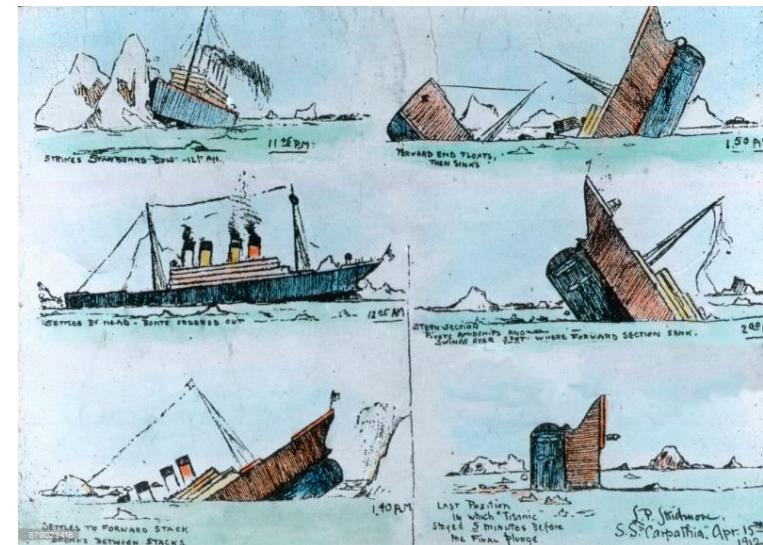


RMS Titanic má zákaz plavby

Snažím sa rozvíjať školské programovanie pre každého

Ivan Kalaš

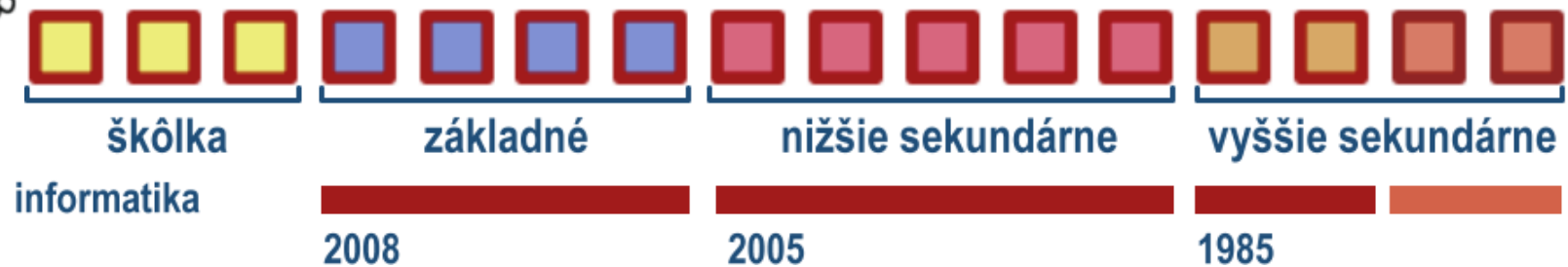
FMFI Univerzita Komenského, Bratislava
UCL Knowledge Lab, London
visiting professor



korene 'novej' informatiky (1/2)

na Slovensku, DVUi... a vo svete

● školská informatika

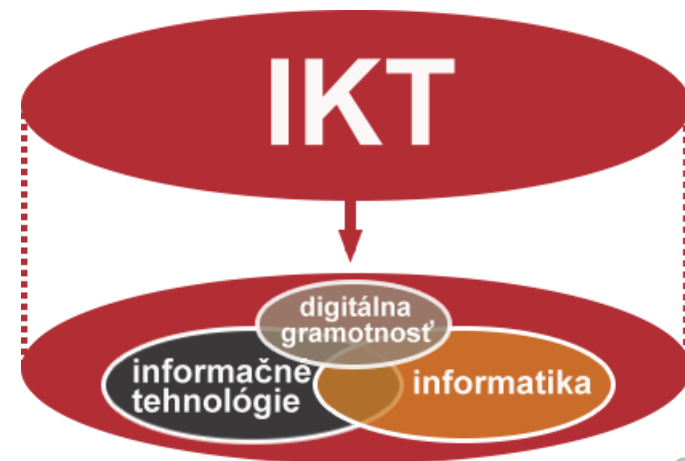


- didaktici informatiky v SR osnovy, učebnice, vzdelávanie učiteľov
vývoj programovacích prostredí pre vzdelávanie

- takže všetko je ideálne (?)
pokúsím sa identifikovať reálnu situáciu

- správa Britskej kráľovskej spoločnosti
2012: *Buď ju zrušme, alebo ju robme poriadne*

- obrovská vlna záujmu o lepšiu informatiku
%



korene 'novej' informatiky (2/2)

na Slovensku, DVUi... a vo svete

čo teda znamená: lepšia informatika? ... čím trpí a aká má byť?

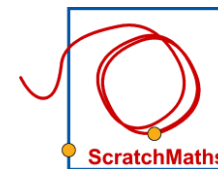
- **patrí do moderného povinného vzdelávania?**

aký je postoj v rôznych krajinách

naša predstava je dosť *unikátna a osamelá*

možno aj preto... spolupráca pri tvorbe

vzdelávacieho obsahu pre programovanie na ZŠ v Anglicku



- **máme odvahu oddeliť IKT a informatiku v škole?**

- **vieme už IKT efektívne používať vo všetkých predmetoch?**

(to bol cieľ projektu Infovek od 1999

ešte stále výrazne nedostatočne naplnený,

predmety prehliadajú cenné „hrozienuku“)

súčasn^é problémy a nedostatky (1/4)

v školskej informatike: vo svete a u nás

- berieme ju vážne?
- má v nej programovanie **primerané** postavenie?
 - často zostáva bokom
 - alebo má „technokratické“ ciele
 - ? primeranosť vzdelávacích cieľov
 - ? posadnutosť objemom, pojmami
 - ? málo fantázie, motivácie, konštruktivizmu...
 - ale pozor... *Urob si vlastnú hru* nemusí byť správna alternatíva
 - ? slabá pedagogika
 - nové prostredia typu Hour of Code
 - IT sektor často nerozumie kognitívnym procesom detí**
 - ... a didaktike informatiky

súčasn^é problémy a nedostatky (2/4)

v školskej informatike: vo svete a u nás

● poskytujeme učiteľom dostatočnú podporu?

>> na 1. stupni --- nová informatika učiteľov skôr odstrašuje

>> kto ju má učiť?

bežný učiteľ

musíme mu pomôcť, aby ju zvládol

aby spoznal jej potenciál a uveril mu

programovanie ako vynikajúca propedeutika matematiky

>> na 2. a 3. stupni

metodické materiály

pracovné listy, učebné materiály pre žiakov

primerané programovacie prostredia

vzdelávanie

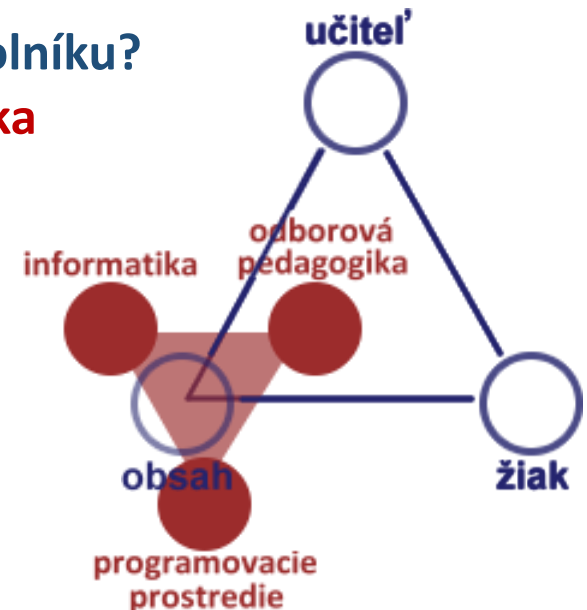
súčasn^é problémy a nedostatky (3/4)

v školskej informatike: vo svete a u nás

- nebezpečné načasovanie programovacích prostredí
priskoro, prineskoro, pridlho...
- poznáme a rešpektujeme **vývinovú primeranosť**?
primerané výpočtové konštrukty
zodpovedajúce kognitívne operácie
a **NAJMÄ** ich kognitívnu náročnosť?
- rešpektujeme rovnováhy v didaktickom trojuholníku?
informatika ● **nástroj** ● **odborová didaktika**

nebezpečné vychýlenia smerom k...

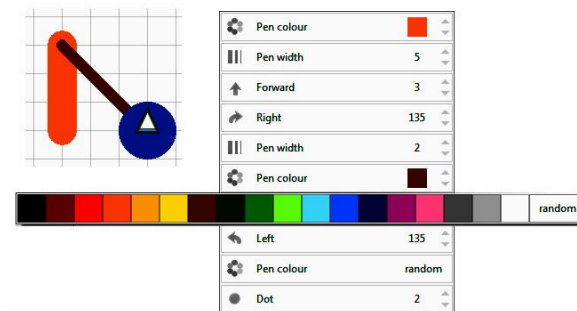
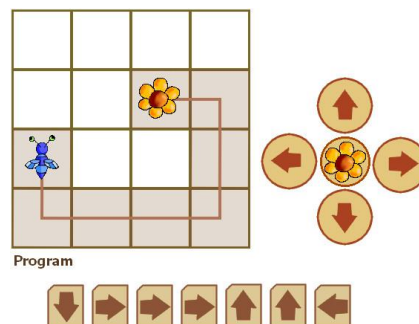
k nástroju < *technocentrizmus*
ku CS < *CS centrizmus*
k didaktike < *neProgramovanie*



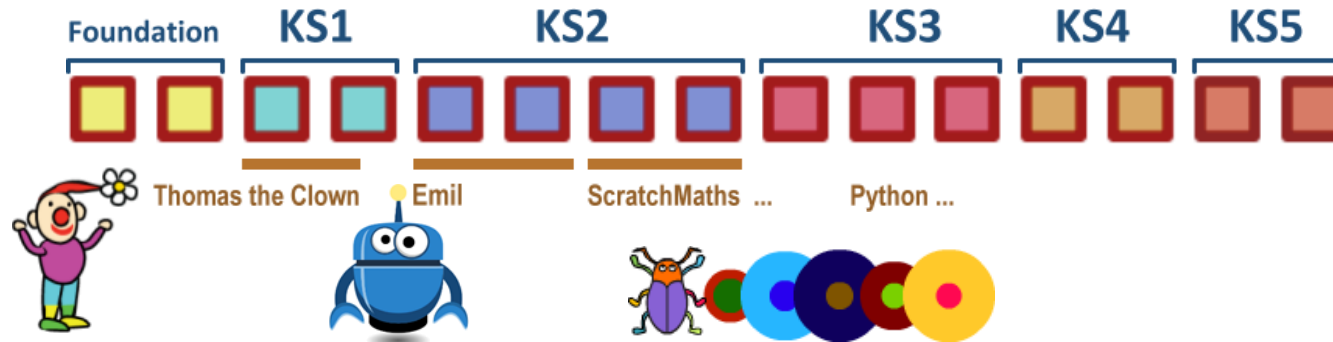
súčasn^é problémy a nedostatky (4/4)

v školskej informatike: vo svete a u nás

- robíme informatiku pre každého žiaka?
skúmanie, objavy, zážitky... postupný vznik poznatkov
pre všetky dievčatá a všetkých chlapcov
- robíme ju tak, aby bola – úprimne – užitočná vo **všeobecnom vzdelaní**?



How we support the process in our latest development (1/4)



Emil (Émile) the Robot: pedagogical framework

- elementary programming for every pupil in the class
as an opportunity and formative learning experience
- implemented/scaffolded by generalist primary teachers
building bridges
- learning together – pair work + whole class discussions
- learning without external feedback
encouraging constructionist/reflective learning
- action/plan for action/programme – an object to think about and think with
doing, planning, reading and interpreting, exploring, comparing...

How we support the process in our latest development (2/4)

goals, procedure, design based research



- developing complex interventions for Y3 and Y4
- 12+ lessons for Y3 and 12+ lessons for Y4 of programming
- designing and developing new programming environments
 - Y3: Emil the Collector; Emil the Caretaker; Emil the Collagist
 - Y4: further extensions of these + 2 new environments
- regularly and closely working with 3 design schools, 5 classes
 - “ordinary” primary teachers
 - our goal is that they accept new interventions **as their own**
- developing
 - software, teacher materials, worksheets, physical stuff, PD strategy

How we support the process in our latest development (3/4)

Emil the Collector: powerful ideas



● Experiencing order

order in objects: collections with order and without order – a **box**, a **shelf**
objects with internal order – numbers, letters

order in steps:
recognizing **patterns** in data collections

● Coping with constraints

static and dynamic

in driving Emil: where to click, **number of clicks...**

in the stage: objects as static or **temporary obstacles**

in the collection: **predefined positions, global state** (sum, word...)

● Exploring control

in the *local direct drive* style

building a record of the steps:

by Emil, by pupils on the paper

planning the steps

locally in the stage

reading and interpreting plans (programs)

How we support the process in our latest development (4/4)

Emil the Caretaker: powerful ideas



- **Mastering control**

in *proper direct drive* arrows for **absolute directions** (vs. relative)

in computational drive: with external plan

affect the stage by instruments/operators

single action

iterated action

some instruments are polymorphic

thinking in 'piles of steps'

consider local state of the stage

- **Planning solutions**

observe the **record** as it is being built

with constraints

number of steps

missing arrow

build a plan – program it, run it

consider global state of the stage

- **Working with plans**

reading, rerunning

comparing, modifying, analysing, filling in, recognising **patterns**

Summing up: some recommendations



- respect and know (primary) pupils, care about all of them
- respect (primary) teachers... they do not have any pre-service background in CS
 - develop with them ... *foster a sense of teacher ownership of an innovation*
 - do not trial it for them
 - more general** – build multiple ways how to get in-service qualification
- be aware of and exploit **affordances** of the tool (and avoid ‘ill affordances’)
- master *pedagogy of computing*
 - respect differences, encourage diversions, follow holistic curriculum**
 - use good metaphors
 - in ScratchMaths: *5E – Explore, Explain, Envisage, Exchange, bridgE*
- continuously support teachers
- offer and accept **bridges** to all subjects
- offer inspiring, challenging and rewarding contexts – constructionism
- make it developmentally appropriate
 - which concepts, practices, tools... and **when**
- make it respected – by teachers, pupils, parents... and lower/higher stages
 - as a sustainable, productive and respectful **instrument for learning**

Ďakujem za pozornosť



Kontakt

kalas@fmph.uniba.sk

References

- Brennan, Resnick (2012) New framework for studying and assessing the development of computational thinking
- Fuller et al. (2007) Developing a Computer Science-specific Learning Taxonomy
- Kabatova, Kalas, Tomcsanyiova (2016) Programming in Slovak Primary Schools
- Kalas (2016) On the Road to Sustainable Primary Programming
- Lister (2016) Toward a Developmental Epistemology of Computer Programming