POSSIBILITIES OF DIAGNOSING THE LEVEL OF DEVELOPMENT OF STUDENTS' COMPUTATIONAL THINKING AND THE INFLUENCE OF ALTERNATIVE METHODS OF TEACHING MATHEMATICS ON THEIR RESULTS

[a]BRYNDOVÁ LUCIE, [b]KVĚTOSLAV BÁRTEK, [c]KLEMENT MILAN

Palacký University Olomouc Faculty of Education, Žižkovo nám. 5 779 00 Olomouc
email: [a]lucie.bryndova01@upol.cz, [b]kvetoslav.bartek@upol.cz, [c]milan.klement@upol.cz

Published within the frame of the University Palacky Faculty of Education grant Implications for the development of computational thinking and computational concepts in primary school pupils with the applied Hejny method of teaching mathematics, GFD_PdF_2023_01.

Abstract: The testing of students' computational thinking and the development of standardized tools for this testing is one of the most debated issues in the practical integration of computational thinking development. Thus, for more than a decade, there have been initiatives aimed at identifying the algorithmic, programming, and information thinking skills of primary and secondary school students.
The research, the progress, and results of which are the subject of the communication of this paper, has been our contribution to the development of testing tools that would allow for the widespread testing of the level of students' computational thinking, and that are not focused on the use of a specific programming language. As part of it, we were also able to identify a possible link between alternative methods of teaching mathematics, such as the Hejny method, and the deeper development of computational thinking in primary school pupils.
Keywords: Computational thinking, diagnostics, didactic test, test tasks, Hejny method

1 Introduction

Accelerating technological development has brought many radical changes in all aspects of life and has undoubtedly affected the functioning of our society in recent decades. The expanding reach of the digital space and technological innovations leading to the modernization of industry, commerce and households have given rise to a plethora of new concepts related to digital and information technologies and their applications. One of these was computational thinking, which was introduced by Jeannette Wing in 2006 as an essential skill for modern humans who are able to make full use of digital technologies and computational methods to solve everyday problems.

According to Wing, computational thinking is a thought process that enables one to formulate a problem and describe its solution in such a way that it can be effectively handled by a computer, machine or even a human (Wing, 2006). In general, therefore, it is a way of solving a problem that focuses on describing, analyzing and finding an effective way to solve it, emphasizing a systematic approach and the use of concepts known in the field of computer science. It is important to emphasize that the development of computational thinking does imply the development of programming skills exclusively. On the contrary, the concept of computational thinking suggests that everyone, not only the professional computer scientists, can use its related competences and skills. Thus, it contributes to the holistic development of students or students, with overlap into the development of their informatics competences. Since the first introduction of the concept of computational thinking (CT), there have been many international discussions about its precise definition, the specification of its dimensions, and the efforts to integrate the development of CT into the curriculum of educational systems all over the world. The introduction of the concept of computational thinking into academic debate has fostered a pedagogical discourse on the role of digital technologies in education and the possibilities of introducing computer science and programming into national curricula, which has existed almost since the early 2000s (Tran, 2017; Klement, 2018).

Although computer literacy and the targeted development of digital and communication skills are still of considerable importance; there is a tendency to move the targeted development of these skills into the cross-curricular domain as part of the modernization of the whole education sector (Balanskat, 2018).

Since the beginning of the international debate on integrating the development of computational thinking into the curriculum, numerous attempts have been made to define specific subdomains of computer science. The primary goal of this process is to specify an otherwise very general definition of the phenomenon of computational thinking, which is not suitable for the practical implementation of CT in the school system. Currently, most national curriculum definitions of the concept of computational thinking are based on or frameworks that align with the 2011 CSTA and ISTE definitions of the characteristics and competencies associated with CT use. These definitions were later simplified by many authors, and reduced to basic elements that summarize the original definitions in their essential principle.

Even for pedagogical and educational purposes, the concretization of the areas defining CT is usually done by a detailed analysis of the text of the CSTA and ISTE documents. In the following comparative Table 1, we list the subcomponents of computational thinking based on the CSTA and ISTE definitions and the key words and phrases used in this definition according to Chen (2017). Based on these listed baseline components we define the corresponding CT skills (Angeli et al., 2020, Bocconi et al., 2016, Wing 2014) that are associated with these concepts, and that computational thinking that students are expected to master.

