

Příprava elektronické učební pomůcky pro předmět Základy informatiky

Michal Ptáček

Bakalářská práce
2006



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Vložit oficiální zadání diplomové/bakalářské práce

Prohlašuji, že jsem na celé diplomové práci pracoval samostatně a použitou literaturu jsem citoval.

Ve Zlíně, 16. 06. 2006

.....

jméno diplomanta

Abstrakt

Hlavním cílem práce bylo doplnit webové stránky předmětu Základy informatiky. Úkolem bylo zpracovat tématický celek Bezpečnostních kódů, speciálně pak kapitoly Systematické a nesystematické kódy. Tyto materiály by měly ulehčit výuku a dopomoci studentům při studiu. Studenti zde naleznou veškeré materiály k dané problematice, jak část teoretickou tak i řešené příklady. Hlavní přínosem bylo vytvoření programů, kde může student snáze pochopit některé kapitoly dané problematiky, nebo si jenom ověřit správnost výpočtů či přenosu kódových slov.

Main object of my work was to add pages of Basic information science to the web pages. My objective supposed to be to work up theme object Safety codes, especialy chapters Systematic & Nosystematic codes. These stuffs should help teaching and help students with learnig. The students can find here all stuff about this problem, theorem practice advices and solved examples. The main contribution was the write programs whith each student can find how to understand to some chapters abou that easily. Or only check calculation of transmition codes words.

OBSAH

ÚVOD.....	5
1 TEORETICKÁ ČÁST.....	6
1.1 CO JE TO WEB.....	6
1.2 HTML.....	7
1.3 PHP.....	7
1.4 JAVASCRIPT.....	8
1.5 CSS – KASKÁDOVÉ STYLY.....	9
2 PRAKTICKÁ ČÁST.....	10
2.1 GRAFICKÝ VZHLED.....	10
2.2 OBSAH KATEGORIE BEZPEČNOSTNÍ KÓDY.....	12
2.2.1 Systematické kódy.....	13
2.2.1.1 Hammingovy kódy.....	14
2.2.1.2 Rozšířený Hammingův kód.....	18
2.2.1.3 Cyklické kódy.....	23
2.2.2 Nesystematické kódy.....	23
3 UMÍSTĚNÍ WEBOVÝCH STRÁNEK.....	25
3.1 KDE STRÁNKY NAJDETE.....	25
3.2 CO JE TO ŠKOLNÍ SYSTÉM MOODLE.....	25
ZÁVĚR.....	27
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	28
SEZNAM OBRÁZKŮ.....	29
SEZNAM TABULEK.....	30
SEZNAM ROVNIC.....	31
SEZNAM POUŽITÉ LITERATURY.....	32
SEZNAM PŘÍLOH.....	33

ÚVOD

Web nebo-li WWW (World Wide Web) je v dnešní době opravdu rozsáhlé médium. Je spousta lidí, podnikatelů nebo škol, které se bez webových stránek neobejdou. Dá se říct že v dnešní době je už nutnost pokud jste podnikatel nebo nějaká organizace, mít své vlastní webové stránky. Používají své webové stránky jako takovou formu reklamy. Ovšem neslouží pouze k reklamě, ale hlavně na nich zveřejňují veškeré informace o své firmě či své činnosti.

Webové stránky mohou mít různé grafické provedení a strukturu. V dnešní době je spousta knih, příruček a internetových článků o tvorbě webových stránek v mnoha používaných jazycích. Jako je XHTML, CSS, PHP, ASP, JavaScript, apod.

Jak jsem se již na začátku zmiňoval, také školy mají své webové stránky. Ulehčuje to komunikaci mezi školou a studentem. Na stránkách můžeme nalézt informace o škole, jejích fakultách, vyučovaných předmětech, seznamy učitelů a jejich kontakty, novinky v daném roce, rozhodnutí děkana a spoustu dalších důležitých věcí, které bychom museli hledat na nástěnkách či někde jinde.

