RAČUNALNO RAZMIŠLJANJE I PROGRAMIRANJE U VIŠIM RAZREDIMA OSNOVNE ŠKOLE U SKLOPU NOVOG KURIKULUMA PREDMETA INFORMATIKA

akademik Leo Budin

Povjerenstvo za uvođenje Informatike kao obaveznog predmeta u osnovnoškolski odgoj i obrazovanja

III. webinar edukacije učitelja u sklopu kurikularne reforme
Ministarstva znanosti i obrazovanja
22. siječnja 2018.

Obavezni predmet Informatika
u 5. i 6. razredu osnovne škole

Dana 16. studenoga 2017. godine Vlada Republike Hrvatske donijela je odluku o uvođenju obaveznog predmeta Informatike u 5. i 6. razrede od školske godine 2018./2109.

Predviđeno je da se obavezna nastava u 5. i 6. razredu, te (za sada) izborna nastava u 7. i 8. razredu, obavlja u skladu s odrednicama prijedloga novog kurikuluma predmeta Informatika pripremljenog u sklopu cjelovite kurikularne reforme.

Posebnu pažnju u pripremi učitelja za te promjene treba usmjeriti na kurikularnu domenu *Računalno razmišljanje i programiranje* što je i osnovna tema ovog webinara.

Povodom odluke o uvođenju Informatike kao obaveznog predmeta u 5. i 6. razredu osnovne škole ministrica prof. dr. sc. Blaženka Divjak imenovala je *Povjerenstvo za uvođenje Informatike kao obaveznog predmeta u osnovnoškolski odgoj i obrazovanje* u sljedećem sastavu:

- Lidija Kralj, prof., Ministarstvo znanosti i obrazovanja, predsjednica
- akademik Leo Budin, Hrvatska akademija znanosti i umjetnosti
- prof. dr. sc. Mislav Grgić, posebni savjetnik rektora Sveučilišta u Zagrebu za područje STEM
- prof. dr. sc. Tomislav Jakopec, Filozofski fakultet Sveučilišta u Osijeku
- prof. dr. sc. Nina Begičević Ređep, Fakultet organizacije i informatike Sveučilišta u Zagrebu, Varaždin
- prof. dr. sc. Vedran Mornar, Fakultet elektrotehnike i računarstva Sveučilišta u Zagrebu
- doc. dr. sc. Goranka Nogo, Prirodoslovno matematički fakultet Sveučilišta u Zagrebu
- dr. sc. Jelena Nakić, Prirodoslovno matematički fakultet Sveučilišta u Splitu

- Vedrana Miholić, dipl. inž., CROZ d.o.o., Zagreb
- Biljana Cerin, dipl. inž., Ostende Consulting, Zagreb
- Mario Stančić, VI. osnovna škola Varaždin
- Darka Sudarević, dipl. inž., III. gimnazija Zagreb
- Nikolina Bubica, prof., Osnovna škola Mokošica, Dubrovnik
- Branka Vuk, Hrvatska akademska i istraživačka mreža CARNet, Zagreb
- Viktorija Vranešić, prof., Agencija za odgoj i obrazovanje
- Dejan Drabić, mag. inf., Mladi informatičari Strahoninca
- Jasmina Martinović, Hrvatska udruga poslodavaca
- Bernard Gršić, dipl. ing., Središnji državni ured za razvoj digitalnog društva
- Darko Tot, Ministarstvo znanosti i obrazovanja
- Alen Čuljak, Ministarstvo znanosti i obrazovanja
- Sanja Đurđek Murković, Ministarstvo znanosti i obrazovanja

Zadaća Povjerenstva je analiza postojećeg stanja u osnovnoškolskim ustanovama i predlaganje modela ispunjavanja nužnih preduvjeta za uvođenje Informatike kao obaveznog predmeta u osnovnoškolski odgoj i obrazovanje i to od sljedeće školske godine 2018./2019. (u okviru postojećeg nastavnog plana).

S obzirom da Informatika postaje obavezni predmet u 5. i 6. razredu osnovne škole, nužni preduvjeti moraju biti ispunjeni <u>u svim školama</u>.

U Strategiji obrazovanja, znanosti i tehnologije stoji:

„Misija hrvatskoga obrazovnog sustava jest osigurati kvalitetno obrazovanje <u>dostupno svima pod jednakim uvjetima</u>, u skladu sa sposobnostima svakoga korisnika sustava.”

U dijelu javnosti je uvođenje obaveznog predmeta Informatika izazvalo veliku polemiku i dočekano je čak i ponekim negodovanjem.

Razlozi za takvo negodovanje nisu opravdani i prouzročeni su nepoznavanjem činjenica.

Analiza nastavnog opterećenja ukazuje da se radi o malim promjenama u nastavnom opterećenju koje nikako ne mogu ugroziti odvijanje ostalih obrazovnih aktivnosti.

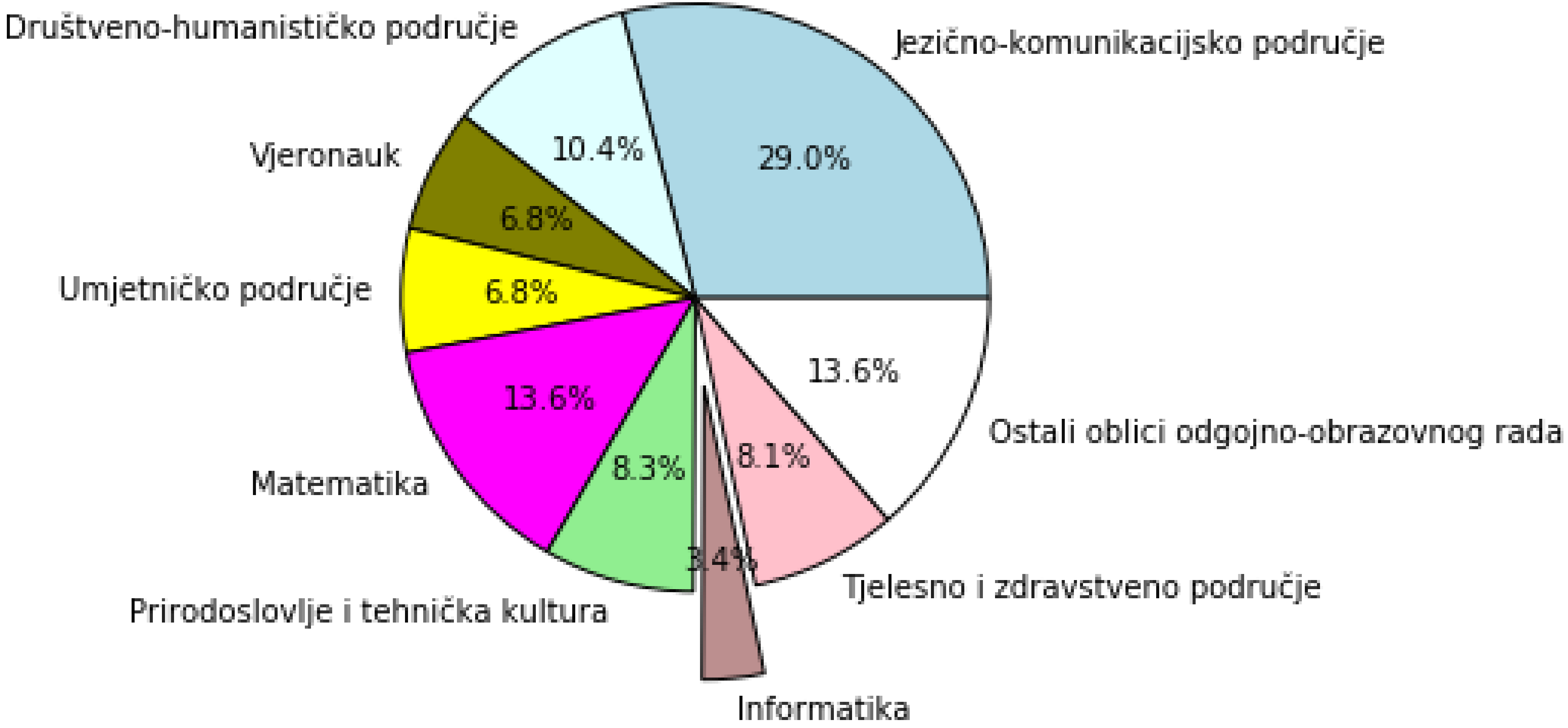
	NASTAVNI PREDMETI	BROJ SATI TJEDNO (GODIŠNJE) PO RAZREDIMA							
		I.	II.	III.	IV.	V.	VI.	VII.	VIII.
OBVEZNI PREDMETI									
1.	HRVATSKI JEZIK	5(175)	5(175)	5(175)	5(175)	5(175)	5(175)	4(140)	4(140)
2.	LIKOVNA KULTURA	1(35)	1(35)	1(35)	1(35)	1(35)	1(35)	1(35)	1(35)
3.	GLAZBENA KULTURA	1(35)	1(35)	1(35)	1(35)	1(35)	1(35)	1(35)	1(35)
4.	STRANI JEZIK	2(70)	2(70)	2(70)	2(70)	3(105)	3(105)	3(105)	3(105)
5.	MATEMATIKA	4(140)	4(140)	4(140)	4(140)	4(140)	4(140)	4(140)	4(140)
6.	PRIRODA					1,5(52,5)	2(70)		
7.	BIOLOGIJA							2(70)	2(70)
8.	KEMIJA							2(70)	2(70)
9.	FIZIKA							2(70)	2(70)
10.	PRIRODA I DRUŠTVO	2(70)	2(70)	2(70)	3(105)				
11.	POVIJEST					2(70)	2(70)	2(70)	2(70)
12.	GEOGRAFIJA					1,5(52,5)	2(70)	2(70)	2(70)
13.	TEHNIČKA KULTURA					1(35)	1(35)	1(35)	1(35)
14.	TJEL. I ZDRAV. KULTURA	3(105)	3(105)	3(105)	2(70)	2(70)	2(70)	2(70)	2(70)
UKUPNO OBVEZNI PREDMETI (REDOVITA NASTAVA)		18(630)	18(630)	18(630)	18(630)	22(770)	23(805)	26(910)	26(910)
IZBORNI PREDMETI									
15.	VJERONAUK	2(70)	2(70)	2(70)	2(70)	2(70)	2(70)	2(70)	2(70)
16.	STRANI JEZIK				2(70)	2(70)	2(70)	2(70)	2(70)
17.	OSTALI IZBOR. PREDMETI					2(70)	2(70)	2(70)	2(70)
UKUPNO IZBORNI PREDMETI (IZBORNA NASTAVA)		2(70)	2(70)	2(70)	4(140)	6(210)	6(210)	6(210)	6(210)
POSEBNI PROGRAM KLASIČNIH JEZIKA									
18.	LATINSKI JEZIK					3(105)	3(105)	3(105)	3(105)
19.	GRČKI JEZIK							3(105)	3(105)
UKUPNO POSEBNI PROGRAMI						3(105)	3(105)	6(210)	6(210)
OSTALI OBLICI ODGOJNO-OBRAZOVNOG RADA									
20.	DOPUNSKA NASTAVA I DODATNI RAD	1+1 (35+35)	1+1 (35+35)	1+1 (35+35)	1+1 (35+35)	1+1 (35+35)	1+1 (35+35)	1+1 (35+35)	1+1 (35+35)
21.	IZVANNASTAVNE DJELATNOSTI	1 (35)	1 (35)	1 (35)	1 (35)	1 (35)	1 (35)	1 (35)	1 (35)
22.	SAT RAZREDNIKA	1(35)	1(35)	1(35)	1(35)	1(35)	1(35)	1(35)	1(35)

Sadašnji nastavni plan za osnovnu školu

U nastavnom programu ne postoje ***ostali izborni predmeti***

Postoji samo program za izborni predmet ***Informatika***

Ukupno 8225 sati iz nastavnog plana (uključujući izborne predmete i ostale oblike odgojno-obrazovnog rada) može se podijeliti na sljedeći način:



Polovina od 3.4% satnog opterećenja Informatike pretvara se u obavezno ! Obavezna Informatika činit će 1.7% ukupnog satnog opterećenja nastavom!

