*2020, **9**(1): 37–52*
DOI: 10.2478/ijicte-2020-0004

THE PROCESS OF DEVELOPMENT AND EXPERT VALIDATION OF THE ATTITUDES SURVEY ON TEACHING OF PROGRAMMING IN COMPUTER SCIENCE LESSONS

Tomáš Hornik[1]

[1]Department of Applied Cybernetics, Faculty of Science, University of Hradec Kralove, Czech Republic { tomas.hornik@uhk.cz }

ABSTRACT

Current global trend of shifting teaching the introductory programming and algorithmization topics into the primary and lower secondary education sphere brings along a wide array of different programming languages, development environments, textbooks, courses and other educational materials. Even though curricular documents, such as Framework Education Programmes, ensure the type of content and the width of discussed topic, the selection of specific educational tools utilized in lessons can significantly influence pupils' attitudes toward the topic of programming and thus subsequently also the progress of the lessons. Present attitudes surveys regarding the topic of programming were identified in course of the literature review and after undergoing comparative analysis they serve as the foundation for creation of a new attitudes survey suitable for secondary school pupils. This paper describes the process of selection, modification and creation of items for a new questionnaire as well as the process of internal validity evaluation based on the input from the panel of experts composed of selected methodological specialists, university lecturers and teachers of informatics with long-time experience with teaching the topic of programming. Pilot testing on selected elementary school is going to commence at the beginning of school year 2020/2021. Based on the results the items will be modified for the last time before final data collection scheduled till the end of the school year.

KEYWORDS

Programming, education at elementary schools, educational programming languages, attitudes survey, attitudes questionnaire, creation of questionnaire, validation process.

1 INTRODUCTION

Establishment of teaching of introductory programming on elementary schools is no longer just a current trend, but with regard to educational curricular documents worldwide (Národní ústav pro vzdělávání, 2018; Štátny pedagogický ústav, 2014; The White House, 2016; Kemp, 2014, p.6) we can state it has become a standard and an integral part of Computer Science subject. By means of various specialized educational tools, such as traditional robotic toy Bee-Bot, it is possible to build the most basic foundations already in the course of pre-primary education (Pekárová, 2008). When addressing the subject from the Czech point of view, terms *Computer Science* and *Informatics* can be used interchangeably, since at the Czech elementary schools the most common designation is still *Informatics* even though the content is mostly overlapping with globally preferred Computer Science. Programming is one of the fundamental topics of

Computer Science and can span anything from a month up to the whole school year, depending on a given school policy and a level of importance the school puts on it.

At the secondary elementary school for pupils at the age of 11 to 15, which is Czech equivalent to ISCED 2, the options that are available for a teacher for this topic are vast. Essentially there are:

- online interactive courses (Hour of Code, Code Combat, Code Monster, Khan Academy, etc.);
- various educational children specific programming languages and their environments, which usually contain only limited introductory tutorial and then depend on the work and ideas of pupils or teachers (Scratch, Snap!, Swift Playgrounds, Kodu Game Lab, Alice, LOGO,...);
- textbooks and other educational materials utilizing aforementioned languages (where from the point of view of Czech education system currently the most relevant sources are results of nationwide project PRIM (Czech acronym for Support of the Development of Informational Thinking), which are freely the official website imysleni.cz);
- robotic educational tools (such as LEGO Mindstorms, LEGO WeDo, Ozobot, mBot, etc.), however, there are always further expenses necessary for the acquisition of selected robotic kit;
- professional full-fledged programming languages and their environments (e.g. JavaScript, Python, Java, Visual Basic, C#, etc.), which are largely not suitable to be employed at elementary schools as a part of programming lessons intended for general population.

Further information can be found in a university textbook *Programming didactics* (Hornik, Musílek, Milková, 2019), which was created within the framework of project PRIM as an introduction to specific selected programming courses, languages, and environments for students of pedagogical faculties with a major in Computer Science.

Curricular documents currently in effect for Czech teachers are Framework Education Programmes that came into effect in 2017 (NÚV). The specification of mandatory content for the subject Informatics is extremely vague and reduced compared to the other subjects, thus leaving the IT teachers almost completely open choice regarding the content of the lessons and selection of general areas of interest. The topic of programming is in this concept not mandatory and it is not included at all in school educational programmes of many schools. However, the current revision proposal of the Framework Education Programme in the Field of Computer Science and ICT does include the topic and defines corresponding framework of expected results (NÚV, 2018, p. 11 and 12). Thus it is possible to claim that when the proposal of curricular revision is accepted, minimal mandatory content of the lessons for the topic of programming will be same at all elementary schools and the nature of the topic itself dictates that all the educational resources are conceptually very similar – see e.g. the comparison conducted by Krejsa in his diploma thesis focused on teaching the foundations of programming in Scratch (2014, p. 35 and 36).

With regard to the fact that such a complex topic originally used to be taught at as a part of upper secondary education and at universities (ISCED 3 and higher), which are all selective schools, it is necessary to take into consideration complications caused by character of elementary schools, which are intended for general population. When the content of the lessons is factically similar it is worth considering their form, since attractiveness of selected materials from the pupils' point of view and user friendliness of the selected language and its environment can have positive influence on pupils' attitudes which further influence their behavior during lessons. The topic can be negatively accepted especially among the weaker pupils and consequently there is a high risk of disruptive behavior. Such a behavior resulting of these attitudes even from a single pupil has negative impact on the rest of the class and even though there are proven methods for effective classroom management (see e.g. Cangelosi, 2006), it still hinders the progress and depth to which the given topic can be introduced in the course of a lesson. Hence it is necessary to focus not only on the pupils' performance, but also their attitudes towards the topic.