Já jsem měl za úkol doplnit existující elektronickou učební pomůcku pro předmět Základy informatiky, která bude sloužit studentům jak kombinované formy studia tak i prezenční formy studia. Úkolem bylo doplnit tématický okruh Bezpečnostní kódy, zejména pak kapitoly týkající se systematických a nesystematických kódů. Vedle teoretických podkladů bude příručka obohacena o názorné řešené příklady, a bude zpracována ve formě WWW stránek.

1 TEORETICKÁ ČÁST

1.1 Co je to web

Web (či WWW, World Wide Web) je jednou ze služeb Internetu. Internet je vlastně tvořen několika desítkami tisíc počítačů, které jsou trvale spuštěny a zapojeny v sítích. Princip webu spočívá hlavně ve zrození takového nového typu klientského programu, který běží na počítači u uživatele. Takovému programu se říká prohlížeč (browser) a jedná se o program, který dekoduje data a jednoduché instrukce zaslané serverem. Pak z nich sestaví stránku a tu zobrazí ve svém hlavním okně.

Web je vlastně taková obrovská otevřená kartotéka s několika milióny stránek, kterou může každý číst a do které může také každý psát.

Toto médium je ve velké šířce využíváno například jako slovník, encyklopedie, učebnice, obrázková kniha, hudební nosič nebo noviny. Ovšem nejdůležitější je, že slouží jako místo pro obchodování s čímkoliv a na spoustu způsobů.

Web má obsah a formu stejně jako každé jiné médium. Například to, co chce sdělit a způsob jakým to chce sdělit. Webová stránka je na serveru (tomu se pak říká web server, též webový server, server WWW) reprezentována skupinou souborů, které jsou takovými stavebními kostkami této stránky. V textových souborech HTML jsou pak takové návody pro sestavení stránky.

Tvorba webových stránek dnes není zas tak složitá, jelikož existuje spousta příruček, podle kterých lze jednoduché stránky vytvořit. K vytvoření stránek je možné použít například již zmíněný HTML kód, PHP kód, DHTML kód nebo XML kód a ještě spousta dalších.

Mým úkolem bylo jedny takové stránky vytvořit pro Fakultu aplikované informatiky na Univerzitě Tomáše Bati ve Zlíně.

1.2 HTML

HTML (Hypertext Markup Language) je jazyk, který je používán k popisu webových stránek. Je to jako bychom vytvářeli takové „recepty“, které po správném „uvaření“ zobrazí stránky podle našich představ. Jeho podstatné dvě složky jsou hypertext, tj. odkazy na externí elementy, a markup, tj. využívání značek (štítku, tagu) pro formátování dokumentu. Základem tohoto jazyka jsou značky, což je text ve špičatých závorkách `< >`. Internetový prohlížeč má za úkol vyhodnotit a zpracovat text v závorkách. Text, který je zapsán mimo tyto závorky je zobrazen na obrazovce.

Důležité základní principy jsou například takové, že HTML je pouze textový formát. Pokud se na stránce vyskytnou binární data, jako jsou třeba obrázky nebo animace, je na ně odkázáno. Tato data pak nejsou umístěna do souboru se základním popisem stránky. Příkazům ve špičatých závorkách se říká tagy. Tagy bývají párové a nepárové. Párové tagy mají počáteční a uzavírací značku (`< >` a `</>`) a slouží k formátování elementu. nepárové tagy se vztahují na celý dokument (např. na jeho pozadí) nebo slouží pro vkládání různých objektů jako je například obrázek. Hned za první otevírací závorkou je jméno příkazu a pak jeho parametry.

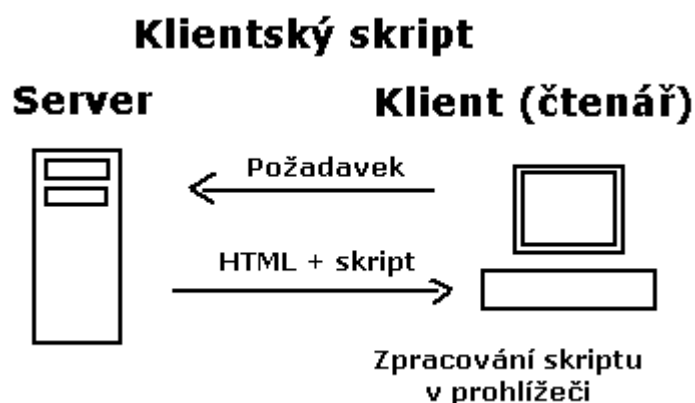
1.3 PHP

PHP je dnes velmi rozšířená technologie umožňující snadné programování na straně serveru (server-side programming). Toho lze využít k tvorbě různých interaktivních webových stránek. Stručně lze říct, že skript napsaný v PHP vygeneruje stránku podle zadaných kritérií a výsledek odešle volajícímu počítači stejným způsobem, jakým se odesílají běžné statické stránky. Například prosté zobrazení data a jména, které v daný den slaví svátek: je bezesporu jednodušší, aby server sám zjistil, kolikátého je, k tomu přiřadil odpovídající jméno z databáze jmenin a toto předal počítači, který si danou stránku vyžádal, než aby správce stránek každý den přesně o půlnoci na server uploadoval novou verzi stránek s opravenými údaji.

PHP je objektově orientovaný programovací jazyk, svojí strukturou velmi podobný jazyku C++. Poslední dostupná verze PHP je 5.1.4. [1]

1.4 JavaScript

JavaScript je klientský skript. To znamená, že se program odesílá se stránkou na klienta (do prohlížeče) a teprve tam je vykonáván (viz obr. 1). (Protikladem klientských skriptů jsou skripty serverové, které jsou vykonávány na serveru a na klienta jdou už jen výsledky.)



Obr. 1 – Klientský skript

JavaScript je interpretovaný, multiplatformní programovací jazyk se základními objektově orientovanými schopnostmi. Univerzální jádro jazyka bylo vloženo do webových prohlížečů a rozšířeno přidáním objektů reprezentující okno prohlížeče a jeho obsah. Tato *klientská* verze JavaScriptu umožňuje vložit do webových stránek proveditelný obsah. Stránky se tak stávají dynamické – mohou obsahovat nejrůznější programy, které komunikují s uživatelem, řídí prohlížeč, či dynamicky vytváří obsah HTML. Při práci skriptu není třeba kontaktovat server, veškerou práci skriptu zajišťuje sám prohlížeč.

Jádro jazyka syntakticky připomíná C++ a Javu. Avšak syntaxí podobnost končí. JavaScript je jazyk bez typové kontroly, což znamená, že proměnné nemusí mít specifikovaný typ. A navíc JavaScript je čistě interpretovaný jazyk, na rozdíl například od kompilovaných C a C++ a na rozdíl od Javy, která je před interpretací kompilována do bajtového kódu.

Co se týče bezpečnosti v JavaScriptu. JavaScript na straně klienta neumožňuje čtení a zapisování souborů, z důvodů které jsou zřejmé. Nebyl by pak problém jednoduchým programkem naprosto znehodnotit obsah pevného disku. Rovněž nepodporuje práci se sítí, až na jednu důležitou výjimku: může donutit webovský prohlížeč k načtení libovolného URL. [2]

1.5 CSS – Kaskádové styly

V počátcích webové technologie se objevila celá řada tagů jazyka HTML. Různé prohlížeče podporovali své vlastní specifické tagy a žádný z nich neuměl všechny. Tyto specifické tagy se obvykle uváděly místo obecně podporovaných tagů. Tím nejvýznamějším problémem, který vznikl bylo to, že se v každém prohlížeči stránka zobrazovala jinak.

Proto konsorcium World Wide Web Consortium(W3C) vyvinulo standart Cascading Style Sheets(CSS), který má tyto problémy eliminovat a vede k jednoduchému modelu, který odděluje styl stránky od jeho obsahu.

Definice stylů pomocí této specifikace může být vložena do stránky, nebo může být použita pro více stránek a definuje tak vzhled(styl) všech tagů ve stránce, nebo všech stránek na serveru. Takhle můžeme jednoduše, rychle a efektivně měnit vzhled stránky. Pomocí CSS je možné měnit cokoliv od velikosti, stylu a barvy textu po mezery mezi znaky a řádky, okraje a mezery kolem prvků, přesné umístění na stránce.

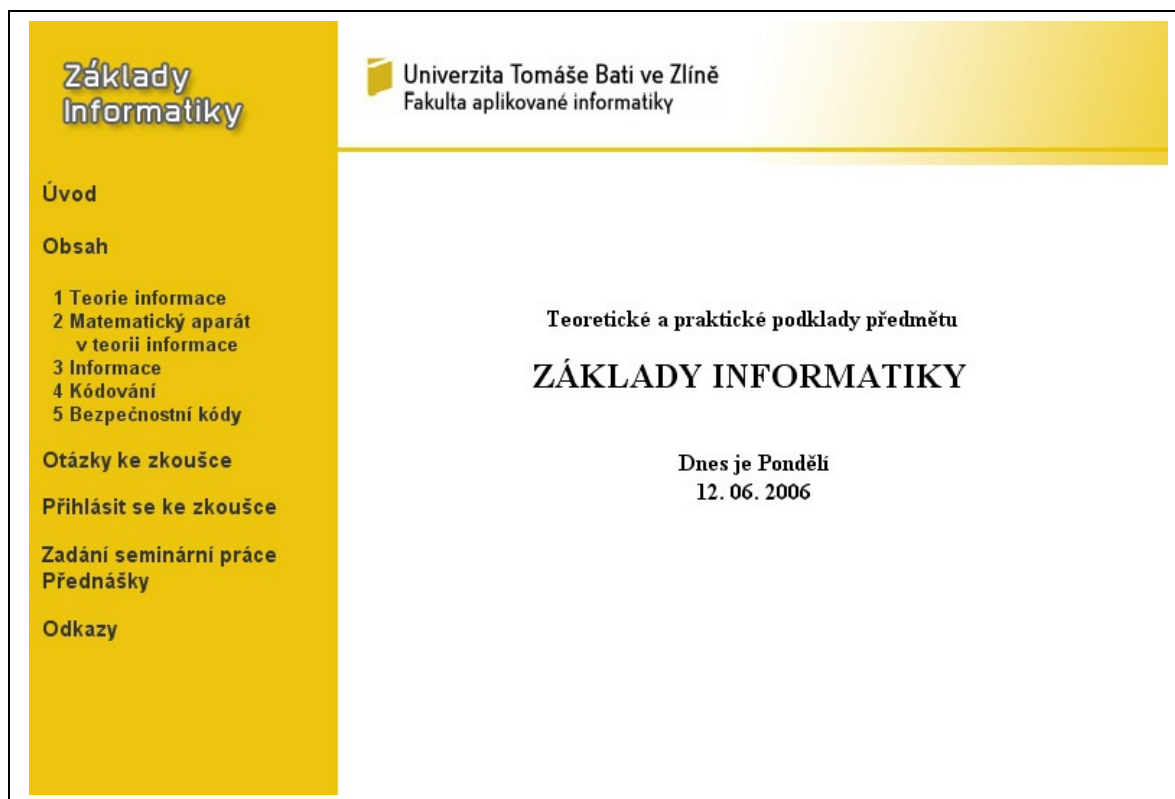
Stejně jako HTML a XHTML atd. se CSS rozděluje na verze př. HTML 4.1, XHTML 1.1, tak CSS se také rozděluje na verze CSS1, CSS2, CSS3, doposud nejpodporovanější verzí nejrozšířenějších prohlížečů je CSS2. [3]

2 PRAKTICKÁ ČÁST

2.1 Grafický vzhled

První co uvidíte po otevření stránek je kombinace žluté a bílé barvy. Žlutou barvu jsem volil proto, protože je to barva fakulty Aplikované informatiky, pod kterou spadá předmět Základy informatiky. Viz. Obr. 2.

Stránka je rozdělena pomocí rámců na tři části. Jeden rámeček je sloupcový na levé straně, který obsahuje menu. Druhý je řádkový v horní části stránky, ten obsahuje logo univerzity. Zbývá část je hlavní rámeček, kde se zobrazují informace po kliknutí na některou položku z menu. Na úvodní stránce je nadpis předmětu, animovaná zelená kostka, která má znázorňovat častou pomůcku při uvádění příkladů, např. u pravděpodobnosti v předmětu Základy informatiky. Pod ní jsem naprogramoval pomocí jazyka PHP zobrazování aktuálního datumu a dne. Viz. tab. 1.



Obr. 2 – Úvodní stránka

```
<?
$datum = date("j. m. Y");
$dnes = getdate();
if ($dnes[wday]==0){$dneska="Neděle";}
if ($dnes[wday]==1){$dneska="Pondělí";}
if ($dnes[wday]==2){$dneska="Úterý";}
if ($dnes[wday]==3){$dneska="Středa";}
if ($dnes[wday]==4){$dneska="Čtvrtek";}
if ($dnes[wday]==5){$dneska="Pátek";}
if ($dnes[wday]==6){$dneska="Sobota";}

echo "<center><strong><font size=\"3\">";
echo "Dnes je " . $dneska. "<br />" . $datum;
echo "</font></strong></center>";
?>
```

Tab. 1 – Kód pro zobrazování datumu a dne

Mým hlavním úkolem bylo upravit a doplnit stávající webové stránky předmětu Základy informatiky o tématický celek Bezpečnostní kódy speciálně pak kapitoly Systematické a Nesystematické kódy.

V první řadě jsem upravil strukturu a vzhled menu pro lepší a přehlednější orientaci na stránkách. Menu je rozděleno na hlavní a vedlejší kategorie. Mezi hlavní patří „Úvod“, který nám po kliknutí zobrazí v hlavním okně Úvodní slovo. Další odkazy jako jsou „Přihlásit se ke zkoušce“, „Zadání seminární práce“ a „Odkazy“, jsou odkazy směřující na jiné stránky, které obsahují přímo danou věc popsanou v menu. Poté menu obsahuje odkaz „Přednášky“, kde jsou ke stáhnutí přednášky na dané témata probírané v předmětu Základy informatiky. Menu obsahuje taky odkaz „Otázky ke zkoušce“, kde se po kliknutí zobrazí aktuální otázky ke zkoušce z předmětu Základy informatiky pro studenty prezenční i kombinované formy studia. Jádrem celého systému je pak položka „Obsah“. Zde se nachází veškeré teoretické podklady nutné pro zvládnutí předmětu Základy informatiky. Materiál je rozdělen podle tématických celků do pěti kapitol. Jedná se o kapitoly Teorie informace, Matematický aparát v teorii informace, Informace, Kódování a Bezpečnostní kódy.

Mým cílem bylo studentům zprostředkovat co nejpodrobnější a nejprehlednější popis kategorie Bezpečnostní kódy do nichž spadají Systematické a Nesystematické kódy.

Kapitola Bezpečnostních kódů je i v menu dále rozvedena viz. obr. 3.

Základy Informatiky

Úvod

Obsah

1 Teorie informace

2 Matematický aparát v teorii informace

3 Informace

4 Kódování

5 Bezpečnostní kódy

5.1 Systematické kódy

5.2 Nesystematické kódy

Otázky ke zkoušce

Přihlásit se ke zkoušce

Zadání seminární práce

Přednášky

Odkazy

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

5. Bezpečnostní kódy

5.1 Systematické kódy

- ♦ Kódové slovo obsahuje vedle symbolů nesoucí informaci také symboly zabezpečující
- ♦ Nezabezpečený kód lze převést na systematický tak, že ke každému kódovému slovu přidáme zabezpečující symboly
- ♦ Má-li kódové slovo m míst, z toho k informačních míst a $m-k$ zabezpečovacích míst, hovoříme o kódu $K(m, k)$.
- ♦ Pro systematické kódy platí:
 - $L_x < L$
 - Hammingova vzdálenost kódových slov je větší než 1 $d(C_i, C_j) > 1$

Optimální kód - takový kód $K(m, k)$ pro nějž neexistuje jiný kód o stejném m a k , který by poskytoval větší pravděpodobnost zachycení chyby, tj. který by měl větší zabezpečovací schopnost.

Těsný kód (perfektní) - takový kód, který pro dané m a k využívá kódová slova vzdálená vzájemně o předpokládanou minimální Hammingovu vzdálenost

$$d(C_i, C_j) \rightarrow \min$$

těsný kód je zároveň kódem optimálním

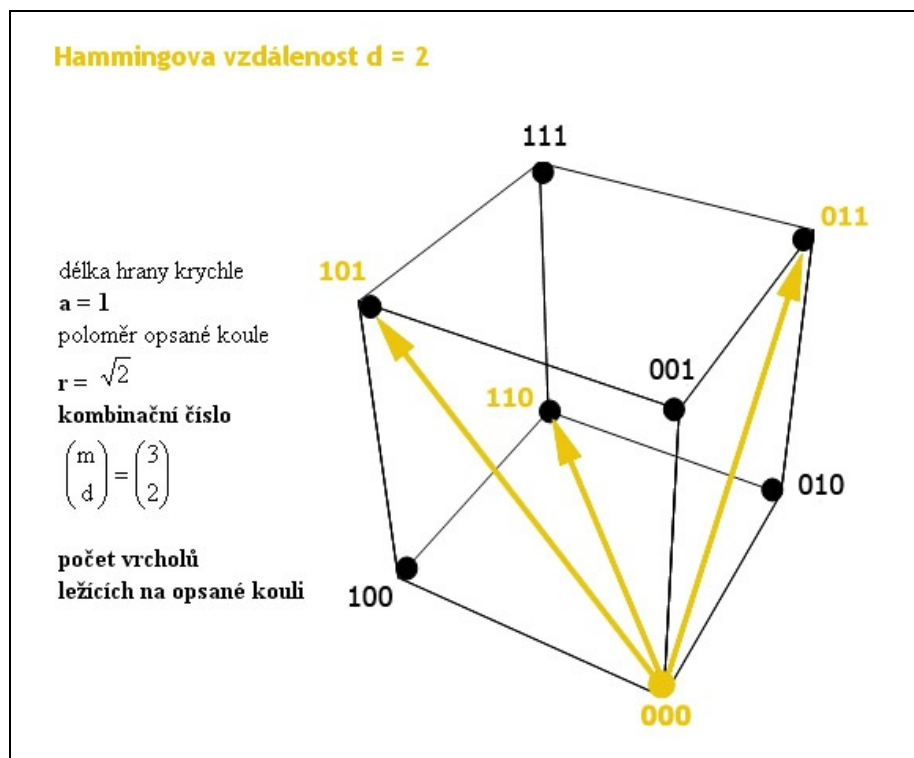
Rozdělení kódů

- ♦ **Lineární** – libovolná kombinace kódových slov je opět kódovým slovem

Obr. 3 – Struktura menu Bezpečnostní kódy

2.2 Obsah kategorie Bezpečnostní kódy

Po kliknutí na odkaz Bezpečnostní kódy se rozbalí podmenu Systematické a Nesystematické kódy a zároveň se v hlavním okně zobrazí obecný popis Bezpečnostních kódů se základními pojmy. Je zde popsán geometrický model kódu s vysvětlením Hammingovy kostky. Dále jsou zde vysvětleny pojmy jako Hammingova vzdálenost kódových slov s grafickým zobrazením Hammingovy kostky pro Hammingovu vzdálenost 1, 2 i 3. Hammingovu kostku pro jednotlivé vzdálenosti jsem pro lepší pochopení překreslil v programu Adobe Photoshop. Hammingovu kostku pro vzdálenost 2 můžete vidět na obr. č. 4



Obr. 4 – Hammingova kostka

Poté je v této části věnován prostor tématům jako jsou Detekční a korekční schopnosti kódu, Hammingova váha kódového slova a Distribuce chybových míst v kódovém slově. Jednotlivé části jsou doplněny i o řešené příklady.

2.2.1 Systematické kódy

Při kliknutí na odkaz „Systematické kódy“ se rozbálí další podmenu s možností přímého výběru z některého ze systematických kódů uvedených na stránkách. V hlavním okně se otevře celá kapitola Systematických kódů. Nerozdělil jsem to do menších podkapitol, protože jsem se domníval, že v celku je to lepší na učení i pokud v tom bude student něco hledat. Taky si to může vytisknout jako celou kapitolu Systematických kódů.

Podmenu Systematický kódů je děláno pomocí záložek.

Příklad:

`Cyklické kódy`

`<h3 id="cyklicke">Cyklické kódy</h3>`

Záložka definovaná pomocí # a za ní návěští 'cyklicke' nám ukazuje kam chceme odskočit na stránce. Samozřejmě musíme ještě nastavit parametr id, který nám udává pozici, kam chceme skočit.

Obsah hlavní stránky kategorie Systematické kódy obsahuje obecný popis Systematických kódů, jejich rozdělení a poté popis hlavních kódů jako jsou kódy Paritní, Iterační, Lineární, Hammingovy, rozšířené Hammingovy, Cyklické, Reed-Mullerovy, BCH, Reed-Solomonovy, Golayův a Nelineární kódy.

Jednotlivé kódy jsou podrobně popsány a většina je doplněna o ukázkové příklady. Podrobnějšímu vysvětlení včetně příkladů pro studenty se dostalo Hammingovým, rozšířeným Hammingovým a Cyklickým kódům.

2.2.1.1 Hammingovy kódy

U Hammingových kódů je jednak uveden obecný popis, ale také ukázkový příklad, který je pro lepší pochopení principu, řešen krok za krokem. Viz obr. 5. Příklad je rozebrán na postup, podle kterého dojdeme k výsledku.

Mým hlavním přínosem bylo vytvoření programu pomocí programovacího jazyka PHP a JavaScriptu, který řeší kontrolu správnosti přeneseného kódového slova zabezpečeného pomocí Hammingova kódu. Student si může kliknout na odkaz a program se mu otevře v hlavním okně. Viz. obr. 6.

Do pole pro kódové slovo student vloží kód u kterého chce kontrolovat jeho správnost přenosu. Poté vybere typ Hammingova kódu. Může si vybrat z nabízeného seznamu možných Hammingových kódů: Jde o kódy:

- $K(3, 1)$
- $K(7, 4)$
- $K(15, 11)$
- $K(31, 26)$
- $K(63, 57)$

Poté kliknutím na tlačítko „odeslat“ se nejprve pomocí JavaScriptu provede správnost odeslaných údajů.

Parametry pro správné odeslání a zpracování požadavku:

- pole pro kódové slovo musí být vyplněno
- kódové slovo nesmí obsahovat písmena
- kódové slovo může obsahovat jenom čísla 1 a 0
- kódové slovo musí mít délku odpovídající zvolenému typu Hammingova kódu
tzn. když student vybere typ $K(7, 4)$ tak kódové slovo musí obsahovat 7 znaků (bitů) a podobně pro jiné typy

Po splnění všech předcházejících požadavků se začne provádět samotný program.

Nejprve se načte do proměnné hodnota kódového slova a poté podle zvoleného kódu, se přiřadí do proměnných m , k a r hodnoty odpovídající danému typu kódu.

Při výběru $K(7, 4)$ je $m = 7$, $k = 4$, $r = m - k = 3$.

Poté se vytvoří a vykreslí kontrolní matice délky $m(7)$.

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Rovnice 1 – Kontrolní matice $K(7, 4)$

Ta se dále překlopí a vznikne matice H^T

$$H^T = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Rovnice 2 – Matice H^T

K určení správnosti či chyby přenosu je potřeba vypočítat syndrom přijatého kódového slova.

$s = p \cdot H^T$ kde p je kontrolované kódové slovo

Jedná se vlastně o násobení jednořádkové matice (kódové slovo) se sloupcovou maticí H^T

Výsledkem bude jednosloupcový vektor o velikosti r , v našem případě při použití kódu $K(7, 4)$ je $r = 3$.

U Hammingova kódu je ta výhoda, že výsledný syndrom přijatého slova nám přímo udává jestli je přenos bezchybný nebo v případě chyby, na kterém místě se chyba nachází.

Pokud je přenos bezchybný tak nám vyjde syndrom přijatého slova 0 resp. nulový vektor.. Pokud se v syndromu přijatého slova objeví jednička, tak nám přímo ukazuje na místo chyby a to v binární soustavě. Po převedení syndromu přijatého slova z binární do desítkové soustavy dostaneme číslo, které udává na které pozici kódového slova nastala chyba.

Hamming odvodil lineární dvojkové $K(m, k)$ kódy, které opravují jednu chybu při vzdálenosti kódových slov $r = 3$ a 4.

To znamená, že pokud při kontrole přenosu objevíme chybu, tak automaticky určíme na které pozici je chyba, a jelikož se jedná o binární přenos tak jednoduchou záměnou jedničky nebo nuly dostaneme správné kódové slovo.

Které z přijatých kódových slov 1100111, 0110101, 0011001 je chybné při použití Hammingova kódu $K(7,4)$. V případě chyby určete místo výskytu chyby.

Pro kód $K(7,4)$
dostáváme kontrolní $\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$
matici ve tvaru:

Syndrom přijatého kódového slova je pak dán:

$$\mathbf{s} = \mathbf{p} \cdot \mathbf{H}^T = [p_1 \ p_2 \ p_3 \ p_4 \ p_5 \ p_6 \ p_7] \cdot \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \mathbf{0}$$

Pro syndrom jednotlivých přijatých kódových slov pak platí:

$$\mathbf{s}(1100111) = [1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1] \cdot \mathbf{H}^T = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \text{ Chybný přenos (poslední místo)}$$

$$\mathbf{s}(0110101) = [0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1] \cdot \mathbf{H}^T = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \text{ Chybný přenos (třetí místo)}$$

$$\mathbf{s}(0011001) = [0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1] \cdot \mathbf{H}^T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ Bezchybný přenos}$$

Obr. 5 – Rozbor příkladu na Hammingův kód

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Hammingovy kódy

Zde vložte přijaté kódové slovo pro kontrolu správnosti přenosu:

Hammingův kód:

Kontrolní matice

```
0001111
0110011
1010101
```

Kódové slovo	Matice H^T	Syndrom přijatého slova
1010010	001	
	010	
	011	1
	100	0
	101	0
	110	
	111	

Přenos s chybou.
Chyba je na místě 4.

Obr. 6 – Program pro kontrolu správnosti přenosu pomocí Hammingova kódu

2.2.1.2 Rozšířený Hammingův kód

Mým dalším cílem bylo zprostředkovat studentům podrobnější informace a seznámit je blíže s pojmem Rozšířený Hammingův kód. Kapitola oět obsahuje obecný popis tohoto kódu, ale také vzorově řešený příklad, aby student pochopil jednotlivé kroky kódování pomocí rozšířeného Hammingova kódu. