МАТЕМАТИКА И МАТЕМАТИЧЕСКО ОБРАЗОВАНИЕ, 2003
MATHEMATICS AND EDUCATION IN MATHEMATICS, 2003
Proceedings of the Thirty Second Spring Conference of
the Union of Bulgarian Mathematicians
Sunny Beach, April 5–8, 2003

COMPUTING CURRICULA 2001 ... КЪДЕ СМЕ НИЕ?

Павел Азълов, Аврам Ескенази

В настоящия доклад накратко са представени основните резултати от новата учебна документация по информатика за висшите училища, известна като CC2001. Разработена е от IEEE и ACM и е публикувана в края на 2001 г. Преминала през широко обсъждане, днес CC2001 е вече работен материал при разработката на учебни планове в много университети по света. В доклада са представени и основните държавни параметри, по които са разработени учебните планове по информатика в нашите университети. Формулирани са няколко общи проблема, отнасящи се до учебните планове и тяхната реализация, за които се предполага дискусия по време на конференцията.

1. Увод. Проблемите на образованието по математика и информатика са били винаги приоритетни теми на Съюза на Математиците в България (СМБ). Точно за това с настоящия доклад на конференция на СМБ искаме да представим и дискутираме основните съвременни тенденции на университетското образование в първата (бакалавърска) степен по информатика. Това е тема, добре позната на колегията от академичните среди и е дискутирана многократно на различни форуми. Повод за поредното ѝ разглеждане е разработката на нова учебна документация по информатика, известна със съкращението CC2001 (Computing Curriculum 2001) [1]. Тя е съвместен проект на две световно признати огранизации в областта на инфорамтика – IEEE (The Computer Society of the Institute for Electrical and Electronic Engineers) и ACM (The Association for Computing Machinery). Има и един втори не по-малко важен повод. През 1997 год. у нас беше приета наредба за единните държавни изисквания за придобиване на висше образование по специалностите от професионално направление „Математика“ [2], към което се причислява и информатиката. По тази наредба бяха изградени учебните планове на катедрите по информатика в нашите университети. По тях учиха и вече завършиха първите бакалаври по информатика. През 2002 год. наредбата от 1997 е отменена и вече се създадоха или сега се изграждат нови учебни планове по информатика. Тъкмо сега е времето да се направи анализ на учебните планове, които минаха през пълния цикъл на реално използване. А може би е и време да се осъвременят, на базата на дискутираната CC2001?

В т. 2 на настоящият доклад се прави много кратко описание на CC2001, който в пълния си обем съдържа 240 страници. За по-подробни справки препоръчваме директно използване на оригиналния текст от [1]. Във т.3. на доклада са дадени основните параметри на държавните изисквания за специалността информатика,

така както са публикувани в [2]. В края на доклада са формулирани няколко групи от актуални въпроси на обучението по информатика.

1.1. Кратки исторически сведения. Историята на информатиката като университетско образование е малка по години, но динамичното развитие на информатиката като научна и приложна област породи изключително голяма динамика в учебните планове и информатичните дициплини. Първите учебни планове по информатика се разработват в началото на 60-те години. През 1968 год. ACM публикува доклад за академично образование по информатика, известно като *Curriculum'68*. В този доклад се правят препоръки за разработка на академични програми, подробно се описват някои основни дисциплини и се дава обширна за времето си библиография. Десет години по-късно този доклад е изместен от появилия се нов, много по-обширен и задълбочен доклад на ACM, *Curriculum'78*. В него, освен общите препоръки за учебен план, се предлагат и стандарти за съдържание на редица основни курсове по информатика. През 1981 год. със съвместните усилия на ACM и IEEE е раработен нов учебен план, публикуван под заглавието Computing Curricula 1991 [CC91]. CC91 бележи съществено нов елемент в структурта на изграждането на учебните планове. В него се формулират основните учебни *микромодули**, които трябва задължително да влязат в учебните планове по информатика. Учебният микромодул е структурна единица, която обединява няколоко единици от знания. Обикновено един микромодул се състои от няколко теми, а една дисциплина се състои от няколко микромодула. Възможно е също така един микромодул да бъде „разпределен“ в няколко дисциплини. В CC91 микромодулите не са обвързани с конкретни дисциплини, което позволява голяма свобода при изграждането на учебните планове.

Разнообразни са причините за появата на CC2001, но те могат да се обособят в две основни групи:

• *Технологични промени.* Големите промени в компютърните и комуникационните технологи са една от съществените причини за силните промени в учебните планове на университетската информатика. Много от трудните проблеми, решавани преди, поради съществено по-мощната компютърна техника сега са вече леко решими. Това промени общата представа за информатиката, промени се посоката на научните и приложните интереси. Създадоха се нови изследователски области, които рефлектираха в нови учебни модули и дисциплини. Като пример само ще спомена, че XML технологията се появи само преди няколко години, но вече навлезе в учебните планове.

• *Културни промени.* Преподаването на информатиката се промени коренно през последното десетилетие (презентации с компютър и прожектор, дистанционно обучение, и т.н.). Засили се индивидуалният достъп на студенти и преподаватели до множество огромни информационни и изчислителни ресурси. Като приложна дисциплина, информатиката масово навлезе в ежедневието на хората, което силно повиши интереса към нея в бизнес средите. Трудно може да се посочи преуспяващ бизнес, в който пряко да не се използват компютърни технологии. Преустановиха се споровете доколко информатиката е академична област.

CC2001 е създадена с усилията на голям колектив от първокласни специалисти

* *Микромодул* е термин въведен от авторите, като превод на английския термин *knowledge unit*

след допитване и консултации с огромен брой университетски преподаватели от САЩ и Канада. Проектът е стартирал през 1998 год., преминал е през етапа на публикуване и обсъждане, а финалната му версия е с дата 15.12.2001. Спонсориран е от Фонд Научни изследвания на САЩ.

1.2. Кратки терминологични бележки. Информатиката е млада научна и приложна област с все още неустановена терминология. Самият термин информатика има различен превод и смисъл в различни държави. Точно това е причината, поради която заглавията на модули, дисциплини и области от информатиката в този материал не са преведени, а са записани така както са дадени в [1]. Понеже настоящият доклад е предназначен преди всичко за българската колегия, той е написан на български език. Това обаче поражда някои терминологични особености, поради което е добре да ги коментираме съвсем накратко.

Computing е може би най-общият термин, който се използва, когато става дума за информатика и е най-близък по смисъл до термина *информатика*, използван у нас. *Computer Science* (CS) или както все по-често се среща и у нас като *Компютърна наука* е част от *Computing*. Към *Computing* в САЩ се причисляват още *Software Engineering* (Софтуерни технологии, Софтуерно инженерство), *Computer Engineering* (Компютърно инженерство) и *Information Systems* (Информационни системи). *Computing Curriculum* е много повече от обикновен учебен план, поради което в този доклад е използван термина *Учебна документация*.

По изложените съображения по-долу ние ще се придържаме към американската терминология, като оставим без превод имената на области, дисциплини и модули от информатиката.

2. Същност и съдържание на CC2001. По същество CC2001 е учебна документация по информатика с препоръчителен, а не със задължителен характер. Следва обаче веднага да се добави, че обикновено (от предишни версии на CC) това е документът, по който се ръководят водещите университети в света. Може и сега да се очаква, че след това продължително подготвяне и обсъждане, CC2001 ще залегне като основен документ при изграждането на учебните планове на информатичните катедри по света.

2.1. Приципи, използвани при създаването на CC2001. В CC2001 са формулирани 11 принципа, на базата на които той е изграден. В свободен превод те са следните:

• *Computing* е термин, който включва в себе си термина CS.

• CS е теоретична и приложна област, включваща множество разнообразни дисциплини.

• Бързите промени в CS изискват постоянно обновяване на учебните планове.

• Разработката на учебните планове трябва да позволява промени, предизвикани от промени в технологиите и методиките на обучение. В същото време трябва да се гарантира обучение, което да позволява бърза пренастройка към предизвикателствата на бъдещето.

• Освен списък от микромодули, CC2001 следва да предоставя примери, подпомагащи създаването на конкретни учебни планове.

• В CC2001 трябва ясно да се идентифицират основните знания и практическите умения, които студентите по информатика трябва да придобият в първата степен

на висшето си образование.

• Поради постоянно нарастващият брой на нови микромодули, задължителното ядро от знания трябва да се сведе до минимум.

• CC2001 трябва да бъде широко отворена за университетите от цял свят като позволява пренастройка за съответните национални особености.

• При създаване на CC2001 трябва да се осигури участието както на представители на индустриални и държавни институции, така и на академичната аудитория, свързана с обучението по CS.

• В CC2001 трябва да се предвиди професионална практика, като част от обучението.

• CC2001 трябва да включва цялостната информация, която да подпомогне академичните организации при изграждането на конкретните учебни планове.

2.2. Основни области и модули в Computer Science. На фиг. 1 е даден списъкът с основните области по CS включени в CC1991. Десет години по-късно този списък в CC2001 е вече друг (фиг. 2). Не само броят на областите от 10 е нарастнал до 14, но и твърде малко от имената са същите.

- Algorithms and Data Structures (AL)
- Architecture (AR)
- Artificial Intelligence and Robotics (AI)
- Database and Information Retrieval (DB)
- Human-Computer Communications (HU)
- Numerical and Symbolic Computation (NU)
- Operating Systems (OS)
- Programming Languages (PL)
- Software Methodology and Engineering (SE)
- Social, Ethical, and Professional (SP)

Фиг. 1. *Основни области на CS* (CC 1991)

- Discrete Structures (DS)
- Programming Fundamentals (PF)
- Algorithms and Complexity (AL)
- Architecture and Organization (AR)
- Operating Systems (OS)
- Net-Centric Computing (NC)
- Programming Languages (PL)
- Human-Computer Interaction (HC)
- Graphics and Visual Computing (GV)
- Intelligent Systems (IS)
- Information Management (IM)
- Social and Professional Issues (SP)
- Software Engineering (SE)
- Computational Science (CN)

Фиг. 2. *Основни области на CS* (CC2001)

2.3. Задължителни и избираеми микромодули. На фиг. 3 е даден списък на всички области по CS и съответните им микромодули. Подчертаните микромодули са задължителни и за всеки един от тях е посочен минималният брой часове. Тук следва обаче да се направят и следните немаловажни бележки. В САЩ един учебен час е 50 мин., а поради възможността за презентации с компютър и прожектор в рамките на един учебен час се преподава много повече материя, отколкото у нас. В рамките на един семестър студентите в САЩ имат около 15-17 часа седмично. У нас седмичната аудиторна натовареност е около 25 часа.

2.4. Варианти за реализиране на CC2001. В CC2001 са разработени различни варианти за реализация на учебни планове. Всеизвестно е, че съществува голямо разнообразие на подходите (стиловете) в преподаването на уводните курсове (първо равнище). На преден план в CC2001 са изведени и подкрепени с идеи за реализация

DS. Discrete Structures (43 core hours)
DS1. Functions, relations, and sets (6)
DS2. Basic logic (10)
DS3. Proof techniques (12)
DS4. Basics of counting (5)
DS5. Graphs and trees (4)
DS6. Discrete probability (6)

PF. Programming Fundamentals (38 core hours)
PF1. Fundamental programming constructs (9)
PF2. Algorithms and problem-solving (6)
PF3. Fundamental data structures (14)
PF4. Recursion (5)
PF5. Event-driven programming (4)

AL. Algorithms and Complexity (31 core hours)
AL1. Basic algorithmic analysis (4)
AL2. Algorithmic strategies (6)
AL3. Fundamental computing algorithms (12)
AL4. Distributed algorithms (3)
AL5. Basic computability (6)
AL6. The complexity classes P and NP
AL7. Automata theory
AL8. Advanced algorithmic analysis
AL9. Cryptographic algorithms
AL10. Geometric algorithms
AL11. Parallel algorithms

AR. Architecture and Organization (36 core hours)
AR1. Digital logic and digital systems (6)
AR2. Machine level representation of data (3)
AR3. Assembly level machine organization (9)
AR4. Memory system organization and architecture (5)
AR5. Interfacing and communication (3)
AR6. Functional organization (7)
AR7. Multiprocessing and alternative architectures (3)
AR8. Performance enhancements
AR9. Architecture for networks and distributed systems

OS. Operating Systems (18 core hours)
OS1. Overview of operating systems (2)
OS2. Operating system principles (2)
OS3. Concurrency (6)
OS4. Scheduling and dispatch (3)
OS5. Memory management (5)
OS6. Device management
OS7. Security and protection
OS8. File systems
OS9. Real-time and embedded systems
OS10. Fault tolerance
OS11. System performance evaluation
OS12. Scripting

