranje je proces prijenosa problema u računalno okruženje pomoću softvera nakon njihovoga modeliranja (Benzer i Erümit, 2017). Lye i Koh (2014) smatraju da programiranje zahtijeva vještine apstrakcije i analize, a ne samo kodiranje. Činjenica da programiranje kao proces stvaranja rješenja za probleme uključuje niz mentalnih vještina (Benzer i Erümit, 2017) izaziva učenike da se suoče s teškoćama u nastavi programiranja i percipiraju programiranje preteškim (Aşkar i Davenport, 2009; Guzdial i Soloway, 2002; Lahtinen i sur., 2005). Danas se grafički alati vizualnoga programiranja za prevladavanje poteškoća, posebice u srednjoškolskoj dobi, kad učenici napreduju od faze konkretnih operacija do faze apstraktnih operacija. U ovu svrhu služe softverski proizvodi kao što su Alice, Scratch, Microsoft Small Basic, Toontalk, Stagecast Creator i Code GameLab. Ovi su alati posebno poželjni za poboljšanje algoritamskih vještina razmišljanja učenika srednjih škola i olakšavanje nastave programiranja.

S pojavom softvera koji olakšava poučavanje programiranja, lako je primijetiti da su akademska istraživanja u ovom polju široko rasprostranjena među različitim dobnim skupinama, školskim predmetima i ishodima učenja. U zadnje vrijeme povećan je interes i znatiželja istraživača za nastavu programiranja koja ima važnu ulogu u poučavanju vještina 21. stoljeća. Istodobno, broj studija usmjerenih na obrazovanje o programiranju znatno je porastao. Ovo je rezultiralo pojavom određenih trendova u dizajnima istraživanja tih studija.

Cilj istraživanja nastave programiranja bio je uglavnom poboljšanje kognitivnih i/ili afektivnih karakteristika sudionika. Istraživanjem se proučilo učinke različitih pedagoških pristupa i nastavnih okruženja kako bi se postigao taj cilj. Shim i sur. (2017) i Tocháček i sur. (2016) naglasili su da robotska programska okruženja pozitivno poboljšavaju stav učenika prema programiranju i pomažu im da nauče programirati. Meerbaum-Salant i sur. (2013) i Oluk i sur. (2018) utvrdili su da je okoliš Scratch softvera učinkovit u poučavanju koncepata programiranja. Vatansever i Baltacı-Göktalay (2018) zaključili su da je programiranje u Scratchu pozitivno utjecalo na razvoj vještina rješavanja problema kod sudionika. Werner i Denning (2009) i Hartl i sur. (2015) otkrili su da programiranje u parovima pozitivno utječe na motivaciju učenika i vještine rješavanja problema. Johnson (2017) i Baytak i Land (2011) izjavili su da je nastava programiranja temeljena na igrama učinkovita u objašnjavanju koncepata programiranja.

Povećanje akademskih studija o poučavanju programiranja dovelo je do zajedničkih točaka i razlika u planiranju dizajna istraživanja. Ipak, razlike i dalje postoje u ciljevima istraživanja, korištenim alatima za programiranje, sadržaju, alatima za prikupljanje podataka i trajanju istraživanja vezanih za nastavu programiranja.

Cilj je ovoga istraživanja analizirati članke koji se odnose na nastavu programiranja u srednjoj školi pomoću metode analize sadržaja kao i ponuditi okvir za dizajniranje istraživanja nastave programiranja u srednjoj školi. Krajnji je cilj istraživanja opisati cjelokupni dojam nastave programiranja i sukladno tome usmjeravati buduća istraživanja. S obzirom na navedeno formulirana su sljedeća istraživačka pitanja:

1. Kako izgleda distribucija objavljenih članaka o učenicima srednjih škola prema određenim kriterijima (godini objave, metodi istraživanja, razredu sudionika, alatima za prikupljanje podataka, alatima za programiranje, sadržaju i trajanju)?
2. Što je zajedničko objavljenim člancima o učenicima srednjih škola s obzirom na osobine dizajna istraživanja?

