ISSEP-2018

The 11th International Conference on Informatics *in Schools*

Saint-Petersburg

Electrotechnical University "LETI"

Saint-Petersburg, Russia

October 10-12, 2018

PROGRAM ISSEP-2018

First conference day	**Wednesday, October 10**	**Opening** **Section 1** **Section 2** **Cocktail Reception** **Hiking tours**	
8.00-8.30	*Registration*		
8.30-9.00	*Opening*		
	Section 1. Sustainable Education in Informatics for Pupils of All Ages. Role of Programming and Algorithmics		
9.00-9.30	№1	**Ivan Kalas,** Andrej Blaho and Milan Moravcik	Exploring Control in Early Computing Education
9.30-10.00	№2	**Jacqueline Staub,** Giovanni Serafini, Urs Hauser and Martina Forster	Autonomous Recovery from Syntactical Errors made by Primary School Children
10.00-10.30	№3	**Angélica Herrera Loyo**	Effects on the School Performance of Teaching Programming in Elementary and Secondary Schools
10.30-11.00	№4	**Jean-Philippe Pellet,** Gabriel Parriaux and Tristan Overney	A Case Study on the Effect of Using an Anchored-Discussion Forum in a Programming Course
11.00-11.30	*Coffee break*		
11.30-12.00	№5	**Sylvia Da Rosa** and Andrés Aguirre	Students Teach a Computer How to Play a Game
12.00-12.30	№6	**Panagiota Chatzipetrou**, Raquel Ouriques and Javier Gonzalez-Huerta	Approaching the Relative Estimation Concept with Planning Poker
12.30-13.00	№7	Lex Bijlsma, **Harrie Passier**, Harold Pootjes and Sylvia Stuurman	Integrated Test Development. An Integrated and Incremental Approach to Write Software of High Quality
13.00-13.30	№8	Tobias Kohn and **Dennis Komm**	Teaching Programming and Algorithmic Complexity with Tangible Machines
13.30-14.30	*Lunch*		

14.30-15.00	№9	**Noa Ragonis** and Ronit Shmallo	A Diagnostic Tool for Assessing Students' Perceptions and Misconceptions Regarding the Current Object "this"
15.00-15.30	№10	**Tatjana Jevsikova,** Valentina Dagienė and Vladimiras Dolgopolovas	On Preferences of Novice Software Engineering Students: Temperament Style and Attitudes Towards Programming Activities
	Section 2. National Concepts of Teaching Computer Science. Research on Informatics in Schools		
15.30-16.00	№11	**Arno Pasternak,** Lutz Hellmig and Gerhard Röhner	Standards for Higher Secondary Education for Computer Science in Germany
16.00-16.30	№12	Vreda Pieterse and **Estelle Taylor**	On the Triviality of the Assignment Statement *The report was shifted from section 6
16.30-17.00	№13	Luca Forlizzi, Michael Lodi, Violetta Lonati, Claudio Mirolo, Mattia Monga, Alberto Montresor, **Anna Morpurgo** and Enrico Nardelli	A Core Informatics Curriculum for Italian Compulsory Schools
17.00-17.30	№14	**Liudmila Bosova**	Comparative Analysis of the Content of School Course of Informatics in Russia and Subjects of the International Competition Bebras
17.30-18.00	№15	**Sixto Javier Castro**, Luis Jácome, Ana Gabriela Concha, Jordy Vásquez. Gustavo Londa, Luis Leonel Córdova, Boris Vintimilla, Cristina Abad, María Fernanda Miño and María De Lourdes Pilay	An Undergraduate Project Combining Computer Science and Arts: An Experience Report of a Multidisciplinary Capstone Design
18.00-19.00	*Cocktail Reception*		
19.00-22.00	*Hiking tours*		

Second conference day	**Thursday, October 11**	**Section 3** **Section 4** **Section 5** **Banquet**	
Section 3. Teacher Education in Informatics			
8.30-9.00	№1	**Noa Ragonis**	Computational Thinking: Constructing the Perceptions of Pre-service Teachers from Various Disciplines
9.00-9.30	№2	**Ebrahim Rahimi,** Ineke Henze, Felienne Hermans and Erik Barendsen	Investigating the Pedagogical Content Knowledge Development of Teachers Attending a MOOC on Scratch Programming
9.30-10.00	№3	Carlo Bellettini**, Violetta Lonati**, Dario Malchiodi, Mattia Monga and Anna Morpurgo	Informatics and Computational Thinking: a Teacher Professional Development Proposal Based on Social-Constructivism
10.00-10.30	№4	**Viktória Heizlerné Bakonyi** and Zoltán Illés	Real Time Classroom Systems in Teachers' Training
10.30-11.00	№5	**Eva Klimeková** and Monika Tomcsányiová	Case Study on the Process of Teachers Transitioning to Teaching Programming in Python
11.00-11.30	*Coffee break*		
11.30-12.00	№6	**Irina Gorbunova** and Ananstasia Govorova	Music Computer Technologies in Informatics and Music Studies at School for Children with Deep Visual Impairments: : From the Experience *The report was shifted from section 6
12.00-12.30	№7	Elisa Reçi and **Andreas Bollin**	The Quality of Teaching – Is There Any Difference Between University Teachers and School Teachers?

Section 4. Contests and Competitions in Informatics			
12.30-13.00	№8	**Christine Lutz**, Marc Berges, Jonas Hafemann and Christoph Sticha	Piaget's Cognitive Development in Bebras Tasks – A Descriptive Analysis by Age Groups
13.00-13.30	№9	Nataša **Grgurina,** Jos Tolboom and Erik Barendsen	The Second Decade of Informatics in Dutch Secondary Education
13.30-14.00	*Lunch*		
14.00-15.00	№10	Andreas Bollin**,** Heike Demarle-Meusel, **Max Kesselbacher**, Corinna Mößlacher, Marianne Rohrer and Julia Sylle	The Bebras Contest in Austria - Do Personality, Self-Concept and Attitudes Play an Influential Role?
15.00-15.30	№11	Lucia Budinská, Karolina Mayerová and **Michal Winczer**	Gender differences in Graph Task - Do They Exist in High School Bebras Categories Too?
15.30-16.00	№12	**Lucia Budinská,** Karolina Mayerová and Václav Šimandl	Differences Between 9–10 Years Old Pupils' Results from Slovak and Czech Bebras Contest
16.00-16.30	№13	Roberto Borchia, Antonella Carbonaro, Giorgio Casadei, **Luca Forlizzi**, Michael Lodi and Simone Martini	Problem Solving Olympics: An Inclusive Education Model for Learning Informatics
16.30-17.00	*Coffee break*		
Section 5. Socio-psychological aspects of teaching informatics			
17.00-17.30	№14	Johannes Magenheim, Kathrin Müller, **Carsten Schulte**, Nadine Bergner, Kathrin Haselmeier, Ludger Humbert, Dorothee Müller and Ulrik Schroeder	Evaluation of Learning Informatics in Primary Education. Views of Teachers and Students
17.30-18.00	№15	**Francesca Agatolio,** Fabio Albanese and Michele Moro	How an Ambitious Informatics Curriculum Can Influence Algebraic Thinking of Primary School Children
18.00-18.30	№16	**Sylvia da Rosa** Zipitría	Piaget and Computational Thinking
18.30-19.00	*Poster session*		
19.00-22.00	*Banquet (Botanic garden)*		

Third conference day	**Friday, October 12**	**Section 6** **Bus excursion** **Closing** **Farewell reception**	
Section 6. Computer Tools in Teaching and Studying Informatics			
8.30-9.00	№1	Fedor Novikov and **Viktor Katsman**	Gamification of Problem Solving Process Based on Logical Rules
9.00-9.30	№2	Isabella Corradini**,** Michael Lodi and **Enrico Nardelli**	An Investigation of Italian Teachers' View on Coding and Programming *The report was shifted from section 3
9.30-10.00	№3	Olga Starunova, **Valeria Nemychnikova** and Anna Dronzik	Computer Modeling of Secretary Problem and its Interesting Results
10.00-10.30	№4	Simone Mora, **Francesco Gianni**, Stefano Nichele and Monica Divitini	Introducing IoT Competencies to First-year University Students with Tiles Toolkit
10.30-11.00	№5	**Yusuf Moosa Motara**	Development of an LCT-Based MOOC Taxonomy
11.00-11.30	№6	**Fayiq Alghamdi,** Arnold Pears and Aletta Nylén	Computer Science Teachers Perspectives on Competencies – a Case Study in the Kingdom of Saudi Arabia *The report was shifted from section 2
11.30-12.00	*Light buffet*		
12.00-16.00	*Bus excursion*		
16.00-17.00	*Closing*		
17.00-19.00	*Farewell reception*		

SECTION 1. SUSTAINABLE EDUCATION IN INFORMATICS FOR PUPILS OF ALL AGES. ROLE OF PROGRAMMING AND ALGORITHMICS

Exploring Control in Early Computing Education

Ivan Kalas[1], Andrej Blaho [1] and Milan Moravcik[2]

[1] Comenius University, 842 48 Bratislava, Slovakia
[2] Edix, Bratislava, Slovakia

kalas@fmph.uniba.sk, blaho@fmph.uniba.sk, milan.moravcik@gmail.com

Abstract. In the paper we reflect on how our design research approach in the current development allows us to study the increasing cognitive complexity of *different levels of control* which pupils conduct when they program Emil, a virtual character on the screen. In our earlier work we outlined conceptual framework for primary programming, which recognised three different levels of control: (a) *direct manipulation*, (b) *direct control* and (c) *computational control (i.e. programming)* an actor. In the present research we managed to get deeper into the complexity of control by identifying four instead of three of its levels. Based on our close collaboration with three design schools we have also found that it is more productive to project and analyse learning progression of pupils connected with control within two-dimensional grid, where the first dimension is control itself and the second explores the way how the control is represented. Along this dimension we have identified five distinct levels of representation: (a) *none*, *(b) as internal record*, (c) *as external record*, (d) *as internal plan for future behaviour*, and finally (e) as *external plan for future behaviour*. In our paper we explain the grid of control by presenting selected tasks from different environments of Emil, our new approach to educational programming for Year 3 pupils.

Autonomous Recovery from Programming Errors Made by Primary School Children

Martina Forster[1], Urs Hauser[1], Giovanni Serafini[1] and Jacqueline Staub[1,2]

[1] Department of Computer Science, ETH Zürich, Universitätstrasse 6, 8092 Zürich, Switzerland
[2] Pädagogische Hochschule Graubünden, Scalärastrasse 17, 7000 Chur, Switzerland

martfors@student.ethz.ch
{urs.hauser,giovanni.serafini,jacqueline.staub}@inf.ethz.ch

Abstract. Programming classes offer unique opportunities to learn both semantic and syntactic precision, even for primary school children without prior knowledge in computer science. In order to make students progress autonomously, programming languages and environments need to be chosen with care to their didactic quality. This paper introduces four classes covering the majority of what we call structural programming errors. These mistakes are either syntactical errors or the result of invocations that do not match the signature of any user-defined command, and therefore prevent the execution of a program. Furthermore, we present a methodology that allows for detecting as many structural programming errors as possible, and show how we integrated this methodology in our Logo programming environment for primary schools. Finally, we reflect on an evaluation we carried out at school that confirms the didactic benefits of the chosen approach.

Effects on the School Performance of Teaching Programming in Elementary and Secondary Schools

Angélica Herrera Loyo

ETH Zürich, ABZ,
angelica.herrera@inf.ethz.ch

Abstract. The aim of this report is to describe some aspects of teaching programming to teenagers from 10 to 13 years old. The examples, data and impressions were taken from some courses given in public and private schools in Mexico, Switzerland and Colombia.

The paper is organized as follows: It starts with a description of the didactic proposal of ABZ-ETHZ under which the courses were given. Then the observations and reflexions are described with respect to several considerations: a) The ubiquitous recreational and social use that children and young people make of the technology, b) their motivation for programming, c) reading and writing in language acquisition and d) strengthening and applying mathematics as a decisive aspect of the ability to think logically.

A Case Study on the Effect of Using an Anchored-Discussion Forum in a Programming Course

Jean-Philippe Pellet[1,2], Gabriel Parriaux[1] and Tristan Overney[2]

[1] University of Teacher Education, Lausanne, Switzerland
fjean-philippe.pellet,gabriel.parriauxg@hepl.ch
[2] École polytechnique fédérale de Lausanne, Switzerland
tristan.overney@gmail.com

Abstract. In the context of a programming course taught to college freshmen, we give an account of the switch from a classical, Moodlebased discussion forum to MIT's NotaBene (NB) platform. One of the defining features of NB is to anchor each discussion thread to a given rectangular zone freely highlightable in any of the course's PDF documents. In doing so, it forces a precise contextualization of every post|be it to a slide from the lectures, to a sentence from the instructions in the exercises, or to lines of code in the given exercise keys. We hypothesize that this feature lowers the contextualization effort needed to ask a question, thus strengthening students' engagement and, ultimately, understanding of the matter. Using historical data on three years of giving the same course, we first examine and classify the students' interventions with both the traditional and the NB-based approach to see if the questions significantly differ qualitatively or quantitatively. We also quantify the contextualization effort needed in both approaches. Finally, we discuss our teacher experience with both platforms and make recommendations on the choice such a discussion forum in a programming course.

Students Teach a Computer How to Play a Game

Sylvia Da Rosa and Andrés Aguirre

Dorelo Institute of Computing, Facultad de Ingeniería, Universidad de la República
{darosa,aaguirre}@fing.edu.uy

Abstract. This paper describes a study into how secondary school students construct knowledge of programming. The study consists of three classroom sessions. In the first session

the students play a simple video game called LumberJack. Then, they are asked to describe the rules of they themselves playing the game as an algorithm in natural language.

In the second session, the students are asked to design an automata for a program that plays the game. In the third and final session, the students write programs that play the game and execute them in the programming language called TurtleBots.

The aim of the study is to help learners establish a correspondence between the algorithm and the elements relevant to the execution of the program. The results obtained in this study offers significant insights which contribute to the development of didactic guidelines for the introduction of programming to novice learners. These results are presented and analysed in section 4.

Approaching the Relative Estimation Concept with Planning Poker

Panagiota Chatzipetrou, Raquel Ouriques
and Javier Gonzalez-Huerta

Software Engineering Research Lab Sweden, Blekinge Institute of Technology,
Karlskrona, Sweden

{panagiota.chatzipetrou,raquel.ouriques,javier.gonzalez.huerta}@bth.se

Abstract. Simulation is a powerful instrument in the education process that can help students experience a reality context and understand complex concepts required to accomplish practitioners' tasks. The present study aims to investigate the software engineering students' perception about the usefulness of the Planning Poker technique in relation to their understanding of the relative estimation concept.

We conducted a simulation exercise where students first estimated tasks applying the concepts of relative estimation based on the concepts explained in the lecture, and then to estimate tasks applying the Agile Planning Poker technique. To investigate the students' perception, we used a survey at the end of each exercise. The preliminary results did not show statistical significance on the students' confidence to estimate relatively the user stories. However, from the students' comments and feedback, there are indications that students are more confident in using Agile Planning Poker when they are asked to estimate user stories. The study will be replicated in the near future to a different group of students with a different background, to have a better understanding and also identify possible flaws of the exercise.

Integrated Test Development. An Integrated and Incremental Approach to Write Software of High Quality

Lex Bijlsma, Harrie Passier, Harold Pootjes
and Sylvia Stuurman

Open Universiteit Nederland, Heerlen, Netherlands

{Lex.Bijlsma, Harrie.Passier, Harold.Pootjes, Sylvia.Stuurman}@ou.nl

Abstract. Creating test cases is a difficult task for students. The number of existing recommendations on how to create test cases is over whelming. There is a lack of guidelines on how to apply those recommendations one step after another. This problem even becomes more complicated when students are taught to refactor their code as a habit. We propose an approach to teach students how to develop and test their code systematically, with refactoring integrated. In our approach, we pay attention to both functionality and robustness.

Teaching Programming and Algorithmic Complexity with Tangible Machines

Tobias Kohn[1] and Dennis Komm[2]

[1] University of Cambridge, UK
[2] Department of Computer Science, ETH Zurich, and PHGR, Chur, Switzerland
tk534@cam.ac.uk, dennis.komm@inf.ethz.ch

Abstract. Understanding the notional machine that conceptually executes a program is a crucial step towards mastery of computer programming.

In order to help students build a mental model of the notional machine, visible and tangible computing agents might be of great value, as they provide the student with a conceptual model of who or what is doing the actual work.

In addition to programming, the concept of a notional machine is equally important when teaching algorithmic design, complexity theory, or computational thinking. We therefore propose to use a common computing agent as notional machine to not only introduce programming, but also discuss algorithms and their complexity.

A Diagnostic Tool for Assessing Students' Perceptions and Misconceptions Regards the Current Object "*this*"

Noa Ragonis [1,2] [0000–0002–8163–0199] and Ronit Shmallo [3] [0000–0002–1783–6109]

[1]Beit Berl College, Beit Berl, Kfar Saba, Israel
[2]Technion Israel Institute of Technology, Haifa, Israel
[3]Shamoon College of Engineering, Ashdod, Israel
noarag@beitberl.ac.il, ronit1@sce.ac.il

Abstract. Understanding of the object concept in Object Oriented Programming (OOP) is obviously the center of the paradigm. Many educators and researchers explored students' difficulties and developed teaching materials targeted at this central concept. The paper presents a diagnostic tool we developed that aims to reveal students' perception and understanding about the current object, referring to it by the *this* annotation. Proper conceptualization of *this* indicates an under-standing of objects in general, and involves aspects of memory allocation and programming approaches. The tool contains five questions, each devoted to cov-ering different aspects in various frameworks, such as: using *this* in constructors, using *this* as a visible parameter, using *this* in inheritance, or making necessary changes in transition from a non-static context that uses *this* to a static context. The questionnaire combines closed questions with a request to explain the an-swers and open questions. In the paper we present the purpose of each question, and address what it comes to examine. The diagnostic tool is based on known educational approaches: Bloom's taxonomy, assessment for, as, and of learning and learning from errors. The tool can be used by educators at high school or academic levels as a teaching tool, as a base for discussions, or as an evaluation tool. A short report on the use of the tool with different populations, including high school teachers, is presented. The paper uses Java as the programming lan-guage, but it easily can be translated to other OOP languages.