NC. Net-Centric Computing (15 core hours)
NC1. Introduction to net-centric computing (2)
NC2. Communication and networking (7)
NC3. Network security (3)
NC4. The web as an example of client-server computing (3)
NC5. Building web applications
NC6. Network management
NC7. Compression and decompression
NC8. Multimedia data technologies
NC9. Wireless and mobile computing

PL. Programming Languages (21 core hours)
PL1. Overview of programming languages (2)
PL2. Virtual machines (1)
PL3. Introduction to language translation (2)
PL4. Declarations and types (3)
PL5. Abstraction mechanisms (3)
PL6. Object-oriented programming (10)
PL7. Functional programming
PL8. Language translation systems
PL9. Type systems
PL10. Programming language semantics
PL11. Programming language design

Фиг. 3. *Модулна структура на учебния план по CS*

HC. Human-Computer Interaction (8 core hours)
HC1. Foundations of human-computer interaction (6)
HC2. Building a simple graphical user interface (2)
HC3. Human-centered software evaluation
HC4. Human-centered software development
HC5. Graphical user-interface design
HC6. Graphical user-interface programming
HC7. HCI aspects of multimedia systems
HC8. HCI aspects of collaboration and communication

GV. Graphics and Visual Computing (3 core hours)
GV1. Fundamental techniques in graphics (2)
GV2. Graphic systems (1)
GV3. Graphic communication
GV4. Geometric modeling
GV5. Basic rendering
GV6. Advanced rendering
GV7. Advanced techniques
GV8. Computer animation
GV9. Visualization
GV10. Virtual reality
GV11. Computer vision

IS. Intelligent Systems (10 core hours)
IS1. Fundamental issues in intelligent systems (1)
IS2. Search and constraint satisfaction (5)
IS3. Knowledge representation and reasoning (4)
IS4. Advanced search
IS5. Advanced knowledge representation and reasoning
IS6. Agents
IS7. Natural language processing
IS8. Machine learning and neural networks
IS9. AI planning systems
IS10. Robotics

IM. Information Management (10 core hours)
IM1. Information models and systems (3)
IM2. Database systems (3)
IM3. Data modeling (4)
IM4. Relational databases
IM5. Database query languages
IM6. Relational database design
IM7. Transaction processing
IM8. Distributed databases
IM9. Physical database design
IM10. Data mining
IM11. Information storage and retrieval
IM12. Hypertext and hypermedia
IM13. Multimedia information and systems
IM14. Digital libraries

SP. Social and Professional Issues (16 core hours)
SP1. History of computing (1)
SP2. Social context of computing (3)
SP3. Methods and tools of analysis (2)
SP4. Professional and ethical responsibilities (3)
SP5. Risks and liabilities of computer-based systems (2)
SP6. Intellectual property (3)
SP7. Privacy and civil liberties (2)
SP8. Computer crime
SP9. Economic issues in computing
SP10. Philosophical frameworks

SE. Software Engineering (31 core hours)
SE1. Software design (8)
SE2. Using APIs (5)
SE3. Software tools and environments (3)
SE4. Software processes (2)
SE5. Software requirements and specifications (4)
SE6. Software validation (3)
SE7. Software evolution (3)
SE8. Software project management (3)
SE9. Component-based computing
SE10. Formal methods
SE11. Software reliability
SE12. Specialized systems development

CN. Computational Science and Numerical Methods (no core hours)
CN1. Numerical analysis
CN2. Operations research
CN3. Modeling and simulation
CN4. High-performance computing

