RUDN Journal of Informatization in Education
Вестник РУДН. Серия: Информатизация образования

2020 Vol. 17 No. 3 179–189
http://journals.rudn.ru/
informatization-education

DOI 10.22363/2312-8631-2020-17-3-179-189
UDC 373

Scientific article

# Programming as a tool for across-subjects learning in primary school

**Christer Sjöberg[1], Tove Risberg[2],**
**Jalal Nouri[3], Eva Norén[3], Lechen Zhang[3]**

[1]Strandskolan
*14 Lagergrens väg, Tyresö, 135 63, Kingdom of Sweden*
[2]Tyresö school
*6 Tyresövägen, Tyresö, 135 61, Kingdom of Sweden*
[3]Stockholm University
*Frescativägen, Stockholm, 114 19, Kingdom of Sweden*

**Abstract.** *Problem and goal.* Computational thinking has been introduced in many countries around the world and teachers are working intensely to incorporate programming activities in the classroom. However, teachers are faced with several challenges due to the fact that there is still little research conducted focusing on programming education for younger children, that programming didactics is a rather new phenomenon for the K-9 educational system, and that K-9 teachers have little training with regards to programming. In Sweden for instance, programming has been introduced in several subjects and not as a subject in its own, which create a pressure on teachers to utilize programming as an instrument to teach and enhance learning of different subjects such as mathematics.

*Methodology.* In this paper, we report on a larger lesson study conducted in a primary school (sixth grade) in Sweden with a total of 155 participating pupils. The aim of the lesson was to study whether the visual programming languages, *Scratch*, in particular, can be used to teach computational thinking, mathematics and social science in an interdisciplinary way.

*Results.* Thus, the paper more specifically presents findings related to: 1) reflections of the use of lesson study methodology to develop programming education; 2) how programming can be utilized as an instrument to teach mathematics as well as social sciences in an interdisciplinary way; and 3) the didactical strategies employed by the teachers.

*Conclusion.* The evidence from this study suggests that the interdisciplinary character of the lesson which incorporates learning goals of mathematics, social science and programming was highly beneficial. The pupils gained a better understanding of learning material by drawing, digitalizing and animating their ideas in *Scratch*.

**Keywords:** computational thinking, programming, mathematics, social science, learning, primary school

**Problem statement.** Computational thinking (CT), a term coined by Papert [1] in 1980 and revitalized by Wing a decade ago, is a 21st century skill that it is imperative the coming generation develops. Wing stated that "To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability" [2]. That is to say; computational thinking is a set of problem-solving thought processes derived from computer science but applicable to any subject or domain.

This belief has been internationally acknowledged as increasing numbers of educational systems have integrated CT in their compulsory education in recent years. For example, in the west, "Computing" in England, "ICT capability" in the new national curriculum of Australia; in the east, South Korea's new curricula consists of a compulsory subject "Informatics" in Middle School focusing on computational thinking and programming [3]. Sweden is also one of the countries that embraced this change in 2018 as programming became compulsory in K-12 education. Programming is introduced mainly as part of the mathematics curriculum under the topics "Algebra" and "Problem-solving" [4; 5].

With all the attention on computational thinking and programming, the most crucial question is "do teachers possess adequate pedagogical skills and subject knowledge to deliver the new curriculum"? The Swedish National Union of Teachers' questionnaire [6] pointed out that 70% of sixth–ninth grades teachers do not have any experience in integrating programming. For teachers facing curriculum change, how to teach it is very pertinent. Successful integration of the new curricula and practice require appropriate pedagogies, particularly in those aspects of computer science that relate to algorithms, programming and the development of computational thinking skills [7]. For example, in UK where programming has been made a compulsory part of primary school, an inspiring research project, *Scratch Maths*, showed that programming and mathematics could be successfully bridged [8; 9]. Visual programming languages, such as *Scratch*, has been recommended in primary educational cross-curricular implementation [10].

Lesson study have been proposed as a method to scientifically and systematically develop lessons and curricula, and at the same time support professional development of teachers [11]. The focus of the methodology is to support teachers in gaining knowledge and making didactical adjustments through collaborative investigation of individual lessons [12].