Kurikulum nastavnog predmeta Informatika

Ministarstvo znanosti i obrazovanje je dana 5. siječnja 2018. otvorilo javnu raspravu sa zainteresiranom javnosti o nacrtu prijedloga Odluke o donošenju kurikuluma za nastavni predmet Informatike za osnovne škole i gimnazije u Republici Hrvatskoj.

https://esavjetovanja.gov.hr/Econ/MainScreen?EntityId=6720

Sastavni dio te odluke je i nacrt Kurikuluma nastavnog predmeta Informatika za osnovne i srednje škole s prilozima.

Od priloga tog kurikularnog dokumenta treba posebno istaknuti
Prilog 1.
Odgojno-obrazovni ishodi, razrade ishoda, razine usvojenosti i preporuke za ostvarenje odgojno-obrazovnih ishoda po razredima i domenama s popisom literature

Kurikulum nastavnog predmeta Informatika za osnovne i srednje škole s prilozima izvedenica je dokumenta nastalog u okviru prijedloga cjelovite kurikularne reforme.

Pripremila ga je stručna radna skupina u sastavu *:

- Predrag Brođanac, dipl. ing., V. gimnazija, Zagreb
- Nikolina Bubica, prof., Osnovna škola Mokošica, Dubrovnik
- Zlatka Markučič, dipl. ing., XV. gimnazija, Zagreb (voditeljica)
- Marina Mirković, dipl. ing., Tehnička škola Požega, Požega
- Maristela Rubić, prof., I. gimnazija Split, Split
- Darka Sudarević, dipl. ing., III. gimnazija, Zagreb

* U pripremi izvornog dokumenta voditeljica stručne radne skupine bila je Lidija Kralj, prof.

U nacrtu kurikuluma uvodno se kaže:

Uz tradicionalne znanstvene discipline kao što su matematika, fizika ili kemija, informatika se nameće kao dodatno područje koje je nužno izučavati. Poznavanje temeljnih informatičkih koncepata kao što su programiranje, algoritmi ili strukture podataka postaje neophodno kako ne bismo bili samo korisnici informacijske i komunikacijske tehnologije (IKT) nego i stvaratelji.

Većina poslova 21. stoljeća zahtijeva razumijevanje i primjenu računalne znanosti s ciljem što veće produktivnosti i konkurentnosti. Informatičke kompetencije nužne su u rješavanju različitih izazova u svim područjima ljudskoga djelovanja i u svim područjima znanosti.

U nacrtu kurikuluma dalje piše:

Pod nazivom Informatika u obrazovnom sustavu podrazumijeva se:

- *stjecanje vještina za uporabu informacijske i komunikacijske tehnologije (digitalna pismenost) kojom se oblikuju, spremaju, pretražuju i prenose različiti multimedijski sadržaji;*
- *uporabu informacijske i komunikacijske tehnologije u obrazovnom procesu (edukacijska tehnologija, e-učenje);*
- *rješavanje problema računalom uporabom nekog programskog jezika pri čemu su prepoznatljivi sljedeći koraci: specifikacija i raščlamba problema, analiza problema i odabir postupaka za njegovo rješavanje, priprema i izrada programa, ispitivanje programa i uporaba programa (rješavanje problema i programiranje).*

U nacrtu kurikuluma se posebno naglašava:

Težište obrazovnog procesa u predmetu Informatika treba biti na rješavanju problema i programiranju kako bi se poticalo razvijanje računalnog načina razmišljanja koje omogućuje razumijevanje, analizu i rješavanje problema odabirom odgovarajućih strategija, algoritama i programskih rješenja. Takvi se načini razmišljanja trebaju prenositi i u druga područja posebice matematičko i prirodoslovno, kao i u svakodnevni život.

Nacrt Kurikuluma nastavnog predmeta Informatika za osnovne i srednje škole sadrži *Popis izvora i literature* sa šezdeset navoda.

Iz tog je popisa vidljivo da je stručna radna skupina pažljivo pregledala i uzela u obzir svjetske i posebice europske preporuke za unapređenja informatičkih kurikuluma.

Može se ustvrditi da zadnjih četiri do pet godina postoji cijeli svjetski pokret koji se bavi inoviranjem nastave Informatike.

Tako je u 2016. i 2017. objavljen veći broj dokumenata koji nisu mogli biti uzeti u obzir za pripremu našeg kurikuluma pa ih i nema ih u popisu izvora i literature. U nastavku ove prezentacije komentirat će se neki od tih dokumenata.

Bilo bi vrijedno neke od novih spoznaja iskoristiti za doradu obrazovnih ishoda posebice u domeni Računalno razmišljanje i programiranje.

O nazivlju *

*Prije pregleda nekih odabranih dokumenata potrebno je usuglasiti nazivlje.

Mnogi nesporazumi u raspravama nastaju zbog neusklađenosti naziva.

- Riječ informatika prvi puta se spominje 1957. godine u Njemačkoj u naslovu eseja ***Informatik: Automatische Informationsverarbaitung***.
- Godine 1966. *l'Academie française* službeno odobrava riječ ***l'informatique*** koja označava ***„science du traitement de l'information"***.
- U Njemačkoj se 1968. službeno uvodi naziv ***Informatik*** kao novo znanstveno područje i otvaraju se prvi sveučilišni studiji koji se njime bave.
- U Italiji i Španjolskoj koristi se naziv ***Informatica***,
- Engleski naziv ***Informatics*** rabi se u Velikoj Britaniji i diljem Europe kada se govori i piše engleskim jezikom.
- U S.A.D. se umjesto naziva ***Informatics*** rabi naziv ***Computer Science***. Taj se naziv susreće i u europskim okruženju kao sinonim naziva ***Informatics***.

U hrvatskom se jeziku se pod nazivom **informatika** uobičajeno podrazumjevaju (kako je to uvodno navedeno u kurikularnom dokumentu):

- *vještine uporabe informacijske i komunikacijske tehnologije (**digitalna pismenost**);*
- *uporaba informacijske i komunikacijske tehnologije u obrazovnom procesu (**edukacijska tehnologija**, **e-učenje**);*
- *rješavanje problema računalom uporabom nekog programskog jezika (**računalno razmišljanje i programiranje**).*

U engleskom jeziku se za vještine uporabe informacijske i komunikacijske tehnologije rabi naslov ***digital literacy*** (digitalna pismenost)

Digitalna pismenost stječe se upoznavanjem ***informacijske tehnologije*** (***Infomation Technology, IT***) odnosno ***informacijske i komunikacijske tehnologije*** (***Information and Communication Technology, ICT***).

Sve je više u uporabi i naziv ***computing*** koji obuhvaća kako teorijsku podlogu tako i načela izgradnje i primjenu suvremenih računalnih i komunikacijskih sustava.

U Hrvatskoj je naziv ***computing*** preveden kao ***računarstvo.***

U znanstvenom području ***tehničkih znanosti*** postoje polje ***računarstvo***.

U znanstvenom području ***društvenih znanosti*** postoji polje ***informacijske i komunikacijske znanosti***
(s granama: ***informacijski sustavi i informatologija***, ***organizacija i informatika***, ***informacijsko i programsko inženjerstvo***).

U znanstvenom području ***prirodnih znanosti*** u polju ***matematika*** postoji grana ***matematička logika i računarstvo.***

U svojevrsnom terminološkom kaosu u raznim zemljama rabe se različiti nazivi za istovrsne ili vrlo slične sadržaje. To su:

Computing,
Computational thinking,
Informatics,
Computer science,
Algorithmitic Thinking,
Coding,
...

Međutim, pod tim različitim nazivima pojavljuju se vrlo slični sadržaji:

- ***specifikacija i raščlamba problema (dekompozicija i apstrakcija),***
- ***odabir postupaka za njegovo rješavanje (algoritamsko razmišljanje),***
- ***priprema i izrada programa (programiranje),***
- ***ispitivanje i ispravljanje programa,***
- ***poopćavanje rješenja.***

To su sadržaji koje obuhvaća domena Računalno razmišljanje i programiranje u našem kurikulumu informatike

Informatics Europe

Informatics Europe je udruženje od preko 120 sveučilišnih i istraživačkih institucija iz 30 europskih zemalja. Članice udruženja iz Hrvatske su Fakultet elektrotehnike i računarstva Sveučilišta u Zagrebu i Odjel za informatiku Sveučilišta u Rijeci.

Informatics education:

Europe cannot afford to miss the boat

Report of the joint
Informatics Europe & ACM Europe Working Group
on Informatics Education
April 2013

Governments and the public all too often satisfy themselves that teaching digital literacy is enough to prepare the citizenry for the "Information Society" that Europe has decided to become. **It is not**. Digital literacy is a practical skill, not a scientific topic or an adequate intellectual preparation for the challenges of a digital world.

For a nation or a group of nations to compete in the race for technology innovation, the general population must in addition to digital literacy understand the basics of the underlying discipline, *informatics*[1]. On the road to an information society, informatics plays the same enabling role as mathematics and physics in previous industrial revolutions.

