opción

Revista de Antropología, Ciencias de la Comunicación y de la Información, Filosofía, Lingüística y Semiótica, Problemas del Desarrollo, la Ciencia y la Tecnología

Año 34, 2018, Especial N° 16

Revista de Ciencias Humanas y Sociales

ISSN 1012-1587/ ISSNe: 2477-9385

Depósito Legal pp 198402ZU45

Universidad del Zulia

Facultad Experimental de Ciencias

Departamento de Ciencias Humanas

Maracaibo - Venezuela

Opción, Año 34, Especial No.16 (2018): 311-335
ISSN 1012-1587/ISSNe: 2477-9385

Research into attitudes of subjects of education process in teaching programming

Alma Abylkassymova
Kazakh National Pedagogical University after Abay,
rector@kaznpu.kz

Manargul Mukasheva
National Academy of Education named after Y. Altynsarin,
info@nao.kz

Zaure Zhumagulova
Center for Secondary Education, National Academy of Education named after I. Altynsarin
nao_2011@mail.ru

Abstract

The article presents some outcomes of probing into attitudes and needs of subjects of the education process in teaching programming at school via a set of complementary research methods like theoretical, comparative and questionnaire survey methods. As a result, the majority of female respondents favored the programming skills exactly in terms of brain building. Teachers and IT (Information Technology) specialists would regard the Scratch facility as the most suitable tool to teach programming at primary school, Python as applicable for middle school, and C as applicable for the senior level of a secondary school for designing software.

Key words: Education Process, School, Subjects, Teaching.

Recibido: 04-12--2017 •Aceptado: 10-03-2018

Investigación de actitudes de sujetos del proceso educativo en programación docente

Resumen

El artículo presenta algunos resultados del sondeo de las actitudes y necesidades de los sujetos del proceso educativo en la enseñanza de la programación en la escuela a través de un conjunto de métodos de investigación complementarios como los métodos teóricos, comparativos y de encuesta por cuestionario. Como resultado, la mayoría de las encuestadas favorecieron las habilidades de programación exactamente en términos de desarrollo del cerebro. Los maestros y los especialistas en TI (Tecnologías de la información) considerarán que la instalación de Scratch es la herramienta más adecuada para enseñar programación en la escuela primaria, Python como aplicable para la escuela intermedia y C como aplicable para el nivel superior de una escuela secundaria para diseñar software.

Palabras clave: Proceso educativo, Escuela, Asignaturas, Docencia.

1. INTRODUCTION

Scientific, technological, engineering and math disciplines (STEM-education) are nowadays worldwide observed to be wedged in secondary school education programs. Technology driven future calls for a school training course (informatics) related pressing issues to be discussed with school teaching programming in particular. Research and methodology reference materials keep addressing the need of teaching kids on basic grasp of programming with IT scholars, teachers and specialists involved. Back in the 80s of the last century, one of the founders of the artificial intelligence theory, the Logo author Seymour Papert in his book Mindstorms: Children, Computers, and Powerful Ideas claims: “And learning languages is one of the things children do best. Every normal child learns to talk. Why then should a child not learn to talk to a computer?” (Papert, 1980: 18). Today’s information era realities suggest school education content with science and natural math education needs to be revised. This is why programming has recently been listed among school subjects referred to in school curriculum.

For example, since 2014 the programming basics have been taught in the UK. Aided by MIT’s Scratch, Kodu, Logo software the Britain’s primary school attendees learn to piece together basic programs by blocks. By the time they enter middle school students are expected to get hold of a basic grasp of the fundamental algorithmic structures and use them when developing simple applications. The Finnish Koodi 2016 project and training plans developed in Australia, France, Estonia and South Korea also advocate training kids on programming basics starting out at primary school. It is worth noting that the trends of early school teaching programming are observed by startup projects as well as the majors, the likes of the LEGO Group, Microsoft and others. Not only do these companies furnish comprehensible programming tools, they completely and utterly support the idea of launching programming at schools. A vast majority of those using MIT’s Scratch, Codecademy, Code.org and other resources is indicative of youth taking a growing interest in gaining knowledge and appreciating the arts of programming.