	CS111r. Intro to Programming	CS112r. Data Abstraction	CS115. Discrete Structures	CS210r. Algorithm Analysis	CS220r. Computer Architecture	CS225r. Operating Systems	CS230r. Net-centric Computing	CS260r. Artificial Intelligence	CS270r. Databases	CS280r. Social and Prof Issues	CS290r. Software Development	CS490. Capstone Project	Total	Extra hours
DS1. Functions, relations, and sets			6										6	
DS2. Basic logic			10										10	
DS3. Proof techniques			9	3									12	
DS4. Basics of counting			5										5	
DS5. Graphs and trees		2		4									6	+2
DS6. Discrete probability			6										6	
PF1. Fundamental programming constructs	9												9	
PF2. Algorithms and problem-solving	3			3									6	
PF3. Fundamental data structures	6	6		3									15	+1
PF4. Recursion		5											5	
PF5. Event-driven programming							2				4		6	+2
AL1. Basic algorithmic analysis		2		2									4	
AL2. Algorithmic strategies				6									6	
AL3. Fundamental computing algorithms	2	4		6									12	
AL4. Distributed algorithms						3							3	
AL5. Basic computability	1			6									7	+1
AR1. Digital logic and digital systems			3		3								6	
AR2. Machine level representation of data	1				3								4	+1
AR3. Assembly level machine organization	2				9								11	+2
AR4. Memory system organization and architecture					5								5	
AR5. Interfacing and communication					3								3	
AR6. Functional organization					7								7	
AR7. Multiprocessing and alternative architectures					3								3	
OS1. Overview of operating systems						2							2	
OS2. Operating system principles						2							2	
OS3. Concurrency						6							6	
OS4. Scheduling and dispatch						3							3	
OS5. Memory management						5							5	
NC1. Introduction to net-centric computing							2						2	
NC2. Communication and networking							7						7	
NC3. Network security							3						3	
NC4. The web as an example of client-server computing							3						3	
PL1. Overview of programming languages	1	1											2	
PL2. Virtual machines		1											1	
PL3. Introduction to language translation		2		2									4	+2
PL4. Declarations and types	1	2											3	
PL5. Abstraction mechanisms	2	1											3	
PL6. Object-oriented programming	3	7					2						12	+2
HC1. Foundations of human-computer interaction									2		6	2	10	+4
HC2. Building a simple graphical user interface											2		2	
GV1. Fundamental techniques in graphics	2										2		4	+2
GV2. Graphic systems											1		1	
IS1. Fundamental issues in intelligent systems								1					1	
IS2. Search and constraint satisfaction								5					5	

IS3. Knowledge representation and reasoning								4					4	
IM1. Information models and systems									3				3	
IM2. Database systems									3				3	
IM3. Data modeling									4				4	
SP1. History of computing	1									1			2	+1
SP2. Social context of computing										3			3	
SP3. Methods and tools of analysis										2			2	
SP4. Professional and ethical responsibilities										3			3	
SP5. Risks and liabilities of computer-based systems										2			2	
SP6. Intellectual property									3	3			6	+3
SP7. Privacy and civil liberties									2	2			4	+2
SE1. Software design	2	2									2	4	10	+2
SE2. Using APIs		2									3	3	8	+3
SE3. Software tools and environments	1	2									2	3	8	+5
SE4. Software processes												2	2	
SE5. Software requirements and specifications	1										2	2	5	+1
SE6. Software validation	1										1	3	5	+2
SE7. Software evolution											2	2	4	+1
SE8. Software project management											2	3	5	+2
Общ брой часове на отделен курс	39	39	39	35	33	21	19	10	17	16	29	24		

Фиг. 4. *Императивен подход. Таблица на разпределението на микромодулите по дисциплини*

шест такива подхода: *императивен*, *обектно-ориентиран*, *функционален*, *алгоритмичен*, *хардуерен* и *общообразователен*. Всеки един подход има своите особености и поражда различни варианти на учебни планове.

2.5. Примерен учебен план. Посочените шест подхода са добре известни и някои от тях се използват от дълго време в нашите университети. Най-популярен е императивният подход. По тази причина на фиг.4 е даден един конкретен учебен план от [1], построен на основата на императивния подход.

Микромодулите са дадени в хоризонталните редове, а дисциплините са представени по колони. Това дава съответствието между микромодули и дисциплини изразено в броя на часовете, с които всеки микромодул се включва в сответните дисциплини. Целта на уводните курсове е да изградят солидна база от знания за второто и третото равнище.

CS210T. Algorithm Design and Analysis
CS220T. Computer Architecture
CS225T. Operating Systems
CS230T. Net-centric Computing
CS260T. Artificial Intelligence
CS270T. Databases
CS280T. Social and Professional Issues
CS290T. Software Development
CS490. Capstone Project

Фиг. 5. *Дисциплини от второ равнище*

За курсовете от второто равнище се предлагат четири примерни модела. На фиг. 5 е дадена примерна реализация на предметно-базирания модел, който масово2 се прилага и у нас.

