International Journal of Learning Spaces Studies, 2024. 2(3): 81-89.

The International Journal of Learning Spaces Studies (IJLSS)

Homepage: https://journals.artahub.com/

ORIGINAL RESEARCH ARTICLE

Investigation of the Effectiveness of Enhanced Computational Thinking Through Scratch on Self-Regulation

Elham Rostami[1], Javad Hatami[2*]

[1] *Rostami, Elham. Graduated MA in Educational Technology, Faculty of Humanities, Tarbiat Modares University, Tehran, Iran. e.rostamii@modares.ac.ir*

[2] *Hatami, Javad, Professor in Educational Technology, Faculty of Humanities, Tarbiat Modares University, Tehran, Iran. J.hatami@modares.ac.ir*

ARTICLE INFO

Article History:
Received: 2023.12.08
Revised: 2024.01.06
Accepted: 2024.04.20
Published Online: 2024.09.01

Keywords:
Self-regulation, Computational thinking, Scratch, Metacognition

Number of Reference: 35
Number of Figures: 0
Number of Tables: 4

DOI:
10.22034/lss.2024.490581.1027

Publisher:

Ayande Amoozan -e- ATA (AAA)

ABSTRACT

Abstract

This study investigated the effectiveness of computational thinking education in enhancing self-regulation skills. The population of this study consisted of sixth-grade students, and the sample was drawn from sixth-grade students in Alborz County, Qazvin Province, Iran, during the 2023-2024 academic year. A quasi-experimental design was employed, with two groups (control and experimental) participating. Each group consisted of 30 students. A convenience sampling method was used, but participants were randomly assigned to the control and experimental groups. The experimental group received instruction using the Scratch programming language. Both groups underwent seven training sessions, conducted by the same instructor. Pre and post-tests were administered to measure changes in self-regulation skills. The Buford self-regulation questionnaire, a 14-item instrument designed to assess self-regulation based on Bandura's social cognitive theory, was used. The questionnaire's items were measured on a Likert scale and assessed two factors: cognitive and metacognitive self-regulation strategies. The reliability and validity of the Buford questionnaire have been established in previous studies. The results of a covariance analysis indicated that there was no significant difference between the two groups in terms of the overall impact on self-regulation. However, the experimental group scored higher on the metacognitive self-regulation dimension. Therefore, it can be concluded that the type of educational media used for teaching computational thinking is not critical for increasing self-regulation. Computational thinking education, regardless of the medium, can enhance self-regulation skills. **©authors**

►**Citation:** Rostami, E., & Hatami, J. (2024). Investigation of the Effectiveness of Enhanced Computational Thinking Through Scratch on Self-Regulation. *The International Journal of Learning Space Studies (IJLSS),* 2(2): 81-89.10.22034/lss.2024.490581.1027

Introduction

According to the World Health Organization (WHO, 2018), adolescence is defined as the period between the ages of 10 and 19. Adolescents during this period often face various challenges. One of the challenges during this period is the low self-regulation abilities of students. Self-regulation is a multidimensional construct that relies on several psychological processes and can be defined as the ability to control and monitor attention, thoughts, emotions, and behaviors .

Self-regulated learners actively engage in their learning at metacognitive, motivational, and strategic levels (Zimmerman et al., 1995).

Effortful control (self-regulation) involves "sustained attention, shifting attention, inhibitory control, perceptual sensitivity, and low-intensity pleasure." Children who exhibit high levels of effortful control are able to maintain their arousal without it becoming excessive and possess strategies for self-soothing. Conversely, children who have difficulty with effortful control are often unable to regulate their arousal. They are easily aroused and experience intense emotions. A recent study of school-aged children in the United States and China revealed that, across both cultures, low effortful control was associated with externalizing problems such as lying, cheating, defiance, and excessive aggression (Zhou, 2009).

Various methods have been proposed to enhance adolescents' self-regulation. One such method is to promote higher-order thinking skills such as critical thinking, problem-solving, and computational thinking. The development of self-regulation is influenced by many factors, including modeling and self-efficacy (Bandura, 2010).

Learners need self-regulation in the learning process to effectively learn and master relevant knowledge and skills (Greene et al., 2018). Computational Thinking processes can be viewed as monitoring and goal-directed processes, and self-regulated learning theory can be used as a framework to assess and enhance Computational Thinking (Burton, Cleary, & Kitsantas, 2015).