Attitude is in Pedagogical dictionary (Průcha, Mareš and Walterová, 2003, p. 171) defined as "evaluative relationship maintained by an individual towards the surrounding world, other subjects, even oneself. It includes a disposition to behave or react in a given relatively stable way." Nevertheless, Albarracin et al. (2018) state that the definition of attitude must be sufficiently exhaustive while remaining universal and taking into consideration modern trends. In their research they claim that "what has been consistent in the multiple conceptualizations of the attitude construct is that evaluation is the key component" (p. 4), based on which they consequently simplify the whole definition to equation of attitude with evaluation.

2 RESEARCH DESIGN AND RESEARCH QUESTIONS

Basic tool for data gathering is questionnaire survey. Research design is based on two questionnaires with introductory programming lessons in-between, where the impact of the lessons on pupils' attitudes is measured by comparing results obtained in pre-test with the post-test answers. In order to be able to compare the attitudinal shift before and after the lessons, pupils are given a random code that they write in both pre-test and post-test questionnaires.

The introductory part of the questionnaire consists of only five questions that are gathering general information about the pupil, namely aforementioned random code, sex, age, class and school. In the post-test questionnaire there is a sixth question that identifies specific materials and languages used in the course of lessons. Even though attitudes can be influenced by a vast range of variables (e.g. personal interests, family environment, amount of free time, parents' professions, etc.) the goal of this questionnaire is not an attempt to detect and record all these influences, but an evaluation of the impact of undergone Informatics lessons in course of which the topic of programming has been presented. This impact is determined by examination and comparison of answers obtained before and after the programming lessons regardless the aforementioned personal variables of individual pupils.

The primary focus of the survey is to evaluate pupils' attitudes towards their programming lessons, i.e. answer the question whether the specific lessons in specific programming environment and language with given coursebook or tasks with their IT teacher was received positively or not and if not, indicate possible reasons. This is achieved by comparison of attitudes obtained from the questionnaire given to the pupils before their programming lessons with the answers from the second questionnaire after the lessons. Secondary research questions are focused on mapping the situation of pupils' attitudes towards the teaching of programming, specifically on evaluation of the influence of sex and age on the attitudes; on comparison of pupils' attitudes towards the topic of programming and the subject of Informatics in general; and on identification of specific aspects of the lessons that could significantly influence the attitudes of pupils towards the topic.

3 RESEARCH METHODS

Based on the literature review (Hornik, 2019) all the questionnaires published until 2019 focused on the topic of attitudes towards the teaching of programming were researched and analyzed. Every questionnaire in the conducted review used four or five-point Likert scales as their only method of attitudes evaluation, with the exception of attitudes evaluation tool created for the purpose of evaluation of programming lessons on high schools by Klement et al. (2012), which was based solely on dichotomous yes/no questions. Most of the items in our new questionnaire survey is based on existing surveys (see table 1) and as such they also utilize five-point Likert scale.

Table 1 Attitudes Questionnaires Focused on Programming Selected in the course of Literature Review

Author(s)	Publication year	Name of the research/questionnaire	Number of Attitudes Items	Respondents Sample size
Phillips & Brooks	2017	Impact on Attitudes and Self-Efficacy with CS	4	**8040** elementary and high school pupils
Asad, Tibi, & Raiyn	2016	Attitudes toward Learning Programming through Visual Interactive Environments	**29**	**24** elementary school pupils
Du, Wimmer & Rada	2016	Attitudes towards computer programming and knowledge of programming	**4**	**116** university students
Klement, Klement & Lavrinčík	2012	Metody realizace a hodnocení výuky základů programování	**12** (for students) **16** (for teachers)	**321** grammar school students and **12** teachers
Korkmaz & Altun	2014	Attitude Scale of Computer Programming Learning (ASCOPL)	**20**	**496** university students in first phase and **262** in the second
Tew, Dorn & Schneider	2012	Computing Attitudes Survey	**10**	**447** university students
Baser	2013	Programming Attitude Scale	**35**	**179** university students
Wiebe, Williams, Yang & Miller	2003	Computer Science Attitude Survey	**57**	**162** university students

Prior the new questionnaire initial evaluation by a panel of experts the possibility of employment of the semantic differential instead of the Likert scale was considered in order to simplify understanding of the items and to ensure pupils' answers unambiguity. Only two of the eight experts in their assessment mentioned that the use of Likert scale can be problematic and in both cases only in relation to a single item (see chapter 5) and subsequently the idea was abandoned. The selection of items regarding the attitudes was done in three consecutive steps described in detail in chapter 4.

Even though attitudes questionnaires usually contain same question in more than one phrasing (potentially supplemented by their negative formulation) in order to ensure that the respondent understands it and the concordance in answers supports internal reliability, there are substantial drawbacks to this approach especially when dealing with young respondents. Křeménková and Novotný (2016) state that in their study among 7th grade pupils apparently recurring items lead to frustration, anger and significant slowdown in finishing the questionnaire. Other problems discernible in the pupils' answers were focusing too much on detailed/literal accuracy of the question, inability to answer too general questions and a lack of understanding complex words combined with fear to ask for their explanation. Křeménková and Novotný (2016, p. 88) conclude that *"it is appropriate to apply some principles for questionnaire items when creating assessment tools for children and adolescents. These principles include use of 1) simple, specific and unambiguous questions/items that minimize the risk of multiple possible interpretations, 2) items with low demands on abstraction and generalization, 3) fewer semantically identical or similar items, and 4) minimal use of foreign terms. The verification of clarity of understanding the items by respondents shows also as necessary."*

The author of this paper has the same experience from previous research on secondary school pupils (Hornik, 2016) and from his four years of experience as a teacher on secondary elementary school. As such, the questionnaire is trying to avoid multiple occurrences of the same items and general ambiguity. However, in order to avoid random answers from pupils who do not read the items at all, we found inspiration in item 19 from Computing Attitudes Survey (Tew, Dorn & Schneider, 2012) and we added simplified version of it: *"This item verifies whether you are reading the questions. In order to prove it please select from the given options precisely the option four, which is I Agree."* If any of the pupils selects anything else (including option 5 – I completely agree), their whole questionnaire is discarded as biased data, without any information content, obtained by random clicking of the pupil.