Digital technologies are not "just another technology" like the steam engine, the telegraph, the aeroplane, and penicillin. All other technologies invented by mankind, are technologies that stretch physical abilities. Digital technology and its scientific basis, Informatics, radically challenge the way we think about, understand, and organise the World. The impact on society is pervasive and profound, e.g. politically, economically, legally, medically, scientifically, and educationally.

Therefore, it is of profound importance that Informatics becomes part of general education so that all children are educated to become critical, competent and reflective citizens who can contribute in the broadest sense to shaping the future of our society.

To bring informatics education to the level that their schools deserve, European countries will have to take both long-term and short-term initiatives:

- Universities, in particular through their informatics departments, must put in place comprehensive programs to train informatics teachers, able to teach digital literacy and informatics under the same intellectual standards as in mathematics, physics and other sciences .

- The current chicken-and-egg situation is not an excuse for deferring the start of urgently needed efforts. Existing experiences conclusively show that it is possible to break the deadlock. For example, a recent New York Times article [6] explains how IT companies such as Microsoft and Google, conscious of the need to improve the state of education, allow some of their most committed engineers and researchers in the US to pair up with high school teachers to teach computational thinking. In Russia, it is common for academics who graduated from the best high schools to go back to these schools, also on a volunteer basis, and help teachers introduce the concepts of modern informatics. All these efforts respect the principle that outsiders must always be paired with current high-school teachers.

Report by

The Committee on European Computing Education (CECE)

Jointly established by

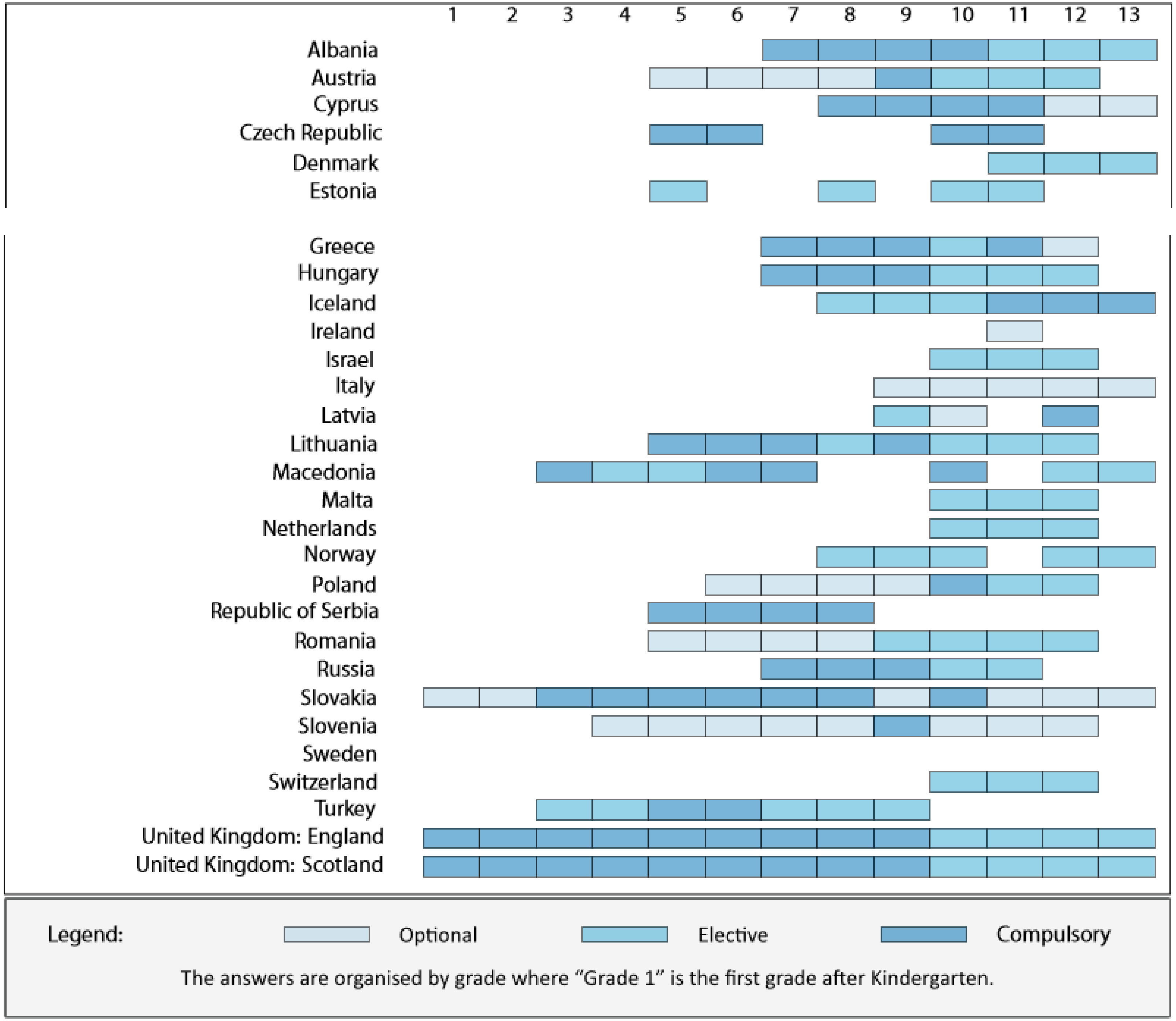
Informatics Europe & **ACM Europe**

May 8th, 2017

Association for Computing Machinery

INFORMATICS EUROPE

acm Europe Council



Predmet Informatika u obaveznom školovanju

Priča iz Ujedinjenog kraljevstva

siječanj 2012. godine

President's Foreword
Sir Paul Nurse FRS

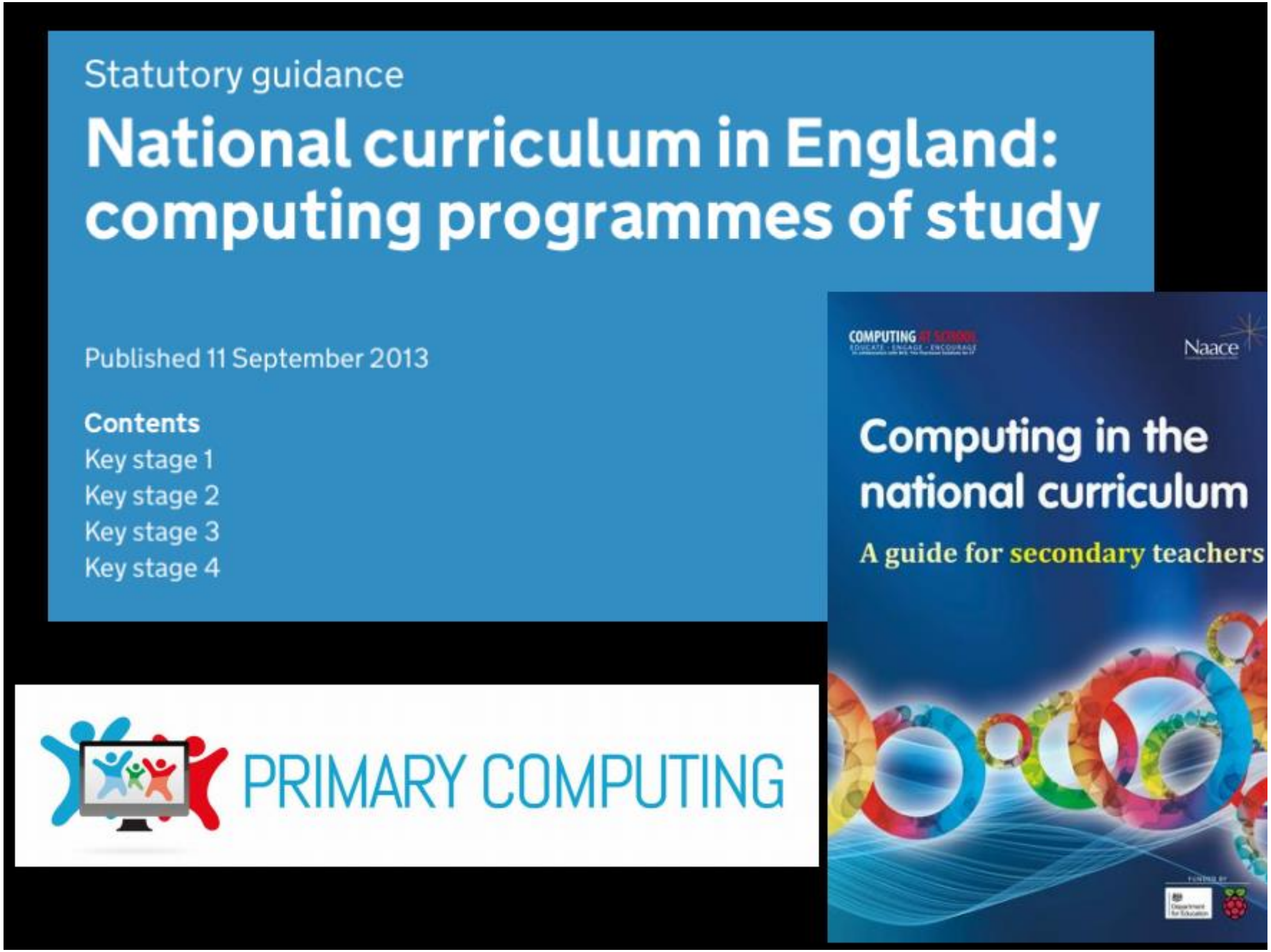
This report analyses the current state of Computing education in schools and sets out a way forward for improving on the present situation. With support from the Royal Academy of Engineering and others the Royal Society has used its 'convening' role to bring together a wide range of distinguished Computer Scientists and stakeholders to explore problems and propose solutions.

Computing is of enormous importance to the economy, and the role of Computer Science as a discipline itself and as an 'underpinning' subject across science and engineering is growing rapidly. This alone is motivation enough, but as this report shows, the arguments for reforming Computing education are not purely utilitarian. It is becoming increasingly clear that studying Computer Science provides a 'way of thinking' in the same way that mathematics does, and that there are therefore strong educational arguments for taking a careful look at how and when we introduce young people to the subject.

The Government has recognised the need for more high quality Computer Science teaching, and has committed to exploring the best ways to achieve this. Our report therefore provides a particularly timely source of evidence that will be needed to inform important policy decisions relating to the National Curriculum in England and to support a drive towards improving Computing education throughout the UK.

I am grateful to Professor Steve Furber for leading this study and to the project Advisory Group for their hard work on this report. I hope that the Royal Society's work will provide a solid foundation which the community can build on, to ensure that the next generation of young people in this country can be creators of technology – not just consumers of it. Just as describing and analysing the problems in this report has been a joint activity, it is clear now that ensuring that the solutions are taken forward is a shared responsibility.