On Preferences of Novice Software Engineering Students: Temperament Style and Attitudes Towards Programming Activities

Tatjana Jevsikova, Valentina Dagienė and Vladimiras Dolgopolovas

Vilnius University Institute of Data Science and Digital Technologies,
Akademijos 4, 04812 Vilnius, Lithuania

tatjana.jevsikova@mii.vu.lt, valentina.dagiene@mii.vu.lt, vladimiras.dolgopolovas@mii.vu.lt

Abstract. Educators' experience shows that learning programming is in many aspects problematic for novice software engineering students. On the other hand, software engineering processes and the view of programming has been changing during the recent years. In this paper, we address socio-cognitive aspects of computer science and software engineering in order to contribute to programming education enhancement: the research is focused on students' temperament style and favorite programming learning activities. The study of 158 first and second year students, studying programming specialties in five higher education institutions, has been presented. The "psychological portrait" of the surveyed students reflects the evolution of the temperament style in programming during last decades. The attitudes towards the programming activities, presented in this paper, may contribute to the development of enhancement of existing programming courses in higher education.

SECTION 2. NATIONAL CONCEPTS OF TEACHING COMPUTER SCIENCE. RESEARCH ON INFORMATICS IN SCHOOLS

Standards for Higher Secondary Education for Computer Science in Germany

Arno Pasternak[1], Lutz Hellmig[2] and Gerhard Röhner[3]

[1] Fritz-Steinhoff-Schule Hagen, Technische Universität Dortmund
[2] Universität Rostock
[3] Lichtenberg-Schule Darmstadt, Studienseminar Darmstadt

arno.pasternak@cs.tu-dortmund.de, lutz.hellmig@uni-rostock.de, groehner@t-online.de

Abstract. In this paper we will report on the Standards for Higher Secondary Education for Computer Science in Germany. After the first results of the PISA-studies in Germany the administration in the central and the federal states switched standards and curricula from setting teaching aims and cognitive learning objectives to describing students' learning outcome with practices.

On the central level this work was only done for the major subjects like Mathematics, Mother Tongue, Foreign Language and Natural Science. This is a disadvantage for the other subjects in school. Therefore the German CS society – Gesellschaft für Informatik (GI) installed a working group to develop standards for higher secondary education in 2013. By the end of 2015 this group finished the task and in January 2016 the board of the GI published this academic work as the oficial standards in Germany for higher secondary CS education [2].

It is to be hoped that these standards will have a positive impact on the standardisation of computer science teaching in the various German federal states, just as the standards for lower secondary education had almost 10 years earlier.

Computer Science Teachers Perspectives on Competencies – A Case Study in the Kingdom of Saudi Arabia

Fayiq Alghamdi[1], Arnold Pears[2] and Aletta Nylén[3]

[1] Uppsala University, Uppsala, Sweden Al-Baha University, Al-Baha, Saudia Arabia
[2] KTH Royal Institute of Technology, Stockholm, Sweden
[3] Uppsala University, Uppsala, Sweden

fayiq.alghamdi@it.uu.se, fpears@kth.seg, aletta.nylen@it.uu.se

Abstract. The Kingdom of Saudi Arabia (KSA) has recently adopted the Saudi Teaching Competencies Standard (STCS). This paper tries to answer how these competencies are achieved, how they are maintained, and what support exists to support teaching CS competently in the KSA. This paper presents the results of an investigation of teacher awareness of, and attitudes to, the STCS in the Kingdom. Through the study reported here, we address an urgent need in the Kingdom to understand teacher preparedness in terms of CS teaching competencies. The study draws on interviews with ten CS teachers in _ve di_erent cities in the KSA. A thematic coding analysis approach was used. This study explores the CS teaching competencies held by teachers in three areas of CS teaching, focusing on connection to society, professional practice and professional development. The results of the study highlight the CS teaching competencies that CS teachers feel they currently grasp well in the

KSA. By enhancing awareness of what teachers currently do well we contribute to the adjustment and improvement of the STCS and help to build a program which addresses the current in-service training needs of CS teachers. The outcomes also help to raise awareness of the challenges of implementing the Computer Education curriculum in KSA schools.

A Core Informatics Curriculum for Italian Compulsory Education

Luca Forlizzi[1][0000–0002–3923–7668], Michael Lodi[2][0000–0002–3330–3089], Violetta Lonati[3][0000–0002–4722–244X], Claudio Mirolo[4][0000–0002–1462–8304], Mattia Monga[3][0000–0003–4852–0067], Alberto Montresor[5][0000–0001–5820–8216], Anna Morpurgo[3][0000–0003–0081–914X] and Enrico Nardelli[6][0000–0001–9451–2899]

[1] Università degli Studi dell'Aquila, Italy, *luca.forlizzi@univaq.it*
[2] Alma Mater Studiorum – Università di Bologna & INRIA Focus, Italy, *michael.lodi@unibo.it*
[3] Università degli Studi di Milano, Italy, *http://aladdin.di.unimi.it*
[4] Università degli Studi di Udine, Italy, *claudio.mirolo@uniud.it*
[5] Università degli Studi di Trento, Italy, *alberto.montresor@unitn.it*
[6] Università degli Studi di Roma "Tor Vergata", Italy, *nardelli@mat.uniroma2.it*

Abstract. In order to bring informatics, its ideas and ways of thinking of major educational value to all primary and secondary school students, the Italian Inter-universities Consortium for Informatics (CINI), in collaboration with the academic associations who gather together researchers in informatics (GRIN) and computer engineering (GII), has recently proposed a core informatics curriculum for all the levels of compulsory school. This paper summarizes the proposed curriculum, highlights the key underlying motivations, and outlines a possible strategy to ensure that its implementation in schools can be effective.

Comparative Analysis of the Content of School Course of Informatics in Russia and Subjects of the International Competition Bebras

Liudmila Bosova

Moscow Pedagogical State University, Moscow, Russian Federation

Abstract. During of the first twenty five years of its existence, the informatics course in Russian school was structured around following crosscutting content lines: information and information processes; representation of information; computer; modeling and formalization; algorithmization and programming; Information Technology; computer telecommunications, social informatics. Recently thematic block Mathematical foundations of computer science has been more clearly formed in it. In addition traditional line of algorithmization and programming, which was transformed into the thematic block Algorithms and programming elements, which includes robotics and mathematical modeling, has developed significantly. In general content of the course of informatics is stable, its fundamental component is the basis of the state final certification of graduates of primary and senior schools. At the same time students, their parents as well as representatives of the higher education and IT industry express concern about content of the modern school informatics course, rightly believing that this discipline has much greater potential for mastering such key competences of the digital economy as basic programming, basics of working with data, communication in modern digital environments. Since 2012, Russian students successfully participate in the international distant competition Bebras, in the tasks of which priority is given to the issues of theoretical informatics as the basis

of modern information technologies. The subject field of the competition is in many respects consonant with the content lines of Russian informatics course in school, the main differences lie in approaches to the representation of key concepts, ideas, methods and algorithms. Materials of the international competition Bebras can become a basis for the modernization of Russian informatics course in school by expanding its theoretical base as well as ensuring the unity of its theoretical and practical components.

An Undergraduate Project Combining Computer Science and the Arts An Experience Report of a Multidisciplinary Capstone Design

Sixto Javier Castro, Luis Jácome, Ana Gabriela Concha, Jordy Vásquez, Gustavo Londa, Luis Leonel Córdova, Boris Vintimilla, Cristina L. Abad, María Fernanda Miño and María De Lourdes Pilay

Escuela Superior Politécnica del Litoral, ESPOL, Guayaquil, Ecuador

{sjcastro,lejacome,agconcha,jgvasque,glonda,lulecord, boris.vintimilla,cabadr,mfmino,mdpilay}@espol.edu.ec

Abstract. The jobs pursued by many Computer Science graduates often require an ability to work effectively in multidisciplinary teams (e.g., game development teams). While this skill can be strengthened throughout the CS curricula, the Capstone Course offered in many programs provides an invaluable opportunity to expose students to a multidisciplinary team.