Table 1 Definition of domains for the development of computational thinking

Original definitions of CSTA and ISTE	Keywords	Matching skill CT
Formulate problems for machine solutions	Formulation	Syntax, coding
Logically organize and analyze data	Data	Data processing
Represent data using abstractions	Representation	Modelling
Automate solutions using algorithmic thinking	Algorithmic thinking	Algorithmic thinking, automation
Analyzing possible solutions to achieve the most efficient combination	The most effective combinations	Abstraction, optimization
Generalize and apply a specific process to a solution problem	Generalization	Evaluation, debugging, generalisation

Therefore, continuous research on material conditions, analysis of educational content, forms and methods of teaching, as well as the readiness of teachers, including the necessary competences for the development of computational thinking in their pupils and students, is necessary. In Europe, the European Schoolnet project, for example, has been mapping the problems accompanying the introduction of CT development in schools. According to the results of this research, the most significant shortcomings are the lack of teacher qualifications (Balanskat, 2018) and the absence of the necessary diagnostic tools for identifying the level of computational thinking in students (Tikva & Tambouris, 2021). Thus, a number of research activities in this area can be noted in the definition of CT content (e.g., Brennan, 2012, Kanemune, 2017, Moller & Crick, 2018, etc.), methods of teaching CT (e.g., Rubio et al., 2015, So, Jong, & Liu, 2020, etc.), and forms of teaching CT (Román-Gonzáles et al., 2017, Tran, 2017, Tang et al. 2020, etc.). However, less research has focused on the area of developing tools for testing students' levels of computational thinking (e.g., Hadad et al., 2020; Klement et al. 2020; Denning, 2017; Brennan & Resnick, 2012, and others).

2 Opportunities To Test The Level Of Computational Thinking

Consequently, many researchers are currently trying to develop specific diagnostic tools that aim to directly test computational thinking development within students. The purpose of those tools is evaluation of both, the domestic state of development of computational thinking, and determining the position of the results of the state educational system at the international level. In Europe, the pioneer of standardized CT testing is Spain, where the question of measuring the level of CT development in primary school students has been discussed since 2015. Within the global discourse, the United States has come to the academic forefront on the issue of targeted testing.

Probably the most widespread tool of measurement computing skills through didactic testing, which is implemented in the form of a test combining closed and open questions, is a computing challenge Bebras. The challenge focusses on students' general computational skills; however, its main aim is to popularize and promote computer science rather than to diagnose it. In the context of a didactic test measuring computational thinking with open-ended questions, it is worth mentioning the Psychometric Computational Thinking Test, or PCT test, by Julio Santisteban and Jennifer Santisteban-Muñoz (2018). In addition to those, the first standardized test of computational thinking in Europe was the CT-test by Román-González in 2015. This test was aimed at Spanish primary school pupils working in the Scratch environment and was therefore linked to a specific programming environment that the students were used to working with. A similar approach to the level of CT development testing with the use of a specific programming environment was a testing tool constructed by Chen et al. in 2017. This test combined open and closed questions, was designed for fifth-grade students and primarily focused on the use of CT in practical activities of daily living.

Thus, globally, four types of evaluation of student development and performance are typically encountered when testing computational thinking in education, regardless of the level of education. Tang (2019) divides these categories according to the form of pupil work they work with, specifically the form of didactic test composed of open or closed questions, analysis of pupil portfolio evaluation, interview and survey. The following table elaborates the distribution of testing options and with examples of specific applications of the method:

Table 2 Possibilities of determining the level of computational thinking

Type		Practical use
Didactic tests	CT test with open questions	Román-González (2015) Santisteban (2018)
	CT test with closed questions	Dolgopolovas (2015)
Analysis of student work	Analysis of the student's portfolio	Román-González (2015)
	Evaluation of the student's performance in solving the task	Angeli (2020)
Interview	Usually a supplementary form of test	Gülbahar (2018)
Questionnaire	Determines the student's attitude towards CT and related skills	Sáez-López a kol. (2016)

The mentioned tools for testing the level of students' computational thinking, however, may not always be suitable for widespread use, as they are either closely tied to one environment (CT-test) or are not primarily intended for use in mainstream education and focus more on talented individuals (BEBRAS). Another shortcoming of the assessed testing tools appears in the context of open-ended answers, where functional solutions may conflict with poor syntax, making it difficult to evaluate the results. The student is able to arrive at a solution to the problem presented, but this solution, although correct, does not correspond syntactically to any formal programming language. It is important to note, therefore, that the concept of CT does not relate to the learner's ability to use specific digital and communication technologies. At present, the school system is set up in such a way that the school chooses its own programming language, and there are therefore no official guidelines for selecting this programming language. It is therefore impossible to determine with certainty which programming environment the tested pupils will be able to work in. The use of tasks based on existing environments may favor or disadvantage pupils depending on their experience. Therefore, when designing a test instrument aimed at a large number of respondents, it is necessary to consider the complexity of the task and to simplify terminology that may be unfamiliar to the students.

Taking into account the basis mentioned above, the aim of our research work was to develop a test instrument that would be able to measure the level of development of computational thinking in primary school pupils, to validate and standardize this test on a sufficiently large sample of primary school pupils, to process the obtained results statistically, and to prepare the test in a form suitable for widespread use in primary schools. The constructed test tasks can also serve as suggestions for methodological training of teachers, as well as a possible evaluation tool to determine the general level of development of computational thinking at the time of the general implementation of the new curriculum revision.

3 Design Of A Didactic Test As A Diagnostic Tool

Didactic tests are pedagogical diagnostic tools that are generally used to measure learning outcomes in schools in such a way that they can be subsequently evaluated and interpreted. The term didactic test has quite diverse definitions, depending on the author, but in general, it can be said that it is a test that is oriented to objectively determine the level of mastery of the curriculum in a certain group of people (Chráska, 2016). A quality didactic test must meet certain criteria and exhibit certain characteristics. Typical criteria that a didactic test must meet are validity, reliability and practicality, or objectivity, sensitivity, economy, etc. These criteria should already be considered in the design of the test itself, since the starting point of any test instrument design is, as a rule, the determination of the purpose of the test. The actual verification of the elementary properties of the instrument is then a matter of standardizing it in the context of validation, evaluation and interpretation according to certain predefined rules.

To assess the validity of the questions, the Ebel method was used; in which experts divide the questions into two groups according to importance and difficulty. Each question was rated in terms of importance on a four-point scale with decreasing relevance, i.e. essential, important, useful and irrelevant, with irrelevant questions being excluded from the set. The difficulty scale is a standard three-step scale of decreasing difficulty, i.e. from difficult before medium to easy. The set of questions was then constructed based on expert judgement of the level of difficulty from the easiest to the most difficult questions in sets of three questions.

3.1 Definition Of The Tested CT Framework And Its Dimensions

Due to the nature of computational thinking and the characteristics of its constructs, we anticipated that different test items would have different importance in relation to different CT domains (Abstraction and Decomposition, Algorithmic Thinking, Generalization and Optimization, Evaluation and Debugging, Syntax), while some items may relate to several different CT domains). For this reason, in the validation of the test items, we included the domain-specific determination, which according to the assessing experts was predominantly needed to solve the individual task. Determining the specific CT domain construct that the component measures is then a matter of determining the validity of the individual test items and the test itself. The internal structure-based validity assumes that each item has different importance in relation to the constructs and that some items are more related to another construct. Therefore, the items were weighted prior to final scoring, as we discuss in the diagnostic instrument validation methodology.

Therefore, for the design of the test, the creation of test items, their validation, standardization, and their formal processing, we drew on a search of the literature dealing with the creation and methodology of didactic tests (Black, 1998; Chráska, 2016; Ackerman, 2019 and others), and especially on the methodological procedure developed by G. Chen (2017). The set of test questions, based on the prior premises, was designed as a level test. The time limit was set so as not to imply an interruption for the vast majority of students. Therefore, the test questions were ordered from easiest to hardest. In this case, there is no statistical bias in the results as the slowest pupils generally do not perform better on level tests when the time is extended (Chráska, 2016).

Based on the set objectives, individual test tasks were designed that would be relevant and adequate to measure the required skills and knowledge in each CT domain. The individual tasks design was based on the intersection of the domestic definition with the internationally understood standard (Abstraction and Decomposition, Algorithmic Thinking, Generalization and Optimization, Evaluation and Debugging) and the component that is specific to the Czech primary school environment according to the National Institute for Education recommendations (Syntax):

- Abstraction and Decomposition - tasks will focus on the ability to simplify a problem to its basic form so that essential information is not lost, and then work with a diagrammatic representation of the problem,
- Algorithmic thinking - tasks will focus on the ability and skill to find an effective and efficient solution to a particular problem and to formulate the solution adequately, quite independently of practical programming,
- Generalization and Optimization - tasks will focus on the ability to break down the whole into sub-components and to work with these sub-components, for example by optimizing functions,
- Evaluation and Debugging - tasks will focus on the ability to analyze the problem, debug it to predict the outcome of the situation and the operation of the algorithm based on a critical analysis of the situation,
- Syntax - in the context of the education of pupils in primary schools in the Czech Republic, we include this dimension following the original definition of CT according to the National Institute of Education. We define it as the ability to write a solution using an adequate programming language or code, at a level appropriate to the age of the pupil, understanding the principle of this writing, and the procedure of the problem solution, compliance with the laws of computer programming, and the ability to rewrite the solution so that it can be understood by a computer or an adequate machine.

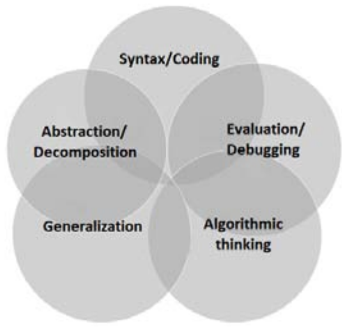

Figure 1 Final definition of dimensions of computational thinking for the purpose of testing the development of CT in primary schools in the Czech Republic

3.2 Design Of The Didactic Test And Its Items

In the development of the set of initial phases of the test items, a total of thirty test items of increasing difficulty were created, corresponding in equal proportion to the five predefined CT domains, from which a number of questions suitable to the age category of the tested pupils were selected after expert judgment.

Each test item was designed to be a closed-ended multiple-choice question. At the time of the beginning of the research, there was no validated computational thinking that corresponded to the legislative definition of the CT concept within the EU and the Czech Republic curriculum. For this reason, the tasks were principally designed on the basis of previous foreign research dealing with the creation of didactic tasks for the development of computational thinking, such as the CT-test by Román-González (2017) and the Beaver of Informatics contest. The content of the test items focused primarily on the areas of computational thinking that are defined by the Ministry of Education within its concept, especially from the revised Primary School Curriculum Framework (2022) and the recommendations of the European Commission, developed by CSTA/ISTE (2012).

The test items were designed so that they were not dependent on a specific programming language or environment and therefore allowed for widespread deployment in the classrooms without the need for specialized software (graphical assignments with the option of printing and manual completion). Each test item consisted of a complex task whose solution required the use of a particular dimension of computational thinking.

To avoid the problem of guessing the correct answers, each item of the test had a choice of four answers. Each question had to be answered within the test, so it was impossible to omit the answer. Each correct answer was scored with one point, and no points were deducted for a wrong answer. The final set of test items was sorted with increasing difficulty, with all CT dimensions of interest represented equally.

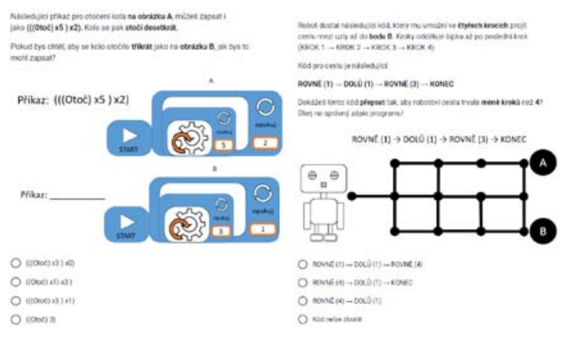

Figure 2 Selected test tasks for determining the level of development of computational thinking

Based on expert judgments regarding the difficulty and validity of the test items, we constructed a test set that consisted of 12 questions, which corresponded to the three main domains of the proposed CT theoretical framework; namely Abstraction and Decomposition, Algorithmic Thinking, and Syntax and Coding. For the questions, which were originally designed to determine the level of learners' development in the dimensions of Generalization, and Evaluation and Debugging, the required condition of the consensus of expert assessments regarding the individual task, which had to be over 70 %, was not met. Moreover, while the average agreement of the experts' assessments for the items focused on Syntax and Coding was 69 %, Abstraction and Decomposition corresponded to 68 %, and Algorithmic Thinking reached an average of 62 %, the average agreement of the Generalization and Optimization component reached only 32 % average agreement, and the Evaluation, Debugging and Solution Evaluation component only 32 %. Therefore, it can be suggested, that it is not entirely possible to identify exactly which dimension of CT is predominant within solution of those tasks. This tendency might also be supported by the fact that while evaluating the test items that were originally aimed on exploring the last two low-ranked CT dimensions, the experts did not identify their intended CT dimension as a dominant focus of those test items and gave more priority to Algorithmic Thinking, and Abstraction and Decomposition. For this reason, the Generalization and Optimization, and Evaluation and debugging dimensions were not considered as separate concepts in the final test but were included only as components of the other CT dimensions.

Table 3 Concordance of expert assessments in the validation of test items according to CT dimensions

Targeted CT dimension	Average consensus of expert assessments	Median consensus of expert assessments
Evaluation/Debugging	32 %	31 %
Generalization/Optimizatio	35 %	33 %
Algorithmic thinking	62 %	64 %
Abstraction/Decomposition	68 %	67 %
Syntax/Coding	69 %	71 %

4 Diagnostic Tool Validation Methodology

The validity of the test items was verified from the perspective of experts dealing with the didactics of computer science and the development of computational thinking, who had the opportunity to assess the validity of the test items, their clarity, difficulty, and appropriateness in relation to the age and context of the test. These experts were from independent departments. This phase of preparation included checks for concurrent validity (comparison with other tests measuring the same ability) and predictive validity (comparison with performance in practice). In this phase of the research, 22 expert practitioners were involved in the validation of the instrument and were able to assess the test item's relevance in the CT domain, its level of difficulty, and its appropriateness to the target population.

Determining the specific CT domain construct that the component measures, is a matter of determining the validity of the individual test items and the test itself. Internal structure-based validity assumes that each item has different importance in relation to the constructs and that some items are more related to another construct. Therefore, items will need to be weighted before final scoring.

Construct validity was further assessed using factor analysis, a technique that "groups observed variables (in our case, test items) into latent variables, (here, the related domain) based on common features in the data (e.g., Atkinson et al, 2011). There are two main approaches for factor analysis (exploratory and confirmatory). Confirmatory factor analysis (CFA) is used when there is an assumption about the underlying structure of the data and to confirm the structural model of the instrument (de Souza et al., 2019), while exploratory factor analysis is typically used to explore the dimensionality of the data. In our case, which aims to evaluate whether questions focused on each CT domain form coherent groups, we used CFA analysis.

The selected test items that passed expert review were further pilot tested on a selected sample of primary school students and then evaluated.

5 Diagnostic Tool Validation Results

In the course of validating the test instrument, the traits of respondent's gender (boys - girls), year (4th and 5th grade of primary school) and two different schools were monitored among others. The test results were subjected to cluster analysis, which divided the study population into three clusters. The criterion was the overall success rate in the test. The first cluster contained respondents with test scores between 3 and 6. This cluster had the lowest occupancy, namely 10 respondents. The mean score in this cluster was $\bar{x}$=5.3; SD=1.059. The other two clusters had similar frequencies. The cluster with scores ranging from 7-8 had 41 respondents, $\bar{x}$=7.56; SD=0.502 and the last cluster with scores ranging from 9-12 had 43 respondents, $\bar{x}$=9.9; SD=0.867.

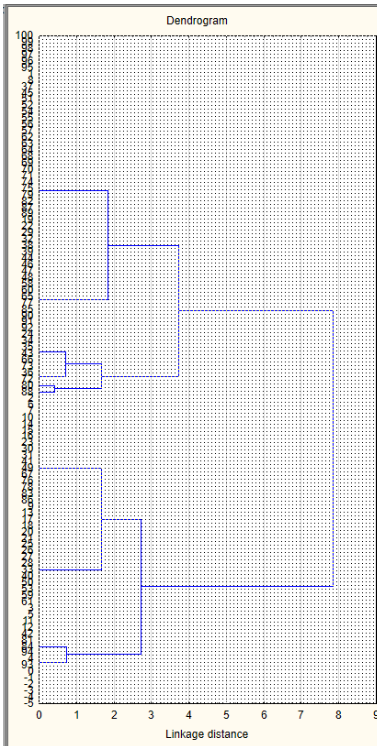

Figure 3 Cluster analysis: results of pupils from both schools

We looked at which variables might affect the partitioning of the data into clusters. First, we used a two-sample Student's t-test of the agreement of means to test whether differences in test scores were influenced by respondents' gender. Forty-four girls and 50 boys participated in the test.

We established the following hypotheses:

- Null hypothesis: H0: $\mu 1 = \mu 2$; i.e., there are no statistically significant differences between girls' and boys' performances.
- Alternative hypothesis: HA: $\mu 1 \neq \mu 2$; i.e., there are statistically significant differences between girls' and boys' performances.

From the above table (p = 0.058), it is clear that the null hypothesis cannot be rejected. Therefore, the result is statistically insignificant at the level of $\alpha = 0.05$.

Table 4 T-test results for the “gender” variable

	Variable	points
T-tests; grouping: B=0; G=1 (Group 1=0; group 2=1)	Mean0	8.750000
	Mean1	8.080000
	t-value	1.913691
	df	92
	p	0.058769
	ValidN 0	44
	ValidN 1	50
	SD0	1.366118
	SD1	1.936123
	F-ratio Variances	2.008580
	P Variances	0.021398

Next, we tested whether the differences between the test scores were statistically significant when comparing students in grades 4 and 5. The test was administered to 51 4th grade students and 43 5th grade students.

We established the following hypotheses:

- Null hypothesis: H0: $\mu 1 = \mu 2$; i.e., there are no statistically significant differences between the performance of students in grades 4 and 5.
- Alternative hypothesis: HA: $\mu 1 \neq \mu 2$; i.e., there are statistically significant differences between the performance of students in grades 4 and 5.

Table 5 T-test results for the “grade” variable

	Variable	points
T-tests; grouping: 4th=0; 5th=1 (Group 1=1; group 2=0)	Mean0	8.837209
	Mean1	8.019608
	t-value	2.354369
	df	92
	p	0.020681
	ValidN 0	43
	ValidN 1	51
	SD0	1.675172
	SD1	1.679169
	F-ratio Variances	1.004777
	P Variances	0.994092

From the above table (p = 0.020681), it is clear that we reject the null hypothesis and accept the alternative hypothesis. Thus, the result is statistically significant at α= 0.05 level.

We were not surprised by this result; it was expected that there would be the expected difference in performance affected by the age of the pupils.

The final feature that we believe may have influenced the differences in performance on the submitted test was the school that the students attended. The test was taken by 31 pupils from School 1 and 63 from School 2.

We set the following hypotheses:

- Null hypothesis: H0: $\mu 1 = \mu 2$; that is, there are no statistically significant differences between the performance of the students of School 1 and School 2.
- Alternative Hypothesis: HA: $\mu 1 \neq \mu 2$; that is, there are statistically significant differences between the performances of the pupils of School 1 and School 2.

Table 6 T-test results for the “method of mathematics teaching” variable

	Variable	points
T-tests; grouping: H=0; C=1 (Group 1=0; group 2=1)	Mean0	9.064516
	Mean1	8.063492
	t-value	2.748125
	df	92
	p	0.007213
	ValidN 0	31
	ValidN 1	63
	SD0	1.388896
	SD1	1.776849
	F-ratio Variances	1.636673
	P Variances	0.141141

From the above table (p = 0.007213), it is clear that we reject the null hypothesis and accept the alternative hypothesis. Thus, the result is statistically significant at α= 0.05 level. This result has already surprised us more. Therefore, we studied the above variables in more detail. In the next graph, the differences between the results of the studied groups are evident.

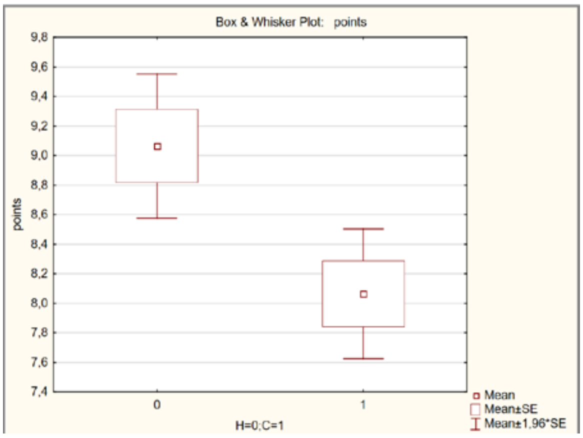

Figure 4 Box Plot comparing success rates in the test

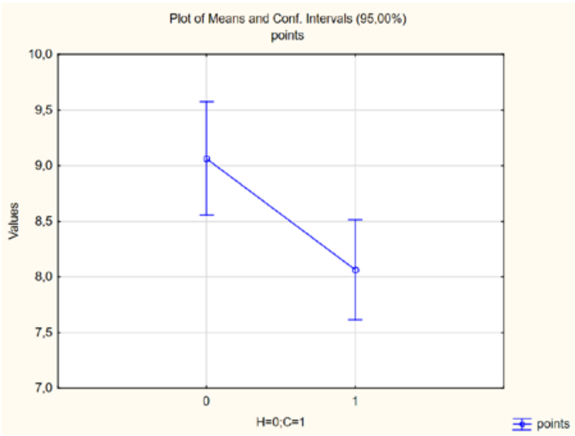

Figure 5 Box Plot comparing means and confidence itervals

We searched for the causes of this phenomenon. The influence of the teacher on students' performance on the observed test offers a logical possibility. However, in this case, we did not develop this reasoning further; the respondents in both sets of observations had received instruction with one, i.e., identical, teacher. We therefore asked what further might account for these differences, focusing on science instruction for computer science and other supporting science. In the course of the investigation, we found that in one of the schools studied, mathematics was taught using an alternative method - Hejny method.

6 Discussion

On the basis of the research results described above, a possible link between teaching mathematics using the Hejny method and the deeper development of computational thinking in primary school pupils was revealed (Bryndová, 2021). In the developed test of computational thinking, pupils who are taught using the Hejny method achieve statistically conclusive better results than students with classical computer science teaching. This increased development is also observed in parallel classes in terms of specific abilities and skills associated with the CT concept. The currently tested sample of pupils shows better skills in algorithmizing (6% better than the parallel sample of pupils taught with classical mathematics), abstraction (5.8% better) and syntax (15% better).

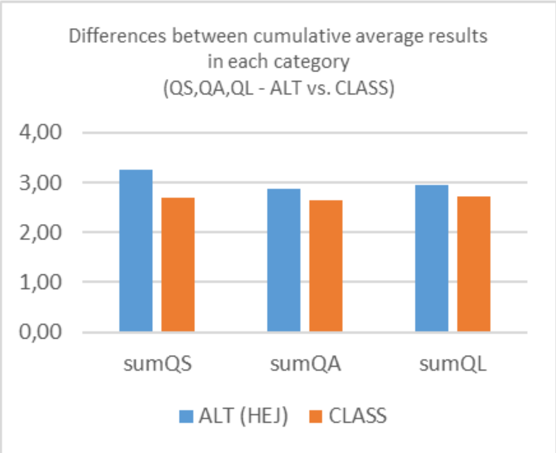

Figure 6 Differences between cumulative average results in each category (alternative teaching vs. classical teaching)

The data also suggest a greater development of computational thinking in pupils taught with HM compared to the global sample tested (about 8%). However, the sample mentioned above is currently very small (for the school with combined mathematics teaching, $n^{HM}=31$; $n^{KM}=63$, so these data have not yet been published. The data are visualized in the following graphs for the complete relevant set of questions (12 test items) with increasing difficulty.

Figure 7 Differences between the averages of the results in each question (ALT vs. CLASS) – blue line= Hejny method; red line = classical teaching of mathematics

Q1, Q4, Q7, Q10 – Questions – Logical thinking
Q2, Q5, Q8, Q11 – Questions - Syntax
Q3, Q6, Q9, Q12 – Questions – Algorithmization

7 Conclusion

The development of computational thinking and the modern teaching of informatics in primary schools is currently, in many ways, the subject of extensive discourse worldwide (Li et al., 2021; Bryndová, Klement, 2021; Tripon, 2022 and others). The implemented revision of the informatics curriculum has universally introduced fundamental changes in the concept of teaching informatics in primary schools and introduced a new educational area of Informatics. The primary purpose of this revision is the development of computational thinking, i.e., a set of certain computational skills, qualities and attitudes to ensure that graduates of primary education understand the basic principles of digital and information technologies, and possibly further development in this area.

Thus, at present, many researchers are trying to develop specific diagnostic tools that would be aimed directly at testing computational thinking and would allow evaluating both the domestic state of development of computational thinking and determining the position of the results of the state educational system at the international level. Our contribution in the field of developing testing tools that would allow for the widespread testing of the level of students' computational thinking and not be focused on the use of a specific programming language was the research whose progress and results are the subject of the communication of this paper.

Based on the findings, a possible link between teaching mathematics using the Hejny method and a deeper development of computational thinking in primary school pupils was revealed (Bryndova, 2021). In the developed test of computational thinking, pupils who are taught using the Hejny method achieve statistically conclusive better results than pupils with classical computer science teaching. These results suggest that there is a potentially neglected area of cross-curricular development of computational thinking in the current school system. The interdisciplinary development of computational thinking is directly supported by its original conception, which defines it as a modern problem-solving competency using practices and methods that are primarily prominent in computer science and computing but offer applications beyond them (Wing, 2014).
While these results cannot be considered entirely significant, given the size of the research sample, they do suggest a possible direction for further research efforts by the authors' collective. These may focus on research into the impact of alternative methods of mathematics education, such as the Heine method, on the development of knowledge in specific CT domains (decomposition, debugging, abstraction, (data) modelling and algorithmizing).

Literature:

1. Ackerman, T. A. A *Didactic Explanation of Item Bias, Item Impact, and Item Validity from a Multidimensional Perspective.* Journal of Educational Measurement, 2019. 29(1), pp. 67-91.
2. Angeli, Ch. & Nicos V. *Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy.* Computers in Human Behaviour. 2020. DOI: 10.1016/j.chb.2019.03.018. ISSN 07475632.
3. Atkinson, T. M, Rosenfeld, B. D., Sit, L., Mendoza, T. R., Fruscione, M., Lavene, D., et al. *Using confirmatory factor Analysis to evaluate construct validity of the brief pain inventory (BPI). J Pain Symptom Manage*, 2011. 41(3), pp. 558–565.
4. Balanskat A., Engelhardt K. & Licht A.H. *Strategies to include computational thinking in school curricula in Norway and Sweden- European Schoolnet's 2018 Study Visit.* European Schoolnet, Brussels. 2018.
5. Black, P. J. *Testing, friend or foe: the theory and practice of assessment and testing.* Washington: Falmer Press, 1998.
6. Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A. & Engelhardt, K. *Developing computational thinking in compulsory education* – Implications for policy and practice. 2016.