4 THE PROCESS OF PRIMARY ITEMS SELECTION

The process of new questionnaire survey focused on pupils' attitudes towards programming was executed in three steps. In the first step, questions that appeared in more than one of these questionnaires were selected automatically. In the second step, the questions that appeared always just in a single questionnaire were evaluated and if considered meaningful for the purpose of this study and especially for the target audience (secondary school pupils), they were also selected. Because there is no validated and reliability tested attitudes questionnaire in the area of programming that could be used on secondary school pupils, the Computer Science Attitude Survey by Wiebe et al. (2003) was also included in this step as a possible base ground for question modifications. Even though the questionnaire is officially intended for the subject of Computer Science/Informatics, it was actually deployed as a part of a study exploring the benefits of pair programming (Williams et al., 2002) and its items are not only focused mostly on the topic of programming, but they are also simple enough to be used with secondary school pupils.

Selected questions that meaningfully complete our questionnaire, were modified according to a generally acknowledged process used e.g. by Baser (2013, p. 251) or Tew et al. (2012). The modification process consisted of elementary changes that focused generally oriented questions into more specific topic (word *"courses"* was replaced by word *"lessons,"* phrase *"computer science"* was replaced by *"programming,"* etc.) and if the question was too ambiguous and could be asked in a simpler and shorter way it was rephrased as such, even though there could be a slight informational loss or shift (e.g. *"Errors generated by computers are random, and when they happen there's not much I can do to understand why."* that appeared in the Computing Attitudes Survey by Tew et al. (2012) was shortened and rephrased to *"When the computer generates an error during programming, I don't know why and I don't understand it."*).

For the third step the overall composition of the questionnaire was considered while focusing on the research questions and data applicability. New items were created to fulfill the missing information concerning the pupils' attitudes toward the programming lessons and their content, including the specific programming language and its development environment.

4.1 Selection of Items from Established Questionnaires

Since five out of eight established questionnaires are focused on university students, most of the items were not suitable for secondary school pupils and thus the first selection step was concluded with only five items. The specific phrasing of questionnaire items was never exactly the same and the questions were compared based on their semantic content. The most repeated questions were about pupils' desire to continue learning about programming (Klement et al., 2012; Korkmaz & Altun, 2014; Tew et al., 2012; Baser, 2013) and to join a programming course (Du et al., 2016; Korkmaz & Altun, 2014; Baser, 2013). Remaining three questions appeared always in only two questionnaires. Following repeated questions were whether pupils consider the lessons difficult (Klement et al., 2012; Korkmaz & Altun, 2014), whether the pupils believe they can learn the topic (Phillips & Brooks, 2017; Korkmaz & Altun, 2014) and whether they enjoy solving challenging programming problems (Asad et al., 2016; Tew et al., 2012).

In the second step, questions were selected from individual questionnaires based on their suitability for secondary school pupils. All the criteria introduced by Křeménková and Novotný (2016) were observed in course of this selection process and only thirty-five items were considered satisfactory from the questionnaires focusing strictly on the topic of programming with additional forty-six questions from more general subject oriented Computer Science Attitude Survey (Wiebe et al., 2003). All the selected questions were further analyzed and questions that were too similar and thus could be confusing for pupils (even though their semantic content was different) as well as questions that were considered outside the scope of this research were omitted, leaving twenty-seven questions in the second step.

4.2 Addition of Completely new Items

The last step consisted of assessing missing information and subsequent creation of entirely new items. These items were necessary since none of the questionnaires was focused on pupils' evaluation of the selected programming language and its environment as well as the specifics of a given primary source of information (which could have been an online course, a textbook or a teacher). The new set of questions is focused on the pupils' impressions regarding not the general topic, but their specific lessons, because even the best language and the nicest development environment can be ruined by e.g. poor choice of tasks or incomprehensible explanations of new programming topics (and vice versa).

The impact of how the pupils perceive the connection of the topic with the real world is undeniable (see Arthur et al. (2018), which was focused on the same question in the area of mathematics) and an item about this perception regarding the general Computer Science was already among the questionnaires (Tew et al., 2012). However, an extra question had to be created anew for the particular topic of programming: *"The topic of programming is strongly connected with the everyday world around us."* which was based on the experts assessments changed to *"Programs are used in everyday life all around us, not only on PCs, laptops and mobile phones."* This is also the only new question that does not focus on details regarding the lessons pupils experienced.

Remaining new questions can be divided into two categories, namely questions evaluating pupils' attitudes toward the programming language and its environment and the questions aimed at their perception towards the source of information and tasks (be it their teacher, an online course or a textbook). The specification of the source of information is part of the introductory questions in the post-test. Questions focused strictly on the programming language and its environment were:

- *The work in the programming environment was without any problem.*
- *When I was looking for something in the programming environment, I usually found it immediately or very fast.*
- *I liked the look of the programming environment (the placement of everything, the look of buttons, etc.)*
- *I liked the pictures (e.g. backgrounds and sprites) in the programming environment.*
- *If I could, I would use completely different pictures for backgrounds, character sprites and things.*
- *I think it is possible to create anything I could think of in this programming language.*

These questions should identified possible sources of frustration and other problems that could be hindering the pupils' progress and attitudes, regardless the quality of given explanations or practice tasks, which is the aim of the second set of new questions:

- *Usually I understood explanations how everything works without any problems.*
- *There was always enough simple examples for a new topic.*
- *Textbook/materials were always easy to understand.*
- *In every task I always knew what is the goal.*
- *In some new topics I would like more tasks to try the new things.*
- *Usually I had enough time for my work.*
- *I very often needed help from my teacher.*
- *I was totally OK with the way my teacher handled the lessons.*

Even though the personality of teachers and their teaching styles have very strong influence on the pupils' attitudes toward the subject (Blazar and Kraft, 2017; Hashim et al. 2014), this questionnaire identifies possible problems related to the teacher in a single question. If there are such problems, they are not connected specifically with the topic of programming and if a teacher wants to identify and rectify them, there are other tools, as for example the Teacher Self-Assessment Tool (Teacher Leadership, 2017).

