

# Ako funguje algoritmus odporúčaného obsahu v aplikáciách?\*

Samuel Kohút

Fakulta informatiky a informačných technológií  
[xkohuts@stuba.sk](mailto:xkohuts@stuba.sk)

13. októbra 2021

## Abstrakt

...

## 1 Abstrakt

V mojom článku sa plánujem zamerať na „Recommendations algorithm“, v preklade algoritmus odporúčania. Čo to vlastne znamená? V rôznych aplikáciach ste sa mohli stretnúť s odporúčaním obsahom na vašej domovskej stránke, ktorý verte či nie, bol vybraný len pre vás. Ako je to možné, že len pre vás? Na to sa zameriavam v svojom článku. Posnažím sa vám priblížiť model takéhoto algoritmu, ako funguje, ako spracúva všetky informácie, ktoré informácie spracúva, ako vyhodnocuje tieto informácie a nakoniec ako implementuje získane poznatky do vašich aplikácií.

$$v = s/t \quad (1)$$



## 2 Nejaká časť

Z obr. 1 je všetko jasné.

## 3 Iná časť

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

\*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2021/22, vedenie: Ing. Ján Lúčanský

<sup>1</sup>Niekedy môžete potrebovať aj poznámku pod čiarou.

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.

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 Nejaké vysvetlenie

Niekedy treba uviesť zoznam:

- jedna 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 hned za nadpisom.

## 4 Dôležitá časť

## 5 Ešte dôležitejšia časť

## 6 Záver

## Literatúra

- [1] James O. Coplien. *Multi-Paradigm Design for C++*. Addison-Wesley, 1999.
- [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/](http://www.sei.cmu.edu/productlines/frame_report/).