

Zlepšenie matematických schopností študenta pomocou logických počítačových hier*

Filip Chromek

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
`xchromek@stuba.sk`

11. október 2022

Abstrakt

Iba málo študentov v dnešnej dobe má rado matematiku. Na druhej strane má veľa mladých ľudí rado hranie hier. Či už hier na počítači alebo na mobilnom telefóne.

Počítačové hry zväčša využívajú prirodzenú zvedavosť a súťaživosť ľudskej mysle, aby presvedčili človeka, aby sa im venoval. Hry taktiež využívajú zvýšené vylučovanie endorfinov a dopamínov počas hrania, čo vedie k ešte väčšiemu záujmu hrať.

Tento fakt je vhodné využiť pre niečo dobré. Pri hraní hier, ktorým obsahom sú informácie z reálneho sveta, si daný hráč dokáže podvedome tieto informácie zapamätať. V hrách musí tieto informácie využívať pre dosiahnutie progresu, musí sa nad nimi zamýšľať a vďaka tomu sa mu „uložia“ v pamäti, aj keď sa ich pôvodne učiť nechcel. Takýto štýl učenia je veľmi účinný najmä pri vedomostiach z oblasti matematiky, keďže sa ich väčšinou nedá „nauciť naspamäť“, ale je potrebné sa nad nimi zamyslieť.

1 Úvod

Motivujte čitateľa a vysvetlite, o čom píšete. Úvod sa väčšinou nedelí na časti.

Uveďte explicitne štruktúru článku. Tu je nejaký príklad. Základný problém, ktorý bol naznačený v úvode, je podrobnejšie vysvetlený v časti 2. Dôležité súvislosti sú uvedené v častiach 4 a 5. Záverečné poznámky prináša časť 6.

2 Nejaká časť

Z obr. 1 je všetko jasné.

*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2022/23, vedenie: Ing. Vladimír Mlynarovič, PhD.

Aj text môže byť prezentovaný ako obrázok. Stane sa z neho označný plávajúci objekt. Po vytvorení diagramu zrušte znak % pred príkazom `\includegraphics` označte tento riadok ako komentár (tiež pomocou znaku %).

Obr. 1: Rozhodujúci argument.

3 Iná časť

Základným problémom je teda... Najprv sa pozrieme na nejaké vysvetlenie (časť 3.1), a potom na ešte nejaké (časť 3.1).¹

Môže sa zdať, že problém vlastne nejestvuje [1], ale bolo dokázané, že to tak nie je [2,3]. Napriek tomu, aj dnes na webe narazíme na všelijaké pochybné názory [4]. Dôležité veci možno *zdôrazniť kurzívou*.

3.1 Njaké vysvetlenie

Niekedy treba uviesť zoznam:

- jedna vec
- druhá vec
- x
- y

Ten istý zoznam, len číslovaný:

1. jedna vec
2. druhá vec
- (a) x
- (b) y

3.2 Ešte nejaké vysvetlenie

Veľmi dôležitá poznámka. Niekedy je potrebné nadpisom označiť odsek. Text pokračuje hneď za nadpisom.

4 Dôležitá časť

5 Ešte dôležitejšia časť

6 Záver

Literatúra

[1] James O. Coplien. *Multi-Paradigm Design for C++*. Addison-Wesley, 1999.

¹Niekedy môžete potrebovať aj poznámku pod čiarou.

Nasledujúca tabuľka zobrazuje HTML náhľad na vybraný rozvrh. Tlačovú verziu získate voľbou výstupu vo formáte PDF.

Deň	8.00-8.50	9.00-9.50	10.00-10.50	11.00-11.50	12.00-12.50	13.00-13.50	14.00-14.50	15.00-15.50	16.00-16.50	17.00-17.50	18.00-18.50	19.00-19.50
Po	-1.61 (Aula Magna) (BA-FIT-FIT) Algebra a diskretná matematika J. Šlagová		-1.61 (Aula Magna) (BA-FIT-FIT) Matematická analýza Z. Minnerchová					-1.61 (Aula Magna) (BA-FIT-FIT) Principy počítačového inžinierstva (1) J. Pluhák				
Ut		-1.37/b (U806) (BA-FIT-FIT) Matematická analýza (2-3) M. Omachelová				-2.01/a(CPUa) (BA-FIT-FIT) Procedurálne programovanie (2,3) M. Čelich		-2.01/a(CPUa) (BA-FIT-FIT) Metódy inžinierstva (2,3) V. Mlynárovic	-2.01/c(CPUc) (BA-FIT-FIT) Principy počítačového inžinierstva (4-5) J. Lando			
St		-1.61 (Aula Magna) (BA-FIT-FIT) Procedurálne programovanie J. Zelenka										
St		-1.61 (Aula Magna) (BA-FIT-FIT) Matematická analýza (4) Z. Minnerchová							-1.61 (Aula Magna) (BA-FIT-FIT) Metódy inžinierstva práce (4) J. Lang			
Pi		1.38 (U206) (BA-FIT-FIT) Algebra a diskretná matematika (5) E. Stanečková			1.40 (U40) (BA-FIT-FIT) / 5 Anglický jazyk I Z. Suliková							

Legenda:

prednáška cvičenie

Ak nie je v poznámke uvedené inak, prebieha výučba v areáli Bratislava - FIT.

Poznámky:

(1) Ďalej vypočítu: K. Jelemská

(2) Určené pre: 5

Obr. 2: A boat.

- [2] Krzysztof Czarnecki, Simon Helsen, and Ulrich Eisenecker. Staged configuration through specialization and multi-level configuration of feature models. *Software Process: Improvement and Practice*, 10:143–169, April/June 2005.
- [3] Krzysztof Czarnecki and Chang Hwan Peter Kim. Cardinality-based feature modeling and constraints: A progress report. In *International Workshop on Software Factories, OOPSLA 2005*, San Diego, USA, October 2005.
- [4] Carnegie Mellon University Software Engineering Institute. A framework for software product line practice—version 5.0. http://www.sei.cmu.edu/productlines/frame_report/.