Metodologija

U ovom istraživanju pregledani su objavljeni članci o poučavanju i korištenju programiranja za učenike srednjih škola. Za analizu podataka korištena je metoda analize sadržaja, odnosno, pregled dokumenata, kako je to zahtijevala priroda istraživanja. Analiza sadržaja je metoda istraživanja koja se temelji na sastavljanju, klasifikaciji, usporedbi i teorijskom tumačenju dokumenata. Ova je metoda odabrana kako bi se povezali relevantni podatci te kako bismo ih predstavili na čitljiv i jasan način čitateljima ovoga istraživanja. Članci su pregledani kako bismo saznali godinu objave, metodu, razinu uzorka, alat za prikupljanje podataka, korišteni alat za programiranje, trajanje, sadržaj, svrhu i rezultate. Podatci istraživanja prikupljeni su pomoću recenzentskoga obrasca, a zatim analizirani pomoću deskriptivnih statističkih metoda.

Populacija i uzorak

Uzorak istraživanja sastoji se od članaka o nastavi i korištenju programiranja za učenike srednjih škola objavljenih u referiranim časopisima.

Prikupljanje i analiza podataka

Tijekom pregleda literature, ključne riječi „računalo", „programiranje" i „srednja škola" istodobno su korištene u bazama podataka Google Scholar, ERIC i Web of Science za članke objavljene nakon 2000. godine. Pretraživanje je ograničeno na nastavne skupine koje se sastoje od srednjoškolaca od 5. do 8. razreda ili učenika u dobi od 10 do 15 godina, budući da pojam „srednja škola" obuhvaća različite godine u različitim zemljama.

Glavni dijelovi članaka kao što su naslov, sažetak i ključne riječi korišteni su kao kriteriji za uključivanje u ovu studiju. U slučajevima kada se odlučujuće informacije nisu mogle dobiti iz tih dijelova, razmotrili su se i drugi dijelovi članaka. Preliminarnim pregledom za daljnju analizu dobivena su ukupno 62 članka koji zadovoljavaju ove kriterije te su uključeni u istraživanje. Članci pregledani u našem istraživanju analizirani su i kodirani prema kriterijima godine, sadržaja, cilju i rezultatima putem recenzentskoga obrasca. Odabir i kodiranje članaka provodila su dvojica istraživača. Statistika Cohenova Kappa-koeficijenta korištena je kako bi se utvrdilo je li pouzdanost između dva kodera prihvatljiva. Kappa-koeficijent izračunat je kao 0,92. Prema ovom rezultatu kompatibilnost između kodera je idealna (Landis i Koch, 1977). Rezultati analize prikazani su u obliku tablica i grafikona.

Rezultati

Ova je studija osmišljena kako bi se identificirala distribucija istraživanja nastave programiranja, kako bi se utvrdila sličnost i na kraju razvio okvir koji će se koristiti u razvoju istraživanja nastave programiranja na srednjoškolskoj razini. Kao rezultat analize članaka prema godini objave, metodi istraživanja, dobi skupine uzorka, alatu za prikupljanje podataka, alatu za programiranje, sadržaju, ciljevima i trajanju, rezultati distribucije prikazani su u grafikonima. Pregledani članci prikazani su u dodatku 1. U svjetlu ovih nalaza identificirani su opći aspekti istraživanja vezanih uz dizajn i formiran je okvir dizajna istraživanja. Ovaj je okvir prikazan na slici 11.

Analiza članaka prema godini objave

Distribucija istraživanja usmjerenih na nastavu programiranja kod srednjoškolaca analizirana je i distribuirana prema godini objave, a dobiveni rezultati prikazani su na slici 1 u nastavku.

Slika 1.

Prema slici 1, broj istraživanja povezanih s nastavom programiranja na srednjoškolskoj razini povećao se tijekom godina, a najveći broj studija objavljen je 2018. godine.

Analiza članaka prema metodama istraživanja

Distribucija članaka prema metodama istraživanja prikazana je na slici 2 u nastavku.

Slika 2.

Slika 2 pokazuje da se većina razmotrenih članaka sastojala od eksperimentalnih studija. Prema našoj analizi, od ukupno 62 članka 41 članak (66 %) uključuje eksperimentalnu studiju, 14 studija slučaja (22 %), 4 korelacijske studije (6 %) i 3 opisne studije (4 %).