In this paper, we present an experience report in multidisciplinary capstone design, where the goal was to design an interactive film whose outcome can be changed by the viewer by using a second screen mobile app (a concept known as transmedia). We report on our implementation of this idea, including details on the composition of the team, the development methodology and end results. We highlight our lessons-learned and discuss possible improvements to the project. Our experience and methodology can help faculty at other institutions implement similar multidisciplinary capstone projects.

SECTION 3. TEACHER EDUCATION IN INFORMATICS

Computational Thinking: Constructing the Perceptions of Pre-Service Teachers from Various Disciplines

Noa Ragonis [1,2] [0000-0002-8163-0199]

[1]Beit Berl College, Beit Berl, Kfar Saba, Israel
[2]Technion Israel Institute of Technology, Haifa, Israel
noarag@beitberl.ac.il

Abstract. In the last two decades, educators have been following the terminology of Computational Thinking first posed by Wing. Different viewpoints and commentaries have been adopted, and accordingly course syllabi and learning materials were developed, particularly for K-12. The field has become a mandatory part of the curriculum in various countries, even for preschool age. The paper presents an academic course for pre-service teachers with the main aim to facilitate and instruct students in the process of building their understanding and interpretation of Computational Thinking, in the context of teaching their own discipline. The course pedagogical approach emphasizes the adoption of Computational Thinking while identifying significant, non-trivial, computational processes in different disciplines. The course model was implemented with three pre-service teacher populations studying for their teaching certificate in: (1) sciences for high school; (2) humanities and social sciences for high school; and (3) various disciplines for elementary school. The course allows future teachers to experience for themselves learning activities that are recommended for implementation with their future students. The course pedagogical approach and rationale are presented, followed by detailed course structure and learning assignments. The teaching, learning, and assessment approach yielded impressive achievements, although not without obstacles and difficulties. The details of the course presentation enable its implementation with different populations of pre-service and in-service teachers, and can also be implemented in schools.

Investigating the Pedagogical Content Knowledge Development of Teachers Attending a MOOC on Scratch Programming

Ebrahim Rahimi[1], Ineke Henze[2], Felienne Hermans[3] and Erik Barendsen[4]

[1] Radboud University and Open University, The Netherlands, *e.rahimi@cs.ru.nl*
[2] Delft University of Technology, The Netherlands, *f.a.henze-rietveld@tudelft.nl*
[3] Delft University of Technology, The Netherlands, *f.f.j.hermans@tudelft.nl*
[4] Radboud University and Open University, The Netherlands, *e.barendsen@cs.ru.nl*

Abstract. The goal of this study is to investigate changes in PCK (Pedagogical Content Knowledge) of Dutch primary and secondary computer science teachers participating in a MOOC about Scratch programming. We captured the teachers' PCK using identical pre- and post-questionnaires and conducted a qualitative deductive-inductive content analysis to identify changes in the PCK of the MOOC attendees. We relate the observed differences between PCK before and after the MOOC to Clarke and Hollingsworth's model of teacher professional growth and Van Driel and Henze's model of PCK development. Our analysis gives rise to four design principles meant to inform the pedagogical design of such MOOCs and improve their pedagogical affordances with regard to PCK development of their attendees.

Informatics and Computational Thinking: a Teacher Professional Development Proposal Based on Social-Constructivism

Carlo Bellettini[0000–0001–8526–4790], Violetta Lonati[0000–0002–4722–244X], Dario Malchiodi[0000–0002–7574–697X], Mattia Monga[0000–0003–4852–0067] and Anna Morpurgo[0000–0003–0081–914X]

Universitá degli Studi di Milano, Italy
http://aladdin.di.unimi.it

Abstract. Teaching informatics with a socio-constructivist approach is the theme of the Professional Development (PD) proposal for teachers we present in this paper. This proposal is built upon the expertise we developed in the last few years by designing and delivering enrichment activities to school students, where constructivist strategies are used to let students discover informatics as a scientific discipline, and to promote computational thinking skills and problem solving competences. Starting from the analysis of teachers' training needs, we structured the proposal into different units. We highlight their goals and contents, and illustrate some of the proposed activities. We held some training sessions to test our proposal; we report our findings and the feedback from the participants who amount to a total of ninety-five in-service and prospective teachers.

Real Time Classroom Systems in Teachers Training?

Viktória Heizlerné Bakonyi[1] [0000–0001–5093–8492] and Zoltán Illés [0000–0002–6623–5721]

Eötvös Loránd University, Budapest, Hungary
{hbv,illes}@inf.elte.hu, http://www.inf.elte.hu

Abstract. Nowadays students are surrounded by plenty of information resources and they use these multimedia streams parallel. This multitask working environment appears during lessons as well. This hyper attention symptom blocks students in the effective participating of the common work. Therefore more and more ICT tools appear and are used in schools from elementary schools to higher education. One of these possibilities is CRS (Classroom Response System) of which benefits are undoubted.

Their usage makes lessons more diversified, more motivating for students who are used to and prefer to use applications and smart devices, more interactive. However we must not forget about the disadvantages either so in this paper we should like to give a look around this topic, present our own newly developed CRS system and speak about how we can use it in teachers training.

Case Study on the Process of Teachers Transitioning to Teaching Programming in Python

Eva Klimeková and Monika Tomcsányiová

Department of Informatics Education, Comenius University in Bratislava, Slovakia
{klimekova, tomcsanyiova}@fmph.uniba.sk

Abstract. The aim of our research was to investigate the process of teachers transitioning to teaching programming in Python, with respect to the challenges they face and support they

require. Through the methods of qualitative research, we analysed a number of cases where computer science teachers transitioned from teaching in Pascal to Python.

Based on the analysis of these cases, we propose a categorization for the transformation process. We identified inuencing factors and present recommendations to support teachers transitioning to teaching a new language. We believe our research will contribute to improved support for teachers transitioning to teaching programming basics in new programming languages.

An Investigation of Italian Primary School Teachers' View on Coding and Programming

Isabella Corradini[1], Michael Lodi[2] and Enrico Nardelli[3]

[1] Themis Research Centre, Rome, Italy, *isabellacorradini@themiscrime.com*
[2] Univ. Bologna & INRIA Focus, Bologna, Italy, *michael.lodi@unibo.it*
[3] Univ. Roma "Tor Vergata", Rome, Italy, *nardelli@mat.uniroma2.it*

Abstract. This paper reports the results of an investigation involving almost a thousand primary school teachers in Italy, to explore their views on the terms "coding" and "programming", and how they are related to their ideas on "computational thinking".

When directly asked "if coding is different from writing programs", roughly 2 out of 3 teachers answered "no". Among the teachers who answered "yes", almost 160 tried to motivate the difference: a few of them gave admissible explanations, while the others showed various misunderstandings, which we classify and discuss.

By contrast, when asked about their idea of "what coding is", only 4 out of 10 of the teachers explicitly linked coding to programming, but an additional 2 out of 10 cited an information processing agent executing instructions. The remaining part of the sample did not provide explicit or implicit links between coding and programming.

Our investigation shows that untrained teachers hold misconceptions regarding CS and its related terms. Given the general public and media attention on "coding" in schools, currently taught by existing teachers – mostly not appropriately trained, professional development actions focusing on CS scientific principles and methods are therefore a top priority for the effectiveness of CS education in schools.

The Quality of Teaching – Is There Any Difference Between University Teachers and School Teachers?

Elisa Reçi and Andreas Bollin

Alpen-Adria-Universität Klagenfurt, Universitätstr. 65–67, Klagenfurt, Austria
{elisa.reci, andreas.bollin}@aau.at
https://www.aau.at/en/informatics-didactics/

Abstract. An important aspect of the profession of an educator is the assessment and the improvement of the quality of the underlying teaching process, but does this hold for all types of teachers? By collecting best practices of computer science teachers in school we created a teaching maturity model (called TeaM) and recently demonstrated its benefit.

The paper now takes this maturity model as a basis and investigates the question about the differences in teaching at Universities and secondary schools. To do so, we randomly selected computer science lectures at our university, assessed them based on the Team Model and looked at the practices in more detail. In our setting it turned out that not all practices are covered at both types of institutions, and especially practices needing documentation and methodologies are lacking at university teaching.

SECTION 4. CONTESTS AND COMPETITIONS IN INFORMATICS

Piaget's Cognitive Development in Bebras Tasks – A Descriptive Analysis by Age Groups

Christine Lutz, Marc Berges, Jonas Hafemann and Christoph Sticha

Technical University of Munich, School of Education,
Arcisstr. 21, 80333 Munich, Germany,
christine.lutz@tum.de, berges@tum.de, jonas.hafemann@tum.de, christoph.sticha@tum.de

Abstract. The Bebras contest started as a small computer-science competition in Eastern Europe and is now well-established all around the world. In the meantime, the challenge also addresses primary school students; in Germany since 2015. In the light of the ongoing discussion on introducing computer science in primary education, the question of whether the tasks differ between the age groups moved to the focus. So, we analyzed how Bebras tasks look like, especially comparing the different age of the participants. Here, only characteristics of the task descriptions are examined, showing that both, text and pictures differ through the age groups. Their complexity increases and also, addressed objects and subjects manifest a clear separation of Piaget's stage of cognitive development between primary school and secondary school students. All the findings enable to concisely characterize Bebras tasks for their use in regular computer-science classes or for further research on the cognitive processes involved in solving those tasks.