Five questions repeated in several questionnaires combined with twenty seven questions appearing in single questionnaires further complemented by additional fifteen new questions leads to the total number of forty-seven questions. Such amount of questions would make this questionnaire the second longest of them all, with only Computer Science Attitude Survey by Wiebe, Williams, Yang & Miller (2003) being longer with its fifty-seven questions. Even though the questions are based only on Likert scales, the questionnaire should be suitable for children in the age of 11 to 15, whereas especially the younger pupils wouldn't be able to cope with this extent. Preliminary selected forty-seven questions are meant to be reduced in the course of their evaluation by the panel of experts (see chapter 5) and during the pilot testing in selected focus groups (which is planned for September / October, 2020).

4.3 Preliminary Organization of Items Into Categories

Four of the eight questionnaires that emerged from the original literature review, utilize categories for organization of questions by their subject matter, although with the exception of Wiebe et al. (2003) and Baser (2013), the categories are always completely different (see table 2).

Table 2 Categories of Items Identified in Other Authors' Surveys

Questionnaire and its author(s)	Categories of questions
The Attitude Scale of Computer Programming Learning by Korkmaz & Altun (2014)	- Willingness - Negativity - Necessity
Attitudes toward Learning Programming through Visual Interactive Environments by Asad, Tibi, & Raiyn (2016)	- Motivation category - Competition category - Challenge category
Computer Science Attitude Survey by Wiebe, Williams, Yang & Miller (2003)	- Confidence in learning computer science and programming - Attitude toward success in computer science - Computer science as a male domain - Usefulness of computer science and programming - Effective motivation in computer science and programming
Programming Attitude Scale by Baser (2013)	- Confidence in learning computer programming - Usefulness of computer programming - Attitudes toward success in computer programming - Effective motivation in computer programming

Since none of the existing divisions of questions is suitable for the goals of the new questionnaire and do not fit the research questions, six new categories were identified based on our selected questions. The new questionnaire does not only focus on general attitudes (categories 1 and 2) and pupils inner motivation (category 5), but more importantly analyzes what could cause the pupils' attitudes shift in the course of school Computer Science lessons focused on the topic of programming (categories 3 and 4).

Despite the attempts of projects such as Hour of Code to equalize gender in programming (Du, 2019), women are still minority in this area of expertise. Study on one of the major programmers' forum Stack Overflow showed that as of 2018 only 6,3% of their users were females (Griffin, 2018). Four items were selected from Computer Science Attitude Survey by Wiebe et al. (2003) regarding the issue of gender perception (category 6) because of that. Categories 3 and 4 are only in post-test questionnaire.

1. The attitudes toward the Computer Science as a subject in general
2. The attitudes toward the topic of programming in general
3. The attitudes toward the specific programming language and environment
4. The attitudes toward the specific source of information (course/textbook/teacher)
5. Subjective personal aspects of the pupil
6. Gender prejudices with regard to the topic of programming

5 ASSESSMENT BY THE PANEL OF EXPERTS

Preliminary version of the whole questionnaire was sent to the panel of eight experts for evaluation of its internal validity and assessment of the exact phrasing of the questions. The panel of experts consisted of specialists focused on methodology testing and computer science didactics, as well as teachers with long term experience in teaching the topic of programming at elementary and high schools. The experts were approached before the beginning of school year 2020/2021. They were given the questionnaire in both Czech and English language mutation accompanied by six queries that should be considered for each of the questionnaire questions:

1. *Is general phrasing of the statement properly characterizing the point of the question, in other words, is the question clear?*
2. *Is the specific word choice suitable for secondary school pupils?*
3. *Does the statement ascertain thematically relevant information connected with programming?*
4. *Can the use of Likert scale with this statement be confusing for children?*
5. *Is the statement classified in the suitable category?*
6. *Should the statement remain or be completely discarded?*

The divisions of questions that are present in the preliminary version for experts' evaluation are omitted in the questionnaires for pupils. All the statements that were not accompanied by any kind of commentary from the experts were considered as without objections. One of the goals of the questionnaire modification was overall reduction of the survey, because two experts' assessments described the survey as far too long, which could increase the probability of implausible answers. Such a distortion can be caused by a wide variety of reasons, ranging from misunderstanding, inattention, boredom, fear up to deliberate prank-like reactions (Kohoutek & Mareš, 2012, p. 6). If the opportunity arose to reduce the length, it was in most cases taken. Following minute modifications in formulations and major changes were conducted based on the experts' assessments:

- Items *"The topic of programming interests me."* and *"From all the Computer Science topics, programming is the most interesting for me."* were labeled as duplicates with only difference in the level of stress put on the intensity of interest. Both items were merged and averaged to *"Programming is one of the most interesting Computer Science topics for me."*
- Similarly duplicate were items *"The work in the programming environment was without any problem."* and *"When I was looking for something in the programming environment, I usually found it immediately or very fast."* The core of both items is to determine how user friendly is the programming environment for pupils. Items were also merged and the second item was used as an example in brackets: *"The work in the programming environment was without any problem for me (e.g. when I was looking for something, I found it very quickly)."*
- It was pointed out that specific examples of subjects in brackets can strongly influence the pupils' answers in item *"Computer Science is useful in other subjects (e.g. biology, arts and crafts, foreign languages)."* Since it is also an item that can be modified to target behavioral aspect of attitudes, which were originally underrepresented, the item was changed to *"The things I learn in Computer Science I also use in other subjects."*
- The problem of pupils responding to questions from block B (see chapter 6) even before the programming lessons was brought to attention. Because the questionnaire did not incorporate an explanation of what programming is, the answers could have been unintentionally misdirective. Programming is a fundamental term and as such it is very difficult to define it accurately in a few sentences. At the beginning of the questionnaire an explanation taken from the Czech translation of a Python textbook was included: *"Computer Programming is the art of making a computer do what you want it to do. At the very simplest level it consists of issuing a sequence of commands to a computer to achieve an objective."* (Gauld, 2005) Instructions for teachers will also warn teachers they have to draw pupils' attention to the elementary explanation.
- Item *"I want to attend an after-school programming course."* was identified as misleading, because a positive answer could be limited by factual hindrances (e.g. there is no such course at the local school or its nearby surroundings or the pupils could be limited by the amount of their free time or their family financial situation). Two experts recommended simply use conditional mood *"I would like to attend an after-school programming course."*
- Two of the experts described the item *"I'm sure I can learn programming."* as difficult to answer on Likert scale. The intensity was lowered by rephrasing to *"I believe I can learn programming."* but the item was kept since it is a relevant indicator of so called self-efficacy (which is confidence of individuals in their own abilities) that has fundamental influence on acquisition of new knowledge, performance and behavior (Smetáčková a Vozková, 2016). This item will be pointed out in teachers' instruction manual and there will be an example for pupils of how to work with Likert scales.

- Item *"Everybody should learn how to program because it teaches you how to think."* was highlighted as unsuitable combination of two questions – should everybody learn to program and does programming teaches how to think? The answer to the second question was implied as a fact based on the formulation of the item. This problem was addressed by two experts who suggested a simple modification to *"People learn how to think logically by learning programming."* which was accepted and incorporated into the survey.
- Phrases *mostly, in most cases, usually* and *very often* were described as far too general, because every pupil has different notion of it and it would be better to not use them at all. These phrases were replaced by unified *usually.* However, it was decided to keep the word in all the items, because regardless the exact understanding of the word *usually*, it lowers their absoluteness, see e.g. *"I usually had enough time for my work."* compared to *"I had enough time for my work."* (where in the second case there seem to be implied *always*).
- Formulation "*I think it is possible to create anything I could imagine in this programming language."* did not include any examples and word *anything* is far too general and practically unrealistic at the same time. Based on the remarks of two experts the item was modified to *"In this programming language I can try programming without any restrictions."* The goal of this item is to find out whether pupils perceive the limitations while working in online courses such as Hour of Code.
- Based on the advice of two experts and despite the effort to lower the number of items, one item was added into the first part of the questionnaire regarding the subject Informatics *"The things we learn in Computer Science are interesting for me."* This item complements the information about the importance of the subject and thus allows to ascertain if the interest correlates with the subjective perception of the subjects' importance.
- Item *"The topic of programming is strongly connected with the everyday world around us."* was expanded to *"Programs are used in ordinary life every day, not only on PCs, laptops and mobile phones."* The point of the item is to find out whether pupils realize that programs are controlling even items and processes impacting every aspect of human life (e.g. traffic lights, cars, appliances,...)
- Item *"A big problem in learning programming is being able to memorize all the information I need to know."* was labeled by three experts as confusing and they all recommended to omit it. The questionnaire is intended for secondary school pupils, where the most common environment is based on visual blocks and pupils don't have to remember specific formulations of e.g. cycles or conditions such as *if...elif...else*. If the target respondents were high schools or technically oriented schools that utilize text based programming, rephrasing the item would be important. Here the item was removed.
- Item *"Generally I was not worried about attempting to solve computer programming problems."* which was intended for establishing if the pupils weren't afraid to try different approaches or to do major changes in their programs was deemed incomprehensible by two experts. Since the reformulation with further explanation would be too long and there already is an item about pupils' fears or worries, the item was eliminated altogether.
- Category of items preliminary labeled as *Subjective personal aspects of the pupil concerning the topic of programming* was based on the recommendation of three experts divided into two parts – one that can be answered even in pre-test and the other is only in the post-test.

- *Men and women* in the first gender oriented item were replaced by boys and girls. In order to not insinuate the prejudice that boys are better than girls at programming, genders were swapped in the item to *"Boys are as good as girls at programming."*
- Out of the four initial gender oriented items *"Women are certainly logical enough to do well in programming."* was eliminated. Remaining three gender oriented items were moved from their own category to category B that was renamed to *The attitudes and prejudices toward the topic of programming in general* and the separate category of gender prejudices was removed.
- Items *"I liked the pictures (e.g. backgrounds and sprites) in the programming environment."* and *"If I could, I would use completely different pictures for backgrounds, characters sprites and other things."* were pointed out by one expert as contradictory to the intention of the survey universality. Due to the prevalent gamification of the topic it is reasonable to expect that standard lessons (and thus not specialized programming oriented extracurricular courses and clubs) will utilize some form of visual block language or other graphical elements (such as Logo, which is a text based language focused on controlling turtles drawing geometrical patterns). Both items remained unchanged.

One of the experts raised a question whether to include also the answers from pupils with special educational needs, because the handicap of some pupils can significantly influence their answers. Pupils with individual study plans and other specific educational needs are still pupils of the school, teachers must work with them in regular lessons and the pupils have to be familiarized with the topic to a certain extent. Because these pupils are inseparable part of the school, exclude them would bias the data. Addition of an extra question to the introductory part of the questionnaire regarding the special needs was considered, but since it would confuse pupils without it, it was not included.