Analiza članaka prema ispitanicima

U nastavku se prikazuje distribucija članaka prema dobi nastavne skupine na slici 3.

Slika 3.

Slika 3 pokazuje da je u srednjoj školi 6. razred najčešće preferirani razred za istraživanje nastave programiranja (36 %), nakon kojega slijedi 5. razred (20 %) i 7. razred (13 %). Međutim, neka su istraživanja uključila i mješovite razrede koji obuhvaćaju više od jednoga specifičnog razreda. Slično tome, mnoga su istraživanja ukazala samo na dobni raspon, bez navođenja razreda (14 %).

Analiza članaka prema alatima za prikupljanje podataka

Distribucija istraživanja prema alatima za prikupljanje podataka prikazana je na slici 4.

Slika 4. Distribucija istraživanja prema korištenim alatima za prikupljanje podataka

Kao što se vidi na slici 4, utvrđeno je da su najčešće korišteni alati za prikupljanje podataka u istraživanjima koja su proučavala nastavu programiranja u srednjoj školi bili upitnici/ljestvice (31 %), nakon čega su uslijedili testovi postignuća (17 %) i intervjui (15 %). Osim toga, većina istraživanja koristila je više od jednoga alata za prikupljanje podataka.

Analiza članaka prema alatima za programiranje

Slika 5 u nastavku prikazuje distribuciju istraživanja prema odabranim alatima za programiranje.

Slika 5.

Slika 5 pokazuje da je Scratch najpopularniji alat za programiranje u istraživanjima nastave programiranja na srednjoškolskoj razini (45 %). Nakon Scratcha slijede ostali alati kao što su Alice (7 %) i Logo (6 %). Osim ovih, neka su istraživanja koristila više od jednoga alata za programiranje. U gornjem kružnom grafikonu, sekcija „ostalo" (18,18 %) sadrži alate koji su se koristili u jednoj specifičnoj studiji i samo jednom.

Analiza članaka prema sadržaju

Slika 6 pokazuje distribuciju istraživanja prema sadržaju.

Slika 6.

Slika 6 pokazuje da je velik dio istraživanja u našem uzorku povezan s nastavom programiranja. Međutim, neka su istraživanja obratila pažnju na više od jednoga specifičnog sadržaja.

Analiza članaka prema trajanju eksperimenta

Distribucija ovdje pregledanih istraživanja prema trajanju eksperimenta prikazana je na slici 7.

Slika 7.

Kao što se vidi na slici 7, većina ovdje razmatranih istraživanja provedena je u razdoblju od jednog do tri mjeseca. Osim toga, utvrđeno je da je trajanje istraživanja koja su provedena kako bi se razvile i kognitivne i afektivne osobine bilo jednako dugo, 1 – 3 mjeseca, kao i istraživanja provedenih za razvoj samo kognitivnih osobina.

Analiza članaka prema ciljevima

Utvrđeno je da je opći cilj istraživanja koja se ovdje razmatraju proučavanje ili razvoj kognitivnih ili afektivnih domena sudionika. Ciljevi ovih istraživanja podijeljeni su u dvije skupine: kognitivna i afektivna istraživanja, s obzirom na ciljeve i alate za prikupljanje podataka koji su uporabljeni u relevantnim istraživanjima. Rezultati i njihova distribucija prikazani su na slici 8.

Slika 8.

Na slici 8 vidljivo je da 34 od 62 istraživanja ima za cilj analizirati ili poboljšati kognitivne i afektivne osobine sudionika. Slijede 22 istraživanja (35 %) usmjerena na proučavanje ili razvoj samo kognitivnih osobina i 6 primjera (9 %) usmjerenih na proučavanje ili razvoj samo afektivnih osobina.

Naime, analiza istraživanja koja raspravljaju o kognitivnoj domeni pokazala je da se 40 od 56 studija bavi akademskim postignućima, a 21 kognitivnim vještinama. S druge strane, 35 od 40 studija koje su istraživale afektivnu domenu usredotočile su se na stavove (interes, angažman i mišljenje učenika), 4 na samoučinkovitost i 2 na samopouzdanje.