Paul Nurse
President of the Royal Society



rujan 2013. godine

Issued: November 2017 DES4633

ISBN: 978-1-78252-297-3

studeni 2017. godine

RECOMMENDATION 5

Governments should introduce quality-assured computing conversion courses for existing teachers, equivalent to those in physics and mathematics. Individual teachers or schools should not have to contribute to the costs of this training.

RECOMMENDATION 6

Governments should work with higher education providers and the British Computer Society to develop and accredit pre-service subject content courses to enable more people from a wider variety of backgrounds to become computing teachers.

Existing initiatives to support and develop computing degree courses with qualified teaching status should be continued and, if successful, expanded.

RECOMMENDATION 7

Higher education providers need to promote careers in computing education to a wide range of students.

RECOMMENDATION 8

Industry and academia should support and encourage braided[9] careers for staff who want to teach as well as work in another setting.

THURSDAY, JANUARY 4, 2018 JOBS BOARD | ABOUT | SUBSCRIBE | ADVERTISING | CONTACT US | FESTIVAL OF EDUCATION SCHOOLSWEEK

SCHOOLS WEEK

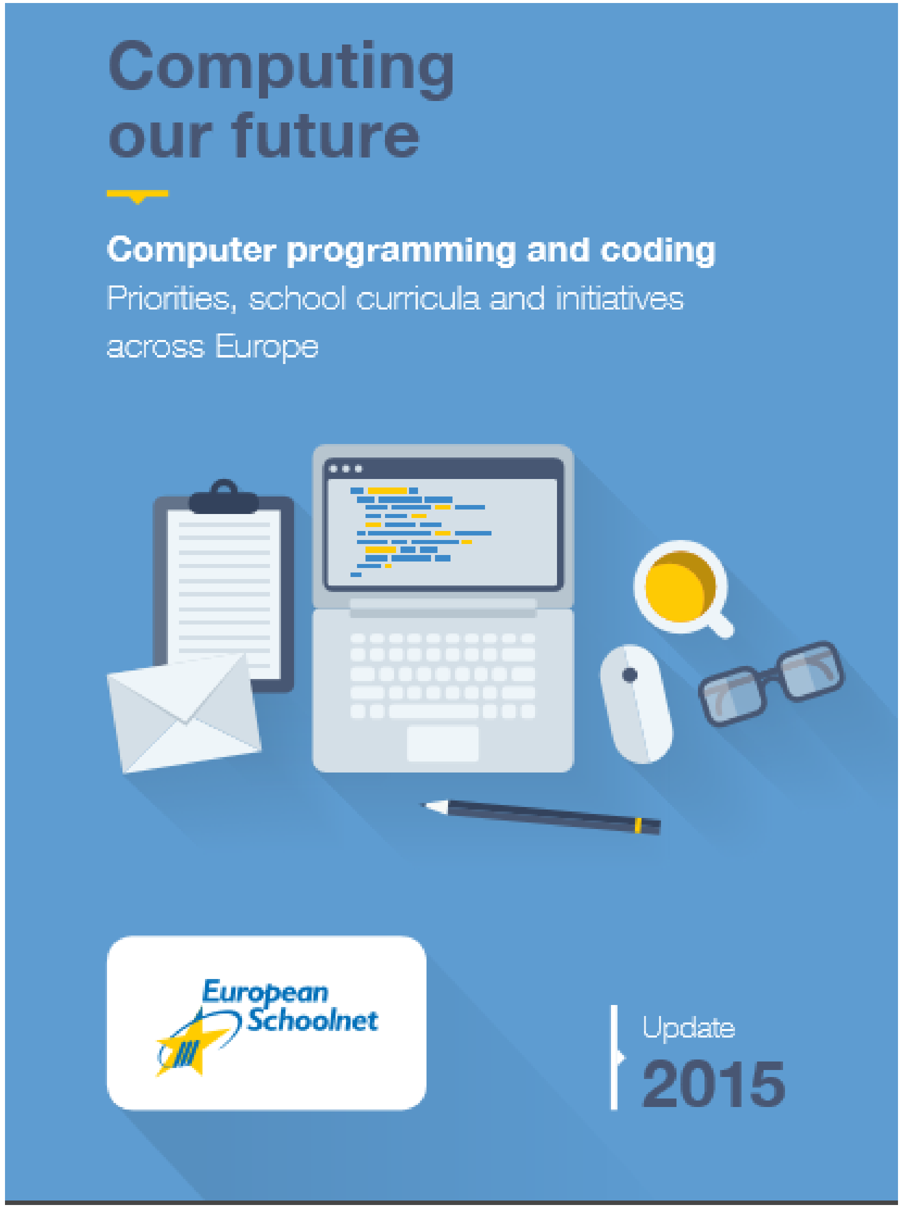
The government will train 8,000 extra computer science teachers at a new £100 million National Centre for Computing, Philip Hammond is expected to announce this week.

In Wednesday's budget, the chancellor will set out his vision for a "hi-tech Britain", and acknowledge the need to train more teachers in computer science.

The shortage of computing teachers was officially recognised in January when the role was added to the "shortage occupation list" by the Migration Advisory Committee, which lifted restrictions on hiring specialists in the subject from outside the EU.

siječanj 2018. godine

Nekoliko europskih dokumenata



COMPUTER PROGRAMMING

Computer programming is the process of developing and implementing various sets of instructions to enable a computer to perform a certain task, solve problems, and provide human interactivity. These instructions (source codes which are written in a programming language) are considered computer programs and help the computer to operate smoothly.

In this report the terms computer programming and coding are used interchangeably, and are in general referred to as coding. They refer to activities that enable children not only to know how to use specific programmes but to learn how to programme computers, tablets, or other electronic devices.

Computational thinking is typically associated with coding and computer programming, but is more than that, involving "solving problems, designing systems, and understanding human behaviour", according to the Carnegie Mellon University.


JRC SCIENCE FOR POLICY REPORT

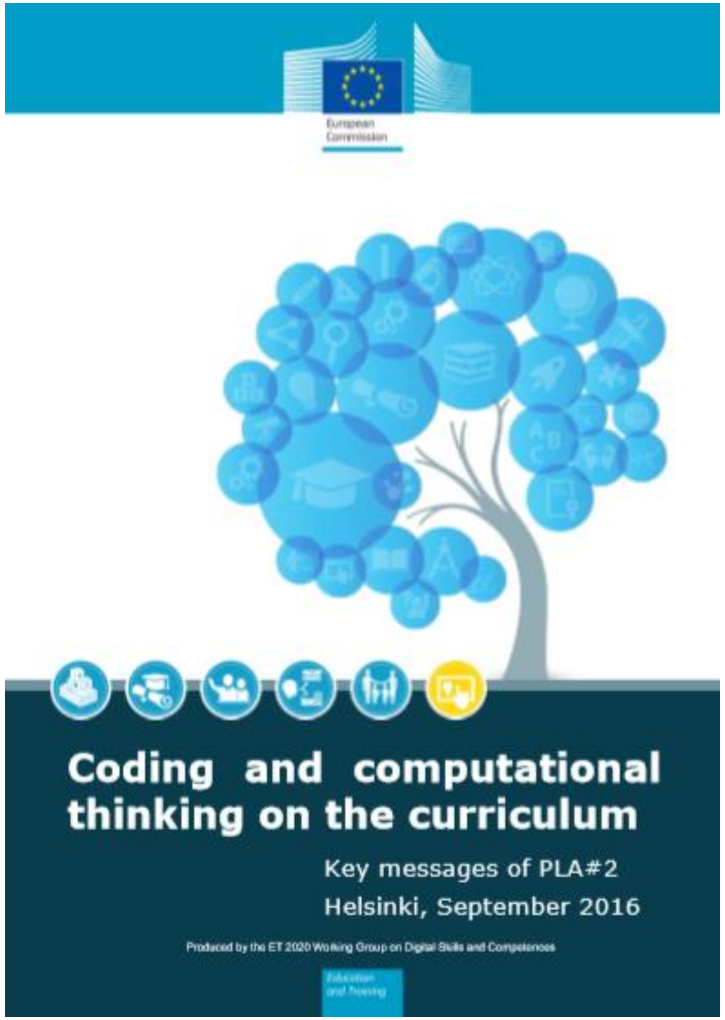
Developing Computational Thinking in Compulsory Education

Implications for policy and practice

Authors: Stefania Bocconi, Augusto Chioccariello, Giuliana Dettori, Anusca Ferrari, Katja Engelhardt
Editors: Panagiotis Kampylis, Yves Punie

2016

A number of prominent institutions inside and outside Europe have intervened in the debate about the introduction of CT skills in compulsory education. In 2012, the Royal Society published the report "*Shut down or restart? The way forward for computing in UK schools*". The Académie des Sciences (2013) intervened on this subject in the report "*L'enseignement de l'informatique en France – Il est urgent de ne plus attendre*". Moreover, Informatics Europe and the Association of Computing Machinery (ACM) Europe, Working Group on Informatics Education (2013), urged Europe "not to miss the boat" on this subject. All those reports call for a change in curricula to make room for CS as a discipline. Industry also supports this position: following the Next Generation Report and Eric Schmidt's 2011 speech on the UK's education, the Department for Education decided to introduce CS teaching in primary and secondary state schools.



Computational Thinking is understood as shorthand for "thinking like a computer scientist", i.e. using concepts of computer science to formulate and solve problems. In the past decade Computational Thinking has increasingly gained attention in the educational field for its potential to teach logical thinking, problem-solving and digital competence.

Teachers need a good understanding of what CT is and how to teach it. Introducing CT requires new training, possibly at large scale, as CT does so so far not often feature in teachers' initial training. Support services for teachers that provide concrete advice and examples can support teachers to use coding and computational thinking in class.

In delivering CT education, teachers' knowledge and starting point should be considered; e.g. the similar focus on logical steps between mathematics and CT can be exploited.

Teachers need examples and good practices relevant for their specific teaching context in order to confidently approach CT.[2] Teaching CT may require new pedagogical approaches that put students at the centre of the learning process.