The Second Decade of Informatics in Dutch Secondary Education

Nataša Grgurina[1], Jos Tolboom[2] and Erik Barendsen[3]

[1] Teaching and Teacher Education, University of Groningen, Groningen, the Netherlands,
[2] Stichting Leerplanontwikkeling, Enschede, the Netherlands,
[3] Radboud University and Open University, the Netherlands,
n.grgurina@rug.nl, j.tolboom@slo.nl, e.barendsen@cs.ru.nl

Abstract. In 1998, informatics was introduced as an elective subject for all students in the upper grades of senior general secondary education and pre-university education in the Netherlands. Rather than focusing on digital literacy or the use of office applications, it focuses on informatics as a scientific discipline.

In its first decade, it faced growing pains while fighting for recognition and necessary facilities from the stakeholders: students, parents, school administrators, politicians and the general public. In 2007, the curriculum was slightly streamlined but not updated. In its second decade, informatics reached adulthood with established teacher training programs and a new curriculum which is to be introduced in 2019. In this paper we describe the events and processes that led to the renewal of the curriculum, the curriculum itself with the principles it is based on and its aims, the current process of teaching material development, the related research, the teacher training, curriculum reform in primary and lower secondary education, and the current situation of informatics as an upper secondary school subject, together with the challenges it still faces.

The Bebras Contest in Austria – Do Personality, Self-Concept and General Interests Play an Inuential Role?

Andreas Bollin, Heike Demarle-Meusel, Max Kesselbacher, Corinna Mößlacher, Marianne Rohrer and Julia Sylle

Alpen-Adria-Universität Klagenfurt, 9020 Klagenfurt, Austria
{Andreas.Bollin, Heike.Demarle-Meusel, Max.Kesselbacher, Corinna.Moesslacher, Marianne.Rohrer, Julia.Sylle}@aau.at
https://www.aau.at/en/informatics-didactics

Abstract. The Bebras (Beaver) contest aims at testing of and motivating for Informatics and Computer uency, and as such it is designed to be a contest for all pupils between 8 and 19. But, does it really attract and favor all types of children likewise? This paper takes a closer look at different types of personality, self-concept and interests of the winners of the Bebras contest in Austria and discusses those factors that might contribute to a successful participation. It concludes with some recommendations that might help in increasing the number of participation at the event.

Gender Differences in Graph Tasks – Do they Exist in High School Bebras Categories Too?

Lucia Budinská, Karolina Mayerová and Michal Winczer

Department of Informatics Education, Comenius University in Bratislava, Slovakia
{lucia.budinska, mayerova, winczer}@fmph.uniba.sk

Abstract. This paper explores gender differences across the abilities of Junior (15–17 year old) and Senior (17–19 year old) students in terms of solving graph problems. As a basis to our assessment, we look at the graph tasks from the Slovak Bebras competition in 2012 to 2017 across both Junior and Senior categories. In our earlier research on this topic, we introduced a new method of categorising graph problems. This was based on an in-depth analysis of the various problem types aimed at 8–15 year old students, whereby a set of indicators were defined to predict whether a task was more likely to be successfully solved by girls or boys. In this paper, we apply the new categorisation onto graph tasks aimed at Junior and Senior students with an aim to verify whether the same predictors of girls' and boys' success remain valid. A qualitative analysis indicates that our categorisation of graph tasks is suitable for Junior and Senior categories with minor adaptation only. As a result of a subsequent quantitative analysis, we find a significant difference in the solution success rates between girls and boys in 38 out of 65 analysed graph tasks. In 35 tasks boys were significantly more successful and these were tasks with an overall lower success rate. Furthermore, a few tasks with contradictory results concerning girls' and boys' solution success were identified. We selected one of these tasks where a higher error tendency in older students was apparent, and further analysed it together with the students themselves (15–19 years old) in order to better understand the methods used by them while solving this task. Our findings can be used to enable authors of task sets and lesson plans to define problems in a manner that will minimize gender success differences such as the ones described in this study.

Differences Between 9–10 Years Old Pupils' Results from Slovak and Czech Bebras Contest

Lucia Budinská[1], Karolína Mayerová[1] and Václav Šimandl[2]

[1] Comenius University, 842 48 Bratislava, Slovakia

[2] University of South Bohemia, 371 15 České Budějovice, Czech Republic

{lucia.budinska, mayerova}@fmph.uniba.sk, simandl@pf.jcu.cz

Abstract. The education system in Czechia and the education system in Slovakia are very similar but while in Slovakia the education reform (together with the reform of the curriculum for Informatics) was implemented some years ago, in Czechia it is currently being prepared.

Informatics in Slovakia is taught from primary school, unlike in Czechia where it only appears in some types of high school. Nevertheless, both countries organise the Bebras challenge – the international Informatics contest. Therefore, we were interested in the achievement of pupils from the two countries, expecting Slovakian contestants to be more successful. We analysed the results from both competitions, focusing on the age category Little Beavers/Mini, which includes younger primary school pupils. This paper presents a case study, in which we compare Year 4 contestants (9 to 10 years old) from the two countries. Their results from 15 tasks with the same form and wording (to minimise the inuence of other factors) were studied. As it results from the study, Slovakian Year 4 pupils are more successful in digital literacy tasks and in algorithmic tasks and they are slightly more successful in statement logic tasks and in programming tasks. In logic tasks dealing with graph theory no significant differences between among Year 4 pupils in Slovakia and Czechia were revealed. For each from the 15 tasks' results, gender differences were also analysed – dividing tasks into three groups (girls' tasks, boys' tasks, neutral tasks), with almost the same distribution for both countries.

Problem Solving Olympics: an Inclusive Education Model for Learning Informatics

Roberto Borchia[1], Antonella Carbonaro[2], Giorgio Casadei[2], Luca Forlizzi[3], Michael Lodi[4] and Simone Martini[4]

[1] Istituto Tecnico Industriale "Q. Sella", Biella, Italy

[2] Universitá di Bologna, Dip. di Informatica{Scienza e Ingegneria, Italy

[3] Universitá dell'Aquila, Dip. di Ingegneria e Scienze dell'Informazione e Matematica

[4] Universitá di Bologna, Dip. di Informatica{Scienza e Ingegneria, Italy;
and INRIA Sophia-Antipolis, France

Abstract. The paper presents the Olimpiadi di Problem Solving, a mild and inclusive competition aimed to promote computational thinking and general problem-solving in Italian schools. We describe motivation, teaching strategies behind the initiative, as well as its structure, organization and give some sample of the problems proposed to students. We also present some data that show the broad participation of Italian schools to the initiative, and a preliminary analysis of the results obtained by the students in the last 5 editions, which suggests that the competition fosters deep learning of computational thinking knowledge and skills.

SECTION 5. SOCIO-PSYCHOLOGICAL ASPECTS OF TEACHING INFORMATICS

Evaluation of Learning Informatics in Primary Education Views of Teachers and Students?

Johannes Magenheim[1], Kathrin Müller[1], Carsten Schulte[1], Nadine Bergner[2], Kathrin Haselmeier[3], Ludger Humbert[3], Dorothee Müuller[3] and Ulrik Schroeder[2]

[1] University of Paderborn, Germany, *{firstname.lastname}@uni-paderborn.de*
[2] RWTH Aachen University, Germany, *{lastname}@cs.rwth-aachen.de*
[3] University of Wuppertal, Germany, *{khaselmeier,dmueller,humbert}@uni-wuppertal.de*

Abstract. How does learning about informatics in primary education unfold in the eyes of teachers and their students? We report on the evaluation results from a distributed project in Germany (North-Rhine Westphalia) and some implications of these findings for future improvements.

Three modules for informatics in primary education and a corresponding concept for teacher training were developed collaboratively by researchers and practitioners. The project was assessed with methods of formative and summative evaluation. The first results indicate that teaching and learning of informatics in primary education are possible.

It is regarded overall as rewarding and exciting, but it was difficult for primary school students and teachers as well to discern the bigger picture of informatics education and thus to relate the topics taught and learned to everyday experiences.