Two experts further pointed out the issue of setting the topic of programming in context of other topics that should be covered in the school year. Framework Education Programmes do not set the order of topics and because programming is often strongly gamified, it is potentially very entertaining and there is a high chance of employment of the topic at the beginning of the school year with the intention to motivate pupils. In that case it is impossible for pupils to compare programming with other topics and all items related to this issue lack the information value. For the purpose of the survey itself, optimal delivering of the topic is at the end of a school year. If it is not possible and the teachers want to discuss the topic earlier, it is recommended to outline all the topics for the school year briefly beforehand, which gives pupils at least limited perspective and enables substantially more objective answers.

Four experts independently strongly stressed the necessity of pilot testing, which was already arranged and at the time of the writing of this paper at the beginning of school year 2020/2021 it is in progress.

6 CURRENT VERSION OF THE SURVEY BEFORE PILOT TESTING

Based on the experts' evaluation five items were completely removed, one was added and twelve items were rephrased or otherwise modified. The scale of the questionnaire was thus decreased from original forty-seven items to forty-three items in the post-test. Pre-test is significantly shorter with only nineteen items. The author of this paper majored with honors in English language and the whole set of questions with regard to the correctness of their translations was checked by two more English language teachers and one Czech language teacher. Complete questionnaire in English version is in table 3.

Table 3 Unabridged Version of the Questionnaire Before Pilot Testing

A.) **The attitudes toward the Computer Science as a subject in general**

1. I like Computer Science.
2. Computer Science is an important subject.
3. The things we learn in Computer Science are interesting for me.
4. The things I learn in Computer Science I also use in other subjects.
5. I can see the things, I learn in Computer Science, being used in everyday life all around me.

B.) **The attitudes and prejudices toward the topic of programming in general**

6. Programming will be important for my future life.
7. Programming is one of the most interesting Computer Science topics for me.
8. I would like to attend an after-school programming course.
9. I worry that mistakes I make when writing a program may damage my computer.
10. People learn how to think logically by learning programming.
11. Programs are used in ordinary life every day, not only on PC, laptops and mobile phones.
12. Boys are as good as girls at programming.
13. It makes sense that there are more men than women in programming.
14. Women who enjoy programming are a bit peculiar.

C.) **The attitudes toward the specific programming language / environment** (only post-test)

15. The work in the programming environment was without any problem for me (e.g. when I was looking for something, I found it very quickly).
16. I liked the look of the programming environment (the placement of everything, the look of buttons, etc.)
17. I liked the pictures (e.g. backgrounds and sprites) in the programming environment.
18. If I could, I would use completely different pictures for backgrounds, character sprites and other things.
19. In this programming language I can try programming without any restrictions.
20. When the computer generates an error during programming, I don't know why and I don't understand it.

D.) **The attitudes toward the specific course/textbook/teacher** (only post-test)

21. I enjoyed the programming lessons.
22. Programming lessons were the worst Computer Science lessons.
23. Programming lessons were difficult for me.
24. Completing the lessons changed my opinion towards programming for the better.
25. I want to continue learning programming in our computer science/informatics lessons.
26. Even though I work hard, for some reason programming is unusually hard for me.
27. Usually I understood explanations how everything works without any problems.
28. There was always enough simple examples for a new topic.
29. Textbook/materials were always easy to understand.
30. In every task I always knew what is the goal.
31. In some new topics I would like more tasks to try the new things.
32. I usually had enough time for my work.
33. When working, I usually needed someone's help.
34. I was totally OK with the way my teacher handled the lessons.

E1.) **Subjective personal aspects of the pupil concerning the topic of programming**

35. I am afraid of programming lessons.
36. Programming is boring.
37. I believe I can learn programming.
38. I can get good grades in programming.

E2.) **Subjective personal aspects of the pupil concerning the topic of programming** (only in post-test)

39. I like to solve difficult programming problems that are challenging for me.
40. It would be really great to have an opportunity to take part in a programming competition.
41. When I am stuck, I do not spend more than five minutes before giving up or asking someone for help.
42. I do as little work in programming lessons as possible.

43. This item verifies whether you are reading the questions. In order to prove it please select from the given options precisely the option four, which is I Agree. *(This item is going to be inserted randomly somewhere in the second half of the questionnaire.)*

7 CONCLUSION

Modified version of the questionnaire is currently undergoing pilot testing in a selected group of pupils as the first topic in Computer Science lessons at the beginning of the new school year (September/October 2020) and their remarks are going to be considered for the last stage of questionnaire modification. As a part of the pilot testing, validity will be also evaluated by the measures of criterion group subjects, i.e. *"those who have been proved to possess the construct."* (Maurer, 1983 in Simonson & Maushak, 2001) As a part of this method pupils who are very keen and very negative are identified and their answers should account for both positive and negative extremes within the pilot group. The influence of individual items on overall reliability of the survey is going to be verified by Cronbach's alpha reliability test score.

Final version of the survey should not only bring a large amount of data usable for modification of existing educational materials and resources, but it should also enable an objective comparison of different programming languages and courses on pupils' attitudes. This should facilitate more effective choice of suitable educational tools for the teachers.

8 ACKNOWLEDGMENTS

The article has been supported by Specific Research Project of Faculty of Education, University of Hradec Kralove, 2020. The author would like to express gratitude to all the experts who were involved in the survey assessment as well as linguistic experts who verified the translation of the survey for its potential international use.

9 REFERENCES

Albarracin, D., Sunderrajan, A., Lohmann, S., Chan, S. & D. JIANG. (2018). The Psychology of Attitudes, Motivation, and Persuasion. *Handbook of Attitudes, Volume 1: Basic Principles*, 2nd ed. (D. Albarracin & B. T. Johnson, Eds.), 1-105. New York: Psychology Press.