Третото равнище включва курсовете, чието съдържание надхвърля рамките на задължителните микромодули. Един примерен списък на курсове от трето равнище

е даден на фиг. 6.

Discrete Structures (DS)
- CS301. Combinatorics
- CS302. Probability and Statistics
- CS303. Coding and Information Theory

Computational Science (CN)
- CS304. Computational Science
- CS305. Numerical Analysis
- CS306. Operations Research
- CS307. Simulation and Modeling
- CS308. Scientific Computing
- CS309. Computational Biology

Algorithms and Complexity (AL)
- CS310. Advanced Algorithmic Analysis
- CS311. Automata and Language Theory
- CS312. Cryptography
- CS313. Geometric Algorithms
- CS314. Parallel Algorithms

Architecture and Organization (AR)
- CS320. Advanced Computer Architecture
- CS321. Parallel Architectures
- CS322. System on a Chip
- CS323. VLSI Development
- CS324. Device Development

Operating Systems (OS)
- CS325. Advanced Operating Systems
- CS326. Concurrent and Distributed Systems
- CS327. Dependable Computing
- CS328. Fault Tolerance
- CS329. Real-Time Systems

Net-Centric Computing (NC)
- CS330. Advanced Computer Networks
- CS331. Distributed Systems
- CS332. Wireless and Mobile Computing
- CS333. Cluster Computing
- CS334. Data Compression
- CS335. Network Management
- CS336. Network Security
- CS337. Enterprise Networking
- CS338. Programming for the World-Wide Web

Programming Languages (PL)
- CS340. Compiler Construction
- CS341. Programming Language Design
- CS342. Programming Language Semantics
- CS343. Programming Paradigms
- CS344. Functional Programming
- CS345. Logic Programming
- CS346. Scripting Languages

Human-Computer Interaction (HC)
- CS350. Human-Centered Design and Evaluation
- CS351. Graphical User Interfaces
- CS352. Multimedia Systems Development
- CS353. Interactive Systems Development
- CS354. Computer-Supported Cooperative Work

Graphics and Visual Computing (GV)
- CS355. Advanced Computer Graphics
- CS356. Computer Animation
- CS357. Visualization
- CS358. Virtual Reality
- CS359. Genetic Algorithms

Intelligent Systems (IS)
- CS360. Intelligent Systems
- CS361. Automated Reasoning
- CS362. Knowledge-Based Systems
- CS363. Machine Learning
- CS364. Planning Systems
- CS365. Natural Language Processing
- CS366. Agents
- CS367. Robotics
- CS368. Symbolic Computation
- CS369. Genetic Algorithms

Information Management (IM)
- CS370. Advanced Database Systems
- CS371. Database Design
- CS372. Transaction Processing
- CS373. Distributed and Object Databases
- CS374. Data Mining
- CS375. Data Warehousing
- CS376. Multimedia Information Systems
- CS377. Digital Libraries

Social and Professional Issues (SP)
- CS380. Professional Practice
- CS381. Social Context of Computing
- CS382. Computers and Ethics
- CS383. Computing Economics
- CS384. Computer Law
- CS385. Intellectual Property
- CS386. Privacy and Civil Liberties

Software Engineering (SE)
- CS390. Advanced Software Development
- CS391. Software Engineering
- CS392. Software Design
- CS393. Software Engineering and Formal Specification
- CS394. Empirical Software Engineering
- CS395. Software Process Improvement
- CS396. Component-Based Computing
- CS397. Programming Environments
- CS398. Safety-Critical Systems

Фиг. 6. *Примерен списък на курсове от трето равнище*

3. Представяне на държавните изисквания за специалност информатика у нас. Документално у нас специалността „Информатика“ с професионална квалификация „информатик“ на държавно равнище е представена с четири документа – наредби на МС. Първата се появи във връзка с изграждането на учебните планове на две равнища – бакалавар и магистър. С втората (2000 г.) се правят някои допълнения и изменнения, а третата наредба отменя първите две. От август 2002 действаща е четвърта наредба. Интерес представлява [2], по която бяха създа-

дени първите учебни планове за бакалаври и която беше в действие в продължение на около 5 години. Накратко параметрите ѝ са дадени чрез таблицата от фиг. 7. Тя включва списък за задължителните дисциплини, които са в обем от 1500 часа. Допълнително се уточнява, че сборът от часовете по задължителните и избираемите дисциплини е не по-малък от 2200 часа, обучението е 4 учебни години с общ минимален хорариум 3000 академични часа.