As is well recognized, trainees being keen about programming, parents' attitude towards the training course, not to mention teachers' and IT experts' vision are all equally important in contributing to school programming shaping up. School programming training with primary school, among other things, therefore, require integrated studies on the stance and the needs of the process parties concerned, namely, students and their parents, teachers and developers of training resources (Zamaletdinova et al., 2018).

2. METHODOLOGY

2.1. School Programming, Training and Development

The human intellectual development being affected by programming as a subject domain was first ever mentioned in the latter part of the last century. Regarding programming as the construction art one of the great made a note whereby programming is about wide-ranged and multifaceted activities quite often calling for sophisticated intellectual efforts (Wirth, 1978). According to Wirth (1978), evolving from a craft trade to an academic discipline, programming paved the way for a future school concept has brought forward by (Papert, 1980). The concept is stemming from one of the background factors being a model of teaching without a syllabus, implying aiding a child to build up its own intellectual world through utilizing surrounding materials. He would scribble: "In this model, educational intervention means changing the culture, planting new constructive elements in it and eliminating noxious ones. This is a more ambitious undertaking than introducing a curriculum change, but one which is feasible under conditions now emerging" (Papert, 1980: 18). Nowadays, many popular programming tools transformed the idea of a child as a builder of its own intellectual world into reality, laying opportunity even for primary schoolchildren to imagine and see their world of intellectual development as aided by rudimentary programming. It should be noted that following this concept, the approaches to shape up and hone physical skills in the training process also differ from the conventional one keep trying - one day you will just get it. Papert (1980) concept strategy is about making it accessible and obvious even for children that picking physical skills is largely ongoing as a scientific theory build-up. This approach is based on an idea of certain knowledge being not acquired through action or some way, but rather being adopted intuition-aided and steered up, when the world is learned using words – characters (schemes) under the certain theory of (Bruner, 1973).

The cognitive theory related power of symbolism and descriptive languages functions are well expressed and predict human intellectual development. Further, to supporting Bruner's (1973) theory, Third World of K. Popper highlights a pivotal role of descriptive language in human culture (Popper, 1984). There are also background factors about universal and ubiquitous modern technologies and culture facilitating a universal language emerging for the whole world. A new language will possibly be identical with math aiming to avoid ambiguous interpretations and deliver symbols close to real events unfolded in the material world. Such a descriptive language is anticipated to ultimately be developed by artificial intelligence to further be constantly brought up to date in line with existing and new situations worldwide (Jacque, 2002).

On the other hand, the history of science proves that many technological breakthroughs were based on formal descriptive languages. For instance, symbolic techniques of descriptive languages prompted the development of math sections, the likes of analytical geometry, the theory of numbers and others. Given that those and other factors, namely, training technical facilities boosting word-symbol (scheme) capacities and computer models allowing to explain in a plain language complicated events and processes, Papert (1980) reckoned that formal languages of programming could also prove to be computer controls as well as programming languages as new and productive descriptive languages of brainwork. He claimed that the developed descriptive formal systems might as well describe or clarify on real processes and forecast behavioral aspects of the objects thereof, thereby transforming into a flexible and efficient tool to acquire physical skills. His idea was further upheld by other researchers. A Soviet-era scholar, a member of the Soviet Academy of Sciences Ershov would state:

> …ten years ago Professor of Massachusetts Institute of Technologies Seymour Papert, one of those psychologists and pedagogues who first employed the programming concept, was very convincing in a series of his works claiming that a child is only taught to do something after the child gets to the bottom of things being done. Only after such think-through understanding is worked out, a repeated training session reaches success. It is worth noting: not only does it concern programs representing chains of logical reactions to in-a-priori known stimuli, but also programs of

real behavior, including all kinds of motor skills (going in for sports, music, playing games etc. **(**1980: 19).

