Elektronické podklady pro předmět Základy informatiky

Electronic groundwork for subject Basics of informatics

Jan Králík

Bakalářská práce
2009

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2008/2009

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Jan KRÁLÍK**
Studijní program:	**B 3902 Inženýrská informatika**
Studijní obor:	**Informační technologie**
Téma práce:	**Elektronické podklady pro předmět Základy informatiky.**

Zásady pro vypracování:

1. **Zpracujte literární rešerši na dané téma.**
2. **Seznamte se s existující elektronickou příručkou pro předmět Základy informatiky.**
3. **Navrhněte vhodnější strukturu a nový grafický vzhled dané příručky ve formě webové prezentace.**
4. **Doplňte příručku o algoritmy návrhu efektivních kódů. Vybrané algoritmy naprogramujte ve formě online webové aplikace. Výstupem bude i stromová struktura kódu a zhodnocení efektivity.**
5. **Umístěte vytvořenou příručku na webový server.**

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **Zelinka, I. Základy informatiky. Skriptum UTB Zlín, 2005. ISBN 80-7318-299-8.**
2. **Java tutorials [online] [cit. 2009-02-02]. Dostupný z WWW: < http://java.sun.com/docs/books/tutorial >.**
3. **PHP, Manuál PHP [online] [cit. 2009-02-02] Dostupný z WWW: < http://www.php.net/manual/cs/ >.**
4. **Jak psát web, návod na html stránky [online] [cit. 2007-05-19] Dostupný z WWW: < http://www.jakpsatweb.cz >.**
5. **Popelková Kateřina: Současná česká grafická úprava knih, možnost propagace literatury, elektronická kniha. Bakalářská práce. FT UTB Zlín, 2001.**

Vedoucí bakalářské práce: **Ing. Bronislav Chramcov, Ph.D.**
Ústav aplikované informatiky

Datum zadání bakalářské práce: **20. února 2009**

Termín odevzdání bakalářské práce: **1. června 2009**

Ve Zlíně dne 13. února 2009

L.S.

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky

prof. Ing. Vladimír Vašek, CSc.
děkan

doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Tato bakalářská práce se zabývá přípravou elektronických podkladů pro předmět Základy informatiky, nástroji e-learningu a metodami pro návrhy efektivních kódů. Teoretická část popisuje nejčastěji používané e-learningové nástroje na univerzitách v České republice. Další kapitola teoretické části se věnuje metodám návrhu efektivních kódů. Následně jsou uvedeny základní informace o použitých technologiích. Praktická část se zabývá úpravami stávající webové příručky a popisem webových aplikací pro návrhy efektivních kódů. Efektivní kódy jsou navrhovány Shannon-Fanovou metodou a Huffmanova metodou. Aplikace byly naprogramovány v programovacích jazycích Java a PHP.

Klíčová slova: e-learning, efektivní kódy, Java, WWW

ABSTRACT

This Bachelor thesis is engaged in preparation of electronic groundwork for subject Basics of informatics, instruments of e-learning and methods of developing effective codes. The theoretical part subscribes most frequently used instruments of e-learning at the Universities in Czech Republic. Another chapter of theoretical part is engaged in methods of developing effective codes. Afterwards are given basic information about used technologies. The practical part is engaged in modifications of existing web guide and description of web applications of developing effective codes. Effective codes are developed by Shannon-Fanov coding and Huffman coding. Applications were programmed in programming languages Java and PHP.

Keywords: e-learning, effective codes, Java, WWW

Rád bych zde poděkoval panu Ing. Bc. Bronislavu Chramcovovi, Ph.D. za cenné rady, připomínky a vedení při tvorbě této práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.

Ve Zlíně …….…….

podpis diplomanta

OBSAH

ÚVOD

Elektronické podklady mohou nabývat mnoha různých forem. Jednou z nich je forma webové příručky. Velkou výhodou je dostupnost a možnost procvičování si daného učiva pomocí interaktivních prvků dostupných on-line. Důležité ovšem je, aby webová příručka sloužící k výuce neměla rušivý vzhled, tak aby se čtenář mohl plně věnovat studování předkládaných materiálů. Dále je nezbytné, aby obsah webové příručky byl přehledný, tak aby se v něm student mohl snadno orientovat.

O dalších formách elektronických podkladů a nástrojích e-learningu pojednává literární rešerše v teoretické části.

Mým úkolem bylo upravit stávající webovou příručku podle výše zmíněných požadavků. Také jsem měl navrhnout lepší strukturu stávající webové příručky.

Další důležitou úlohou bylo doplnit webovou příručku o on-line aplikaci pro návrh efektivních kódů Shannon-Fanovou metodou a Huffmanovou metodou. Ve výsledku se zobrazí nejen navržená kódová slova, ale i postup, tak aby student lépe pochopil dané algoritmy. Aplikace byla naprogramována pomocí technologií Java a PHP.

I. TEORETICKÁ ČÁST

1 LITERÁRNÍ REŠERŠE

Tato bakalářská práce částečně spadá i do oblasti e-learningu. Proto jsem se rozhodl podat základní informace o této oblasti a nastínit některé možné aplikace e-learningu.

Existuje řada definic e-learningu, které vznikaly v různých dobách. Vzhledem k nepřetržitému dynamickému vývoji e-learningu samotného, i souvisejících informačních a komunikačních technologií, se často výrazně liší. Některé jsou až příliš jednoduché a naopak některé příliš akademické, některé jsou velmi široké, některé zužují význam až příliš. Uveďme čtyři z nich, použité v různých materiálech v poslední době:

1. E-learning je výuka s využitím výpočetní techniky a internetu.
2. E-learning je v podstatě jakékoli využívání elektronických materiálních a didaktických prostředků k efektivnímu dosažení vzdělávacího cíle s tím, že je realizován zejména/nejenom prostřednictvím počítačových sítí. V českém prostředí spojován zejména s řízeným studiem v rámci LMS.
3. E-learning je vzdělávací proces, využívající informační a komunikační technologie k tvorbě kursů, k distribuci studijního obsahu, komunikaci mezi studenty a pedagogy a k řízení studia.
4. E-learning je forma vzdělávání využívající multimediální prvky - prezentace a texty s odkazy, animované sekvence, video snímky, sdílené pracovní plochy, komunikaci s lektorem a spolužáky, testy, elektronické modely procesů, atd. v systému pro řízení studia (LMS).

Ze všech citovaných definic vyplývá, že e-learning v sobě zahrnuje řadu dílčích aktivit, které mohou být propojené do uceleného systému, ale také nemusejí. Může se jednat o rozsáhlé kurzy plně distančního charakteru a propracované nástroje kolaborativního učení, naopak ale může jít jen o doplnění prezenční výuky. Vhodných ICT nástrojů je řada: vystavení studijních materiálů na internetu nebo intranetu, nabídka k nim vztažených autotestů, komunikace prostřednictvím diskusních fór, e-mailů a dalších synchronních nebo asynchronních komunikačních nástrojů. Všechny uvedené nástroje je vhodné integrovat, pro tyto účely proto slouží specializované aplikace pro řízení procesu vzdělávání - LMS (Learning Management System). Těchto systémů je řada, kromě několika desítek nejznámějších existují stovky systémů s nejrůznějším rozsahem. [1]

Z citovaného textu vyplívá, že tak jako existuje mnoho pohledů na pojem e-learning, tak existuje mnoho nástrojů, kterými lze e-learning aplikovat. Stěžejní jsou pro nás tyto:

1. LMS

   Learning Management System je řídicí výukový systém (systém pro řízení výuky), tedy aplikace řešící administrativu a organizaci výuky v rámci e-learningu. LMS jsou aplikace, které v sobě integrují zpravidla nejrůznější on-line nástroje pro komunikaci a řízení studia (nástěnka, diskusní fórum, chat, tabule, evidence ad.) a zároveň zpřístupňují studentům učební materiály či výukový obsah on-line nebo i off-line. LMS aplikací je řada - od těch jednoduchých přes nejrůznější LMS z akademické sféry až po rozsáhlé a složité komerční aplikace. Řada LMS je šířených i jako free nebo open source software. [2]

   Jedním z těchto velmi využívaných LMS systémů je systém Moodle.

   Moodle je softwarový balíček pro tvorbu výukových systémů a elektronických kurzů na internetu. Je vyvíjen jako nástroj podporující sociálně konstruktivistický přístup ke vzdělávání. Moodle je poskytován zdarma jako Open Source software. [3]

2. Webové stránky bez interaktivního obsahu

   Z těchto webových stránek je možné stáhnout přednášky a to většinou ve formátu pdf či prezentace v programu PowerPoint nebo formě video záznamu. Webové stránky většinou neobsahují interaktivní učebními prvky. Příkladem takovýchto webových stránek na Fakultě aplikované informatiky jsou stránky předmětu Mikroelektronika (http://www.mikroelektro.utb.cz).

3. Webové stránky s interaktivním obsahem

   Webové stránky tohoto typu obsahují interaktivní studijní materiály fungující on-line, na kterých si může student ověřit znalost dané látky a vyjasnit si případné problémy.

Z průzkumu používaných e-learningových systémů na univerzitách v České republice bylo zjištěno, že jsou nejvíce používány LMS. Dále jsou hojně využívány webové stránky bez interaktivního obsahu. Mohou samozřejmě existovat i různé kombinace e-learningových nástrojů.

Webové stránky s interaktivním obsahem na vysokých školách v České republice se vyskytují velmi zřídka, byly nalezeny například webové stránky:

http://www.stringology.org/DataCompression. Je možné, že některé interaktivní studijní materiály jsou umístěny přímo v LMS, ale bohužel jsou kurzy zaheslovány, takže jsem nemohl tuto myšlenku ověřit. Každopádně jsou interaktivní studijní materiály pro studenta užitečné. Právě proto jsou tímto směrem směřovány webové stránky upravované v rámci této bakalářské práce.

2 NÁVRHY EFEKTIVNÍCH KÓDŮ

2.1 Efektivní kódy

Efektivní kódy jsou takové, kde průměrný počet kódových znaků připadajících na jeden zdrojový znak je minimální.

Z důvodu možnosti porovnání kódů z hlediska úspornosti, byly zavedeny tyto proměnné:

- Průměrná délka kódového slova

$$\overline{n} = \sum_{i=1}^{n} n_i \cdot p_i \tag{1}$$

- Efektivnost

pro zdrojovou abecedu A={a_1, a_2, … , a_N} s pravděpodobností jednotlivých znaků $P(a_i) = p_i$, $i=1, \ldots, N$ je:

$$\eta = \frac{-\sum_{i=1}^{n} p_i \cdot \log_2 \cdot p_i}{\sum_{i=1}^{n} n_i \cdot p_i} \cdot 100\% \tag{2}$$

2.2 Shannon-Fanova metoda

2.2.1 Historie

Tato metoda pochází z roku 1949. Publikovali ji nezávisle na sobě Claude Elwood Shannon (ten je označován jako otec teorie informace) s Warrenem Weaverem a Robert Mario Fano. [4]

2.2.2 Algoritmus

1. Zdrojové znaky se uspořádají postupně podle pravděpodobnosti p_i v klesajícím pořadí.
2. Zdrojové znaky se rozdělí na dvě podskupiny tak, aby součet pravděpodobností v obou skupinách byl přibližně stejný.
3. První skupině se přiřadí kódový znak 1 a druhé skupině 0.
4. Každou podskupinu opět rozdělíme na dvě podskupiny s přibližně stejným součtem pravděpodobností.
5. Prvním podskupinám se přiřadí znak 1 a druhým 0. Tak získáme druhý kódový znak.
6. V dělení na skupiny pokračujeme dokud v každé podskupině nezůstane pouze jeden zdrojový znak.

2.2.3 Příklad

Zadané hodnoty pravděpodobností 0.1, 0.3, 0.4, 0.2

1. Seřazení hodnot pravděpodobností (Tab. 1)

0.4	0.3	0.2	0.1

Tab. 1 Seřazení hodnot

2. Rozdělení na dvě podskupiny (Tab. 2)

0.4	0.4
0.3	0.6
0.2	
0.1	

Tab. 2 Rozdělení na podskupiny

3. Přiřazení kódového znaku (Tab. 3)

0.4	0.4	1
0.3	0.6	0
0.2		0
0.1		0

Tab. 3 Přiřazení kódového znaku

4. Rozdělení podskupin (Tab. 4)

0.4	0.4	1	
0.3	0.6	0	0.3
0.2		0	0.3
0.1		0	

Tab. 4 Rozdělení podskupin

5. Přiřazení kódového znaku (Tab. 5)

0.4	0.4	1		
0.3	0.6	0	0.3	1
0.2		0	0.3	0
0.1		0		0

Tab. 5 Přiřazení kódového znaku

6. Další dělení a přiřazování (Tab. 6)

0.4	0.4	1				
0.3	0.6	0	0.3	1		
0.2		0	0.3	0	0.2	1
0.1		0		0	0.1	0

Tab. 6 Další dělení a přiřazování

2.3 Huffmanova metoda

2.3.1 Historie

Kalendář ukazoval rok 1951. Na univerzitách a vysokých školách po celém světě se řeší desítky podobných problémů jako na té, kde studoval David Huffman. Profesor umožnil svým studentům vyhnout se zkoušce, když vyřeší složitý problém. Spočíval v dosažitelnosti nejkratšího prefixového kódování (tzv. ideální kódování ve vztahu k informační entropii). Jednalo se zatím o nevyřešenou úlohu, což ovšem studenti nevěděli. V té době byly známy jen docela neúčinné metody (analýzou shora dolů). Davidu Huffmanovi (narozen 1925) na univerzitě v Ohiu se ke zkoušce nechtělo. Nikoliv snad proto, že by látce nerozuměl, ale chtěl se zkoušce prostě vyhnout. Úloha se zpočátku zdála téměř neřešitelná. Když už chtěl nechat bádání a regulérně se na zkoušku připravit, zadíval se na papír s poznámkami, které zlostně vyhodil do koše a v tu chvíli ho to napadlo...

Později už jako magistr Huffman publikoval svůj nápad v práci nazvané Metoda pro vytvoření kódu s minimální redundancí (A Method for the Construction of Minimum Redundancy Codes). Jeho řešení pomocí binárního stromu bylo velmi prosté a zároveň

elegantní. Ale také velmi účinné v praxi. Svůj geniální nápad David Huffman však nikdy nepatentoval (ani nechtěl). Svému synovci totiž říkával: "Vždyť je to jen algoritmus." [5]

2.3.2 Algoritmus

1. Zdrojové znaky se uspořádají postupně podle pravděpodobnosti p_i v klesajícím pořadí.
2. Sečteme poslední dvě pravděpodobnosti a výsledek zařadíme podle velikosti mezi ostatní pravděpodobnosti - redukce.
3. Znovu sečteme dvě poslední pravděpodobnosti a výsledek opět zařadíme podle velikosti.
4. Sečítání pravděpodobností provádíme tak dlouho, až dojdeme k součtu 1.
5. Posledním dvěma znakům přiřadíme kódové znaky 0 a 1.
6. Zpětným postupem přiřazujeme jednotlivým sčítancům vždy kódové znaky 0 a 1, dokud nepřiřadíme kódové znaky všem zdrojovým znakům.

2.3.3 Příklad

Zadané hodnoty pravděpodobností 0.1, 0.3, 0.4, 0.2

1. Seřazení hodnot pravděpodobností (Tab. 7)

0.4	0.3	0.2	0.1

Tab. 7 Seřazení hodnot

2. Součet dvou posledních hodnot a seřazení podle velikosti (Tab. 8)

0.4	0.4
0.3	0.3
0.2	0.3
0.1	

Tab. 8 Součet posledních dvou hodnot

3. Další součet dvou posledních hodnot a seřazení podle velikosti (Tab. 9)

0.4	0.4	0.6
0.3	0.3	0.4
0.2	0.3	
0.1		

Tab. 9 Další součet dvou posledních hodnot

4. Součet posledních dvou hodnot pravděpodobností je roven 1.

5. Přiřazení kódových znaků (Tab. 10)

0.4	0.4	0.6	0
0.3	0.3	0.4	1
0.2	0.3		
0.1			

Tab. 10 Přiřazení kódových znaků

6. Postupné přiřazování 0 a 1 (Tab. 11) a (Tab. 12)

0.4	0.4	1	0.6	0
0.3	0.3	00	0.4	1
0.2	0.3	01		
0.1				

Tab. 11 Krok 2 zpětného chodu

0.4	1	0.4	1	0.6	0
0.3	00	0.3	00	0.4	1
0.2	000	0.3	01		
0.1	001				

Tab. 12 Krok 3. zpětného chodu

2.3.4 Další využití Huffmanovy metody

Dnes se nejrůznější varianty Huffmanova kódování (například adaptivní varianta) používají v mnoha produktech, zejména v některých komprimačních algoritmech. [5]

3 POUŽITÉ TECHNOLOGIE

3.1 HTML

Základní jazyk používaný pro vytváření hypertextových dokumentů (internetových stránek), přesněji k vymezení jejich struktury, na jejímž základě prohlížeč zobrazuje obsah. Dříve sloužil k formátování vzhledu, dnes je spíše využíván k definování základní obsahové kostry. [6]

3.2 CSS

CSS je zkratka pro anglický název Cascading Style Sheets, česky tabulky kaskádových stylů. Je to jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML. Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu.

3.3 PHP

PHP je v současnosti velmi rozšířená technologie umožňující snadné programování na straně serveru (server-side programming). Toho lze využít k tvorbě různých interaktivních webových stránek. Stručně lze říci, že skript napsaný v PHP je proveden na serveru podle zadaných kritérií a výsledek je odeslán volajícímu počítači stejným způsobem, jakým se odesílají běžné statické (XHTML) stránky. Jakmile je však stránka načtena klientem, pomocí PHP ji již není možné dále měnit. [7]

3.4 Java

Java je vyspělý programovací jazyk,obsahující všechny vlastnosti, které jsou vyžadovány v moderním programování, od modularity programu, řídících konstrukcí, přes silnou typovou kontrolu, multithreading, ošetření vyjímek, správu paměti, i silnou podporu pro databáze, XML a síťové operace.

K jejím výhodám patří kromě již zmíněné multiplatformity, patří robustnost, škálovatelnost a vysoká bezpečnost, která jí profituje pro používání na kritické aplikace na mainfraimových počítačích. [8]

3.4.1 Java applet

Applet je takové aplikace, která se spouští v okně internetového prohlížeče, který ji ovšem kontroluje. Applet je de facto součástí stránky prohlížeče jako komponenta. Je mu vymezena velikost i umístění. Parametry mu jsou zadány staticky ve stránce. Svůj výstup vypisuje do dané oblasti. [9]

3.5 Inkscape

Inkscape je open source vektorový grafický editor, schopnostmi podobný programům jako Illustrator, Freehand, CorelDraw, nebo Xara X a to za použití W3C standardu škálovatelné vektorové grafiky (SVG). [10]

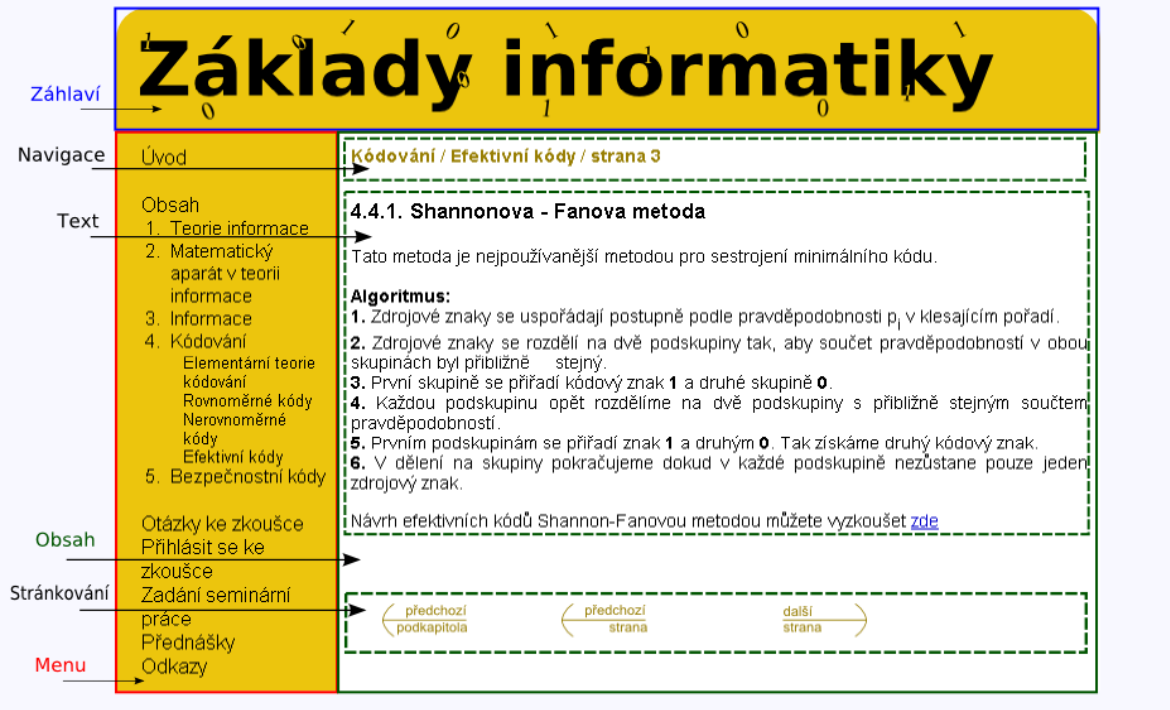
II. PRAKTICKÁ ČÁST

4 ÚPRAVA WWW PŘÍRUČKY

V této kapitole, jsou popsány nejdůležitější změny struktury a grafického vzhledu www prezentace. Největší úlohou bylo rozdělit nepřehledný dlouhý text, na přehlednější stránky, mezi kterými je možné se přepínat v rámci podkapitol. Skripty pro dynamické změny obsahu webové prezentace jsou oddělné od vlastního učebního textu.

4.1 Rozložení obsahu webové příručky

Rozložení obsahu webových stránek je realizováno pomocí tabulky rozdělené na buňky záhlaví, menu, obsah. Buňka obsah se dělí dále na oblasti navigace, text, stránkování. Viz obrázek (Obr. 1).



Obr. 1 Rozložení obsahu webové příručky

4.2 Dynamická změna obsahu webové příručky

Tato podkapitola popisuje jakým způsobem je realizován výpis učebního textu, zobrazení podmenu, rozdělení textu na stránky a navigaci.

4.2.1 Základní údaje

Dynamická změna obsahu je realizována skriptovacím jazykem PHP. Ke změně obsahu webové příručky slouží proměnné *k* a *s*. Proměnná *k* určuje číslo kapitoly. Proměnná *s* určuje číslo stránky. Tyto proměnné jsou skriptu, který vykonává změnu obsahu, předávány superglobální proměnnou GET.

4.2.2 Adresářová struktura

Pro lepší rozřazení souborů byla zvolena následující adresářová struktura:

kČísloKapitoly/pČísloPodkapitoly/sČísloStránky.html

Například k02/p03/s5.html je 5. stránka kapitoly 2.3

4.2.3 Grafické prvky

Záhlaví webové příručky a obrázky pro realizaci stránkování byly vytvořeny v grafickém editoru Inscape.

4.2.4 Zobrazení podmenu

Po kliknutí na položku menu proběhne refresh stránky, dojde k přepsaní hodnoty proměnné *k*. Pokud pro danou položku existuje podmenu, tak se z prvních dvou znaků proměnné *k* pomocí podmínky určí, které položky podmenu se mají zobrazit.

4.2.5 Změna obsahu v oblasti text

Pomocí proměnných *k* a *s* skript určí soubor jehož obsah se zobrazí v oblasti text buňky obsah. Při prvním načtení webové příručky jsou proměnné *k* a *s* nastaveny tak, aby se v oblasti text zobrazila první stránka úvodní kapitoly. Soubory načítané do oblasti text jsou ve formátu HTML.

4.2.6 Stránkování

Stránkování slouží k přecházení mezi jednotlivými podkapitolami a stránkami podkapitol.

Skript z adresářové struktury zjistí počet podkapitol pro aktuálně zobrazenou kapitolu. Pro aktuálně zobrazenou podkapitolu zjistí počet stránek. Potom skript vyhodnotí zda-li je možné přejít na následující a předchozí podkapitoly a stránky. Po vyhodnocení se v oblasti

stránkování zobrazí příslušné odkazy pro přechod na existující podkapitoly a stránky. Po kliknutí na příslušný odkaz proběhne skript popsaný v podkapitole 4.2.5

Přecházení mezi podpodkapitolami je řešeno jako přecházení mezi jednotlivými stránkami podkapitoly.

4.2.7 Navigace

Navigace slouží k lepší orientaci při procházení kapitol a podkapitol. Pro zobrazování navigace je využit soubor, ve kterém se nacházejí k číslům kapitol a podkapitol příslušné názvy. Skript pomocí proměnných *k* a *s* zjistí ve které kapitole se uživatel nachází, potom skript prohledá soubor a vypíše příslušný název kapitoly, podkapitoly, případně podpodkapitoly, a číslo stránky.

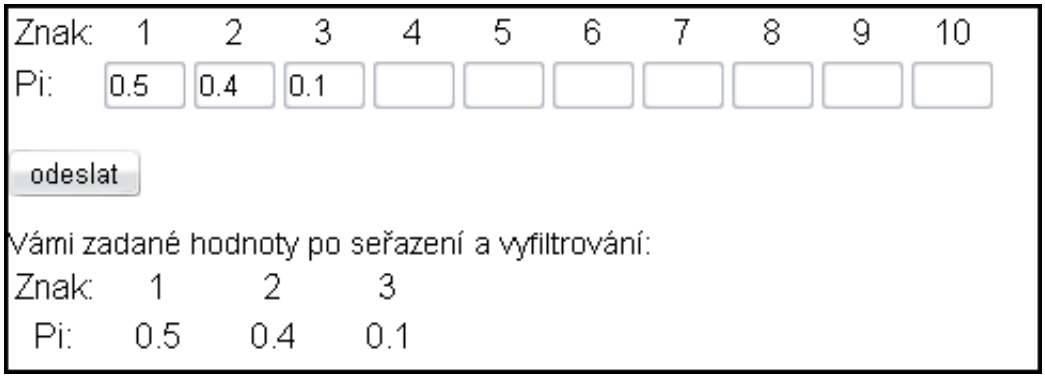
5 ALGORITMY NÁVRHŮ EFEKTIVNÍCH KÓDŮ

Hlavním cílem bakalářské práce bylo sestavit webovou aplikaci pro návrh efektivních kódů Huffmanovou a Shannon-Fanovou metodou.

Tato kapitola se zabývá popisem této webové aplikace. Webová aplikace je naprogramována v programovacích jazycích PHP a Java.

5.1 Zadávání hodnot

Hodnoty pravděpodobností jsou uživatelem zadávány přes HTML formulář. Správnost zadaných hodnot kontroluje PHP skript, který se vykoná po stisku tlačítka odeslat. Pokud jsou všechny hodnoty v pořádku, předají se jako parametry Java appletu. Ten se následně spustí a zobrazí výsledky. Dále se zobrazí tabulka se seřazenými hodnotami pravděpodobností (Obr. 2). Jinak se vypíše chybová hláška a Java applet se nezobrazí (Obr. 3).



Obr. 2 Příklad správně zadaných hodnot

Obr. 3 Příklad špatně zadaných hodnot

5.2 Java applet Shannon-Fanova metoda

5.2.1 Struktura appletu

Applet se skládá z pěti tříd:

1. Hlavni
2. Algoritmus
3. Platno
4. Tabulka
5. UpravaTabulky

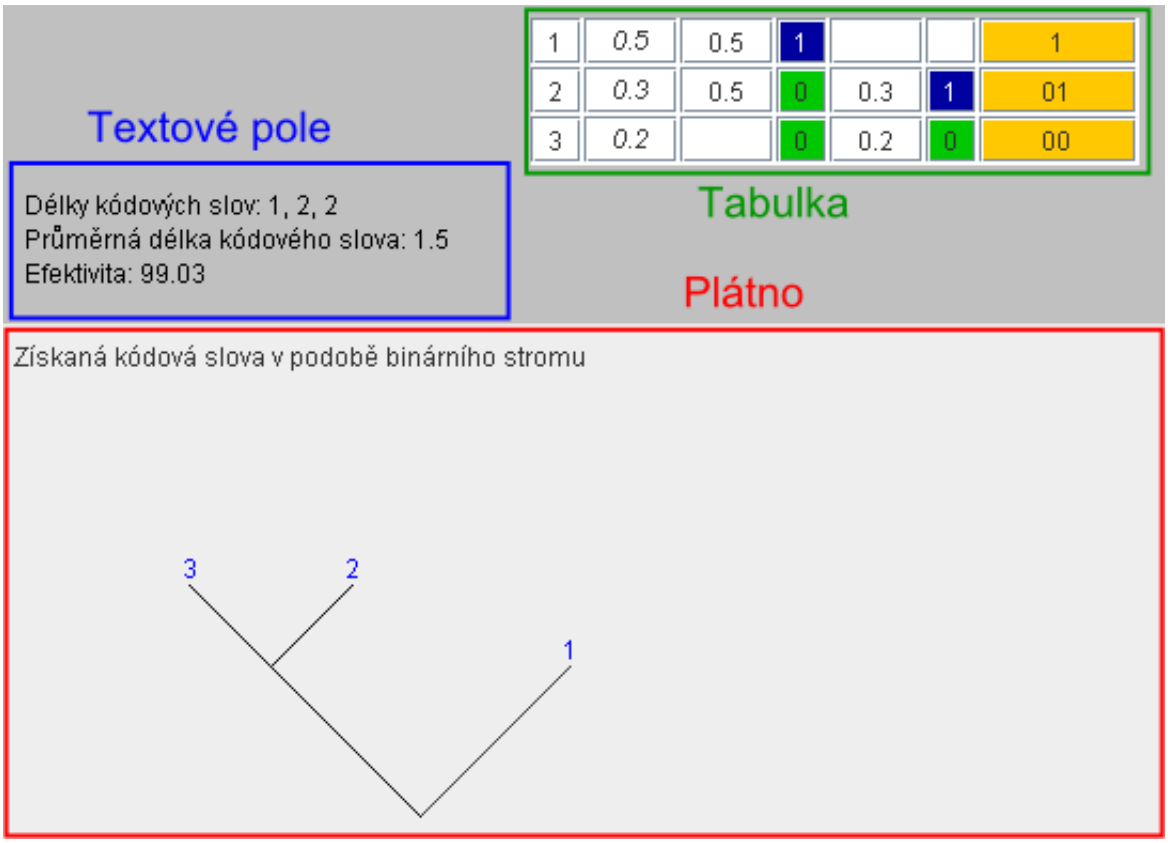
5.2.2 Třída Hlavni

Třída Hlavni obsahuje metodu *start()*. Tato metoda získává hodnoty parametrů appletu, ukládá je do pole pro další využití a stará se o vykreslení položek appletu. Položky jsou 3:

1. Tabulka – slouží pro zobrazení postupu a výsledných slov.
2. Textové pole – slouží pro zobrazení délek kódových slov, průměrné délky kódového slova a efektivity.
3. Plátno – slouží pro vykreslení binárního stromu.

Rozložení je uvedeno na obrázku (Obr. 4).

Dále vytváří instanci třídy *Algoritmus* a volá její metodu *algoritmus()*, která se stará o výpočet efektivního kódu.



Obr. 4 Prvky appletu

5.2.3 Třída Algoritmus

Třída algoritmus vypočítává efektivní kódy Shannon-Fanovou metodou. Obsahuje následující proměnné a metody. Ze všech proměnných a metod jsou vybrány nejdůležitější pro pochopení fungování algoritmu.

- proměnná *pole* - jednorozměrné pole typu float, obsahuje seřazené hodnoty pravděpodobností.
- proměnná *pole2* - dvojrozměrné pole typu integer. Zde se ukládají hodnoty 0 a 1. Po skončení algoritmu jsou v poli zapsána výsledná kódová slova.
- proměnná *krok* - obsahuje aktuální číslo kroku algoritmu. Krok algoritmu symbolizuje rozdělení všech intervalů vzniklých v předchozím kroku a přidělení hodnot 0 a 1 pro vzniklé podintervaly.
- proměnná *velikost* - obsahuje počet nenulových hodnot pravděpodobností.

- proměnná *hranice* - Obsahuje číslo pořadí znaku, za kterým leží pomyslná hranice, rozdělující interval pole na dva podintervaly.
- proměnná *chyba* - Obsahuje popis chyby. Pokud je proměnná prázdná nedošlo k žádné chybě.
- metoda *ziskejZnaky()* - postupně prochází zadané hodnoty pravděpodobností, zjišťuje zda odpovídají podmínkám. Pokud ano ukládá je do proměnné pole. Pokud ne do proměnné chyba se přiřadí text chyby. Filtruje nulové hodnoty.
- metoda *zjistiVelikost()* - určí počet správně zadaných hodnot a uloží výsledek do proměnné *velikost*.
- metoda *chyby()* - testuje zda je součet pravděpodobností roven 1 a zda-li je počet zadaných pravděpodobností větší roven 3. Vrací proměnnou *chyba*.
- metoda *rozdeleni()* - pomyslně rozdělí interval v argumentu zadaného pole na další dva intervaly na základě součtu hodnot v každém podintervalu. Vrací proměnnou *hranice*.
- metoda *prideleni()* - přiřazuje do pole hodnoty 0 a 1 podle proměnné *hranice* na zadaném intervalu.
- metoda *hlavniKrok()* - pomocí metod *rozdeleni()* a *prideleni()* zjistí pro, který podinterval má přiřadit hodnotu 1 a pro který hodnotu 0. Dále zjistí součty pravděpodobností pro oba podintervaly, kvůli pozdějšímu výpisu do tabulky. Výsledek metody *prideleni()* je uložen v proměnné *pole2*.
- metoda *vypisDat( )*- stará se o vyplnění tabulky vypočítanými hodnotami. Dále vypočítává a vypisuje efektivitu, délky kódových slov a průměrnou délku kódového slova.

- metoda *algoritmus()* - používá předchozí metody pro návrh efektivního kódu následujícím způsobem.

1. Provedou se metody, které zajistí seřazení hodnot pravděpodobností, vyfiltrování nulových hodnot, a zjištění chyb.
2. Pokud nedošlo k žádným chybám nastaví proběhne metoda *hlavniKrok()*.
3. Probíhá cyklus while tak dlouho dokud bude možno dělit nově vzniklé intervaly na další podintervaly.
4. V tomto cyklu algoritmus pomocí cyklu for projde intervaly vzniklé v předchozím kroku a provede metodu *hlavniKrok()*.
5. Po skončení cyklu while se vytvoří instance třídy *Tabulka* a provede metoda *vypisDat()*.
6. Nastaví se proměnná vykresli (tato proměnná určuje zda-li se provede vykreslení binárního stromu a vykreslení tabulky) na true.

5.2.4 Třída Platno

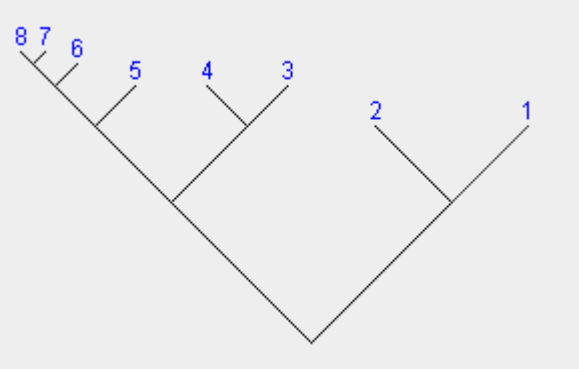
Tato třída obsahuje metodu *paint()*, která se stará o vykreslení binárního stromu. Strom je kreslen následujícím algoritmem:

1. Nejprve je zjištěna délka nejdelšího slova. Slouží pro rozhodnutí, jakým způsobem budou určovány vzdálenosti větví stromu v závislosti na délce vykreslovaného kódového slova.
2. Vykreslení větví každého slova začíná vždy ve pevně daném výchozím bodě.
3. Algoritmus postupně prochází číslice každého kódového slova.
4. Pokud je číslice rovna 1 vypočítá algoritmus další bod šikmo vpravo od výchozího bodu a vykreslí mezi těmito body přímku.
5. Pokud je číslice rovna 0 vykreslí přímku šikmo nalevo.
6. Novým výchozím bodem je zvolen bod, který byl vypočítám pro předchozí číslici.
7. Přímky jsou vykreslovány tak dlouho dokud algoritmus nenarazí na konec slova.

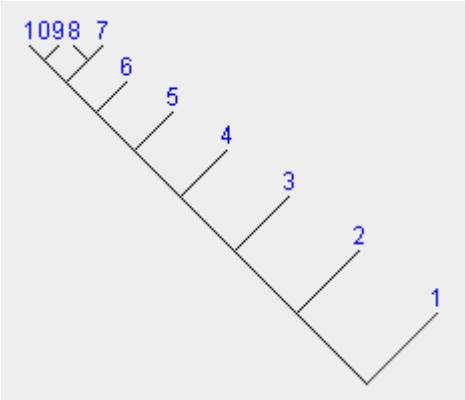
8. Nad poslední větev se vypíše číslo znaku, odpovídající příslušnému kódovému slovu.
9. Pro další slovo je pak nastaven znovu pevně zvolený výchozí bod.
10. Pokud je délka nejdelšího slova menší rovna 5, potom s každou přibývající přímkou je vzdálenost mezi body zkracována přibližně na polovinu. Zkracování je zvoleno kvůli tomu, aby se nekřížily větve uprostřed stromu. Ukázka stromu na obrázku (Obr. 5)
11. Pokud je délka nejdelšího slova větší než 6, potom s každou přibývající přímkou je vzdálenost mezi body zkracována o konstantu. Ukázka stromu na obrázku (Obr. 6)

Tento postup byl zvolen, protože délka větví pro kódová slova delší než 5 je příliš malá, větve jsou špatně viditelné a může dojít k překrývání čísel. Také je menší pravděpodobnost, že by se mohly větve křížit.

Problém křížení větví a překrývání čísel je velmi složitý a proto může i při tomto řešení dojít k některému z nevhodných jevů. Jedná se však o celkem malé procento případů.



Obr. 5 Strom s největší délkou kódového slova rovnou pěti.



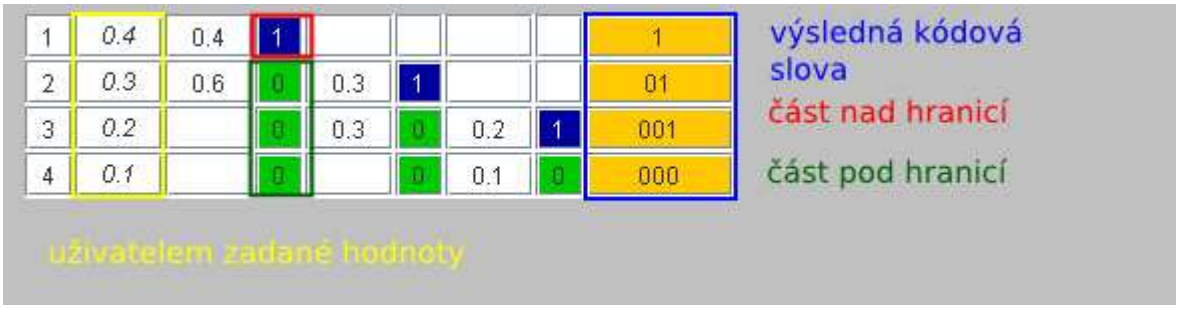
Obr. 6 Strom s největší délkou kódového slova rovnou osmi.

5.2.5 Třída Tabulka

Vytváří tabulku a nastavuje její parametry. Především šířky buněk, tak aby odpovídaly délce textu v nich.

5.2.6 Třída UpravaTabulky

Obsahuje metodu *getTableCellRendererComponent()*, která umožňuje měnit vlastnosti buněk tabulky podle zadaných podmínek. V našem případě jsou nastavována pozadí buněk, aby bylo rozlišeno, kde došlo k rozdělení intervalu. Část nad hranicí má modré pozadí a bílý text. Část pod hranicí má zelené pozadí a černý text. Také jsou barevně (oranžově) odlišena výsledná kódová slova. Kurzívou jsou napsány uživatelem zadané hodnoty pravděpodobností. Ukázka upravené tabulky je na obrázku (Obr. 7).



Obr. 7 Tabulka s postupem a výslednými kódovými slovy Shannon-Fanova metoda.

5.3 Java applet Huffmanova metoda

Struktura appletu a některé třídy pro Huffmanovu metodu jsou stejné jako struktura a třídy appletu pro metodu Shannon-Fanovu. Proto zde budou popsány pouze třídy, které se odlišují.

5.3.1 Třída Algoritmus

Třída algoritmus vypočítává efektivní kódy Huffmanovou metodou. Obsahuje následující proměnné a metody. Ze všech proměnných a metod jsou vybrány nejdůležitější pro pochopení fungování algoritmu.

- proměnná *polePoc* - jednorozměrné pole typu float, obsahuje seřazené hodnoty pravděpodobností.
- proměnná *polePravd* - dvojrozměrné pole ty float. Uchovává hodnoty pravděpodobností po součtu nejměnších dvou hodnot a seřazení v každém sloupci.
- proměnná *poleSlov* - Jednorozměrné pole typu String. Po skončení algoritmu obsahuje výsledná kódová slova.
- proměnná *poleJe* - Slouží k zjišťování, kde se nachází hodnota pravděpodobnosti, která je složena ze dvou hodnot pravděpodobností z předchozího sloupce.
- metoda *pripravPole()* - Vezme hodnoty pravděpodobností z proměnné *polePoc* a přepíše je do prvního sloupce proměnné *polePravd* Potom sečte nejmenší hodnoty prvního sloupce a do následujícího sloupce tento součet zapíše. Pak seřadí pole on největší hodnoty po nejmenší. To se provádí tak dlouho dokud pokud počet hodnot ve sloupci je menší roven 2. Vrací proměnnou *polePravd*.
- metoda *algoritmus()*

Používá předchozí metody pro návrh efektivního kódu následujícím způsobem.

1. Provedou se metody, které zajistí seřazení hodnot pravděpodobností, vyfiltrování nulových hodnot, a zjištění chyb.
2. Pokud nedošlo k žádným chybám do posledního sloupce proměnné poleSlov() se do první buňky zapíše 0 a do druhé 1.

3. Pro každý sloupec směrem od posledního k prvnímu pomocí pole *poleJe()* se vyhodnotí, která hodnota pravděpodobnosti vznikla z nejmenších dvou hodnot v následujícím sloupci.
4. Do příslušných buněk pole *poleSlov* se zapíše hodnota kódového slova odpovídající vzniklé hodnotě pravděpodobnosti K hodnotě prvního zapsaného kódového slova se připíše 0 a k hodnotě druhého zapsaného kódového slova se připíše1.
5. Vytvoří se instance třídy *Tabulka* a provede metoda *vypisDat()*.
6. Nastaví se proměnné vykresli (tato proměnná určuje zda-li se provede vykreslení binárního stromu a vykreslení tabulky) na true.

Následující metody a proměnné jsou popsány v třídě *Algoritmus* appletu Shanon-Fanova metoda.

- proměnná *velikost*
- proměnná *chyba*
- metoda *ziskejZnaky()*
- metoda *zjistiVelikost()*
- metoda *chyby()*
- metoda vypisDat()

5.3.2 Třída UpravaTabulky

Obsahuje metodu *getTableCellRendererComponent()*, která umožňuje měnit vlastnosti buněk tabulky podle zadaných podmínek. V našem případě jsou nastavována pozadí buněk barva textu. Červeně obarvená hodnota pravděpodobnosti značí že tato hodnota vznikla součtem dvou hodnot pravděpodobností z předchozího sloupce. Růžově jsou podbarveny hodnoty pravděpodobností, ze kterých je složena hodnota pravděpodobnosti v následujícím sloupci. Oranžově jsou podbarvena výsledná kódová slova. Ukázka upravené tabulky je na obrázku (Obr. 8).

Obr. 8 Tabulka s postupem a výslednými kódovými slovy Huffmanova metoda.

5.4 Návod pro uživatele

5.4.1 Zadávání hodnot

Zadejte hodnoty pravděpodobnosti, ze kterých bude navržen efektivní kód. Hodnoty se zadávají do formuláře a platí pro ně následující podmínky:

- Minimální počet hodnot je 3
- Součet pravděpodobností musí být roven 1
- Pravděpodobnost musí být z intervalu (0;1>
- Pro nulovou pravděpodobnost se kód nesestavuje
- Pravděpodobnost musí být zadána s desetinnou "." např. 0.5
- Počet desetinných míst je maximálně 2

Pokud jsou všechny hodnoty v pořádku, zobrazí se tabulka se seřazenými hodnotami pravděpodobností, postup a výsledky viz obrázek (Obr. 2). Jinak se vypíše chybová hláška viz obrázek (Obr. 3).

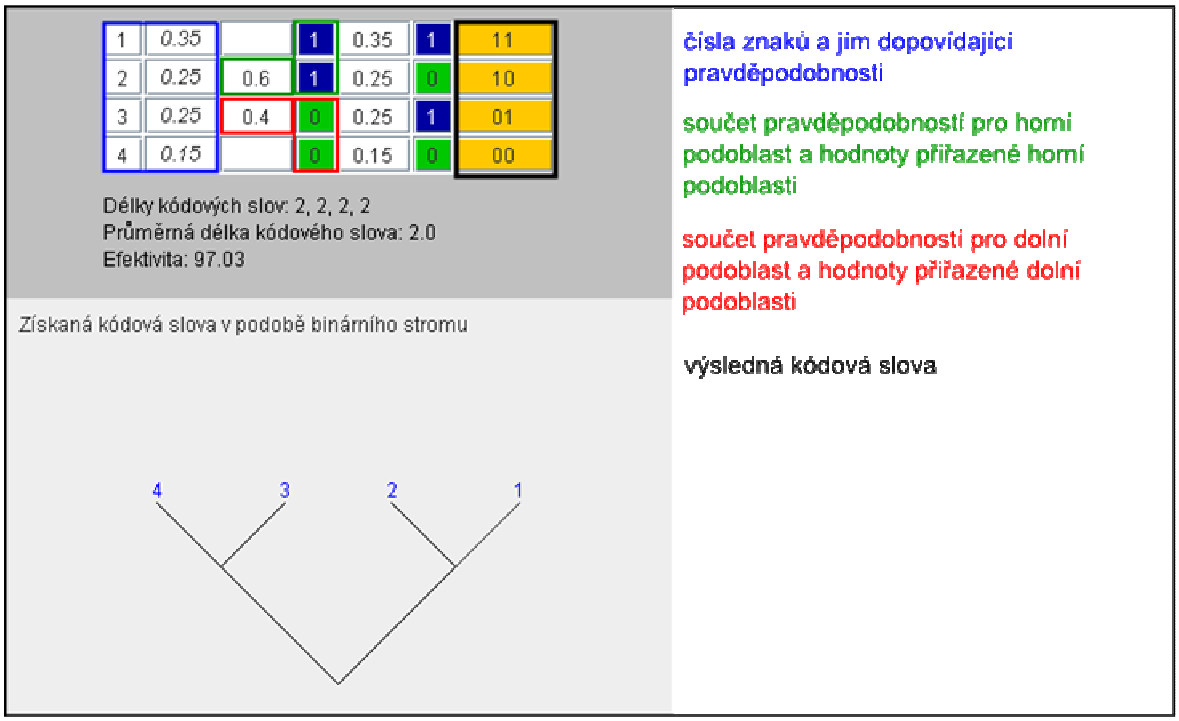
Může chvíli trvat než se načte applet s výsledky, je proto nutné počkat.

5.4.2 Zobrazení výsledků Shanon-Fanova metoda

- V prvním sloupci tabulky s postupem jsou zobrazena čísla znaků, kterým odpovídají hodnoty pravděpodobností v druhém sloupci.

- Pro následující dvojice sloupců platí, že v prvním sloupci jsou zapsány součty hodnot pravděpodobností pro vzniklé podskupiny. V druhém sloupci jsou zapsány hodnoty přiřazené k podskupinám. Hodnoty přiřazené k první podskupině jsou podbarveny modře. Hodnoty přiřazené k druhé podskupině jsou podbarveny zeleně.
- V posledním sloupci jsou zobrazena výsledná kódová slova.
- Kódová slova jsou reprezentována binárním stromem.

Ukázka výsledků a postupu je uvedena na obrázku (Obr. 9).