Computational Thinking (CT) is a crucial cognitive skill for navigating our technology-reliant world. Educators are increasingly incorporating CT into early childhood curricula globally. While there are numerous learning methods to cultivate CT in young children, the most effective approach remains unclear (Hsu et al., 2018). Unplugged learning is a prevalent strategy for fostering CT in young learners (Caeli & Yadav, 2020).

CT is a fundamental cognitive skill essential for navigating our technology-driven world. As a result, it is increasingly being integrated into early-stage school curricula globally (Shute et al., 2017). While a universal definition of CT remains elusive, various frameworks exist to define its components. Selby and Woollard (2013) propose a framework that includes abstraction, algorithmic thinking, decomposition, evaluation, and generalization. Another influential framework is the 3D framework by Brennan and Resnick (2012), which categorizes CT into three dimensions: computational concepts, practices, and perspectives.

While numerous learning strategies exist to cultivate Computational Thinking (CT) in young learners, the most effective approach remains unclear. Recent research indicates that problem-based learning, collaborative learning, project-based learning, and game-based learning are among the most widely used and promising strategies for fostering CT skills (Hsu et al., 2018).

Nowadays society requires individuals with strong technological skills to keep pace with rapidly evolving technologies. It's clear that these skills will be crucial for future career success (Rodríguez-Martínez, González-Calero & Sáez-López, 2020).

Wing (2008) predicted that computational thinking (CT) would become a crucial skill for students in the future. While the concept of CT was introduced by Papert (1980), its definition and scope continue to evolve. CT involves recognizing computational patterns and applying computer science techniques to understand natural and artificial systems (Furber, 2012). More specifically, it encompasses problem-solving, system design, and understanding human behavior through computational lenses (Wing, 2008). DiSessa (2000) viewed CT as a new form of literacy, applicable across various subjects and domains (Weintrop et al., 2016).

Technology has revolutionized the way we interact, communicate, learn, and work (Cabero & Llorente, 2010). To fully harness the potential of digital technology, it is crucial to introduce programming languages to young learners. This will equip them with the essential digital skills needed for the 21st century (Maloney, Peppler, Kafai, Resnick, & Rusk, 2008).

While programming languages are more accessible to young learners than ever before, challenges remain in fully harnessing their educational potential. One common issue is the disconnect between programming activities and real-world contexts, which can hinder student engagement (Resnick et al., 2009). Past technical limitations have also overshadowed the potential of programming to enhance early education (Garneli & Chorianopoulos, 2018) .

Alic, Scratch, Cod.org, and MIT App Inventor are four widely used visual programming platforms and tools, as highlighted in a recent study by Sun, Lee, Su, and Chen (2023). Given the challenges associated with learning text-based programming languages, visual programming languages (VPLs) have been increasingly adopted in computational thinking (CT) education, especially within K-12 settings. VPLs facilitate programming learning by allowing learners to focus on CT concepts and practices rather than syntax (Weintrup & Wilensky, 2017). Consequently, there is a growing interest among educators to introduce CT and VPLs into K-12 classrooms.

Scratch (Resnick et al., 2009) is a visual programming platform designed for children aged six and older. It provides a website where users can share projects and collaborate with other young programmers. With over seven million registered users and ten million shared projects, Scratch has gained significant global popularity. Beyond teaching programming, Scratch aims to develop various skills and enhance learning in other subjects (Resnick, 2013). As a result, it's used in diverse educational settings, from extracurricular activities (Kafai, Fields, & Burke, 2012) to primary, secondary, and even university-level education (Moreno-León & Robles, 2015; Meerbaum-Salant, Armoni, & Ben-Ari, 2013; Malan & Leitner, 2007).

While digital games and applications are commonly used to foster Computational Thinking (CT), unplugged activities, which don't require electronic devices, have proven equally effective, especially in early childhood and primary education (Brackmann et al., 2017).

Literature Review

Chen et al. (2024) conducted a study investigating the impact of a professional development intervention in metacognitive science and learning environment on two groups of teachers and students. Employing a pre-test and post-test design, data was collected through validated observational tools, surveys, and reflective journals. The results indicated a statistically significant increase in students' self-regulated learning scores and in the effectiveness of science teaching and teachers' metacognitive knowledge.