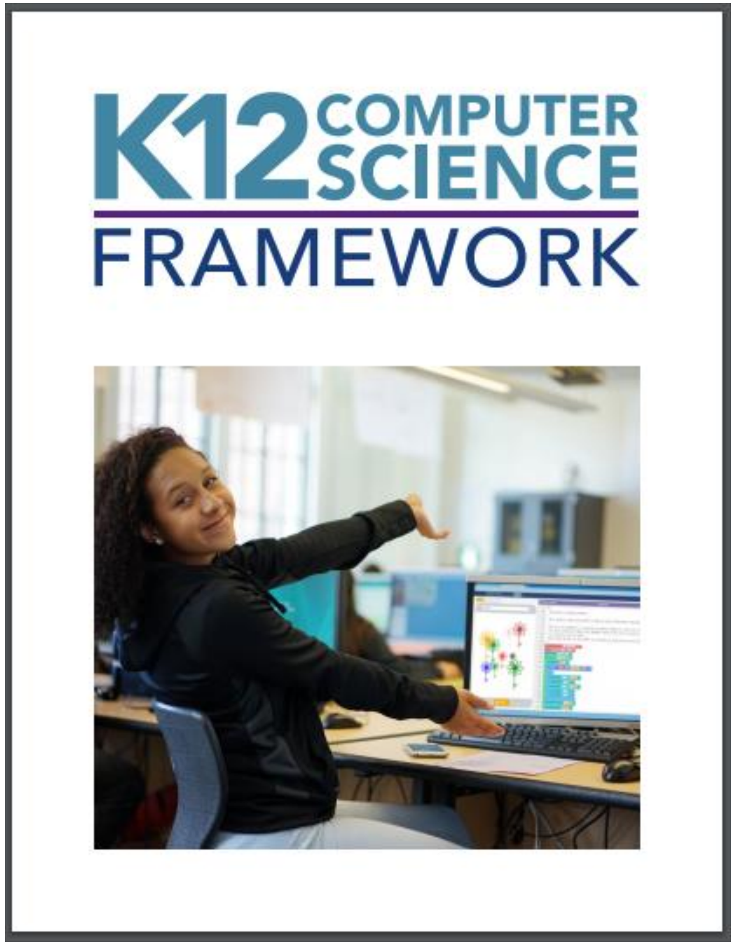
Good quality learning materials for coding/CT are crucial

Good learning materials and tools and guides for delivery of CT are crucial. Naturally the teaching of CT must be adapted for young children; e.g. children learn through games and tasks that can be digital (e.g. visual programming), but could also be paper-based or physical ("unplugged" teaching).

Developing a comprehensive and forward-looking curriculum to engage students across compulsory education is a challenge, in particular as technologies continue to change. Core questions include when and how to start coding/programming in the early years, and at what stage to switch from visual to more textual programming.

To keep children engaged, careful planning and choices of pedagogical approaches and quality learning materials that help enhancing students' understanding of core concepts related to CT sequentially over several school years are necessary; an approach that will need to go much beyond offering a few hours of coding.

Pogled iz S. A. D.



travanj 2016. godine

K–12 Computer Science Framework Steering Committee

Association for Computing Machinery

CODE

CSTEACHERS.ORG
COMPUTER SCIENCE TEACHERS ASSOCIATION

cyber innovation center
Collaboration • Research • Technology

NATIONAL MATH + SCIENCE INITIATIVE

The influence of computing is felt daily and experienced on a personal, societal, and global level. Computer science, the discipline that makes the use of computers possible, has driven innovation in every industry and field of study, from anthropology to zoology. Computer science is also powering approaches to many of our world's toughest challenges; some examples include decreasing automobile deaths, distributing medical vaccines, and providing platforms for rural villagers to participate in larger economies, among others.

Computer science is powering approaches to many of our world's toughest challenges.

As computing has become an integral part of our world, public demand for computer science education is high. Most parents want their child's school to offer computer science (Google & Gallup, 2015), and most Americans believe computer science is as important to learn as reading, writing, and math (Horizon Media, 2015). Many of today's students will be using computer science in their future careers, not only in science, technology, engineering, and mathematics (STEM) fields but also in non-STEM fields (Change the Equation, 2015).

- Americans believe computer science is as important to learn as reading, writing, and math (Horizon Media, 2015).
- Most parents want their child's school to offer computer science (Google & Gallup, 2015b).
- Since 2010, computer science ranks as one of the fastest growing undergraduate majors of all STEM fields (Fisher, 2015), and Advanced Placement (AP®) Computer Science is the fastest growing AP exam, despite being offered in only 5% of schools (Code.org, 2015).
- Jobs that use computer science are some of the highest paying, highest growth (Bureau of Labor Statistics, 2015), and most in-demand jobs that underpin the economy (The Conference Board, 2016).
- Computer science is defined as part of a "well-rounded education" in the Every Student Succeeds Act (2015).

Computational thinking is essentially a problem-solving process that involves designing solutions that capitalize on the power of computers; this process begins before a single line of code is written. Computers provide benefits in terms of memory, speed, and accuracy of execution. Computers also require people to express their thinking in a formal structure, such as a programming language. Similar to writing notes on a piece of paper to "get your thoughts down," creating a program allows people to externalize their thoughts in a form that can be manipulated and scrutinized. Programming allows students to think about their thinking; by debugging a program, students debug their own thinking (Papert, 1980).

Computational thinking refers to the thought processes involved in expressing solutions as computational steps or algorithms that can be carried out by a computer.

Computer Science Practices and Other Subject Areas

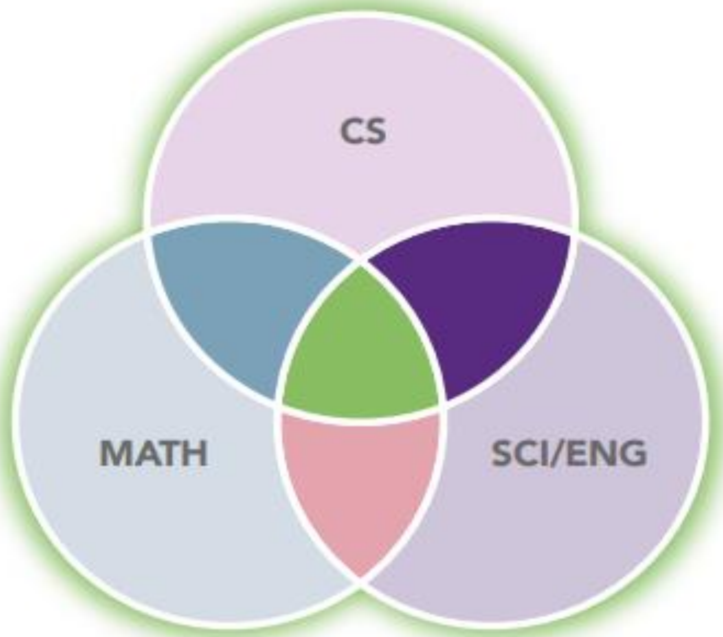
The framework is grounded in the belief that computer science offers unique opportunities for developing computational thinking and that the framework's practices can be applied to other subjects beyond computer science. As Barr and Stephenson (2011) have noted, the "computer science education community can play an important role in highlighting algorithmic problem solving practices and applications of computing across disciplines, and help integrate the application of computational methods and tools across diverse areas of learning" (p. 49).

While computational thinking is a focus in computer science, it is also included in standards for other subjects. For example, computational thinking is explicitly referenced in the practices of many state science standards[1] and implicitly in state math standards.[2] Additionally, the recent revision to the International Society for Technology in Education Standards for Students (ISTE, 2016) describes computational thinking in a similar way as the framework. All of these documents share the vision that computational thinking is important for all students.

Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability (Wing, 2006, p. 33).

CS + Math

- **Develop and use abstractions**
  M2. Reason abstractly and quantitatively
  M7. Look for and make use of structure
  M8. Look for and express regularity in repeated reasoning
  CS4. Developing and Using Abstractions
- **Use tools when collaborating**
  M5. Use appropriate tools strategically
  CS2. Collaborating Around Computing
- **Communicate precisely**
  M6. Attend to precision
  CS7. Communicating About Computing



CS + Sci/Eng

- **Communicate with data**
  S4. Analyze and interpret data
  CS7. Communicating About Computing
- **Create artifacts**
  S3. Plan and carry out investigations
  S6. Construct explanations and design solutions
  CS4. Developing and Using Abstractions
  CS5. Creating Computational Artifacts
  CS6. Testing and Refining Computational Artifacts

CS + Math + Sci/Eng

- **Model**
  S2. Develop and use models
  M4. Model with mathematics
  CS4. Developing and Using Abstractions
  CS6. Testing and Refining Computational Artifacts
- **Use computational thinking**
  S5. Use mathematics and computational thinking
  CS3. Recognizing and Defining Computational Problems
  CS4. Developing and Using Abstractions
  CS5. Creating Computational Artifacts
- **Define problems**
  S1. Ask questions and define problems
  M1. Make sense of problems and persevere in solving them
  CS3. Recognizing and Defining Computational Problems
- **Communicate rationale**
  S7. Engage in argument from evidence
  S8. Obtain, evaluate, and communicate information
  M3. Construct viable arguments and critique the reasoning of others
  CS7. Communicating About Computing

Teacher Development

Teacher development is a critical part of the computer science education infrastructure. Teacher development is used here as a broad term that includes preservice teacher preparation, certification, licensure, and ongoing professional development. It concerns stakeholders in higher education, state agencies, school districts, and organizations that provide professional development.

Professional development should attend to novice teachers' anxiety over their lack of content knowledge.

Given the introduction of computer science into many education systems, it is natural that many teachers attending professional development may not already have a background in computer science. While not diminishing the importance of pedagogical content knowledge or general pedagogical practice for teaching computer science, professional development providers should attend to teachers' anxiety about content knowledge by helping them see that many teachers are in the same situation. Professional development can instill a growth mindset in participants, in which learning builds over time, during a workshop as well as the school year while teachers deliver instruction. Professional development should be viewed as a safe space to try new or difficult things.

More Teachers, Fewer 3D Printers: How to Improve K–12 Computer Science Education

By Prachi Patel
Posted 2 Oct 2017 | 13:00 GMT

Last week, the Trump White House announced $200 million in federal funding to improve K–12 computer science education every year. The next day, tech leaders including Amazon, Facebook, Google, Microsoft, and Salesforce added another $300 million to the bag, spread over five years. The goal of that initiative is to support K–12 STEM education, focusing on computer science.

Where the money should not be spent? On hardware and equipment. Laptops, robots, and 3D printers are important, says Code.org's Yongpradit, "but they don't make a CS class. A trained teacher makes a CS class. So money should be focused on training teachers and offering robust curriculum."

listopad 2017. godine

Odabir programskog jezika

Računala su elektroničke naprave koje mogu obavljati raznovrsne poslove.

Poslove koje računalo mora obaviti opisujemo računalnim programima koji se pripremaju uporabom programskih jezika.

Ti se jezici razlikuju od jezika kojima se ljudi služe u međusobnoj komunikaciji. Programski jezici se sastoje od *skupa naredbi* ili *instrukcija* kojima ljudi određuju koje operacije mora obaviti računalo.

Niz naredbi kojima naređujemo obavljanje nekog složenijeg posla nazivamo *računalnim programom*, a postupak pripreme računalnog programa zovemo *programiranjem*.

Tradicionalni pristup poučavanju programiranja polazio je uobičajeno od opisa svojstava računalnog sustava, upoznavanja internog prikaza načina pohranjivanja osnovnih tipova podataka i detaljnog upoznavanja sintakse programskog jezika. Tek nakon toga pripremali su se jednostavni programi koji su služili su prvenstveno za ilustraciju svojstava pojedinih elemenata programskog jezika.

Moderni programski jezici i njihovo radno okruženje omogućuju da poučavanje programiranja može početi s izučavanjem pojedinih naredbi i postupnom izgradnjom vlastiti ***misaonog modela*** računalnog sustava koji objašnjavanja djelovanje te naredbe.

Takav misaoni model računalnog sustava (engl. ***notional machine***) postepeno se izgrađuje u glavi svakog učenika i prilagođava razini složenosti problema.

Zanimljivo je pogledati neke od radova koji se bave tom temom.

The State Of Play: A Notional Machine for Learning Programming·

Berry, Michael and Kölling, Michael (2014) The State Of Play: A Notional Machine for Learning Programming. In: The 19th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2014), June 2014, Uppsala, Sweden.

It is well understood that programming is a fundamental activity in computer science; it is the process by which conceptual ideas are mapped to instructions that can be understood and interpreted by a machine. The teaching of introductory programming within computer science is essential, and mastery of this skill necessary for students to progress. To be successful in programming, students have to be able to form a valid and consistent mental model of the machine executing their instructions. Forming such a model is not easy, and the computing education community has no agreed, shared abstract model in widespread use. Often, ad-hoc models are formed by instructors or students, but these are not guaranteed to be consistent or correct. A shared, accepted and valid mental model – a notional machine – would benefit both instructors and students in their attempts to teach and learn programming.

Notional Machines and Introductory Programming Education

JUHA SORVA, Aalto University

Article *in* ACM Transactions on Computing Education · June 2013

DOI: 10.1145/2483710.2483713

Since a notional machine is tied to a way of programming, different kinds of programming languages will have different notional machines. An object-oriented Java notional machine can be quite different from a functional Lisp notional machine. Most notional machines that execute Prolog are likely to be quite different again. Similar languages may be associated with similar or even identical notional machines. Some notional machines may not be very 'machine-like' at all if they are based on, for example, mathematics and lambda calculus.

Not only are there different notional machines for different languages and paradigms, but even a single language can be associated with different notional machines. After all, there is no one unique abstraction of the computer for describing the execution of programs in a language. Let us consider, for instance, the following ways of understanding the execution of Java programs.

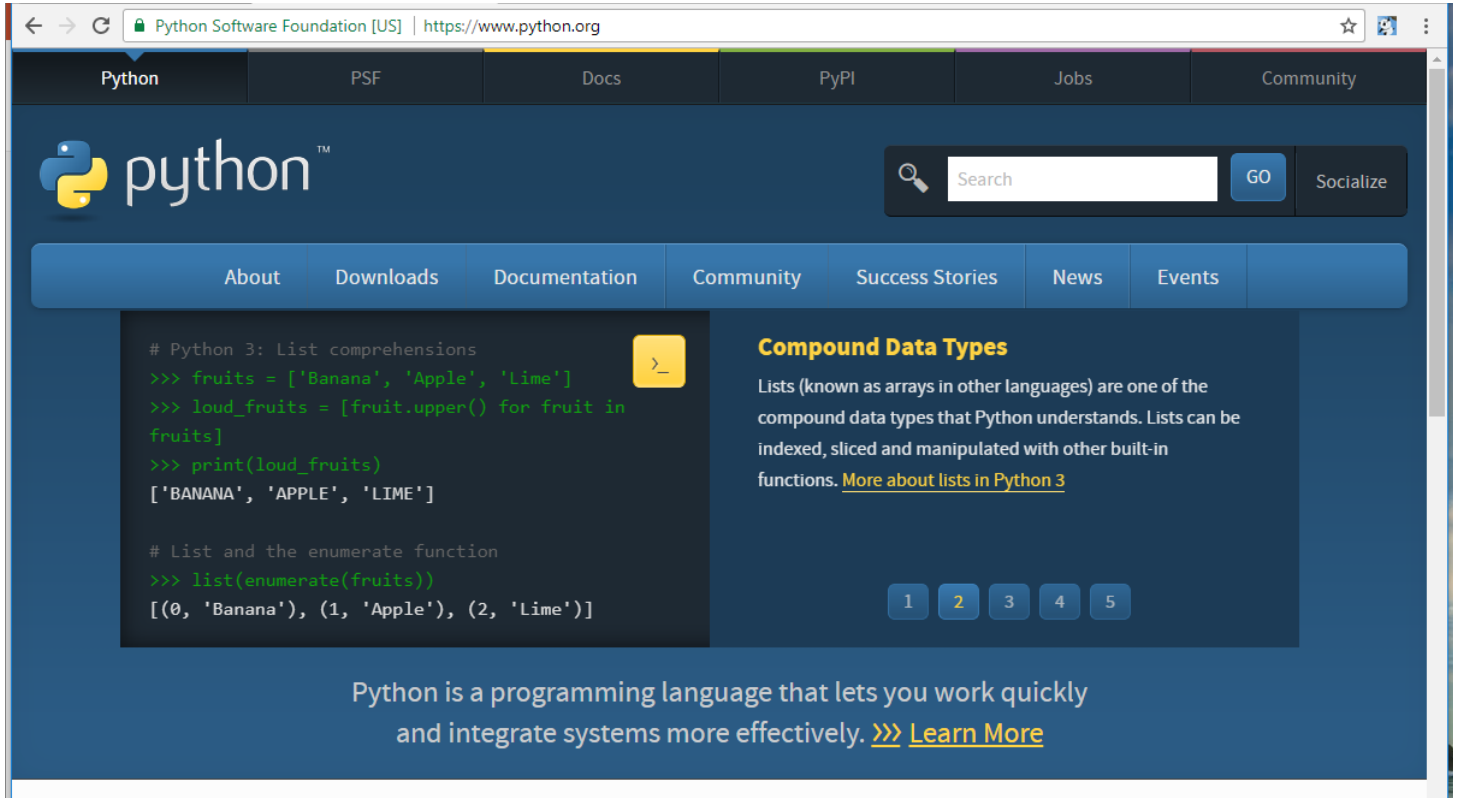
To summarize, a notional machine:

—is an idealized abstraction of computer hardware and other aspects of the runtime environment of programs;
—serves the purpose of understanding what happens during program execution;
—is associated with one or more programming paradigms or languages, and possibly with a particular programming environment;
—enables the semantics of program code written in those paradigms or languages (or subsets thereof) to be described;
—gives a particular perspective to the execution of programs; and
—correctly reflects what programs do when executed.

S obzirom na to da misaoni model ovisi o svojstvima programskog jezika, prikladno je za početno učenje programiranja odabrati jedan programski jezik i sustavno ga rabiti u cijelom razdoblju učenja.

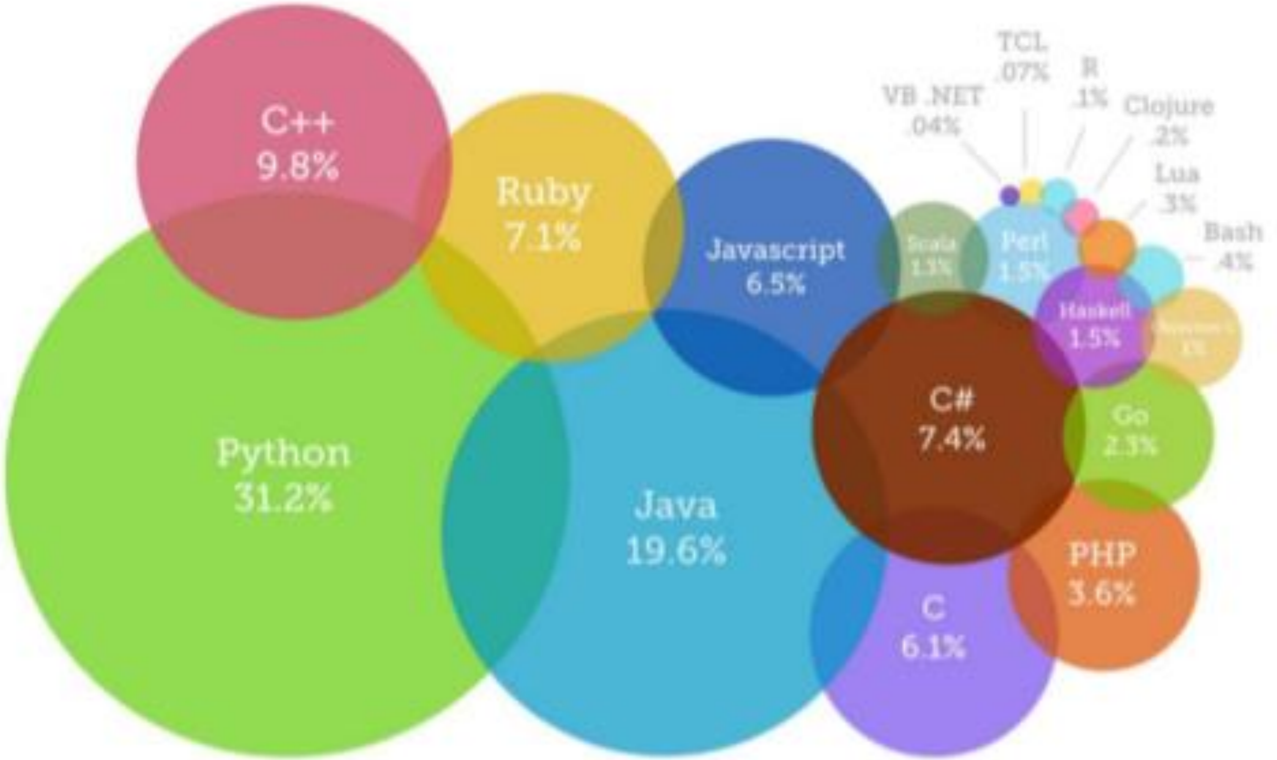
Bilo bi vrlo korisno da odabir jezika bude takav da se on može rabiti i kasnije u višim razinama obrazovanja.

Za poučavanje programiranja u višim razredima osnovne škole, zadnjih se nekoliko godina prikladnim pokazao programski jezik *Python*. On je postao i najrašireniji jezik za početno učenje programiranja.



Programsko okruženje za pripremu i izvođenje programa dobavlja se slobodno i besplatno s mrežne stranice **`http://www.python.org`**

Pregled uporabe programskih jezika u nastavi programiranja (2017. godina)



Python se sve više koristi i za profesionalne primjene. Tako je *Python* jedan od tri službena programska jezika u tvrtki *Google*. *Youtube* je u velikom svom dijelu razvijen u *Pythonu*, a u svemirskoj agenciji NASA mnogi su primjenski programi također pripremljeni u *Pythonu*.

Preporuka
Povjerenstva za uvođenje informatike kao obaveznog predmeta u osnovnoškolski odgoj i obrazovanje

Uz opći zaključak da domena *Računalno razmišljanje i programiranje* mora biti najznačajnije zastupljena u kurikulumu predmeta *Informatika* u osnovnoj školi dodatno se naglašava da za pripremu programa treba odabrati programski jezik *Python*. On je postao i najrašireniji jezik za početno učenje programiranja. Razlozi zato su višestruki:

- *Python* ima čitku i jasnu sintaksu te se programi pisani u *Pythonu* lako čitaju i imaju veliku pedagošku vrijednost.
- Programsko okruženje potiče pisanje programa s jasno izraženom logičkom strukturom programa.
- Podjela problema na manje dijelove i njihovo međusobno povezivanje u cjelovito rješenje obavlja se vrlo jednostavno.
- Programsko okruženje *Pythona* omogućuje jednostavan i brz prijelaz iz faze pisanja u fazu ispitivanja programa i obrnuto.

Stručno usavršavanje učitelja

Učenje programiranja potiče konstrukcijski pristup učenju

Pokazalo se da učenjem programiranja uporabom prikladnog programskog jezika i radnog okruženja za pripremu programskih rješenja učenici razvijaju načine razmišljanja kompatibilne s općim osnovnim postavkama konstrukcijskog modela učenja primjenljivog u ostalim problemskim domenama.

Pokazalo se da aktivno učenje programiranja potiče:

- temeljito razmišljanje, precizno izražavanje i formalni opisa problema (jer se programi zasnivaju na dobro razrađenim algoritmima) ;
- razumijevanje osnovnih koncepata kao što su: formalne procedure, funkcije i varijable (jer se na tim konceptima izgrađuju računalni programi) ;
- heurističke pristupe rješavanju problema kao što su: izrada plana rješavanja, prepoznavanje sličnosti s nekim već riješenim problemima, dekompozicija složenih problema na manje dijelove (jer se u postupcima programiranja postupa na taj način) ;

- postupno pronalaženje rješenja problema metodom pokušaja i pogrešaka (jer se priprema računalnog programa obavlja upravo na taj način – do uspješnog računalnog programa dolazi se najčešće tako da se postupno otklanjaju pogreške u prvotno skiciranoj varijanti programa) ;
- pronalaženje rješenja problema na način da se međusobno povezuju prethodno razrađeni i provjereni manji dijelovi rješenja (jer se računalni programi grade tako da se međusobno povezuju prethodno pripremljene i ispitane funkcije) ;
- prepoznavanje mogućnosti da se neki problem može riješiti na više načina te da se odabir najboljeg rješenja može obaviti temeljem odgovarajućih kriterija usporedbe (jer se pri programiranje između više mogućih rješenja odabire ono koje je po nekim svojstvima – najčešće po trajanju izvođenja ili potrošnji memorije – najprikladnije).

Zbog svega navedenog ustanovljeno je da se učenjem programiranja potiče razvitak računalno-algoritamskog načina razmišljanja (engl. *computational thinking*) koje omogućuje razumijevanje, analizu i rješavanje problema odabirom odgovarajućih strategija i programskih rješenja.

Takav način razmišljanja nadovezuje se na matematički način razmišljanja (engl. *mathematical thinking*) koji se sustavno mora razvijati u matematici. Takvi se načini razmišljanja moraju prenositi i u druga područja, posebice u područje prirodoslovlja, kao i u praktični život.

Inovativni pristup oblikovanju nastave informatike, posebice u osnovnoj školi, svakako bi se trebao odraziti i na poboljšavanje PISA rezultata hrvatskih učenika u području matematike i u području prirodoslovlja.

Zbog toga je potrebno stručno usavršavanje provesti ne samo za učitelje informatike već i za učitelje tehničke kulture, matematike i fizike koji moraju steći potrebna znanja i vještine u rješavanju problema iz svojih domena računalnim razmišljanjem programiranjem.

J Sci Educ Technol (2016) 25:127–147
DOI 10.1007/s10956-015-9581-5

Defining Computational Thinking for Mathematics and Science Classrooms

David Weintrop[1,2] · Elham Beheshti[3] · Michael Horn[1,2,3] · Kai Orton[1,2] ·
Kemi Jona[2,3] · Laura Trouille[5,6] · Uri Wilensky[1,2,3,4]

Abstract Science and mathematics are becoming computational endeavors. This fact is reflected in the recently released Next Generation Science Standards and the decision to include "computational thinking" as a core scientific practice. With this addition, and the increased presence of computation in mathematics and scientific contexts, a new urgency has come to the challenge of defining computational thinking and providing a theoretical grounding for what form it should take in school science and mathematics classrooms. This paper presents a response to this challenge by proposing a definition of computational thinking for mathematics and science in the form of a taxonomy consisting of four main categories: data practices, modeling and simulation practices, computational problem solving practices, and systems thinking practices.

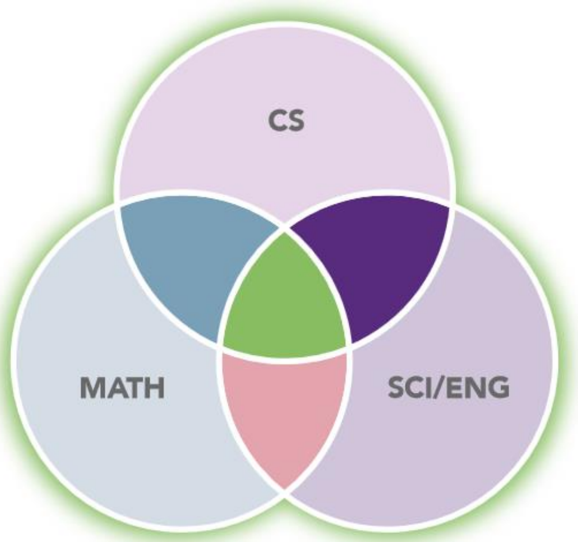
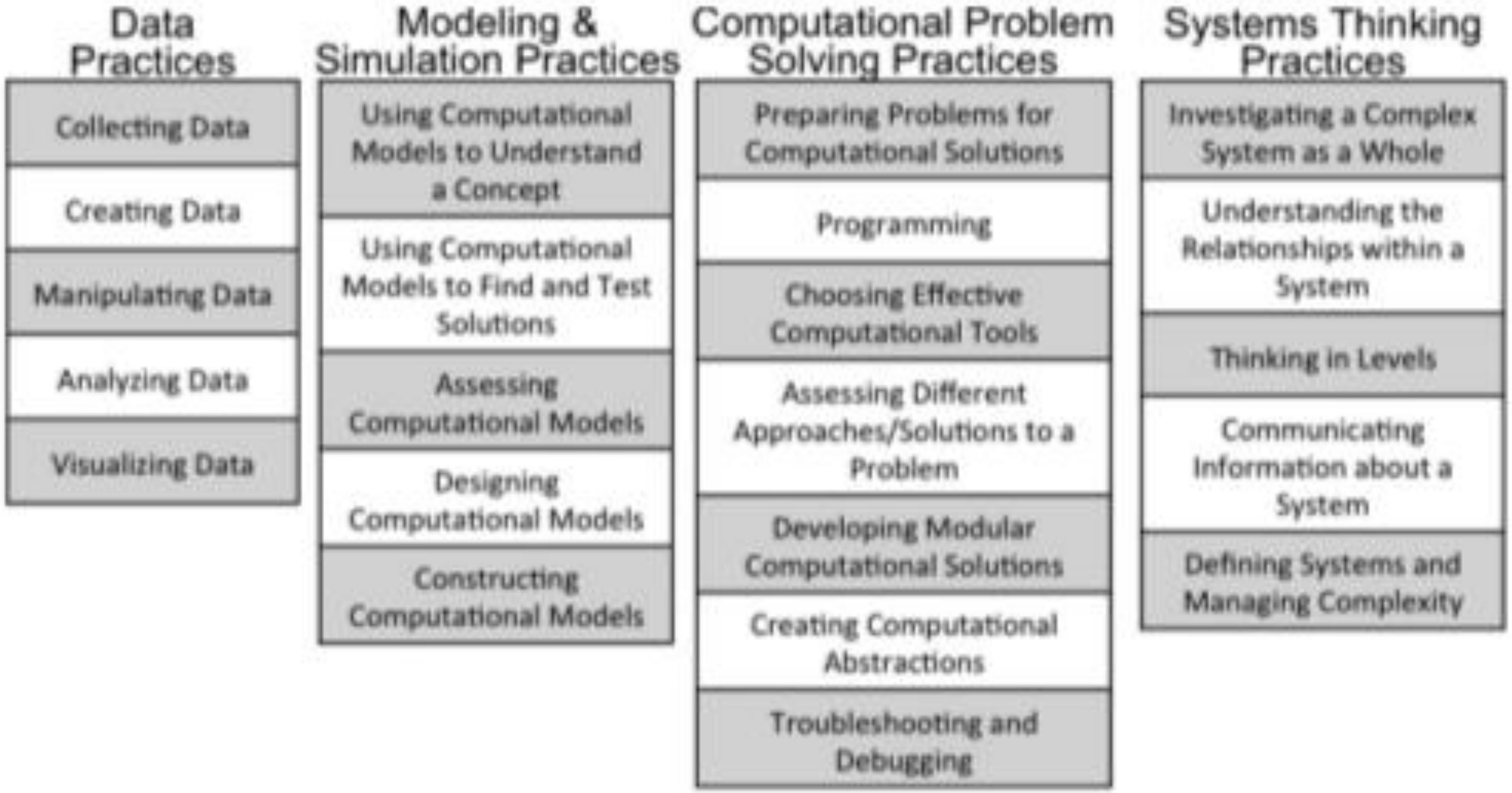


Fig. 2 Computational thinking in mathematics and science taxonomy



Preporuka
Povjerenstva za uvođenje informatike kao obaveznog predmeta u osnovnoškolski odgoj i obrazovanje

Plan stručnog usavršavanja postojećih učitelja

Plan stručnog usavršavanja treba načiniti za dvije skupine postojećih učitelja:

- učitelje informatike,
- učitelje ostalih premeta STEM područja.