His report on Programming as Second Literacy at the 1981 Lausanne hosted the 3rd World Conference of International Federation on Information Processing and UNESCO on Training Computer Application was regarded as hands-on motto inspiring soviet school informatics training for many years to come. Academician Ershov et al. (1985) made a note of major information processing laws, the transition from acquiring knowledge to taking action, the capability to tailor and customize programs and address related issues, forecasting outcomes thereof in the progressive development of human intelligence. He proposed to consider those as fundamental components of general education along with math and linguistic concepts. In other words, in pitching out Programming as a second literacy the scholar assumed, as time goes by, digital technologies would inevitably have an enormous effect on human intellectual development, education content, the philosophy of theory and practice of training.

2.2. Programming and Computational Thinking

Teaching programming at school became a more pressing issue with modern scientific concept – computational thinking – coming to the fore. Every year brings in more clarity on the idea pitched by Professor Wing from Carnegie Mellon University (CMU) (Wing, 2006) about the computational thinking evolving from the philosophical concept phase to reality, which will form an integral part of mankind at the threshold of Fourth Industrial Revolution. He noted that most of people with many parents counting treat computer science as computational thinking in their short range narrow view. In the scholar's opinion, the computational thinking hits the spot, i.e. it is about far reaching guidelines for teachers, researchers and subject matter experts in informatics, aimed at amending society's image in a digital era with a secondary school, teachers, parents and students as stakeholders. It is worth noting that active discussion of computational thinking involving scientific circles, numerous studies by scientists in a variety of countries, dedicated to various aspects thereof suggest of the complexity of the computational thinking concept. The background factors to the effect that algorithmic thinking may differ from the arithmetical one in terms of dependence on the dynamic status of the computer process, structure and quantity of data, parallel implementation of processes, were delivered in an article by a programming classicist (Donald, 1981).

None-the-less, studies regard computational thinking as a more accentuated thinking process encompassing algorithmic and parallel thinking, which stir up other intellectual processes, the likes of structural reasoning, pattern thinking, procedural and recourse intelligence. Researchers would not rule out intersections between computer based and process based, logical and system based reasoning, regarding those as closely intertwined processes, while stressing out that it should not be matched with either algorithmic and/or arithmetical certainty based thinking. Neither should it be identified with either computer literacy or information competence. Algorithmic, logical, system based and information intelligence hooking up with computational thinking would not exhaust it. The majority of those researching into the phenomenon of computational thinking highlight the specific significance and role of programming in its development since the computational thinking concept surfacing was mainly facilitated by computer science, namely, programming. In the meantime, some researchers claim that computer intelligence should not necessarily be linked with computers or programming, which needs to be studied (CS Unplugged, Bell et al., 2009). Another researcher, probing into a more extended interpretation of the computer thinking, composed of the five key components presented in a report on computational thinking delivered by the National Research Council, arrived at the following conclusion (Koh, 2014):

– Computational thinking is not associated with the use of specific programs or programming languages

– Computational thinking is not a computer science, but rather a part thereof

– Computational thinking resulted from the natural evolution of our insight into informatics.

With reference to the above stated, when delivering the above referenced report on computational thinking, the National Research Council repeatedly noted that lack of consensus on the content and structure of computational thinking makes it difficult for its agreed definition to be adopted and to facilitate promotion of this fundamental analytical skill a XXI century representative is distinguished with.