Iz perspektive alata za prikupljanje podataka koji se koriste u kognitivnim istraživanjima, utvrđeno je da su testovi postignuća i analiza artefakata najčešći način mjerenja akademskoga uspjeha. Što se tiče mjerenja kognitivnih vještina (rješavanje problema, računalno razmišljanje), kao najčešće korišteni alati mogu se navesti upitnici/ljestvice, testovi vještina, intervjui, promatranja i audiozapisi. Kod istraživanja koja su istraživala afektivne osobine, upitnici/ljestvice, intervjui i zapažanja uglavnom su korišteni za analizu varijabli poput stavova, odnosno interesa.

Analiza članaka prema rezultatima

Rezultati 51 istraživanja povezanih s različitim učincima nastave programiranja procijenjeni su u smislu kognitivnih i afektivnih ciljeva. Distribucija rezultata kognitivnih i afektivnih intervencijskih studija prikazana je na slici 9, a rezultati istraživanja samo kognitivne domene prikazani su na slici 10. Na kraju, poseban grafikon nije naveden za afektivne studije samo zbog nedovoljnoga broja takvih istraživanja (2).

Slika 9.

Slika 10.

Kao što se vidi na slikama 9 i 10, provedeno je 31 intervencijsko istraživanje s ciljem razvoja kognitivnih i afektivnih osobina sudionika. 26 ih je dobilo pozitivne rezultate u kognitivnim i afektivnim osobinama učenika, dok 4 istraživanja nije uspjelo postići nikakve razlike u kognitivnim osobinama učenika, a pozitivno je utjecalo na njihove afektivne osobine. U preostalom 1 istraživanju intervencija nije dokazala nikakve razlike u afektivnim osobinama učenika, a istodobno je pozitivno djelovalo na njihove kognitivne osobine. Od svih istraživanja, 18 ih je bilo usmjereno na poboljšanje samo kognitivnih sposobnosti sudionika. Kao rezultat toga, 16 istraživanja pokazalo je pozitivnu promjenu u kognitivnim osobinama sudionika istraživanja. Preostala 2 istraživanja završila su bez ikakvoga kognitivnog poboljšanja. Što se tiče druge domene, dva su istraživanja bila usmjerena na razvoj samo afektivnih osobina sudionika, a oba su dala očekivane rezultate.

Rasprava i zaključci

U ovome istraživanju, članci koji se odnose na nastavu programiranja u srednjoj školi ispitani su metodom analize sadržaja prema godini objave, metodi istraživanja, razredu uzorka, alatima za prikupljanje podataka, alatima za programiranje, sadržaju, trajanju eksperimenta, ciljevima istraživanja i rezultatima. Kao rezultat toga, stvoren je okvir istraživačkoga nacrta za provođenje sličnih istraživanja.

Ako uzmemo u obzir istraživanja o nastavi programiranja za učenike srednjih škola tijekom godina, postaje jasno da je, osobito posljednjih godina, broj takvih istraživanja znatno porastao, s najvećim brojem zabilježenim 2018. godine. Čini se da je ovaj rezultat u skladu s nalazima Benzer i Erümit (2017) koji razmatraju poslijediplomske disertacije koje istražuju poučavanje programiranja u turskom kontekstu, Şahiner i Kert (2016) koji proučavaju istraživanja vezana uz računalno razmišljanje, kao i Çatlak i sur. (2015) koji istražuju primjere korištenja Scratcha. Istraživanja koja istražuju utjecaj programiranja na kognitivne procese potječu iz 1960-ih, kad je razvijen programski alat LOGO (Baki i Özpınar, 2007; Feurzeig i sur., 1969; Milner, 1973; Pea, 1983). Što se tiče trendova u akademskim istraživanjima, čini se da je ogroman porast broja istraživanja nastave programiranja posljednjih godina povezan s činjenicom da je važnost i potreba za poučavanjem programiranja uvažena kao bitna za razvoj vještina 21. stoljeća (Gunbatar i Karalar, 2018; Han i sur., 2016; Shim i sur., 2017; Soykan i Kanbul, 2018), većina zemalja dodaje predmete vezane uz programiranje u svoje osnovnoškolsko i srednjoškolsko obrazovanje (Han i sur., 2016; Heintz i sur., 2016; MoNE, 2018; Moreno-Leon i sur., 2016), a programski se alati također posebno dizajniraju za srednjoškolsku nastavu programiranja (Kalelioğlu i Gülbahar, 2014; Korucu i Atıcı, 2018; Lye i Koh, 2014; Resnick i sur., 2009).