Za učitelje informatike stručno usavršavanje treba obuhvatiti:

1) upoznavanje s novim kurikulumom iz nastavnog predmeta Informatika i preporuke oko organizacije i pripreme za provedbu u svim domenama ,
2) cjelovitu edukaciju u domeni *Računalno razmišljanje i programiranje*,
3) pripremu učitelja za mentoriranje učenika na informatičkim natjecanjima.

Za sve učitelje ostalih predmeta iz STEM područja stručno usavršavanje treba obuhvatiti:

4) cjelovitu edukaciju u domeni *Računalno razmišljanje i programiranje* u okviru koje učitelje treba pripremiti za primjenu računalnog razmišljanja i programiranja u konstrukcijskom pristupu poučavanju u njihovim predmetima.

U praktičnoj provedbi stručnog usavršavanja mogu se objediniti točke 2) i 4). Štoviše, takva organizacija stručnog usavršavanja može imati vrlo pozitivan sinergijski učinak.

Povjerenstvo preporuča da Ministarstvo znanosti i obrazovanja i Agencija za odgoj i obrazovanje (u suradnji sa sveučilišnim institucijama) pripremi detaljni plan provedbe posebnog projekta stručnog usavršavanja.

Završne napomene

Nacrt kurikuluma predmeta Informatika predviđa da se na svim razinama predvisokoškolskog obrazovanja u nastavi razrađuje domena računalnog razmišljanja i programiranja.

U ovom je izlaganju naglasak stavljen na više razrede osnovne škole, tj. na razdoblje od 5. do 8. razreda osnovne škole.

Neposredan povod za razmatranje je uvođenje Informatike kao obaveznog predmeta u 5. i 6. razredu, no nastavne sadržaje treba planirati i za 7. i 8. razrede i to bez obzira na to da li će u tim razredima Informatika ostati izborni ili će u doglednoj budućnosti postati obavezni predmet.

Ustanovljeno je da je za to razdoblje školovanja u današnje vrijeme najprikladniji programski jezik *Python*.

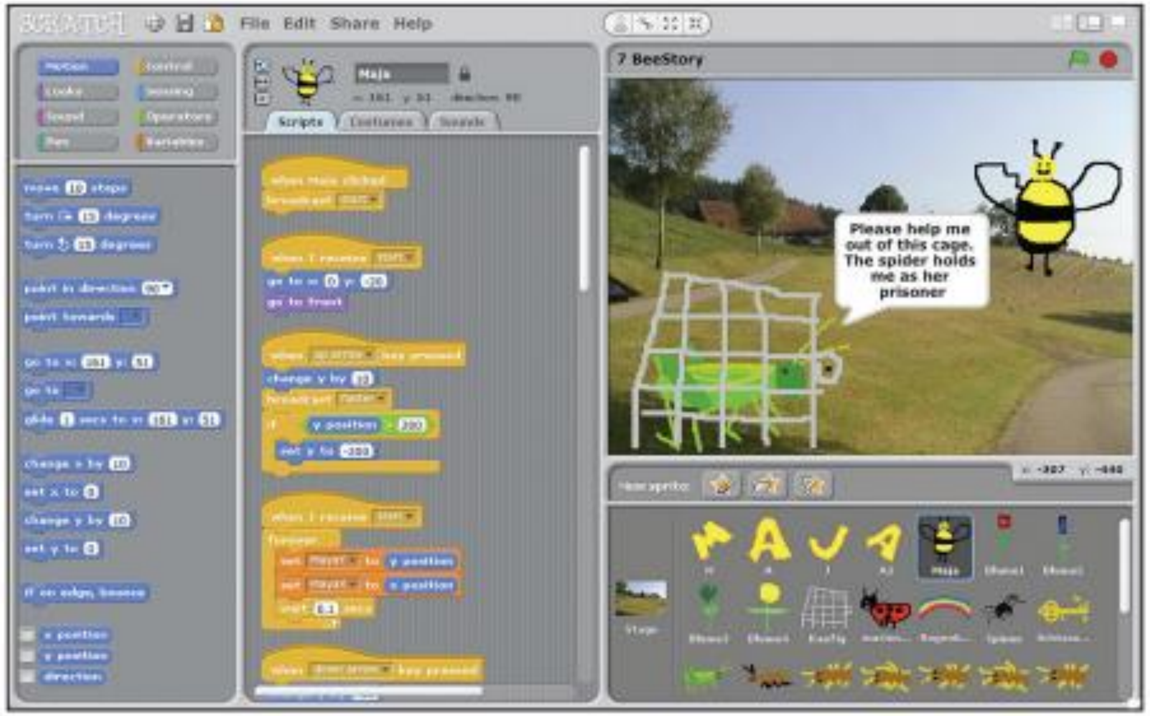
Postavlja se pitanje kako domenu *Računalno razmišljanje i programiranje* obraditi u nižim razredima osnovne škole i u srednjim školama ?

U već spomenutom dokumentu Europske komisije se ustanovljuje:

In order for CT to be integrated comprehensively across all levels of compulsory education, it is necessary to define a clear vision and set specific goals. As CT involves far more than offering a few hours of coding, placing it in the curriculum calls for a robust strategy that takes into account the wide range of factors involved. A key consideration is the extent to which CT is allocated across the full spectrum of subject area studies and, also, in multi-disciplinary and inter-disciplinary contexts. Introducing CT concepts to children early on in school is commonly held to be desirable. These considerations call for a holistic approach to CT integration in compulsory education, which embraces essential aspects such as suitable assessment strategies and adequate teacher training.

Rasprostranjeno je mišljenje da je ***za niže razrede osnovne škole*** prikladno rabiti programske jezike s grafičkim sučeljem koje omogućuje da se program sastavlja tako da se pripremljeni blokovi naredbi kombiniraju u cjelinu pomicanjem pripadnih sličica.

Danas je najrašireniji takav jezik je ***SCRATCH*** razvijen na ***Massachusetts Institute of Technology (MIT)*** koji se može besplatno dabaviti preko mrežnog mjesta **`https://scratch.mit.edu/`**



U gimnazijama i u većini ostalih srednjih razumno je zadržati programski jezik *Python* jer će iskustvo stečeno u osnovnoj školi olakšati računalno razmišljanje pri rješavanju problema više razine složenosti.

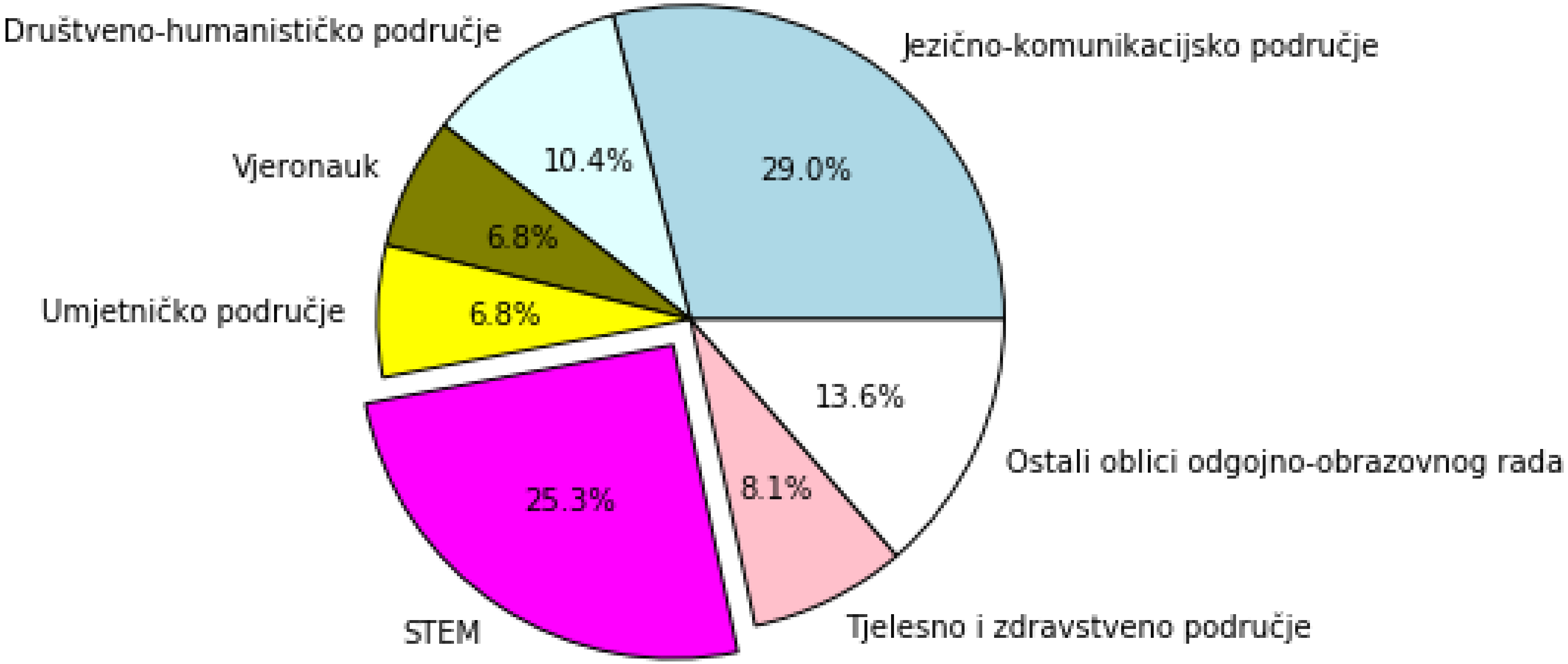
Python se pritom može nadograđivati slobodno dostupnim posebnim specijaliziranim modulima za pojedine vrste problema kao što su:

- numpy – modul za numeričku matematiku
- scipy – modul s naprednim matematičkim postupcima
- pandas – modul za statističku obradu podataka
- matplotlib – modul za pripremu dvodimenzionalnih grafičkih prikaza

Zainteresiranima iz srednjih škola preporuče se pogledati mrežno mjesto **`https://www.anaconda.com/download/`**.

U srednjim školama koje pripremaju učenike za informatičke struke treba uvoditi programske jezike koji se pretežito koriste za pripremu profesionalnih programskih rješenja (*C*, *C++*, *C#*, Java, Javascript, HTML i druge).

STEM područja čine 25.3% nastavnog opterećenja u sadašnjoj osnovnoj školi



Malo je vjerojatno da će se u novom kurikulumu taj udio moći promijeniti! Zbog toga sadržaj predmeta Informatika mora biti pažljivo odabran tako da podupire matematičku i prirodoslovnu pismenost učenika.