COMPUTING IN SCHOOLS – PRESENT

Zuzana Kubincová
Comenius University in Bratislava, Slovakia

Educational Content Reform

Informatics:

- Natural science subject, area: Mathematics and working with information
- Compulsory subject from primary school to grammar school
- Content closer to computation
- National educational program (national curriculum)
  - obligatory → school curricula

Adjusted National Curriculum in Informatics

- Lower grades
  - Acquiring digital literacy
- Higher grades
  - problem solving
  - algorithmic thinking
  - understanding the principles
  - exploring general relationships and information

Thematic Areas

- Representations and tools
- Communication and cooperation
- Problem solving
- Software and hardware
- Information society

- In all grades
- Different levels of difficulty

Time Allocation

Primary school									Grammar school			
1	2	3	4	5	6	7	8	9	I	II	III	IV
-	-	1	1	1	1	1	1	-	3			

Time allocated weekly (in hours) for informatics classes in particular grades.

Structure of national curriculum

- Performance-related standards
- Content standards

- Primary schools – standards specified by two grades (end of the 4th, 6th and 8th year of study)
- Grammar schools – standards specified by particular fields

Limitations and Requirements

- Capabilities and interests of pupils of particular age
  - Working with storybooks, tables/spreadsheets
- Coordination with other school subjects
  - Working with text, e-mail, …

Representations and tools

Representations and tools	Prim 2-4	Prim 5-6	Prim 7-8	Grammar	Graduation
Working with graphics	basic	basic	advanced	advanced	advanced
Working with text	basic	basic	advanced	advanced	advanced
Working with stories	start				
Working with presentations		basic	advanced	advanced	advanced
Working with spreadsheets		start	basic	basic	advanced
Working with multimedia	start		basic	basic	advanced
Evaluation of comput. tools					basic
Information	start	basic	basic	basic	advanced
Structures	start	basic	basic	basic	advanced
Data types and operations					basic

start: they only start to work and collect the skills to get familiar with informatics concepts

basic: they have the basic skills, the emphasis is on the new informatics concepts

advanced: they practice the acquired skills and knowledge, there are only a few new goals

Communication and collaboration

Communication and collaboration	Prim 2-4	Prim 5-6	Prim 7-8	Grammar	Graduation
Working with websites	basic	basic	advanced		
Presentation of information on a website				basic	advanced
Web Search	start	basic	advanced	advanced	advanced
Working with tools for collaboration and information sharing				basic	advanced
Working with communication tools	basic	basic	basic	advanced	advanced

Problem solving

Problem solving	Prim 2-4	Prim 5-6	Prim 7-8	Grammar	Graduation
Problem analysis	start	start	basic	basic	advanced
Interactive creating of solution	start				
Language for solution notation		start	basic	basic	basic
By using sequences of commands	start	basic	advanced	advanced	advanced
By using loops		start	basic	basic	basic
By using branches			basic	basic	basic
By using interactive tools			start	basic	basic
By using variables			start	basic	basic
By using one and two-dimensional arrays					basic
By using files					basic
By using subroutines with parameters					basic
Interpretation of solution notation	start	basic	basic	basic	advanced
Finding and fixing bugs	start	basic	basic	basic	advanced
Solution analysis and evaluation					basic

Software and hardware

Software and hardware	Prim 2-4	Prim 5-6	Prim 7-8	Grammar	Graduation
Working with files and folders	start	basic	basic	advanced	advanced
Working in the operating system	start	start	basic	basic	advanced
Computers and computer peripherals	start	start	basic	basic	basic
Working in computer networks and on the Internet	start	basic	advanced	advanced	advanced
Working against viruses and spying		start	basic	basic	advanced

Information society

Information society	Prim 2-4	Prim 5-6	Prim 7-8	Grammar	Graduation
Security and risks	start	basic	basic	basic	advanced
Digital technology in society	basic	basic	advanced	advanced	advanced
Legality of use	start	start	basic	advanced	advanced

Example – Algorithmic Problem Solving

Performance standard	Content standard
At the end of 4th grade pupils - solve a problem by direct control of an executor (e.g. a robot, a turtle, etc.) - apply elementary commands of a given language (from a commands dictionary) to control the executor	*Attributes and relationships:* direct command - executor's action *Processes:* control of an executor in a direct mode, use of the executor's language

Performance standard	Content standard
At the end of 6th grade pupils - use the language to describe the solution of the problem - apply rules, language constructions	*Attributes and relationships:* bad notation, language constructions as a sequence of commands *Processes:* putting a program together in the language for algorithms writing, running the program

Example – Algorithmic Problem Solving

Performance standard	Content standard
At the end of 8th grade pupils - use the language to describe the solution of the problem - apply rules, language constructions - use mathematical expressions in the language for writing algorithms	*Attributes and relationships*: algorithm - programming language, input - algorithm - result, incorrect notation, language constructs such as: sequence of commands, loop with a fixed number of iterations, conditional command, named command sequence *Processes*: putting a program together in the language for writing algorithms, running the program

Performance standard	Content standard
At a grammar school pupils - use the language to write algorithmic solution of a problem (they use the language constructs and apply the language rules) - use mathematical expressions to represent relationships and conditions - recognize and eliminate errors in the notation - create notations and interpret notations under the new given rules (of syntax) for writing algorithms	*Concepts*: program, programming language *Attributes and relationships*: writing algorithm and execution of the program, input - program execution - output / action *Processes*: putting a program together, identifying, searching, correcting errors

Example – Algorithmic Problem Solving

Performance standard	Content standard
When **graduating**, pupils are able to - recognize the input information from the assignment, describe the expected outputs, results, resulting actions - formulate a formal notation of an algorithm according to various executors criteria	*Concepts*: algorithm, programming language, development environment, declaration (declaration part) and commands (command section), problem, subproblem, generalization, critical situations, restrictions
- formulate the nature of a computational problem - arrange commands in the command constructions according to the rules - identify and eliminate errors in the formal notation of an algorithm - create a solution using prepared subroutines - adjust the solution according to the different requirements and constraints (to change the program inputs, outputs, process, limited set of commands, variables, types) - design a program, which seeks, modifies, creates, writes or draws.	*Attributes and relations*: program (launch, execution, stop), notation rules (syntax - reserved words, identifiers, constants, separators, program constructs), repeating patterns *Process*: a sequence of commands, repeating (types of loops, conditions, loop-counter, body of the loop, nested commands), branching (types of conditional constructs, evaluation mechanism, nested commands) subroutines, parameters (call mechanism, local variables, parameters, return value)

Main focus

- Problem solving approach
  - different solutions
  - most appropriate representation
  - most efficient algorithms and tools
- Higher levels of the Revised Bloom's taxonomy (application, analysis, evaluation, creating)
- Emphasis on algorithmic thinking and programming
  - programming at secondary school: 33-50%

Problems

- Lack of qualified informatics teachers
- Misunderstanding of the real goal of school informatics
- Teachers avoiding programming
- Shortage of textbooks, software and hardware, supporting materials

Thank you!

Zuzana Kubincová
kubincova@fmph.uniba.sk