Većina pregledanih članaka sastoji se od eksperimentalnih istraživanja. Literatura naglašava da nastava programiranja ima važan potencijal u razvoju vještina za 21. stoljeće (Gunbatar i Karalar, 2018; Han i sur., 2016; Kanbul i Uzunboylu, 2017; Papadakis i sur., 2016; Shim i sur., 2017; Soykan i Kanbul, 2018). Ovo je dovelo do povećanja broja eksperimentalnih studija koje istražuju utjecaj nastave programiranja na razvoj stečenih vještina 21. stoljeća kod učenika.

Pregled uzoraka pokazuje da je najpoželjniji razred za istraživanje 6. razred, a mnoga su istraživanja provedena s učenicima iz različitih razreda. Čini se da je ovaj nalaz u skladu s Benzer i Erümit (2017), koji također ističu da je 6. razred najčešće preferirani razred u istraživanjima o nastavi programiranja u srednjim školama. Ovaj razred pokriva određeno razdoblje u kojem djeca prelaze iz faze konkretnih operacija u fazu apstraktnih operacija (Moreno-Leon i sur., 2016). Dobre vještine apstraktnoga razmišljanja smatraju se ključnim čimbenikom uspjeha u učenju programiranja (Koppelman i van Dijk, 2010; Kramer i Hazzan, 2006). Čini se da su ti razlozi istraživačima bili ključni za odabir 6. razreda kao uzorka.

Pod drugim kriterijem za uključivanje u naše istraživanje, kao najučestaliji alati za prikupljanje podataka navedeni su upitnici/ljestvice. Ostali najčešći alati bili su u obliku testova postignuća i intervjua. Ovaj se rezultat čini kompatibilnim s rezultatima nekoliko prethodnih studija, uključujući Şahiner i Curt (2016) koji proučavaju istraživanja računalnoga razmišljanja; Bacca i sur. (2014) koji razmatraju istraživanja o proširenoj stvarnosti u nastavi; Göktaş i sur. (2012) koji istražuju radove povezane s obrazovnim tehnologijama u Turskoj i Şimşek i sur. (2009) koji pregledavaju diplomske radove o obrazovnim tehnologijama u Turskoj. Smatra se da su upitnici/ljestvice najpoželjniji i najčešći zbog praktičnosti jednostavnoga i brzoga prikupljanja podataka iz većih skupina (Büyüköztürk i sur., 2017). Osim toga, utvrđeno je da većina istraživanja koristi više od jednog alata za prikupljanje podataka. U slučajevima gdje je potrebno prikupiti kvantitativne i kvalitativne podatke u istraživanjima, moguće je da će biti potrebno rabiti više od jednoga alata za prikupljanje podataka (Balcı, 2016).

I ovdje pregledana istraživanja upućuju na to da je Scratch najčešće korišten alat za programiranje. Ipak, u nekim se istraživanjima rabilo više od jednoga alata za programiranje. Rezultat je u skladu s rezultatima istraživanja Benzera i Erumita (2017). To se može objasniti činjenicom da tekstualna programska okruženja stvaraju mnoge poteškoće za učenike (Özmen i Altun, 2014); dok se u vizualnim programskim okruženjima suočavaju s manje problema, što rezultira lakšim učenjem i razumijevanjem programiranja (Wilson i Moffat, 2010). Većina istraživanja izvijestila je da Scratch kao vizualni alat za programiranje kod učenika razvija kognitivne i afektivne aspekte učenja programiranja (Brown i sur., 2008; Burke, 2012; Deveci-Topal i sur., 2017; Ferrer-Mico i sur., 2012; Meerbaum-Salant i sur., 2013; Oluk i sur., 2018). Ipak, pitanje kako različiti programski alati utječu na kognitivne i afektivne osobine učenika ostaje poticajna tema za daljnja istraživanja.