Chen et al. (2023) conducted a study to investigate the effects of a game-based digital programming curriculum on the development of computational thinking among third and fourth-grade students. Employing a quasi-experimental design, the study found that computational thinking and overall learning motivation of the students significantly improved, indicating the effectiveness of the implemented curriculum. The study utilized the visual programming language Cod.org for instruction. Additionally, the findings suggested that self-regulated learning without teacher supervision progressed at a significantly slower pace.

Boroza et al. (2023) presented the results of their research on pre-service teachers who had no prior programming experience. The study revealed that teaching programming using Scratch enhanced higher-order thinking skills.

Gao, Yang, and Jiang (2023) examined the relationship between sequencing ability, self-regulation, and computational thinking among Chinese preschoolers. Results indicated a significant positive correlation between sequencing ability and computational thinking. Furthermore, self-regulation was found to be strongly correlated with computational thinking, and children with higher levels of sequencing ability and self-regulation performed better in computational thinking activities.

Lee et al. (2022) presented the results of their research on collaborative programming instruction based on metacognitive functioning. Both male and female high school students aged 15 participated in the experimental and control groups. The training was conducted in seven 45-

minute sessions. Pre- and post-tests were administered to both groups. The results indicated that programming enhanced the students' metacognition and effectiveness. At the end of the study, the experimental group significantly outperformed the control group in terms of academic progress and inclination towards computational thinking.

Alajlan et al. (2022) conducted a study to identify strategies for teaching computational thinking skills. The Delphi technique was used to gather insights from a group of experts in the field of appropriate education for computational thinking in K-12 computer science education. Based on the voting results, the panel considered "problem-based learning" suitable for developing computational thinking skills: data collection, data analysis, data representation, problem decomposition, algorithms and procedures, and parallelization. "Collaborative learning (teamwork)" was suitable for developing five computational thinking skills: data collection, data representation, problem decomposition, algorithms and procedures, and parallelization. "Project-based learning" was suitable for developing four computational thinking skills: data analysis, data collection, data representation, and problem decomposition. Similarly, the results confirmed that "scaffolding" and "visualization techniques" were suitable for developing three computational thinking skills: abstraction, data representation, and algorithms and procedures. Additionally, "inquiry-based learning" was suitable for developing two computational thinking skills: data analysis and data collection .Shou et al. (2022) conducted a meta-analysis to examine the results of previous studies on the effectiveness of programming in enhancing computational thinking skills in K-12 students. According to this study, various programming tools effectively enhance computational thinking skills in learners.Method

Methodology

This study is a quasi-experimental research, aiming to investigate the causal relationship between two variables. The research design is a pre-test-post-test control group design. Before implementing the independent variable (computational thinking training with Scratch), the selected participants in two groups were measured using a pre-test. The diagram of the research design is shown in the table. Two groups of sixth-grade students from Alborz County, Qazvin Province were selected using a convenience sampling method for this study. Before the commencement of the training, the Buford self-regulation questionnaire (1995) was completed by both groups. Seven training sessions were conducted for both groups, with the same instructor. In the experimental group, computational thinking training was conducted using the Scratch software. In each session, a part of the software was introduced, followed by a related project. In the control group, computational thinking was introduced, and in each session, it was presented according to Merrill's Component Display Theory (CDT). After the completion of the training sessions, the Buford self-regulation questionnaire was completed again by both groups. This questionnaire consists of 14 questions and includes questions from two domains: metacognition and cognition. The responses were on a 5-point Likert scale: strongly agree, agree, neither agree nor disagree, disagree, and strongly disagree.

Findings

Since the ratio of skewness and kurtosis to the standard deviation falls within the range of -2to +2, we conclude that the distribution of scores at both stages is not significantly non-normal. This indicates that the assumption of normality has been satisfied.

Descriptive statistics and central indicators, including mean and standard deviation, have been reported. Regarding the first sub-hypothesis, the results in the table indicate that the mean changes in the cognitive component level are not similar between the experimental and control groups. In the experimental group, the mean level of the cognitive component has increased, while in the control group, the mean level of this component has decreased. The standard deviation as a central indicator has increased for the cognitive component in both groups.

In examining the second sub-hypothesis, as shown in the table, the mean changes in the metacognitive component level differ between the experimental and control groups. In the experimental group, the mean level of the metacognitive component has increased, while in the control group, there has been little to no change in this component's mean level. The standard deviation has also increased for the metacognitive component in both groups.