Arain, M., Haque, M., Johal, L., Mathur, P., Nel, W., Rais, A., Sandhu, R., & D. Sharma. (2013). Maturation of the adolescent brain. *Neuropsychiatr Dis Treat*., *9,* 449-61. doi: 10.2147/NDT.S39776

Arthur, Y. D., Owusu, E. K., Asiedu-ado, S. & A. K. Arhin. (2018). Connecting Mathematics To Real Life Problems: A Teaching Quality That Improves Students' Mathematics Interest. *IOSR Journal of Research & Method in Education*. Vol. 8, No. 4, 2018, 65-71. e-ISSN: 2320–7388. https://www.iosrjournals.org/iosr-jrme/papers/Vol-8%20Issue-4/Version-2/J0804026571.pdf

Asad, K., Tibi, M. & J. Raiyn. (2016). Primary School Pupils' Attitudes toward Learning Programming through Visual Interactive Environments. *World Journal of Education*, October 2016, 6(5), 20-26. https://www.researchgate.net/publication/308975053

Baser, M. (2013) Attitude, Gender and Achievement in Computer Programming. *Middle-East Journal of Scientific Research,* 2013, 14, 248-255. https://files.eric.ed.gov/fulltext/ED542330.pdf

Blazar, D. & M. A. Kraft. (2017). Teacher and Teaching Effects on Students' Attitudes and Behaviors. *Educational Evaluation and Policy Analysis*. SAGE Publications, 2017 March, 39(1), 146-170. ISSN 1935-1062. doi.org/10.3102/0162373716670260

Cangelosi, J. S. (2006). *Strategie řízení třídy: jak získat a udržet spolupráci žáků při výuce*, 2nd ed. [Classroom Management Strategies: Gaining and Maintaining Students' Cooperation]. Praha: Portál, 2006. Pedagogická praxe (Portál). ISBN 80-7367-118-2.

Czech Statistical Office. (2016). *Zařazení českých vzdělávacích programů do Klasifikace vzdělání (CZ-ISCED 2011)*. [Placement of Czech educational programmes in the Classification of Education]. Praha: ČSÚ, 27.01.2016. Retrieved July 20, 2020, from https://www.czso.cz/csu/czso/ klasifikace_vzdelani_cz_isced_2011

Dorn, B. & A. E. Tew. (2015). Empirical validation and application of the computing attitudes survey. *Computer Science Education*, 2015, *25*, 1-36. tandfonline.com/doi/full/10.1080/ 08993408.2015.1014142

Du, J., Wimmer, H. & R. Rada. (2016). "Hour of Code": Can it Change Students' Attitudes toward Programming? *Journal of Information Technology Education: Innovations in Practice*, 2016, 15, 52-73. http://www.jite.org/documents/Vol15/JITEv15IIPp053-073Du1950.pdf

Dumontheil, I. (2014). Development of abstract thinking during childhood and adolescence: The role of rostrolateral prefrontal cortex. *Developmental Cognitive Neuroscience*. October 2014, *10*, 57-76.

Ford, M. & S. Venema. (2010). Assessing the success of an introductory programming course. *Journal of Information Technology Education: Research.* 2010, *9*(1), 133-145.

Gauld, A. & P. Přikryl. (2005). Co je to programování? [What is programming?] *Jak se naučit programovat*. 2nd ed. 2005. Retrieved August 22, 2020, from http://jaksenaucitprogramovat.py.cz/

Griffin, R. H. (2018). Stack Overflow 2018 survey: impact of age, gender, and sexuality on inclusion, interest in new tools, and ethics. *Kaggle: Your Machine Learning and Data Science Community*. Kaggle, © 2019, Jun 14, 2018. Retrieved August 15, 2020, from www.kaggle.com/heesoo37/stack-overflow-2018-survey-age-gender-sexuality

Hashim, N. M. H. N., Alam, S. S. & N. M. Yusoff. Relationship between Teacher's Personality, Monitoring, Learning Environment, and Students' EFL Performance. *GEMA Online® Journal of Language Studies*. February 2014, 14(1), 101-116. ISSN 1675-8021.

Hornik, T. (2016). *Možnosti rozvíjení algoritmického myšlení s využitím projektů Hour of Code a Scratch*. [The Possibilities of Development of Algorithmic Thinking with Projects Hour of Code and Scratch], Hradec Králové: Pedagogická fakulta Univerzity Hradec Králové. [Master's thesis]. 135 p. https://theses.cz/id/jf3lnn/19375424

Hornik, T., Musílek, M. & E. Milková. (2019). *Didaktika programování*. [Didactics of Programming]. Jihočeská univerzita v Českých Budějovicích: PRIM. Scriptum. imysleni.cz/images/vyukove_materialy/ UHK_Didaktika_programovani.pdf

Jihočeská univerzita v Českých Budějovicích. (2018). *Informatické myšlení*. [Computational Thinking]. České Budějovice: © 2018. https://imysleni.cz/

Kemp, P. (2014). *Computing in the National Curriculum: A Guide for Secondary Teachers*. Bedford: Newnorth Print, Ltd. www.computingatschool.org.uk/data/uploads/cas_secondary.pdf

Klement, M., Klement, J. & J. Lavrinčík. (2012). *Metody realizace a hodnocení výuky základů programování*. [Methods of Realization and Evaluation of Teaching of Programming Foundations]. Olomouc: 2012. https://www.researchgate.net/publication/280313585

Kohoutek, T. & J. Mareš. (2012). Anketa pro žáky: Anketa škole na míru. [Survey for pupils: Survey tailored for the school]. *Manuály evaluačních nástrojů*. Praha: Národní ústav pro vzdělávání, 2012, p. 36. ISBN 978-80-87652-12-1. Retrieved August 15, 2020, from http://www.nuv.cz/file/49/