In our view, it is too hard to identify as to which human activity or a field of science the computational thinking is applicable to a more or lesser extent. It might as well be unrealistic. Nevertheless, it needs to be taken on board that uninterrupted teaching programming at school, higher education training center or on a self-learning basis would to a larger extent help to shape up skills of processing abstract data, modeling various processes, planning and automating action items, distributing and paralleling processes, and developing computer intelligence at large. The applicable methodology (data abstraction principles): information process – information process models – object of information process – characteristics, methods and events of object, grounded on the principles of encapsulation, polymorphism and heredity – object behavior in other situations in teaching programming is purposefully engaged in facilitating the computer intelligence development. Moreover, expertise regarding abstract structures of data bulk and relations in-between, as well as automated skills of projecting and programing employed when tackling applied challenges might as well be used as a benchmark for pursuing objectives in other activities. Thus, we deem teaching programming at school to be one of the important and key phases of the computer intelligence development, which is a 'must-acquire' item on the agenda of every single member of a digital century society.

2.3. Issues of Teaching Programming at School

Today, with numerous accessible and fool-proof devices and tool coming in to facilitate training programming for different age groups, it is more often discussed about widespread mandatory teaching programming at a secondary school, including primary level. Furthermore, programming is regarded as the most effective tool making studies more interesting, faster, more accurate and simpler, and knowledge and skills acquired in the training process are deemed scientific, robust and encapsulated. However, according to the practice, there are a number of objective and subjective causes getting in the way of the widespread teaching programming. The survey involving students attending universities and colleges identify as such causes complexity of programming, lack of early teaching programming at school, unified approach to building up content and picking teaching tool etc. (Mihci & Ozdener, 2017).

Implementing programming as a syllabus in the former Soviet Union schools would not reap much benefit. A school subject referred to as Basics of Information and Computer Science would be taught at senior classes of a secondary school under a motto Programming as Second Literacy and an informatics textbook would basically be compiled of topics on algorithm development and a basic grasp of programming. Training materials on these topics presented in original textbooks on informatics would be distinguished with a high level of authenticity, scientific insistence on high standards and would cover a wide range of math expertise (Basics of Information and Computer Science, 1985-1986). The Section Algorithms and Briefing on Programming were mainly content-built based on math problems. The basic defined term of training course, referred to as algorithm was presented based on example of figuring values of Euclid’s function and algorithm. In order to walk students through the defined term interim values there would be used a quadratic equation solution algorithm, and with reference to formal running of algorithms, there would be used a step-by-step algorithm development targeted at finding a bisecting point of segment by means of compasses and ruler. Based on learning about those and other math algorithms, students would be expected to develop skills of carrying out new intellectual operations, to learn to plan and precisely describe actions of execution parties, have a basic grasp about evidencing correctness of algorithm, and also pick primary skills of a system-based (oriented) analysis. All these actions would have been understood to mean algorithmic thinking or algorithmic culture. Upon being briefed on the basics of algorithm development, one would be expected to master one of the high level programming languages. It was anticipated that studying algorithms and their functions, algorithm running operation principles, implementing algorithms using one programming language, students would to a certain extent gain knowledge and skills on programming, prompting development of algorithmic thinking and algorithmic culture with school attendees. Thus, it was supposed to introduce teaching programming at school. Afterwards, when it was observed for a few years that school leavers who got enrolled at higher education training centers were spotted to be poorly aware of programming with a number of factors greatly impacting school teaching programming uncovered, namely, poor logistics and software and resources provision, with the balance heavily tipping in favor of math in delivering the programing training course, language proficiency gap, lack of motivation to teach programing on the part of the teachers of informatics (Mukasheva & Zhilbayev, 2016).

The majority of the above referenced researches reviewed by us suggest of the expediency of teaching programming as a priority trend to be observed in terms of teaching the science of informatics at school.

Moreover, there are quite a number of issues still up in the air on the subject matter with one of those concerning probing into the attitude and requirements brought up by the very subjects of the education process as applicable to the school teaching programming at school. With society taking a greater interest in programming as an applied science discipline with special heed being paid to teaching programming at school, there was initiated a search for various approaches to tackle challenges related to syllabuses, training resources, teaching a subject by school teachers and others. Many government and non-government education organizations engaged in delivering programming are keen about getting feedback with training course attendees. Furthermore, a few countries engaged in conducting thorough surveys on use of information technologies and training courses on informatics at a variety of training centers, which revealed that a bulk of training courses resulted in attendees' expectations and needs being not met. To improve the situation, several new curricula were therefore brought forward (Mironova, et al., 2017).