Našim smo pregledom uočili da je značajan dio istraživanja povezan s nastavom programiranja. Osim toga, postoje studije koje istražuju doprinos računalnoga programiranja u nastavi jezika i društvenih znanosti, a osobito u nastavi matematike. Čini se da je ovaj nalaz u skladu s istraživanjima Çatlak i sur. (2015) i Şahiner i Kert (2016). Programiranje ima potencijal da pomogne u nastavi ne samo računalnih znanosti, već i matematike (Baki i Özpınar, 2007; Brown i sur., 2008; Ferrer-Mico i sur., 2012; Ke, 2014), prirode i društva (Baytak i sur., 2011; Navarrete, 2013; Saez-Lopez i sur., 2016), kao i jezika (Burke, 2012; Moreno-Leon i sur., 2016). Budući da je većina istraživanja provedena na satima informatike, ne iznenađuje nas da je sadržaj tih sati povezan s nastavom programiranja.

Što se tiče trajanja istraživanja, može se reći da istraživanja nastave programiranja na srednjoj razini uglavnom traju od jednog do tri mjeseca. Ovaj je rezultat sličan rezultatima postignutim skupom prethodnih istraživanja kao što su Benzer i Erümit (2017); Hazır-Bıkmaz i sur. (2013) koji su istraživali istraživanje disertacije o nastavnim programima i poučavanju; Polat (2015) koji je analizirao istraživanja o vještinama kritičkoga razmišljanja, kao i Dirlikli i sur. (2016) koji su istraživali diplomske radove o kolaborativnom učenju u Turskoj. To se može objasniti činjenicom da je učenje programiranja često ograničeno na jedno polugodište u nastavnim planovima i programima.

Iz perspektive ciljne domene vidi se da istraživanja kognitivnih osobina (postignuća i kognitivnih vještina) nadmašuju istraživanja afektivnih osobina (stavova, percepcija, interesa, motivacija, mišljenja, samopouzdanja i samoučinkovitosti). Čini se da su ti rezultati u skladu s istraživanjem Ulutaş i Ubuz (2008) o matematičkom obrazovanju u Turskoj, kao i istraživanjem Turhan-Türkkan i Arslan-Namlı (2018) koje pruža uvid u turske poslijediplomske disertacije vezane uz korištenje softvera u nastavi matematike. Štoviše, većina ovdje istraženih studija odnosi se i na kognitivne i na afektivne osobine sudionika. U našem se istraživanju uspjeh smatrao najraširenijom kognitivnom varijablom, dok je stav/interes najčešća afektivna varijabla. U širem smislu, cilj obrazovnih istraživanja je analizirati ili poboljšati kognitivne ili afektivne domene sudionika. Kako programiranje nudi velik potencijal za razvoj kognitivnih sposobnosti prvenstveno kao dio vještina 21. stoljeća, čini se da je više nego prirodno istražiti ili razviti kognitivne značajke u istraživanjima programiranja. Naglasak u literaturi o postojanju pozitivne veze između kognitivne i afektivne domene (Anastasiadou i Karakos, 2011; Hawi, 2010; Hongwarittorrn i Krairit, 2010; Korkmaz i Altun, 2013) shvaća se kao opravdanje izvrsnosti istraživanja obje domene.

Kao još jedan kriterij, ispitivanje nalaza istraživanja pokazuje da bi većina mogla postići željene ciljeve. Ponavljamo, većina je primjera zaključena s pozitivnim utjecajem na kognitivne i afektivne osobine sudionika. Stopa uspjeha u afektivnim domenama pokazala se čak i većom nego u kognitivnim. U sličnoj su studiji Çatlak i sur. (2015) pregledali rezultate istraživanja Scratcha kao alata za programiranje. Utvrdili su da su svi rezultati pozitivni u vezi s afektivnom domenom, kao i sa značajnim dijelom kognitivne domene. Kao još jedan primjer, Lye i Koh (2014) analizirali su istraživanja usmjerena na poboljšanje vještina računalnoga razmišljanja programiranjem. I oni su izvijestili o pozitivnim rezultatima u većini istraženih studija. Stoga izjavljujemo da je naš rezultat kompatibilan s rezultatima Çatlak i sur. (2015), kao i s rezulatima Lye i Koh (2014).