Korkmaz, Ö., & H. Altun. (2014). A validity and reliability study of the Attitude Scale of Computer Programming Learning (ASCOPL). *Mevlana International Journal of Education*, 1 April, 2014, *4*(1), 30-43. doi: 10.13054/mije.13.73.4.1

Krejsa, J. (2014). *Výuka základů programování v prostředí Scratch*. [Education of basic programming in Scratch environment]. České Budějovice: Katedra informatiky Pedagogické fakulty Jihočeské univerzity v Českých Budějovicích. [Master's thesis]. 119 p. https://theses.cz/id/b5f11x/DP_Krejsa_Scratch.pdf

Křeménková, L., & J. S. Novotný. (2016). Úskalí tvorby dotazníků pro děti a dospívající: zkušenosti z ohniskových skupin. [Difficulties of children and adolescent questionnaires creation: experience from focus groups]. Psychologie a její kontexty 7 (2), 2016, 87-98.

Narmadha, U. & S. Chamundeswari. (2013). Attitude towards Learning of Science and Academic Achievement in Science among Students at the Secondary Level. *Journal of Sociological Research, 4*(2), 114-124.

Národní ústav pro vzdělávání: Metodický portál RVP. (2017). *RVP pro základní vzdělávání*. [Framework Education Programme for Basic Education]. Praha: Výzkumný ústav pedagogický v Praze, 2017-09-15, 166 p.

Národní ústav pro vzdělávání: Metodický portál RVP. (2018). *Návrh revize RVP v oblasti informatiky a ICT*. [Proposition of Framework Education Programme Revision in the field of Information and Communication Technologies]. Praha: Výzkumný ústav pedagogický v Praze, 2019-02-24, 20 p.

Pejcanovic, B., Holtzman, M., Wong, P. K. & G. Recketenwald. (2017). Assessing student preparedness for introductory engineering and programming courses. *Proceedings of 2017 IEEE Frontiers in Education Conference*, 1-5. doi: 10.1109/FIE.2017.8190539

Pekárová, J. (2008). Using a Programmable Toy at Preschool Age: Why and How? *Workshop Proceedings of SIMPAR 2008 - International Conference on Simulation, Modeling and Programming for Autonomous Robots*. Venice, Italy, 3-4 November 2008, 112-121. ISBN 978-88-95872-01-8.

Peluso, E. M., & G. Sprechini. (2012). The impact of Alice on the Attitudes of High School students Toward Computing. *Journal for Computing Teachers, 7*, 2-11.

Phillips, R. S., & B. Brooks. (2017). *The Hour of Code: Impact on Attitudes Towards and Self-Efficacy with Computer Science*. https://code.org/files/ HourOfCodeImpactStudy_Jan2017.pdf

Poskočilová, M. (2018) "Ajťáci" chybějí dvěma třetinám firem. ["IT Guys and Gals" are Missing in two thirds of Companies]. *Statistika & My: Měsíčník Českého statistického úřadu*. Praha: Český statistický úřad, 10/2018. Retrieved August 7, 2020, from http://www.statistikaamy.cz/2018/10/ajtaci-chybeji-dvema-tretinam-firem/

Průcha, J., Mareš, J. & E. Walterová. (2003). *Pedagogický slovník*, 4th updated ed. [Pedagogical Dictionary] Praha: Portál, 2003. ISBN 80-7178-772-8.

Shabbir, M. A., Asif, I. & M. A. Saeed. (2015). Students' Attitude towards Science and its Relationship with Achievement Score at Intermediate Level. *Journal of Elementary Education, 25*(2), 61-72. ISSN-Online: 2227-1090.

Simonson, M. & N. Maushak. (2001). The Handbook of Research for Educational Communications and Technology: Measuring Attitudes. *AECT: The Association for Educational Communications and Technology*. Bloomington, Indiana: Iowa State University, © 2001, Updated August 3, 2001, 994-1013. Retrieved August 16, 2020, from http://members.aect.org/edtech/ed1/34/34-05.html

Štátny pedagogický ústav (2014). *Inovovaný ŠVP pre 2. stupeň ZŠ – Informatika*. [Innovated School Education Programme for Secondary Schools - Informatics]. Bratislava: ŠPÚ, 2017.

Teacher Leadership: Teacher Self-Assessment Tool. Washington, DC: Center on Great Teachers & Leaders at American Institutes for Research, March 2017. Retrieved August 18, 2020, from https://gtlcenter.org/sites/ default/files/TeacherLeadership_TeacherSelf-Assessment.pdf

Tew, A. E., Dorn, B. & O. Schneider. (2012). Toward a validated computing attitudes survey. *Proceedings of the Ninth Annual International Conference on International Computing Education Research*, 135-142. https://dl.acm.org/ citation.cfm?id=2361303

The White House, Office of the Press Secretary. (2016). *FACT SHEET: President Obama Announces Computer Science For All Initiative*. Washington, DC: January 30, 2016. https://www.whitehouse.gov/sites/whitehouse.gov/files/images/FACT%20SHEET%2BPresident%20Obama%20Announces%20Computer%20Science%20For%20All%20Initiative_0.pdf

Watson, C. & F.W.B. Li. (2014). Failure rates in introductory programming revisited. *Proceedings of the 2014 conference on Innovation & technology in computer science education (ITiCSE '14)*. New York: Association for Computing Machinery, 39-44.

Wiebe, E., Williams, L., Yang, K. & C. Miller. (2003). Computer Science Attitude Survey. https://www.researchgate.net/publication/248503161_Computer_Science_Attitude_Survey

Williams, L., Wiebe, E., Yang, K., Ferzli, M. & C. Miller. (2002). In Support of Pair Programming in the Introductory Computer Science Course. *Computer Science Education*, *12*(3), 197-212.