Studies and review of the current status of teaching programming at senior level (15-16 aged school attendees) conducted at local schools established that one of the main reasons of poor performance on programming is a low level of satisfaction of the attendees with the programing learning process, including a syllabus, a learning course content, learning methods and formats. In 2017, I. Altynsarin National Academy of Education and Kazakh national pedagogical university after Abay launched yet another research into the attitude toward school teaching programming involving school attendees as well as other subjects of education process, namely, parents, teachers, let alone IT experts in the light of teaching programming employing a variety of hardware and software. It was therefore no less important for our research to take on board opinions offered by IT specialists on the problem under study, not to mention those of developers of digital education resources. The research tools were grouped on the following four categories: school attendees, parents, teachers and IT specialists with consideration being given to age, line of business and work experience/service record of respondents.

To identify the attitudes and needs of school attendees, parents, teachers and IT specialists in school teaching programming, various directions were selected:

School Attendees

- Motives stimulating willingness to learn to program: prestigious trade, highly paid job, intellectual work, high level awareness of IT, development of computational thinking and planning

- Level of mastering software (or a digital device, including smartphones)

- Preference in picking platforms and tools to develop applications

- Preference in picking ways to learn to program: attending school classes, independently, aided by others, for example, a tutor, a relative, a friend, an acquaintance programmer.

Parents

- How much time does a child routinely spend on a digital device (PC, notebook, smartphone, i-pad), including surfing net on school assignments, attraction / recreational / fun sites, playing games, social networking etc.

- How does a child use its smart phone (or a cellular) in search of information to handle school assignments (for instance, to handle homework)

- How urgent is the need to teach a child on the basic grasp of programming at school

- How do you assess local school readiness to teach programming to children

- Do you support an idea of teaching programming at primary school

- Would that be an accurate statement that teaching programming from early age (6-7 year olds) positively impacts the development of a child's computer (algorithmic, mathematical) reasoning

- Would that be correct to state that teaching programming may help a child develop such qualities as being disciplined and responsible for its actions.

Teachers, IT Experts

- Vision and perspectives of widespread implementation of programing as a regular syllabus at a secondary school

- What are prospects of early teaching basics of programming at school (starting with teaching 6-7 year olds)

- Does teaching programming at school facilitate the development of a child's general information culture (how to adequately use a digital device, how to behave at a virtual world, how to protect yourself at information space, how to adequately use sources of information etc.)

- Does programming help to develop such skills as computational thinking, being self-disciplined and responsible for one's own actions

- Would that be accurate to state that English language proficiency facilitates the learning of programming

- Would that be accurate to state that to master programming one will be needing sound knowledge on math

- Would that be accurate to state that to the programming education process a teacher's readiness is more important than availability of teaching facilities (for example, computer, software).

3. RESULTS

The research involved 43 senior school attendees aged 15-16, 20 parents, 22 teachers of secondary schools and 12 specialists in the development of software, digital education resources and system administration operating in education organizations.

3.1. School Attendees

The schoolchildren involved in the survey were taught programming under a school training course on informatics within 4 plus years and got hold of a basic grasp of algorithm and programming language, as well as types of data, basic algorithmic structures, the likes of if-the-else, do –while.