Against such a background, this paper reports on a lesson study conducted by teachers in primary school in Sweden, that aimed at developing a cross-subject lesson that incorporate mathematics, social science and programming. As such we contribute with an example of a cross-subject lesson, developed with scientific methods and with a focus on teachers' professional development, to the literature concerning computational thinking that so far have overlooked these issues.

**Method of research.** This study was conducted within the frame of the Swedish national research project, "Programmering i ämnesundervisningen" [13]. Lesson study [11] have been used as the main methodology to develop a cross-subject lesson that incorporates mathematics, social science and programming. Lesson study is a methodology for instructional inquiry and professional development and was originally developed in Japan. The focus of the methodology is to support teachers in gaining knowledge and making didactical adjustments through collaborative investigation of individual lessons [12]. Thus, the procedure of lesson study is as following.

A typical lesson study starts with that teachers jointly plan and design a lesson in detail. Then one member of the teacher team teaches the lesson while the other colleagues participate in the classroom as observers that collect data regarding the teaching and students learning. This phase is followed by an extensive evidence-based team discussion and analysis of the implemented lesson that result in proposed lesson revisions and a redesign of the lesson. Then another member of the team teaches the revised lesson while remaining teachers observe it. In the final phase the teacher team have a final debriefing in which they analyze and reflect about their experiences, document findings, and consider paths for future work [12].

The lesson study described in this paper was conducted in two primary schools. Four teachers from four different schools contributed to the planning of the lesson conducted during a planning session. The lesson was planned in high detail using a template developed (see Table) based on [12]. During the planning phase, the teachers also developed learning material (template for the pupils) needed for the planned lesson.

*Table*

**The template used for lesson study planning**

| Lesson activities | Expected reactions and planned responses | Goals and aim with the activities | Methods of data collection and evaluation | Evaluation |
|---|---|---|---|---|
| Describe in detail the activities that the lesson consists of and what teachers and students should do under these.<br>Try to divide the lesson into as many sub-activities as you can with associated specific sub-goals | How can the pupils react under the different activities?<br>Expected behaviors?<br>What difficulties can different pupils encounter?<br>How should the teacher respond with support? | What are the goals and objectives for each individual activity?<br>Goal achievement should be able to be evaluated/assessed | What methods and techniques do you use to evaluate whether the goals in the activities are met?<br>What kind of evidence and signs for learning should you collect and base the evaluation on?<br>How should you collect the evidence/data?<br>Videos?<br>Audio recordings?<br>Observation notes?<br>Interviews? Tests?<br>Combinations? | This is completed after the lesson has been completed.<br>Evaluate each sub-activity in an evidence-based way.<br>What worked well, what worked less well and based on what indications/evidence? |

The lesson was tested two times (two iterations) in two classes in sixth grade. The total number of participating pupils was 45 (23 during the first lesson, and 22 during the second). Two teachers participated during the first lesson; one as the teacher of the lesson, and the other as observer. The observer wrote observation notes that was used in the evaluation phase that followed the lesson. Two teachers analysed and evaluated the lesson after the first try. The evaluation resulted in three main adjustments of the original lesson. Firstly, we noticed that many pupils had difficulties in understanding which political parties that could collaborate with each other considering their individual ideologies.

These discussions took time and resulted in too little time remaining for the actual programming activity. To address this, in the second iteration we provided the pupils with an informative text that explained the ideologies/opinions of the different political parties. Secondly, we could conclude that we had allocated too little time for the lesson, so we extended the project time with two additional lessons in the next iteration. Thirdly, in the first lesson the pupils worked in groups of three, but we could observe that in many cases only two of the pupils were active. For the second iteration, we decided that the pupils should work in groups of two instead.

The revised lesson was then implemented in another school, and a final evaluation was performed afterwards.

**Results.** The idea of planning a lesson that included content from different subjects, in this case mathematics, social science and programming, was to create an opportunity for the pupils to see how programming can be used as a tool in different domains such as social science and mathematics. More specifically, the learning goals of the lesson were the following:

Lesson goal 1: the pupils should gain an understanding of how societal structures can be analyzed using concepts and models from social science. More specifically, the goal of the lesson was to get a deeper understanding of how governments can be formed based on voting results (social science).

Lesson goal 2: the second goal was that the pupils should deepen their knowledge and skills with regards to interpretation and application of pie charts (mathematics).

Lesson goal 3: the third goal was that the pupils should deepen their knowledge and skills regarding the use of *Scratch*, and more specifically, in terms of using backgrounds and sprites, recording and making use of sound in the programs, and make sprites move in different directions (programming).

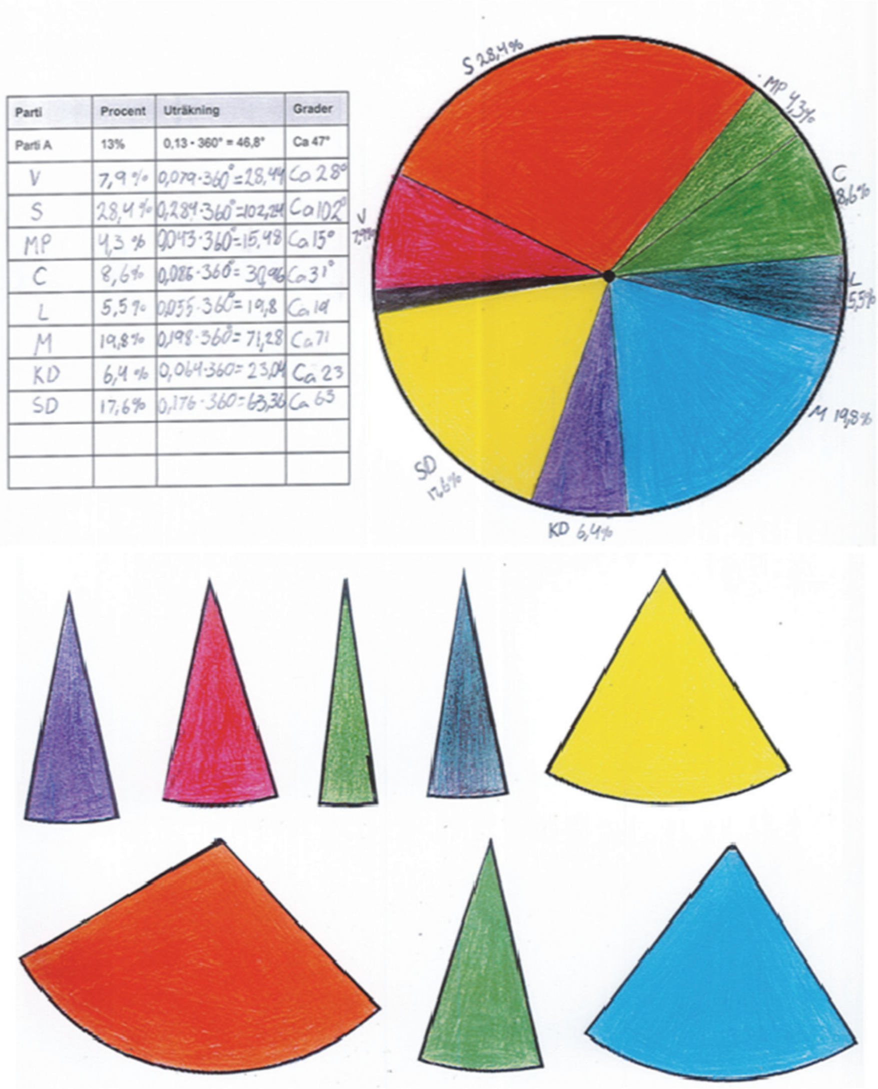
*The final lesson.* The following is a description of the final revised lesson in the lesson study. Before the students started the actual programming activity, they were introduced to the principles of how a government in Sweden can be formed. They also had knowledge of the different political parties and who usually found collaborations and previously ruled together. Prior to the programming activity, the students had learned also learned how to convert a percentage to degrees in order to create circle sectors in a pie chart.

The first sub-activity was for the pupils to create pie charts by hand representing percentages of votes of different parties, and then scan them so they could be digitally transmitted as a sprite in *Scratch* (see Figure 1).

The pupils were randomly grouped into pairs and were tasked with proposing a new government based on the election results. The proposal would be presented as an animation in *Scratch* where a pie chart gradually emerges while a narrative voice explains why this particular government can be possible. The pupils got a bulleted list to follow when they created their speech script. The bulleted list contained the following points:

- Which parties do you think should form the government?
- Will there be a majority government or a minority government and why then?
- Why should these parties form a government?
- Does the government need support from other parties, if so which ones and why?

The pupils were also encouraged to use mathematical concepts such as majority, minority and percentage points. For the programming in *Scratch*, the teacher provided a project template with some fundamental structures in it, that the pupils could remix and make additions to, as it was evaluated that all pupils were not able to code the full program.

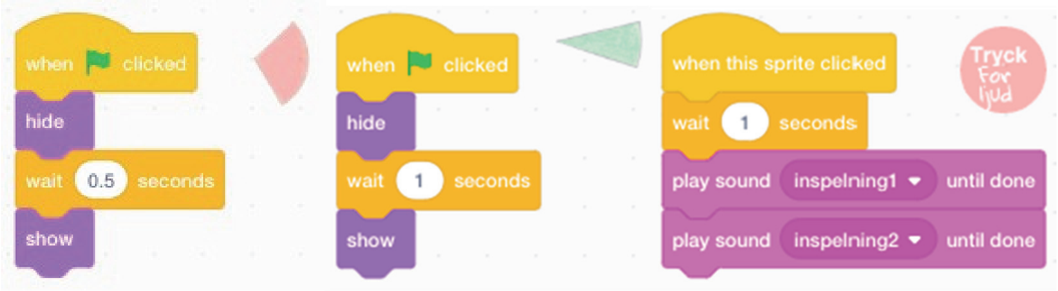


| Parti | Procent | Uträkning | Grader |
|---|---|---|---|
| Parti A | 13% | 0,13 · 360° = 46,8° | Ca 47° |
| V | 7,9 % | 0,079 · 360° = 28,44 | Ca 28° |
| S | 28,4 % | 0,284 · 360° = 102,24 | Ca 102° |
| MP | 4,3 % | 0,043 · 360° = 15,48 | Ca 15° |
| C | 8,6% | 0,086 · 360° = 30,96 | Ca 31° |
| L | 5,5 % | 0,055 · 360° = 19,8 | Ca 19 |
| M | 19,8 % | 0,198 · 360° = 71,28 | Ca 71 |
| KD | 6,4 % | 0,064 · 360 = 23,04 | Ca 23 |
| SD | 17,6% | 0,176 · 360 = 63,36 | Ca 63 |
| | | | |
| | | | |

**Figure 1.** Handmade pie charts that were scanned and used as sprites in *Scratch*

Most followed the template and primarily made the changes that were required to make their proposals, for example, that they put the circle sectors representing the proposed government parties together to graphically show that they form a majority.

Some created some improvements, e.g. downloaded the party symbols as sprites to further clarify the circle sectors' relationship with each party. Each party was represented by a circle sector corresponding to its share of the votes. Figure 2 below shows examples of code for two of the parties. The first parties circle sector (the red) appears after 0.5 seconds and the second parties (the green) after 1 second. Other parties then emerged afterwards and finally a pie chart was created. Figure 3 shows the code for playing the audio files. They are played when you click on the red bar where it says “Tryck för ljud”. Figure 3 shows how the program (interactive animation) could look like when running.



```
when 🏁 clicked        when 🏁 clicked        when this sprite clicked
hide                   hide                   wait 1 seconds
wait 0.5 seconds       wait 1 seconds         play sound inspelning1 ▾ until done
show                   show                   play sound inspelning2 ▾ until done
```

**Figure 2.** The prepared *Scratch* code for the interactive animation

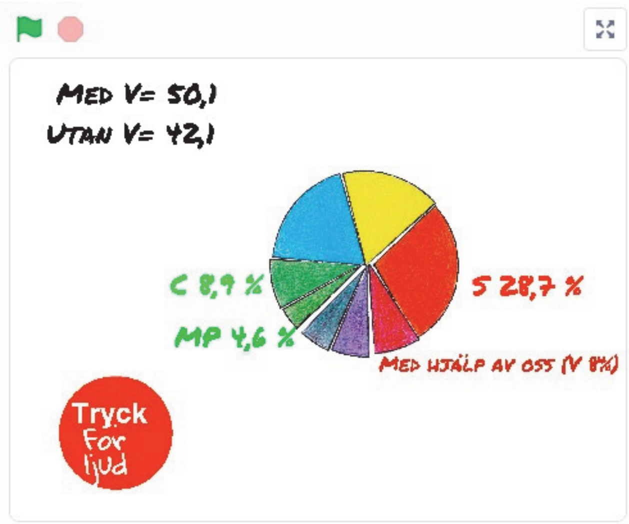


**Figure 3.** The interactive animation when running

The teacher's role during the lesson was to introduce the activity and the lesson goals, as well as the central concepts and tools (templates), and to help and guide the pupils in their work.

*Reflections of goal attainment and experienced difficulties and challenges.* In the preparatory work when the pupils worked analogously to convert the election results to circle sectors, they collaboratively discussed with each other and thus in many cases became learning resources for each other.

Since it was a relatively difficult mathematical task considering the age of the pupils', the discussions were occasionally conducted at a rather advanced level. During the observed lesson, they already had their circle sectors ready and therefore only different options were discussed to get together circle sectors that corresponded to more than 50%, which was not a task that naturally led to equally deep mathematical discussions.

A discussion that arose among several pupils was about plausibility when they had converted the percentages wrong for the small parties, e.g. by writing

4.4% as 0.44 instead of 0.044; i.e. they were concerned about mathematical errors. Some discovered it when they were to draw the circle sectors and noticed that the sum was considerably more than 360 degrees. Others noticed the error because they had the knowledge of about how large the parties should be and then reacted when they learned that actual small parties had been represented as rather large parties in the pie chart.

Thus, considering the mathematical learning goals of the activity, and our observations of the pupils' discussions, we could note that the pupils got a little more in-depth knowledge about how to convert from percentage to decimal. Furthermore, we could observe that the pupils gained knowledge and skill in converting percentages represented in decimal form into degrees/angles; for instance, by calculating 0.044*360° in order to construct a pie sector for the chart. We could also observe that the pupils consolidated the mathematical and social science concepts of majority and minority during the work. These concepts were used extensively in the discussions and the pupils showed understanding when they put together majority governments that would have the best opportunity to pursue their policies.

In terms of the learning goals related to social science, the pupils displayed a good understanding of the democratic processes that govern the formation of government and they could reason about the benefits in appointing a majority government. They also displayed some knowledge of the political parties' political viewpoints and could see which parties were closer to each other than others. Besides, we could observe that the pupils could identify challenges of forming a government that could work together in a majority position.

In terms of the learning goals related to programming, we observed that the pupils were highly motivated and engaged when using *Scratch*. We could also observe that they through collaborative learning and the scaffolding of the teacher, could remixe the provided template with ease, and could work with backgrounds, sprites, sound recordings and movement of sprites without encountering significant challenges.

However, in the first lesson performed, many pupils encountered difficulties in understanding which political parties that could collaborate, which resulted in that a lot of time was spent on such discussions and too little time was available for the programming. Consequently, in the second revised lesson, the teacher provided a document to the pupils that contained information about each political party. The evaluation of the first lesson also led the teachers to conclude that the allocated time for the lesson was not enough, so for the next iteration time was added in form of two additional lessons so that the pupils could complete the projects.

**Discussion.** This paper presented a lesson that was developed by teachers through lesson study methodology. The developed lesson incorporated learning goals from mathematics, social science and programming. From our perspective, the interdisciplinary character of the lesson was beneficial in several ways [10].

One of the advantages of working interdisciplinary, was that the pupils could contextualize mathematics; i.e. mathematics became a concrete tool for analyzing a social phenomenon; and programming became an instrument for learning mathematics and social science. To exemplify, the social science concepts the pupils worked with, like *voting results* and *government*, made meaning together with the mathematical concepts, *minority* and *majority*, and the interpretation and application of pie charts (a mathematical model). The pupils gained a better understanding of how governments can be formed based on the mathematical voting results, by drawing, digitalizing and applying the circle sectors into *Scratch*, and by designing a program that animated their suggestions.

As teachers planning and conducting the lesson, we could clearly see that the pupils were more engaged than usual and based on our observations, it is related to the use of programming as a tool and context for learning mathematics and social science. Using *Scratch* was engaging for the pupils. These findings support earlier research (see [7–9]). We can also conclude that the didactical strategy of letting the pupils use predefined code and remix it was beneficial, in line with previous research (see [14; 15]). By remixing, instead of building everything from scratch, it was possible for all pupils to reach the goals of the lesson in time. In our experience as teachers, when pupils are instructed to work without code templates, many don't know how to start, get stuck, and don't finish the projects as intended. Another benefit of this didactical strategy is that teachers are not familiar with programming can without greater difficulties incorporate programming in their lessons.

**Conclusion.** Concerning the use of lesson study methodology [9], we conclude that it was beneficial for us – teachers to work collaboratively in a focused, systematic and pedagogically/didactically conscious way, when planning, analyzing and improving learning activities. Usually this type of work is done by teachers individually. Furthermore, as teachers, our impression is that many of the conclusions that we draw during the analysis phase could be generalized and be beneficial in other contexts in which we individually plan our lessons.

However, one disadvantage of this method is that it takes time and resources, but in the end, the investment of time and resources that we do as teacher in order to work in accordance with lesson study methodology, will save us time and resources in the long run and produce higher quality education.

## References

[1] Papert S. *Mindstorms.* New York: Basic Books; 1980.

[2] Wing, JM. Viewpoint. Computational thinking. *Communications of the ACM.* 2006; 49(3):33–35.

[3] Heintz F, Mannila L, Färnqvist T. A review of models for introducing computational thinking, computer science and computing in K-12 education. *Frontiers in Education Conference* (pp. 1–9). Pennsylvania, US, IEEE; 2016, October.

[4] Heintz F, Mannila L, Nordén LÅ, Parnes P, Regnell B. Introducing programming and digital competence in Swedish K-9 education. *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (pp. 117-128). Springer, Cham; 2017, November.

[5] National Agency of Education. *Course syllabus in mathematics, compulsory school.* Stockholm: Skolverket; 2017.

[6] Skolvärlden. 2017. Available from: http://skolvarlden.se/artiklar/8-av-10-larare-osakra-pa-att-lara-ut-programmering (accessed: 16.01.2019).

[7] Sentance S, Csizmadia A. Computing in the curriculum: challenges and strategies from a teacher's perspective. *Education and Information Technologies.* 2017;22(2):469–495.

[8] Benton L, Hoyles C, Kalas I, Noss R. Bridging Primary Programming and Mathematics: Some Findings of Design Research in England. *Digital Experiences in Mathematics Education.* 2017:1–24.

[9] Benton L, Saunders P, Kalas I, Hoyles C, Noss R. Designing for learning mathematics through programming: a case study of pupils engaging with place value. *International journal of Child-Computer Interaction*. 2018;16:68–76.

[10] Sáez-López JM, Román-González M, Vázquez-Cano E. Visual programming languages integrated across the curriculum in elementary school: a two year case study using "Scratch" in five schools. *Computers & Education.* 2016;97:129–141.

[11] Yoshida M. Mathematics lesson study in the United States: current status and ideas for conducting high quality and effective lesson study. *International Journal for Lesson and Learning Studies.* 2012;1(2):140–152.

[12] Ermeling A, Ermeling G. Learning to learn from teaching: a first-hand account of lesson study in Japan. *International Journal for Lesson and Learning Studies*. 2014;3(2):170–191.

[13] *Programmering i ämnesundervisning* [*Programming in subject didactic*]. Available from: https://www.ifous.se/programmering-i-amnesundervisningen/ (accessed: 10.04.2020).

[14] Nouri J, Norén E, Skog K. Didactical strategies employed by teachers when teaching programming in K-9 education. *The 12th Annual International Technology, Education and Development Conference (Valencia, Spain, March 5–7, 2018)* (pp. 7983–7989). The International Academy of Technology, Education and Development; 2018.

[15] Sjöberg C, Nouri J, Sjöberg R, Norén E, Zhang L. Teaching and learning mathematics in primary school trough "Scratch". *International Conference on Education and New Learning Technologies, EDULEARN18 Proceedings* (pp. 5625–5632). 2018.

**Article history:**
Received: 10 March 2020
Accepted: 13 April 2020

**For citation:**
Sjöberg Ch, Risberg T, Nouri J, Norén E, Zhang L. Programming as a tool for across-subjects learning in primary school. *RUDN Journal of Informatization in Education.* 2020;17(3):179–189. http://dx.doi.org/10.22363/2312-8631-2020-17-3-179-189

**Bio notes:**
*Christer Sjöberg*, teacher at Strandskolan (Tyresö, Sweden). E-mail: larare.christer@gmail.com

*Tove Risberg*, teacher at Tyresö school (Tyresö, Sweden). E-mail: tove.risberg@utb.tyreso.se

*Jalal Nouri*, PhD in Computer Science, Associate Professor of the Department of Computer and Systems Sciences of the Stockholm University (Stockholm, Sweden). E-mail: jalal@dsv.su.se

*Eva Norén*, PhD in Mathematics Education, Associate Professor of the Department of Mathematics and Science Education of the Stockholm University (Stockholm, Sweden). E-mail: eva.noren@mnd.su.se

*Lechen Zhang*, PhD student of the Department of Computer and Systems Sciences of the Stockholm University (Stockholm, Sweden). E-mail: chen@dsv.su.se

DOI 10.22363/2312-8631-2020-17-3-179-189

Научная статья

# Программирование как инструмент для реализации межпредметных связей в обучении в начальной школе

**К. Шеберг[1], Т. Рисберг[2],**
**Дж. Нури[3], Е. Норен[3], Л. Чжан[3]**

[1]Школа «Strandskolan»
*Королевство Швеция, 135 63, Тюресё, Lagergrens väg, 14*
[2]Школа Тюресё
*Королевство Швеция, 135 61, Тюресё, Tyresövägen, 6*
[3]Стокгольмский университет
*Королевство Швеция, 114 19, Стокгольм, Frescativägen*

**Аннотация.** *Проблема и цель.* Вычислительное мышление было внедрено во многих странах мира, и учителя интенсивно работают над включением программной деятельности в учебный процесс. Однако они сталкиваются с рядом проблем из-за того, что до сих пор мало исследований проводится с упором на обучение программированию детей младшего возраста, а дидактика программирования является довольно новым явлением для образовательной системы К-9, поэтому учителя К-9 имеют мало подготовки в отношении программирования. В Швеции, например, программирование было введено не в качестве отдельной дисциплины, но внедрено в несколько предметов, что создает давление на учителей, поскольку им приходится использовать программирование как инструмент для преподавания и улучшения изучения различных предметов, таких как математика.

*Методология.* В статье сообщается об исследовании урока, проведенном в шестом классе начальной школы в Швеции с участием в общей сложности 155 учащихся. Целью урока было изучение языков визуального программирования, в частности Scratch, которые могут быть использованы для обучения вычислительному мышлению, математике и социальным наукам в междисциплинарном ключе.

*Результаты.* Представлены выводы относительно: 1) применения методологии изучения уроков для развития обучения программированию; 2) использования программирования в качестве инструмента междисциплинарного обучения математике и социальным наукам; 3) дидактических стратегий учителей.

*Заключение.* Данные исследования свидетельствуют о том, что междисциплинарный характер урока, который включал в себя цели обучения математике, обществознанию и программированию, весьма полезен. Ученики получили лучшее понимание учебного материала, рисуя, оцифровывая и анимируя свои идеи в Scratch.

**Ключевые слова:** вычислительное мышление, программирование, математика, обществознание, обучение, начальная школа

**История статьи:**
Дата поступления в редакцию: 10 марта 2020 г.
Дата принятия к печати: 13 апреля 2020 г.

**Для цитирования:**
*Sjöberg Ch., Risberg T., Nouri J., Norén E., Zhang L.* Programming as a tool for across-subjects learning in primary school // Вестник Российского университета дружбы народов. Серия: Информатизация образования. 2020. Т. 17. № 3. С. 179–189. http://dx.doi.org/10.22363/2312-8631-2020-17-3-179-189

**Сведения об авторах:**
*Шеберг Кристер*, учитель в школе «Strandskolan» (Тюресё, Швеция). E-mail: larare.christer@gmail.com

*Рисберг Туве*, учитель в школе Тюресё (Тюресё, Швеция). E-mail: tove.risberg@utb.tyreso.se

*Нури Джалал*, кандидат компьютерных наук, доцент кафедры компьютерных и системных наук Стокгольмского университета (Стокгольм, Швеция). E-mail: jalal@dsv.su.se

*Норен Ева*, кандидат математических наук, доцент кафедры математики и естественно-научного образования Стокгольмского университета (Стокгольм, Швеция). E-mail: eva.noren@mnd.su.se

*Чжан Лечен*, аспирант кафедры компьютерных и системных наук Стокгольмского университета (Стокгольм, Швеция). E-mail: chen@dsv.su.se