How an Ambitious Informatics Curriculum Can Inuence Algebraic Thinking of Primary School Children

Francesca Agatolio[1], Fabio Albanese[2] and Michele Moro[1]

[1] University of Padova, Italy
[2] Collegio Pio X, Italy

francesca.agatolio@phd.unipd.it, fabio.albanese@fondazionecollegiopiox.org, michele.moro@unipd.it

Abstract. In this paper we describe the special case of an informatics curriculum implemented in a primary school, and the observed learning outcomes. The particularity of this curriculum is based on both the variety of the contents and on their apparent complexity. The children experienced different programming languages, including one text-based, and became confident with many educational robotics kits such as Lego WeDo, MBot and Arduino. During this path, children appeared strongly involved also in dealing with complex challenges and seemed to have developed their mathematical thinking. In particular, even if they had never worked with the formal structures of algebra during the math classes, children appeared confident with the concept of variable found in programming. For this reason, we decided to investigate whether these competencies applied even to mathematics, using a well-known national test developed to gauge the level of skills and selecting some questions related to algebra and designed for 8th grade students. The results that emerged were encouraging, suggesting that informatics could be useful in reinforcing algebraic thinking and introducing some mathematical concepts particularly complex for many students, such as variables.

Piaget and Computational Thinking

Sylvia da Rosa Zipitría

Instituto de Computación, Facultad de Ingeniería,
Universidad de la República, Montevideo, Uruguay
darosa@fing.edu.uy

Abstract. In this article I present a theoretical framework for the concept computational thinking. I do so in response to some of the problems and consequences of the lack of viable theoretical foundations; especially in relation to the development in recent years of many educational practices that claim the term computational thinking. I therefore introduce my extension of Jean Piaget's general law of cognition which arose as a result of my empirical research on novice learners knowledge of the concept of a program as an executable object. Said empirical study is briefly described in this paper as a means to highlight the key to my extension of Piaget's general law, which is the insight of how the thought processes and methods involved in cases where the subject must instruct an action to a computer differ from those in which the subject instructs another subject, or performs the action themselves. My theory explains the difference between algorithmic thinking and computational thinking by adequately locating it in the specificities of the subject instructing a computer. Hence, in this article I claim that my extension of Piaget's law offers a more empirically thorough and theoretically sound way forward in the conceptual development of computational thinking than the alternatives that are being debated in academia to the present day.

SECTION 6. COMPUTER TOOLS IN TEACHING AND STUDYING INFORMATICS

Gamification of Problem Solving Process based on Logical Rules

Fedor Novikov and Viktor Katsman

[1] St. Petersburg Polytechnic University, St. Petersburg
fedornovikov51@gmail.com, vikto9494@gmail.com

Abstract. One of the main elements of the modern educational process in the field of IT is the solution of logical-combinatorial problems. When solving such problems, learners need to build a new solution based on the set of basic operations studied before being combined according to certain rules. To consolidate these skills, leaner need to solve a reasonable number of problems that leads to need of verification of a huge number of solutions.

Many systems allow us to automate the verification of solutions. In most cases, these systems check pure answer only, but not the progress of the solution itself. Such a method of verification does not exclude the possibility of obtaining the correct answer and the credited task with an incorrect or inconsistent solution.

We propose a method for verifying the solution of problems, based on search of valid transformations, or stepwise refinements, similar to search for proof in logical calculi. This means that, the system at each step effectively sorts out all possible transformations, trying to find one that the learner applied. In this case, the system not only can find an error in the solution, but also indicate the source of this error, which is wrong transition between steps.

In addition to automatically verifying the solutions, this approach allows us to generate tasks that require the application of specific rules. Also the application of the rules can be interpreted as “moves” in the intellectual game. Then the solution of the problem turns into a game process, and the correct solution corresponds to a sequence of steps leading to the victory in the game.
The proposed approach was tested on students of the junior courses and showed good results. Namely, in the conducted experiments the efficiency of the teacher’s work when checking solutions increased more than 4 times.

Music Computer Technologies in Informatics and Music Studies at Schools for Children with Deep Visual Impairments: from the Experience

Irina Gorbunova[1] [0000–0003–4389–6719]
and Ananstasia Govorova[1,2] [0000–0002–5693–9856]

[1] Herzen State Pedagogical University of Russia, 48, emb. river Moika,
191186 St. Petersburg, Russia
[2] Children’s art school “Okhta center for aesthetic education”, 8 “A”,
Marshal Tukhachevsky st., 195253 St. Petersburg, Russia
gorbunovaib@herzen.spb.ru

Abstract. Music computer technologies (MCT) open up wide opportunities for blind students to more effectively study the diverse content and methods of presentation of educational material

in Informatics, contributing to the achievement of positive learning outcomes in a shorter time. This is especially important for children who study in children's music schools. Mastering of MCT by children with profound visual impairment has a number of characteristic features, which are most clearly manifested in the initial period of studying Informatics. With the help of contemporary computer technologies (speech synthesizers, Braille displays, etc.), as well as due to the possibility of using “hot keys” blind students quickly master many MCT-programs.

The main objectives of the lessons on musical informatics are: to study the basic concepts related to the receipt and use of information, both in school and in everyday life; to obtain skills in working with a computer and some peripherals. In the program considered by the authors of the article it is supposed to study the basic concepts and acquire the skills included in the existing programs in informatics, taking into account the fact that all tasks in informatics are based on materials related to music.

The program is designed in such a way that it is possible to perform the proposed tasks not only with the help of a computer keyboard and “hotkeys”, but also with the help of MIDI-keyboard. This is achieved by the connection between the actual music and informatics.

The authors also propose to consider the application software – a tool that would be accessible to the sighted and blind and would not cause difficulties in working with him/her both in some and in others: the exchange of information between the teacher and the blind student or vice versa – the blind teacher and the ordinary/seeing pupil.

Computer Modeling of Secretary Problem and its Interesting Results

Olga Starunova, Valeria Nemychnikova and Anna Dronzik

Moscow South-Eastern School named after V. I. Chuikov (“Silaeder” classes), Moscow, Russia

Abstract. A methodics of composing research work in programming for middle school students (1–2 years of programming experience) is offered in this article. This methodics is demonstrated using one interesting problem: secretary problem, or marriage problem. This problem is too complex to be theoretically solved in school and comprises probability theory and calculus. However it can be easy modeled using programming and it allows many modifications that can be invented by students.

Moreover, results of such computer modeling allow to make non-trivial practical conclusions about decision making and optimal of choice strategies.

Introducing IoT Competencies to First-Year University Students with Tiles Toolkit

Simone Mora, Francesco Gianni, Stefano Nichele and Monica Divitini

Norwegian University of Science and Technology, Trondheim, Norway

Oslo Metropolitan University, Oslo, Norway

simonem@idi.ntnu.no, francesco.gianni@idi.ntnu.no, stenic@oslomet.no, monica.divitini@ntnu.no

Abstract. Advances in the field of Internet of Things (IoT) are introducing innovations in multipledom a in s including smart cities, health care and transportation. An increasing number of jobs today require IoT competences that university courses need to be prepared to deliver.

Yet, teaching IoT topics is a challenging task due to complexity and unstructured nature of the IoT. It requires to deliver skills in multipledom a ins including design, hardware and software engineering and it is often hard to find an entry point to the field. In this paper we explore using the Tiles ideation tool kit as away to teach bachelor students in IT topics IoT fundamentals. Tiles is composed by a set of 150 cards and a works hop procedure for collaborative ideation. We performed a user study with 60 computer science students to investigate how Tiles can be used as an experiential learning tool to develop basic knowledge in IoT and to train design thinking skills. Results show the tool was accepted as useful and fun top lay with. Nearly all students managed to develop a simple IoT idea during the three-hours workshop. Learning out comes were observed in about half of participants, although time constraints and high stress levels impacted the participants' experience.

Development of an LCT-Based MOOC Taxonomy

Yusuf Moosa Motara

Rhodes University Computer Science, Grahamstown, South Africa
y.motara@ru.ac.za

Abstract. Codecademy is an example of a successful and disruptive player in the online education space. This work describes the structure, content, and experience of working through a Codecademy Pro Intensive module and compares it to the offline approach in a similar domain. Existing taxonomies are insufficiently general to explain its success, and a new taxonomy based on Legitimation Code Theory is developed to compensate. This taxonomy is arguably more meaningful and more general than those presently in use.

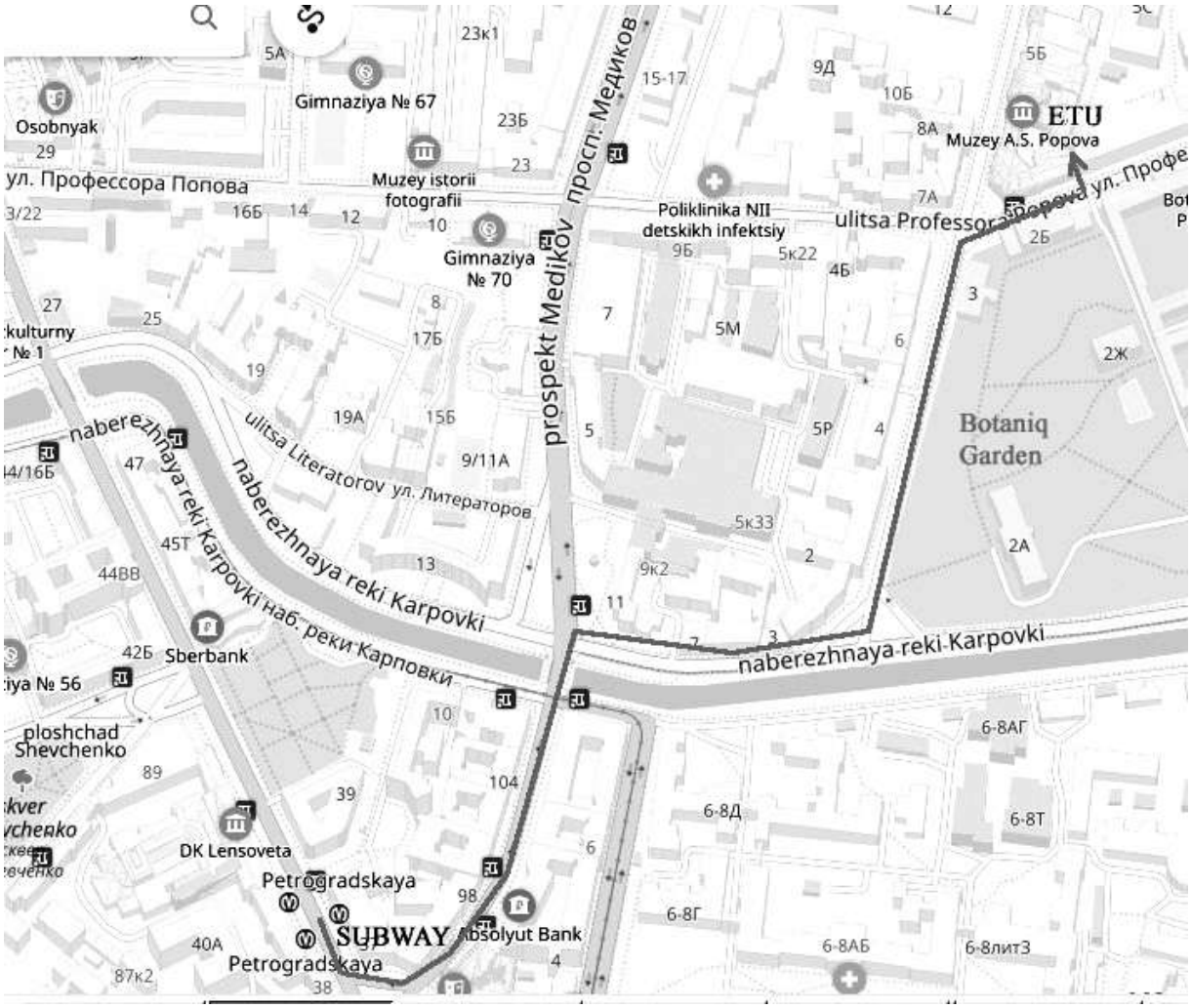
On the Triviality of the Assignment Statemant

Vreda Pieterse[1] and Estelle Taylor[2]

[1] University of Pretoria, Pretoria, South Africa
[2]The North-West University, Potchefstroom, South Africa
vpieterse@cs.up.ac.za, estelle.taylor@nwu.ac.za

Abstract. The assignment statement is a basic programming operation which is used to assign a value to a variable. The assignment statement is needed in all programs which manipulate data. Most programmers assume that this statement is trivial and therefore should not pose conceptual problems for novice programmers. This article investigates the validity of this assumption. We describe a number of misconceptions of the assignment statement in the form of inappropriate mental models for the assignment statement. These were established through a literature review. We used a questionnaire to identify the prior programming experience of the students in a cohort of 1058 students enrolled for an introductory programming course. We classify the respondents into three categories namely those who have never programmed before; those who have previously tried but failed the introductory programming course; and those who claim that they have had prior programming experience. We then asked the students to complete ten questions which ask the outcome of a short program executing assignment statements. The multiple choice questions were designed to identify the mental model of the respondent regarding the assignment statement for a number of cases. We compare the occurrence of correct and wrong metal models in the three groups and discuss our observations as well as their consequences for teaching introductory programming.

HOW TO GET FROM THE METRO STATION PETROGRADSKAYA TO ETU



The University address: Professora Popova str. 5 (улица Профессора Попова, 5)

ENTRANCE TO THE METRO STATION

STATIONS' WORKING SCHEDULES

LINE 2										
№	Station (Lobby)	Station opening time for entrance and exit	Station closing time for entrance	Station closing time for exit	First train to				Last train to	
					Parnas		Kupchino		Parnas	Kupchino
					odd days of the month	even days of the month	odd days of the month	even days of the month	odd days of the month	even days of the month
27	Lobby 1 of Moskovskaya Station	6-30	22-00	22-10	5-41	5-41	5-45	6-04	0-11	0-45
28	Lobby 2 of Moskovskaya Station	5-35	0-41	0-55						
35	Lobby 1 of Nevskiy Prospekt Station	7-00	23-00	23-10	5-58	5-58	5-47	5-47	0-28	0-29
36	Lobby 2 of Nevskiy Prospekt Station	5-36	0-28	0-50						
38	**Petrogradskaya**	**5-35**	**0-28**	**0-45**	**6-04**	**6-04**	**5-42**	**5-57**	**0-33**	**0-23**

The cost of one ticket for metro is 45 rubles. Tickets for ground transportation are purchased directly during the trip with the check-taker and cost 40 rub.
The fare on any type of transport in St. Petersburg is valid for only one trip.

FOR TRAVEL BY PUBLIC TRANSPORT IN ST. PETERSBURG, WE RECOMMEND USING RECHARGE "SINGLE E-TICKET"

The Standard Electronic Ticket can be used to pay for Metro journeys and other kinds of public transportation (including route taxi). The stored-value smart card costs RUB 55 (number of journeys – 0) The smart card can be returned for a full refund within 45 days from the day of purchase.

The Standard Electronic tickets can be purchased at any metro station in ticket window.
Initially, you can put on the card any amount from 45 rubles (the cost of one trip by metro). Then you can recharge the card if necessary. For recharge use automatic machines or ticket window.

You can use one card for pass several people (only for the metro). In other kinds of public transportation, each passenger must have their own card or pay in cash. The cost of the trip decreases from the 10th trip on one map.

TAXI SERVICE (recommended)

TAXOVICHKOF	http://taxovichkof.com/	+7(812)330-00-00
TAXI 777	http://www.777taxi.ru/	+7(812)777-1-777

DRAWBRIDGE SCHEDULE

Palace Bridge 01:10 – 02:50 03:10 – 04:55
Trinity Bridge 01:20 – 04:50
Litejnyj Bridge 01:40 – 04:45

WE RECOMMEND TO VISIT THEATERS

Mariinsky Theatre (Theatre Square, 1)
https://www.mariinsky.ru/en/
Please note that the theater has 3 buildings:
Mariinsky Theater – a historic building (mid 19th century)
Mariinsky 2 -– a modern building (2013 year)
Concert Hall – a modern building, the least interesting to visit.
The named halls are located in the Mariinsky II building

Mariinsky Theater – a historic building

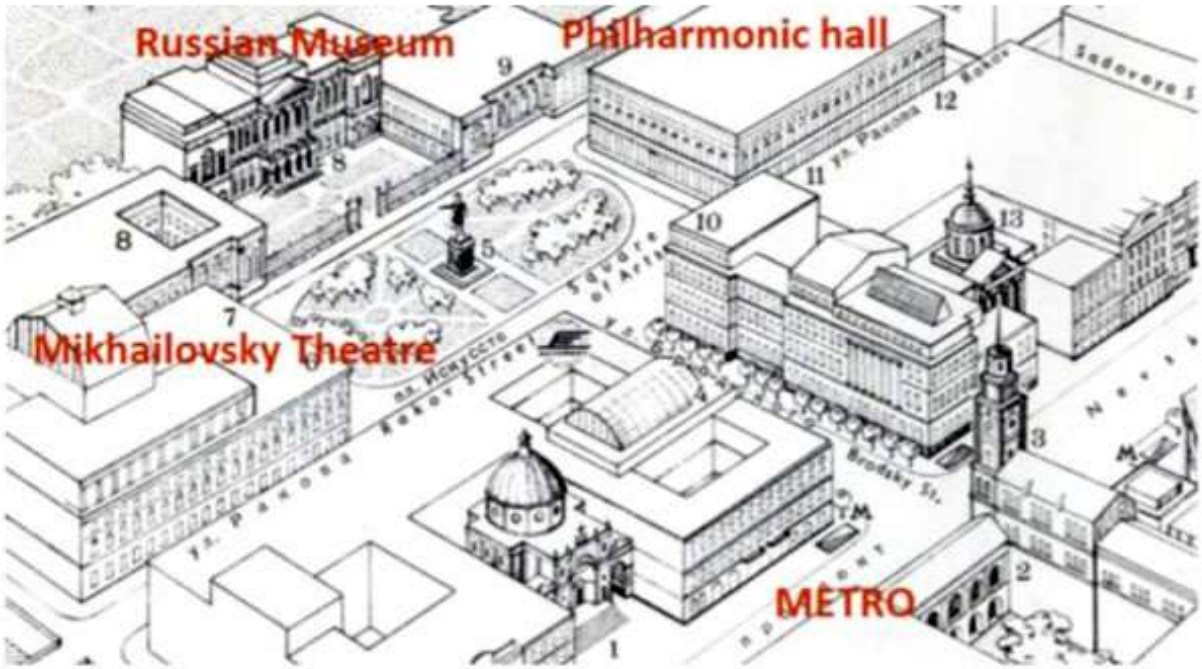
Mariinsky 2 -– a modern building

How to get to the theater: Take the metro to the station Sennaya Ploshchad (the third stop from the metro station Petrogradskaya towards the center). Exit to the square and go through the square to the opposite corner to the embankment of the Griboedov Canal. There, take the taxi bus №1, which goes to the theater.