When questioned would you like to learn to program? The school attendees would answer Yes – 62.8% (27), Most likely Yes rather than No – 25.6% (11), Most likely No rather than Yes – 7% (3), No – 4.6% (2). The chi-squared test figuring $\chi^2 = 37{,}27$ at a number of degrees of freedom $\nu=3$ confirmed that the school attendees made their choice and picked Yes deliberately:

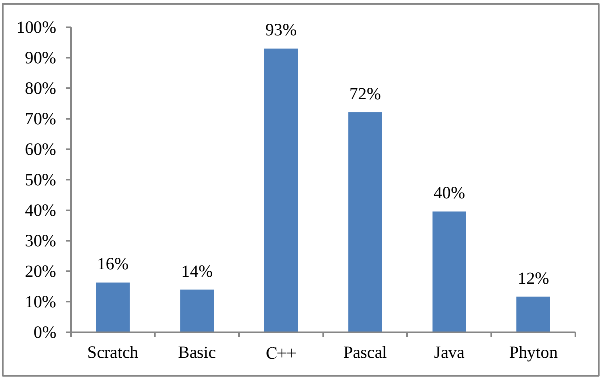
$$\chi 2 = \begin{cases} 7{,}815 \text{ for } P \le 0{,}05 \\ 11{,}345 \text{ for } P \le 0{,}01 \end{cases}$$

Such a choice could be supported by the schoolchildren’s answers to the question about what motivates them to study programming at school (see Table 1). The school attendees would pick one or several answers out of the seven options offered.

Table 1. Senior school attendees' opinions regarding what motivates them to study programming at school

Respondents (school attendees)	Number of respondents	You believe that the programming skills help you (you can pick several options):						
		Brain building	Make use of a digital facility	Adequately behave in a virtual world	Find information in the Internet and utilize it	Protect oneself against unwanted information	Design games	Make money
Respondents	43	74% (32)	44% (19)	12% (5)	14% (6)	9% (4)	37% (16)	44% (19)
Boys	33	70% (23)	45% (15)	12% (4)	18% (6)	12% (4)	39% (13)	52% (17)
Girls	10	90% (9)	40% (4)	10% (1)	0% (0)	0% (0)	30% (3)	20% (2)

In pursuance of assessing the schoolchildren's awareness of programming tools, they were asked a few questions. The schoolchildren's answers to the question Which of the below programming tools are you familiar with? (the schoolchildren would pick one or several answer options out of the 24 listed) suggested that C++ and Pascal (see Drawing 1) were preferable languages to be taught at schools.



Drawing 1. Bar chart with the schoolchildren's answers to the question What programming tool are you familiar with?

The school attendees also confirmed that 16% (7) of them only programmed in one programing environment, 47% (20) managed to work at two environments, 19% (8) in three, 12% (5) in four, only 7% (3) of the schoolchildren tried their hands at designing programs at 5 programming environments. Among those who tried to work at 4 plus programming environments, there were 7 boys and 1 girl.

3.2. Parents

With a view to identifying parent’s attitude toward the school teaching programming, there were questioned parents of 11-15 year old attendees of city and village schools. The questionnaire survey involved 10 parents of attendees of a city school and 10 parents of attendees of a village school. The parents’ age ranged from 22 to 54 years. The parents’ answers to some survey questions are presented in Table 2.

Table 2. Some outcomes on identifying parents’ attitude toward teaching programming at school

№	Questions	Schoolchildren’s parents’ answers								Total			
		City school				Village school							
		Yes	Most likely Yes than No	Most likely No than Yes	No	Yes	Most likely Yes than No	Most likely No than Yes	No	Yes	Most likely Yes than No	Most likely No than Yes	No
1	Is there a need to teach a child on basic grasp of programming at school?	100% (10)	0% (0)	0% (0)	0% (0)	30% (3)	50% (5)	10% (1)	10% (1)	65% (13)	25% (5)	5% (1)	5% (1)
2	Does teaching programing from early age (6-7 year olds) positively impact the development of a child’s computer (algorithmic, mathematical) intelligence?	40% (4)	60% (6)	0% (0)	0% (0)	60% (6)	40% (4)	0% (0)	0% (0)	50% (10)	50% (10)	0% (0)	0% (0)
3	Do you support the idea of teaching programming at primary school?	40% (4)	20% (2)	0% (0)	40% (4)	50% (5)	30% (3)	20% (2)	0% (0)	45% (9)	25% (5)	10% (2)	20% (4)