Na kraju, rezultati intervencijskih istraživanja omogućili su iskazivanje sljedećih tvrdnji s najvećom učestalošću: aktivnosti dizajniranja igara, metoda binarnoga programiranja i aktivnosti robotskoga programiranja pridonose učenju koncepata povezanih s programiranjem, kao i poboljšanju učeničkih stavova i motivacije za programiranje, a time i njihovoga interesa za predmet. Ovaj je zaključak sličan rezultatima istraživanja Çatlak i sur. (2015) o utjecaju Scratcha kao alata za programiranje na proces učenja. Međutim, moguće je primijetiti neke proturječnosti o rezultatima dobivenim istraživanjima koja se bave računalnim razmišljanjem i vještinama rješavanja problema kao osnovnim vještinama potrebnim u 21. stoljeću. U nekim istraživanjima nastava programiranja ima pozitivan učinak na računalno razmišljanje i vještine rješavanja problema (Akcaoglu, 2014; Brown i sur., 2008; Chen i sur., 2017; Vatansever i Baltacı-Göktalay, 2018; Werner i Denning, 2009; Yünkül i sur., 2017); dok neka druga ne spominju prednosti provedbe navedenih vještina (Kalelioğlu i Gülbahar, 2014; Kukul i Gökçearslan, 2014; Oluk i Saltan, 2015). Ne bi bilo previše hrabro pretpostaviti da se vještine na visokoj razini, kao što je rješavanje problema, ne mogu razviti u kratkom vremenu, stoga bi trebalo provesti dugoročno i višenamjensko istraživanje (Kalelioğlu i Gülbahar, 2014; Kukul i Gökçearslan, 2014; Oluk i Saltan, 2015). Zaključno, valja napomenuti da, iako su istraživanja koja se ovdje razmatraju planirana korištenjem različitih alata za programiranje, alata za prikupljanje podataka, trajanja, sadržaja i ciljeva, ona pokazuju slične aspekte, što postavlja temelje za istraživački okvir koji bi mogao privući istraživače koji proučavaju učinke nastave programiranja na srednjoškolskoj razini. Aspekti dizajna istraživanja koji su ovdje podijeljeni predstavljenim istraživanjima definirani su iz distribucija istraživanja, a nakon toga je stvoren okvir dizajna za istraživanje nastave programiranja. Okvir istraživačkoga dizajna prikazan je na slici 11 u nastavku.

Slika 11.

Na slici 11 prikazan je okvir dizajna za istraživanje nastave programiranja u sustavu srednjega obrazovanja. Vjeruje se da će ovaj okvir pomoći u vođenju istraživača u procesima planiranja i dizajniranja njihovih istraživanja nastave programiranja. Model je sastavljen određivanjem klastera metode, veličine, cilja, alata za prikupljanje podataka, trajanja, alata za programiranje i sadržaja nastave u istraživanjima koje se proučavaju. Ono što se može vidjeti jest da su istraživanja nastave programiranja koristila različite metode istraživanja, ovisno o odgovarajućem cilju u vezi s afektivnim ili kognitivnim područjem. Metode istraživanja razlikuju se ovisno o dizajnu istraživanja. Još jedan upečatljiv nalaz iz distribucije istraživanja jest da je u svim istraživanjima poduka programiranja oblikovana pod nadzorom učenika ili istraživača, a ne izgrađena na određenom obrazloženju ili teorijskom standardu.

Iako se istraživanja nastave programiranja provode još od 1960-ih, čini se prikladnim napomenuti da su primjeri nastave programiranja na srednjoškolskoj razini značajno privukli pozornost istraživača i očekuje se da će daljnja istraživanja nastaviti i u budućnosti. Vjerujemo da bi otkrivanje sličnosti i različitosti relevantnih istraživanja moglo usmjeriti istraživače te im pomoći u daljnjem radu.