	Учебна дисциплина	*Общ хорариум (часове)*
1	Алгебра и геометрия	150
2	Анализ и диференциални уравнения	200
3	Приложна математика (числени методи, теория на вероятностите	250
4	Теоретични основи на информатиката (дискретна математика, семантика на езиците за програмиране)	150
5	Програмиране (програмиране на структури от данни, обектно-ориентирано програмиране)	250
6	Изкуствен интелект (логическо и функционално програмиране, изкуствен интелект)	150
7	Компютърни архитектури и мрежи (операционни системи, компютърни архитектури, компютърни мрежи и комуникации)	150
8	Информационни технологии (компютърна графика, бази от данни, софтуерни технологии)	200
	Общо	1500

Фиг. 7. *Извадка от изискванията за професионална квалификация "информатик"*

Последната наредба [4] е обща за всички специалности и е естествено в нея не са представени детайли по отношение на съдържанието, както това е направено в [2]. Ето петте държавни изисквания към образованието в образователно-квалификационната степен „Бакалавър“:

1. Получаване на цялостна представа за характера на професионалното направление и специалността.

2. Овладяване на широкопрофилни теоретични знания и практически умения.

3. Умения за адаптивност в съответствие с изменящите се условия при реализирането на специалистите.

4. Придобиване на умения за самостоятелна професионална работа и за работа в екип.

5. Условия за образователна мобилност и международна сравнимост на получаваните знания и придобитите способности.

Изискванията дадени в [4] не са в противоречие с тези от [2], поради което навярно наредба [2] ще оказва влияние при изготвянето на учебни планове и за напред. Голямата свобода, която се дава на академичните институти с тази наредба е прекрасна възможност за изграждане на съвременни учебни планове. За това може да помогне и CC2001.

4. Заключителни бележки. Информатиката е динамично развиваща се като наука и като университетска специалност. Реформите, които предстоят сега и за напред, са продиктувани преди всичко от мощния поток на научните резултати, бързото им трансформиране в технологии, които постоянно променят ежедневието на хората. Всичко това естествено създава множество проблеми, които ние ще групираме само две групи:

• *учебно съдържание* (учебен план, дисциплини, учебно съдържание и т.н.);

• *преподаване* (*подходи* (императивен, обектно-ориентиран, ...), *форми* на обучение (дистанционно, ...), *среда* (хардуерно и софтуерно оборудване, кадри, ...) ...).

Всяка една от тези групи може да се декомпозира в огромно множество от конкретни въпроси. Надяваме се, поне една малка част от тях да бъдат дискутирани и да намерят решение на дискусията на пролетната конференция на СМБ.

ЛИТЕРАТУРА

[1] CC2001 Computer Science volume Final Report (December 15, 2001) http://www.acm.com.
[2] НАРЕДБА за единните държавни изисквания за придобиване на висше образование по специалностите от професионално направление „Математика“ ДВ, бр. 79 от 12.09.1997 г.
[3] Наредба за държавните изисквания за придобиване на висше образование на образователно-квалификационните степени „бакалавър“, „магистър“ и „специалист“ ДВ, бр. 76 от 6.08.2002 г.,
[4] НАРЕДБА за държавните изисквания за придобиване на висше образование на образователно-квалификационните степени „бакалавър“, „магистър“ и „специалист“ ДВ, бр. 76 от 6.08.2002 г.

Павел Азълов
Пенсилвански Държавен Университет
САЩ
e-mail: pka10@psu.edu

Аврам Ескенази
секция „Софтуерни технологии“
Институт по математика и информатика
ул. акад. Г. Бончев, бл. 8
1113 София
e-mail: eskenazi@math.bas.bg

COMPUTING CURRICULA 2001 ... WHERE WE ARE?

P. Azalov, A. Eskenazi

The main results of the new curriculum for undergraduate programs in computer science, known as CC2001, are presented. CC2001 has been developed by IEEE and ACM and published by the end of 2001. Broadly discussed, today CC2001 is considered as a working tool when developing curricula in many universities all over the world. In this paper we also present the main state parameters according to which the computer science curricula in Bulgarian universities have been developed. Several general problems concerning the curricula and their implementation are formulated. These are expected to be discussed during the conference.