3.3. Teachers and IT Specialists

The teachers involved in the questionnaire survey would be engaged in teaching a variety of school subjects: math, informatics, biology, physics and humanitarian disciplines. The teachers' service record ranged from 5 to 21 years. Out of 12 IT specialists covered by the questionnaire survey, 2 would work as programmers, 7 – as system administrators at education organizations, the rest would be engaged in various service sectors. The answers of the teachers and IT specialists would largely be aligned with a significant difference of opinion on some issues through (see Table 3.)

Table 3. Answers of the teachers and IT specialists to some questions

Questions	Teachers (22)				IT Specialists (12)			
	Yes	Most likely Yes than No	Most likely No than Yes	No	Yes	Most likely Yes than No	Most likely No than Yes	No
Would you agree with the statement that English language proficiency facilitates the learning of programming?	41% (9)	27% (6)	27% (6)	5% (1)	67% (8)	25% (3)	8% (1)	0%
Would you agree with the statement that To master programming one will need sound knowledge on math?	64% (14)	27% (6)	5% (1)	5% (1)	33% (4)	42% (5)	17% (2)	8% (1)

4. DISCUSSION

Most of the attendees (74%) made a note that being capable of programming facilitates development of the computer intelligence. The questionnaire survey outcomes suggested that the majority of female respondents favored the programming skills exactly in terms of brain building. The survey participants also reckon that the ability to program helps make use of a digital facility (44%), make money (44%) and design games (37%). Moreover, school attendees would rather develop applications for personal computers (65%, 28) than mobile devices (44%, 19) and robots (23%, 10), and, when picking a platform for programming, they would prefer iOS (58%, 25) and Android (56%, 24). The parents' questionnaire survey outcomes aggregated with answers to other questions suggested of parent's taking a great interest in their children being taught programming at school. Opinions of the parents of city schools and those of village school vary to a considerable extent. The city based parents are 100% (10) convinced of the need to learn programming at school, whereas the village based parents doubted the need to learn programming at school. However, 30% (6) of all the parents, including 40% (4) of the city school attendees' parents, do not support the idea of teaching programming at primary school. The bulk of the school teachers involved in the survey accounting for 68.1 % (15) were also positive (answer Yes) about the need of teaching a child to the basic grasp of programming at school, 27.2% (6) agreed with the statement (answer Most likely Yes rather than No). Furthermore, 86.4% (19) of the responding teachers would uphold the idea of teaching programming at primary school. When questioned Would you agree with the statement that to the programming education process a teacher's readiness is more important than availability of teaching facilities (for example, computer, software)? The teachers would answer as follows: Yes –54.5% (12), Most likely Yes rather than No – 22.7 % (5), Most likely No rather than Yes – 9.1% (2) and No – 13.7% (3).

5. CONCLUSION

It is worth noting that the outcomes of our research into the attitudes and needs of subjects of the education process in teaching programming at school may state certain peculiarities of a region or a certain country. An education system and a school education process applicable in each country may vary, non-the-less, the key deliverables of our research backed up a positive trend observed on teaching programming at school. Aggregation of the above referenced data and other components suggested that the surveyed subjects of the education process to a large extent assumed mandatory programming training at school, including teaching programming at primary school. Teachers and IT specialists would regard the Scratch facility as the most suitable tool to teach programming at primary school, Python as applicable for middle school, and C as applicable for the senior level of a secondary school for designing software. The teachers' and IT specialists' attitude toward the content of school learning course on programming turned out to be ambiguous thereby upholding the assumption about securing its sustainability laying ahead as a great challenge in the light of rampant development of digital technologies.