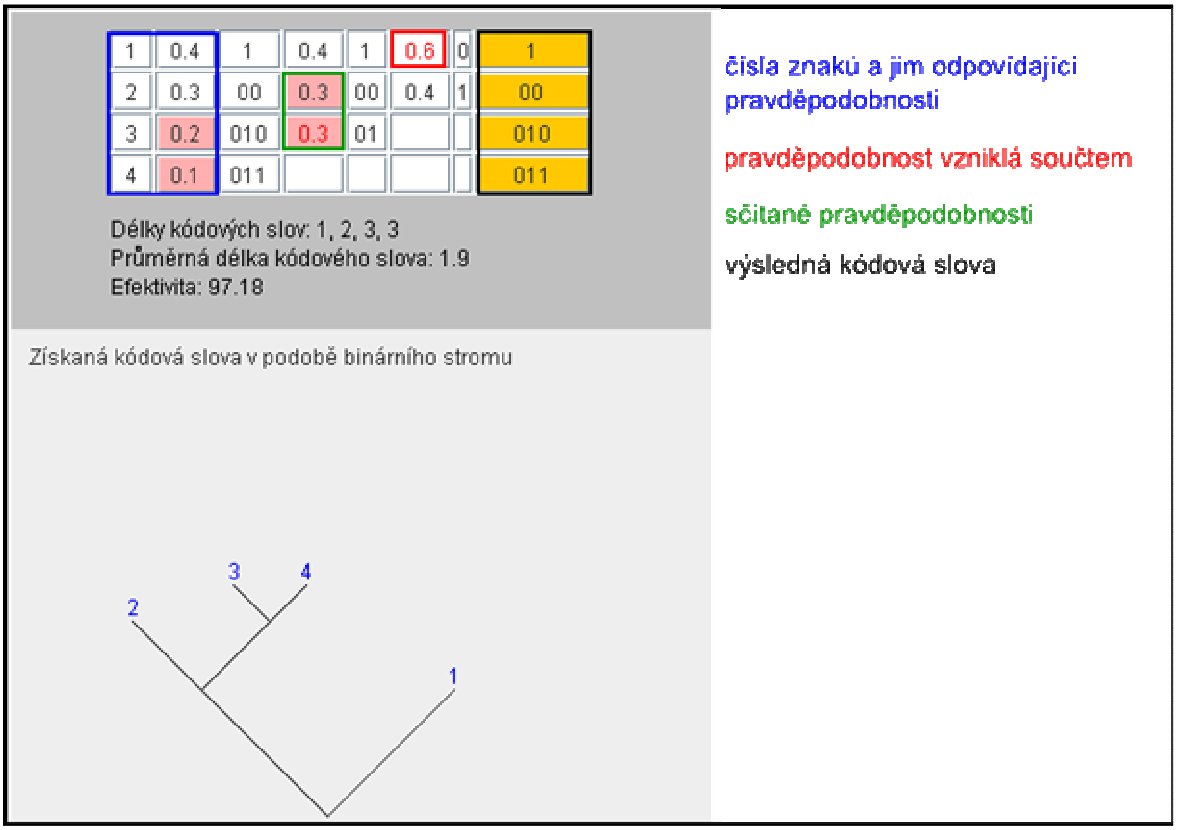


Obr. 9 Popis výsledků Shannon-Fanova metoda.

5.4.3 Zobrazení výsledků Huffmanova metoda

- V prvním sloupci tabulky s postupem jsou zobrazena čísla znaků, kterým odpovídají hodnoty pravděpodobností ve druhém sloupci.
- Červeně obarvená hodnota pravděpodobnosti značí že tato hodnota vznikla součtem dvou hodnot pravděpodobností z předchozího sloupce.
- Růžově jsou podbarveny hodnoty pravděpodobností, ze kterých je složena hodnota pravděpodobnosti v následujícím sloupci.
- Oranžově jsou podbarvena výsledná kódová slova.

- Kódová slova jsou reprezentována binárním stromem.



Obr. 10 Popis výsledků Huffmanova metoda.

5.5 Umístění na webový server

Webové stránky byly umístěny na free-hostingový server ic.cz. Byla zvolena adresa http://www.zin-fai.ic.cz

ZÁVĚR

Prvním z cílů této bakalářské práce bylo navrhnout novou a lepší strukturu stávající webové příručky. Hlavní změnou je dynamické zobrazování obsahu. Dále byl pro lepší přehlednost text rozdělen na více stránek. Bylo vytvořeno přecházení na následující a předchozí stránky. Také byla doplněna navigace pro lepší orientaci v textu. Přecházení mezi stránkami a navigace jsou řešeny dynamicky, skriptovacím jazykem PHP. Byl navržen nový grafický vzhled, tak aby působil příjemně a zároveň nerušil.

Druhým cílem bylo zpracovat téma návrhů efektivních kódů. V teoretické části jsou podány základní informace o metodách návrhů efektivních kódů. Na základě získaných teoretických poznatků byla doplněna stávající webová příručka o algoritmy návrhů efektivních kódů ve formě on-line webové aplikace. Aplikace byla naprogramována pomocí technologií PHP a Java. Ve výsledku je zobrazena tabulka s postupem. Dále je vypsána vypočítaná efektivita, průměrná délka kódového slova a délky kódových slov. Také je vykreslen binární strom pro získaná kódová slova. Při vykreslování binárního stromu může dojít ke křížení některých větví s dalšími větvemi. Ač jsem se snažil tento problém vyřešit pro všechny případy, nepodařilo se mi to. Jedná se ale o malý počet případů. Pro většinu kódových slov je strom vykreslen správně a přehledně.

ZÁVĚR V ANGLIČTINĚ

Proposing of new and better structure of current web guide was first point of this Bachelor thesis. The main change is dynamical displaying of contents. Next for better lucidity was text divided into more pages. Listing between next and previous pages was made. Navigation for better lucidity in text was made too. Listening between pages and navigation are solved dynamical by scripting language PHP. New friendly and not disruptive design was made.

Elaboration of theme methods of developing effective codes was another point of this Bachelor thesis. There is subscribed basic information about methods of developing effective codes in theoretical part. Current web guide was supplemented by algorithms of developing effective codes in the form on-line web application on the basis of the theoretical knowledge. Application was programmed by technologies PHP and Java. Table with procedure is displayed in the end. Calculated efficiency, average length of codeword and codeword lengths are displayed next. Binary tree of gained codeword is displayed too. When is drawing binary tree some nodes may be cross another nodes. Although I strove for finding resolution of this problem for each event, I didn’t find it. However it is only few events. Binary tree is drawn correctly and clearly for majority of codeword.

SEZNAM POUŽITÉ LITERATURY

[1] *Wikipedia: E-learning* [online]. 2009 [cit. 2009-04-26]. Dostupný z WWW: http://cs.wikipedia.org/wiki/E-learning

[2] *Wikipedia: LMS* [online]. 2009 [cit. 2009-04-28]. Dostupný z WWW: http://cs.wikipedia.org/wiki/LMS

[3] *Wikipedia: Moodle* [online]. 2009 [cit. 2009-04-27]. Dostupný z WWW: http://cs.wikipedia.org/wiki/Moodle

[4] *Komprese dat* [online]. 2006 [cit. 2009-04-30]. Dostupný z WWW: http://www.stringology.org/DataCompression/sf/index_cs.html

[5] *Komprese dat* [online]. 2006 [cit. 2009-04-30]. Dostupný z WWW: http://www.stringology.org/DataCompression/sh/index_cs.html

[6] *Slovník pojmů* [online]. c2009 [cit. 2009-05-11]. Dostupný z WWW: http://www.fg.cz/cs/prectete-si/slovnik-pojmu/a-z.shtml

[7] *Wikipedia:Kaskádové styly* [online]. 2009 [cit. 2009-04-17]. Dostupný z WWW: http://cs.wikipedia.org/wiki/Cascading_Style_Sheets

[8] *Interval:PHP* [online]. 2006 [cit. 2009-04-17]. Dostupný z WWW: http://php.interval.cz/clanky/co-je-to-php

[9] *Builder:Java applet* [online]. 2001 [cit. 2009-04-17]. Dostupný z WWW: http://www.builder.cz/art/java/progjava2.html

[10] *Inkscape* [online]. c2009 [cit. 2009-04-25]. Dostupný z WWW: http://www.inkscape.org/index.php?lang=cs

[11] Zelinka, I. Základy informatiky. Skriptum UTB Zlín, 2005. ISBN 80-7318-299-8

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

CSS	Cascading Style Sheets
HTML	HyperText Markup Language
ICT	Information and Communication Technologies
PHP	Hypertext Preprocessor
LMS	Learning Management System
SVG	Scalable Vector Graphics
WWW	World Wide Web
W3C	The World Wide Web Consortium

SEZNAM OBRÁZKŮ

SEZNAM TABULEK