Table 1. Descriptive statistics of the research variables in two stages

Status		N	Estimation error	Standard deviation	Average
Self-regulation pre-test	Experiment	30	1.239	6.788	52.70
	Control	30	0.825	4.516	53.53
Self-regulation post-test	Experiment	30	1.082	5.929	54.43
	Control	30	1.148	6.90	52.53

Table 2. The results of Cronbach's alpha on the overall score of the construct

Components	Critical value	Alpha statistic	Result
Self regulation	0.7	0.725	Acceptance

Table 3. Normality Assessment

Source	F	Mean square	df	Sum of squares	Dependent	Sig/	Partial eta square	Test power
Group	1.829	65.895	1	65.895	Self regulation	0.182	0.031	0.265
Error		36.035	57	2054.010	Self regulation			
Total			60	173849.000	Self regulation			

Table 4. Normality Assessment

Skewness		Kurtosis		Number	Status
Standard error	Statistics	Standard error	Statistics	Statistics	
0.608	0.679	0.309	-0.891	60	Self-regulation pre-test
0.608	-1.032	0.309	-0.491	60	Post_self-regulation test

As observed, the researcher's hypothesis was not confirmed. This suggests that enhanced CT (Computational Thinking) through Scratch software does not effectively impact students' self-regulation. The F-value for group effects was not statistically significant, indicating that, after controlling for the pre-test, there is no significant difference in the post-test mean scores between the two groups. Therefore, we accept the null hypothesis, which posits no significant difference in the post-test means of the two groups after removing the potential influence of the pre-test. If the F-value for the independent variable is not significant, we can claim that, after accounting for the pre-test effect (covariate), no significant difference was observed between the group means.

Discussion

Self-regulated learners demonstrate greater focus during instruction and are less prone to distraction, which can positively impact the quality of their learning. They also excel in establishing connections with others. Self-regulation can influence their effort and progress management, as well as shape their attitudes and beliefs. This skill is essential in fostering lifelong learners, who continuously seek new and improved solutions when facing challenges. Lifelong learners persist in their pursuit of knowledge and aim to enhance their expertise throughout their lives. Technological advancements do not pose a major or ambiguous challenge to them, as they are prepared for these changes and keep their knowledge up to date in response to emerging issues.

Problem-solving is one of the most effective methods for enhancing self-regulation skills. This is because solving problems requires evaluating solutions, breaking down issues into smaller components, and identifying the correct solution among various approaches. One approach to teaching problem-solving is through the use of computational thinking.

The researcher's initial hypothesis was that teaching computational thinking (CT) through Scratch software would be more effective in enhancing self-regulation than teaching CT without tools. However, the study revealed that teaching CT effectively enhances self-regulation in any form. Thus, the alternative hypothesis is supported.

Conclusion

The final conclusion drawn from comparative research in the field of media is that media, simply by virtue of being a medium, do not differ as vehicles through which educational content is transmitted from the source (sender) to the learner (receiver) (Clark, 1983). In other words, if media serve as content transmission tools, then different media, assuming they can fully convey the content from sender to receiver, show no fundamental differences. For example, if a math concept is presented by a teacher, through a 16mm film, or via a computer program, provided that each medium conveys the same content, no differences in student learning outcomes should arise based on the type of media used. Numerous comparative studies conducted on various media types support this conclusion (Fardanesh, 2019).

Practical Suggestions

1 .Developing Blended Learning Programs:

Considering that computational thinking can be effectively taught using various tools, blended programs combining digital methods (e.g., Scratch software) and non-digital approaches can be designed to enhance self-regulation skills.

2 .Focusing on Metacognitive Components in Content Design:

Given the positive impact of Scratch software on metacognitive components, educational content should be designed to strengthen this aspect of self-regulation.

3 .Expanding Computational Thinking Education in Schools:

Introducing computational thinking as a key skill in school curricula, with an emphasis on improving problem-solving abilities and self-regulation.

4 .Conducting Teacher Training Workshops:

Teachers should be trained in methods of teaching computational thinking (with and without digital tools) and understand its role in enhancing cognitive and metacognitive skills.

5 .Developing Localized Software:

Creating software similar to Scratch, tailored to the needs of Iranian students, to increase accessibility and the effectiveness of computational thinking education.

Research Suggestions

1 .Expanding Research to Different Age Groups:

Investigating the effectiveness of computational thinking education on self-regulation skills across other age groups, such as elementary, middle school, or university students.

Studying the impact of computational thinking education on self-regulation in adults and professionals.

2 .Exploring Various Computational Thinking Tools:

Assessing the impact of other tools and educational platforms (e.g., Python, Blockly, or App Inventor) on self-regulation skills.

Comparing traditional and digital methods in teaching computational thinking.

3 .Examining Various Aspects of Self-Regulation Skills:

Conducting in-depth analyses of the effects of computational thinking on self-regulation subcomponents, such as emotional control, time management, or motivation.

Studying the impact of computational thinking education on other cognitive and metacognitive skills.

4 .Broader Applications in Other Fields:

Exploring the effects of computational thinking education on academic performance in other subjects, such as mathematics, science, or languages.

Investigating the role of computational thinking in enhancing social skills and problem-solving abilities.

5 .Using Different Research Methods and Variables:

Designing longitudinal studies to examine the long-term impact of computational thinking on self-regulation.

Investigating the effects of cultural, gender, and economic factors on the effectiveness of computational thinking education.

Exploring the relationship between computational thinking and other learning theories, such as active learning or project-based learning.

6 .Developing Educational Content:

Designing blended learning programs aimed at improving self-regulation through computational thinking.

Examining the impact of game-based or gamified educational content on computational thinking and self-regulation.

7 .Utilizing Modern Technologies:

Studying the impact of artificial intelligence and robotics on computational thinking education and self-regulation skills.

Using augmented reality (AR) or virtual reality (VR) to enhance computational thinking education.

Limitations

1 .Internet Access for Completing Questionnaires:

Due to limited internet access for some participants, printed questionnaires were distributed. This may have reduced the quality of responses and the ease of data collection, as electronic versions could have been completed more quickly and accurately.

2 .Limitations of Scratch Software on Android:

The Android version of Scratch software has fewer features compared to the Windows version. This limitation might have affected the learning experience of the experimental group and created unequal opportunities for participants.

3 .Misinterpretation of Computational Thinking in Educational Texts:

In some resources, computational thinking is incorrectly translated mere

References

1. Alajlan, H., Alebaikan, R., & Almassaad, A. (2023). Computational thinking in K–12 computer education: appropriate pedagogy. Technology, Pedagogy and Education, 32(3), 337-349.DOI:10.1080/1475939X.2023.2211573
2. Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From scratch to "real" programming. ACM Transactions on Computing Education (TOCE), 14(4), 1-15.DOI:10.1145/2677087

3. Bandura, A. (2010). Addressing Population Growth: Social Cognitive Theory Goes Global. The Australian Humanist, (97), 3-5.
4. Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. Proceedings of the 12th Workshop on Primary and Secondary Computing Education, 65–72.DOI:10.1145/3137065.3137069
5. Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 Annual Meeting of the American Educational Research Association, 1, 25.
6. Broza, O., Biberman-Shalev, L., & Chamo, N. (2023). "Start from scratch": Integrating computational thinking skills in teacher education program. Thinking Skills and Creativity, 48, 101285.DOI:10.1016/j.tsc.2023.101285
7. Cabero, J., & Llorente, M. d. C. (2010). Comunidades virtuales para el aprendizaje. Edutec. Revista Electrónica de Tecnología Educativa, 34, 1–10.
8. Caeli, E. N., & Yadav, A. (2020). Unplugged approaches to computational thinking: A historical perspective. TechTrends, 64(1), 29–36.DOI:10.1007/s11528-019-00410-5
9. Chen, H. E., Sun, D., Hsu, T. C., Yang, Y., & Sun, J. (2023). Visualising trends in computational thinking research from 2012 to 2021: A bibliometric analysis. Thinking Skills and Creativity, 47, 101224.DOI:10.1016/j.tsc.2023.101224
10. Chen, S., Sermeno, R., Hodge, K., Murphy, S., Agenbroad, A., Schweitzer, A., ... & Roe, A. J. (2024). Young Children's Self-Regulated Learning Benefited from a Metacognition-Driven Science Education Intervention for Early Childhood Teachers. Education Sciences, 14(6), 565.DOI:10.3390/educsci14060565
11. Clark, R. E. (1983). Reconsidering research on learning from media. Review of Educational Research, 53(4), 445-459.DOI:10.3102/00346543053004445
12. DiSessa, A. A. (2000). Changing minds: Computers, learning, and literacy. Cambridge: MIT Press.
13. Fardanesh, H. (2019). Theoretical Foundations of Educational Technology. Tehran: Samt Publications.
14. Furber, S. (2012). Shutdown or restart? The way forward for computing in UK schools. London: The Royal Society.
15. Gao, H., Yang, W., & Jiang, Y. (2023). Computational thinking in early childhood is underpinned by sequencing ability and self-regulation: A cross-sectional study. Education and Information Technologies, 28(11), 14747-14765.DOI:10.1007/s10639-023-11895-2
16. Garneli, V., & Chorianopoulos, K. (2018). Programming video games and simulations in science education: Exploring computational thinking through code analysis. Interactive Learning Environments, 26(3), 386–401.DOI:10.1080/10494820.2017.1337035
17. Greene, J. A., Copeland, D. Z., Deekens, V. M., & Seung, B. Y. (2018). Beyond knowledge: Examining digital literacy's role in the acquisition of understanding in science. Computers & Education, 117, 141-159.DOI:10.1016/j.compedu.2017.10.003
18. Hsu, T., Chang, S., & Hung, Y. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. Computers & Education, 126, 296–310.DOI:10.1016/j.compedu.2018.07.004
19. Kafai, Y. B., Fields, D. A., Roque, R., Burke, Q., & Monroy-Hernandez, A. (2012). Collaborative agency in youth online and offline creative production in Scratch. Research and Practice in Technology Enhanced Learning, 7(2), 63-87.
20. Lee, S. W. Y., Tu, H. Y., Chen, G. L., & Lin, H. M. (2023). Exploring the multifaceted roles of mathematics learning in predicting students' computational thinking competency. International Journal of STEM Education, 10(1), 64.DOI:10.1186/s40594-023-00449-3
21. Malan, D. J., & Leitner, H. H. (2007). Scratch for budding computer scientists. ACM Sigcse Bulletin, 39(1), 223-227.DOI:10.1145/1227504.1227388
22. Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008, March). Programming by choice: Urban youth learning programming with Scratch. Proceedings of the 39th SIGCSE technical symposium on Computer science education (pp. 367-371).DOI:10.1145/1352135.1352260

23. Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: análisis automático de proyectos Scratch para evaluar y fomentar el pensamiento computacional. Revista de Educación a Distancia (RED), (46).DOI:10.6018/red/46/10
24. Papert, S. A. (1980). Mindstorms: Children, computers, and powerful ideas. New York: Basic Books.
25. Peters-Burton, E. E., Cleary, T. J., & Kitsantas, A. (2018). Computational thinking in the context of science and engineering practices: A self-regulated learning approach. In Digital technologies: Sustainable innovations for improving teaching and learning (pp. 223-240).
26. Resnick, L. B. (2013). Constructing knowledge in school. In Development and learning (pp. 19-50). Psychology Press.
27. Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., ... & Silver, J. (2009). Scratch: Programming for all. Communications of the ACM.
28. Román-González, M., Pérez-González, J., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. Computers in Human Behavior, 72, 678–691.DOI:10.1016/j.chb.2016.08.047
29. Selby, C., & Woollard, J. (2013). Computational thinking: the developing definition. Southampton, UK: University of Southampton.
30. Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. Educational Research Review, 22, 142-158.DOI:10.1016/j.edurev.2017.09.003
31. Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. ACM Transactions on Computing Education (TOCE), 18(1), 1-25.DOI:10.1145/3089799
32. Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology, 25(1), 127–147.DOI:10.1007/s10956-015-9581-5
33. Wing, J. M. (2008). Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366(1881), 3717–3725.DOI:10.1098/rsta.2008.0118
34. Zhou, Q., Lengua, L. J., & Wang, Y. (2009). The relations of temperament reactivity and effortful control to children's adjustment problems in China and the United States. Developmental Psychology, 45(3), 724-739.DOI:10.1037/a0013776
35. Zimmerman, B. J. (1995). Self-regulation involves more than metacognition: A social cognitive perspective. Educational Psychologist, 30(4), 217-221.DOI:10.1207/s15326985ep3004_8