REFERENCES

BELL, T., ALEXANDER, J., FREEMAN, I., GRIMLEY, M. 2009. **Computer Science Unplugged: school students doing real computing without computers**. https://www.researchgate.net. UK.

DONALD, E. 1981. **Algorithms in modern mathematics and computer science**, Lecture Notes in Computer Science 122, 82-89. Berlin. Germany.

ERSHOV, A. 1980. **Programming as Second Literacy**. http://ershov.iis.nsk.su/ru/second_literacy/article. Russia.

ERSHOV, A., MONAHOV, V., BESHENKOV, S., ERSHOVA, A., & MONAHOVA, V. 1985. **Informatics and Computer Science Fundamentals. Textbook for schools**. M.: Prosveschenie, P. 96. http://publ.lib.ru/ARCHIVES/E/ERSHOV_Andrey_Petrovich/_Ershov_A.P..html. Russia.

BRUNER, J. 1973. **Beyond the Information Given: Studies in the Psychology of Knowing by Jerome Bruner**, Jeremy M. Anglin (Editor). Publisher: W. W. Norton & Company. USA.

JACQUE, F. 2002. Th**e Best That Money Can't Buy: Beyond Politics, Poverty, & War Paperback**. Publisher: Global Cyber Visions. Romania.

KOH, K. 2014. **Computational Thinking Pattern Analysis: A Phenomenological Approach to Compute Computational Thinking**. Computer Science Graduate Theses & Dissertations. p. 86. http://scholar.colorado.edu/csci_gradetds/86. USA.

MIHCI, C., & OZDENER, N. 2017. **Teaching GUI-programming concepts to prospective K12 teachers: MIT app inventor as an alternative to text based languages**. International Journal of Research in Education and Science (IJRES), Vol. 3, N° 2: 543-559. DOI:10.21890/ijres.327912. Netherlands.

MIRONOVA, O., AMITAN, I., & VILIPÕLD, J. 2017. **Programming Basics for Beginners**. Experience of the Institute of Informatics at

Tallinn University of Technology .IJEP. Vol. 7, N° 4. http://www.i-jep.org. Greece.

MUKASHEVA, M., & ZHILBAYEV, Z. 2016. **Continuous and Ubiquitous Programming: Learning in Kazakhstani Schools**. Ubiquitous Learning: An International Journal. Vol. 9, N° 2: 13-27. Doi: 10.18848/1835-9795. USA.

PAPERT, S. 1980. **Mindstorms: Children, Computers, and Powerful Ideas**. 1st ed. Basic Books. NewYork: USA.

POPPER, K. 1984. **Evolutionary Epistemology, Evolutionary Theory: Paths into the Future.** Ed. by J. W. Pollard. John Wiley &. Sons, Vol. 10, pp. 239-255. Chi Chester and New York. USA.

WING, J. 2006. **Computational Thinking. Communications of the ACM**. Vol. 49. N° 3: 33–35. https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf. USA.

WIRTH, N. 1978. **Algorithms and Data structures**. Prentice Hall, ISBN-10: 0130220051, ISBN-13: 978-0130220059. USA.

ZAMALETDINOVA, G., KONOPLEVA, N., GLUZMAN, N., & GORBUNOVA, N. 2018. **Development of Electronic Educational Resources for Studying Mathematics by Adobe Flash and HTML5 Systems at Elementary Schools.** The Journal of Social Sciences Research, Vol. 4, pp. 171-174. India.

UNIVERSIDAD
DEL ZULIA

opción
Revista de Ciencias Humanas y Sociales

Año 34, Especial N° 16, 2018

Esta revista fue editada en formato digital por el personal de la Oficina de Publicaciones Científicas de la Facultad Experimental de Ciencias, Universidad del Zulia.
Maracaibo - Venezuela

www.luz.edu.ve

www.serbi.luz.edu.ve

produccioncientifica.luz.edu.ve