7. Brennan, K. & Resnick, M. *New frameworks for studying and assessing the development of computational thinking.* In Proceedings of the 2012 annual meeting of the American Educational Research Association, 2012.
8. Brennan, K., & Resnick, M. *New frameworks for studying and assessing the development of computational thinking.* In Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canada, 2012.
9. Bryndová, L. & Klement, M. (2021). Educational Robotics as a Means of Developing Pupils' and Students' Computational Thinking. In: *Ad Alta – Journal of Interdisciplinary research*. Magmanitas, 2021, Volume 11, Issue 2, pp. 325–329.
10. Bryndová, L. *The Possibilities Of Developing Computational Thinking In Primary Education.* Trends in Education. 2021, 13(2), pp. 5-12.
11. Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. *Assessing elementary students' Computational Thinking in everyday reasoning and robotics programming*. Computers & Education, 2017. 109, 162–175.
12. Chráska, M. *Metody pedagogického výzkumu: Základy kvantitativního výzkumu.* Praha: Grada Publishing, 2016.
13. CSTA & ISTE *Operational Definition of Computational Thinking for K–12 Education*, 2011.
14. de Souza, A. A., Barcelos, T. S., Munoz, R., Villarroel, R., & Silva, L. A. *Data mining framework to analyze the evolution of computational thinking skills in game building workshops*. IEEE Access, 7, 2019. pp. 82848–82866.
15. Denning, P. J. *Remaining trouble spots with computational thinking. Communications of the ACM*, 2017. 60(6), 33–39.
16. Downes, S. *Connectivism and Connective Knowledge: Essays on meaning and learning networks* Toronto: National Research Council Canada. 2012. 616 p
17. Hadad, R., Thomas, K., Kachovska, M., & Yin, Y. (2020). *Practicing formative assessment for computational thinking in making environments*. Journal of Science Education and Technology, 29(1), 162–173.
18. Hejný, M., Kuřina, F. *Dítě, škola a matematika. Konstruktivistické přístupy k matematice.* Praha: Portál. 2009. 232 p.
19. Kanemune, S., Shizuka, S. & Tani, S. *Informatics and Programming Education at Primary and Secondary Schools in Japan. Olympiads In Informatics*, 2017. 11(1), 143-150. DOI: 10.15388/ioi.2017.11. ISSN 18227732.
20. Klement, M. *Traditional topics for the framework educational programme focused on ICT area, and the perception of these topics by the primary school ninth grade pupils.* Journal of Technology and Information Education, 2018. 10(1), 43-62.
21. Klement, M., Dragon, T. & Bryndová, L. *Computational Thinking and How to Develop it in the Educational Process.* 1. ed., Olomouc, Publishing UP, 2020. 216 p.
22. Li, Y., Schoenfeld, A.H., diSessa, A.A., Graesser, A.C., Benson, L.C., English, L.D., & Duschl, R.A. (2020). On Computational Thinking and STEM Education. *Journal for STEM Education Research*.
23. Moller, F. & Crick, T. *A university-based model for supporting computer science curriculum reform.* Journal of Computers in Education. 2018. 5(4), 415-434. DOI: 10.1007/s40692-018-0117-x. ISSN 2197-9987.
24. Román-Gonzáles, M., Pérez-González, J.-C., & Jiménez-Fernández, C. *Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test.* Computers in Human Behavior, 2017. 72, 678-691.
25. Rubio, M. A., Romero-Zaliz, R., Mañoso, C., & de Madrid, A. P. *Closing the gender gap in an introductory programming course*. Computers & Education, 2015. 82, 409-420.
26. So, H., Jong, M.S. & Liu, C. Computational Thinking Education in the Asian Pacific Region. Asia-Pacific Edu Res 29, 2020. pp. 1–8.
27. Tang, X., Yue Y., Lin, Q., Hadad, R. & a Zhai, X. *Assessing computational thinking: A systematic review of empirical studies.* Computers & Education, 148. 2020. DOI: 10.1016/j.compedu.2019.103798. ISSN 03601315.
28. Tikva, C., & Tambouris, E. *Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review*. Computers & Education, vol. 162, 2021. pp. 104-113.
29. Tran, Y. *Computational Thinking Equity in Elementary Classrooms: What Third-Grade Students Know and Can Do.* Journal of Educational Computing Research. 2017. 57(1), 3-31.
30. Tripon, C. (2022). Supporting Future Teachers to Promote Computational Thinking Skills in Teaching STEM – A Case Study. *Sustainability*.
31. Wing, J. M. *Computational thinking.* Communications of the ACM, 49(3), 2006. s. 33-35.
32. Wing, J. M. *Computational thinking benefit society*. Social Issues in Computing blog. 2014.

Primary Paper Section: A

Secondary Paper Section: AM