Mikhailovsky Theatre (Arts Square, 1)

https://mikhailovsky.ru/

How to get to the theater: Take the metro to the station Nevsky Prospekt (the second stop from Petrogradskaya towards the center). Exit to Nevsky Prospect and go through Mikhaylovskaya Street to Arts Square (Ploshchad Iskusstv).



MUSEUM

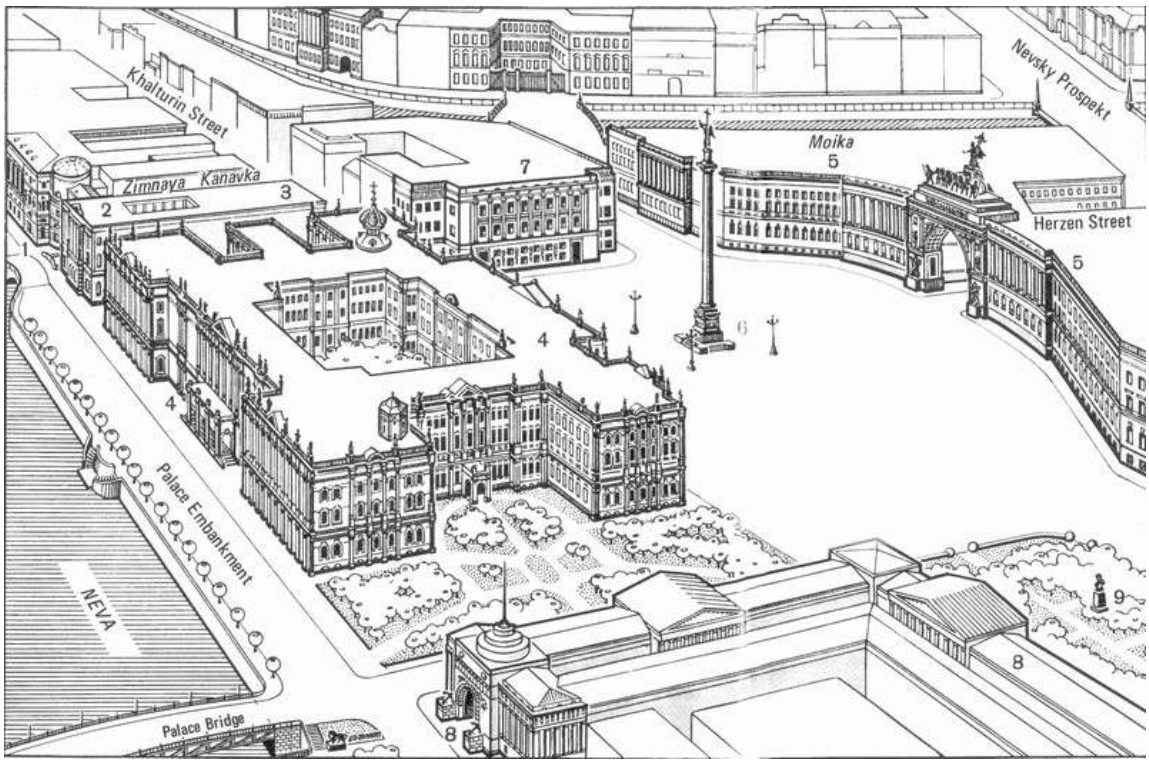
1. **State Hermitage Museum** (2, Palace Square)

http://www.hermitagemuseum.org

Opening Hours
Tuesday, Thursday, Saturday, Sunday:
10.30-18.00
Wednesday, Friday:
10.30-21.00
Closed: Mondays

Online tickets https://www.hermitageshop.org/tickets/ ($17.95)

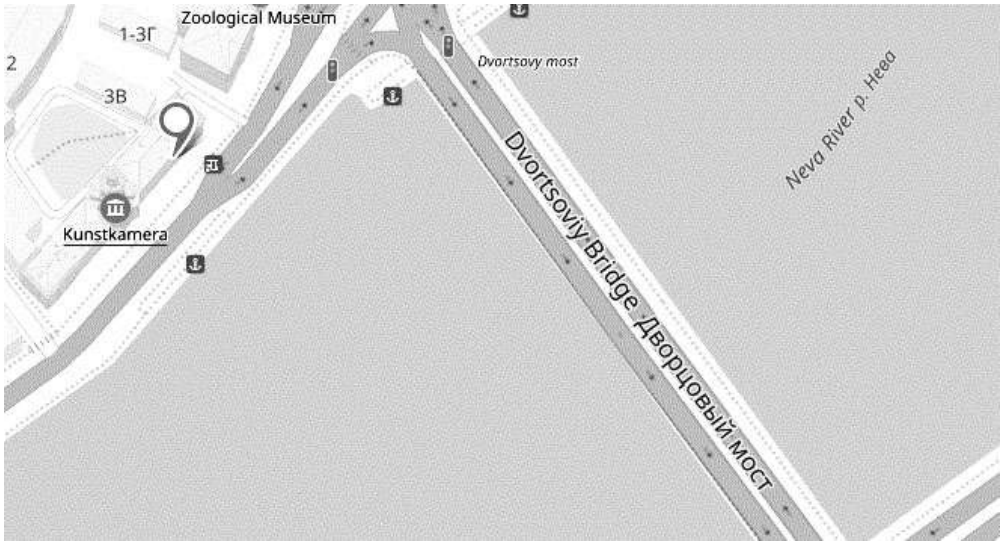
Public Transport
Metro:
Ⓜ ADMIRALTEYSKAYA, Ⓜ NEVSKY PROSPECT / GOSTINNY DVOR
Buses: 7, 10, 24, 191
Trolleys: 1, 7, 10, 11



2. The Kunstkamera (Entrance for the visitors from the Tamozhenny pereulok)
The Kunstkamera: all world knowledge in one building

TICKETS – 300 rub.
OPEN:
Hours: TUESDAY – SUNDAY 11:00 – 18:00
Tickets: till 17:00
MONDAY – A DAY OFF

Public Transport
ADMIRALTEYSKAYA, buses: 7, 24, 191; trolleybuses: 1, 7, 10, 11
NEVSKY PROSPECT / GOSTINNY DVOR, buses: 7, 24, 191; trolleybuses: 1, 7, 10, 11
VASILEOSTROVSKAYA, buses 24, 47
SPORTIVNAYA, buses: 191, 10; trolleybus 1



3. Petrovskaya Akvatoria

ADMIRALTEYSKAYA, Malaya Morskaya str., 4/1, SC ‘Admiral’, 6 floor
http://peteraqua.ru/en
TICKETS – 450 rub.
«Petrovskaya Akvatoria» is the first large historical scale model in Russia at a scale of 1 in 87, 500 square meters in area, opened in September, 2014. Permanent exhibition represents reconstruction of the most significant sights of St. Petersburg and suburbs, related to the city and Russian fleet origin, from Peter the Great’s till Catherine the Great’s era, recreated according to engravings and archive sources.
Unusual format of the museum – scale model – makes it possible to get a bird’s eye view of St. Petersburg.
You may rent binoculars and audio guide.
10:00-22:00 OPEN DAILY

4. The Museum of S.M. Kirov

PETROGRADSKAYA, Kamennoostrovsky Av., 26-28, 5th floor

Kirov's memorial flat – a significant site of the Stalin era. In addition to the memorial five-room apartment (in four of which authentic furniture are fully preserved) you will see two hallways, bathroom and kitchen. In the other rooms of the museum are expositions «The Kirov's office in Smolny","(Thank motherland) For a happy childhood!», and the multimedia exhibition "All Kamennoostrovsky prospect in the box".

Working Hours: Everyday from 11.00 am until 18.00 pm,
Cash Register – from 11.00 am until 17.30 pm
The day off - Wednesday

Prices of the tickets:
Adults – 160 rub.
Student discounts (foreign students included) – 100 rub.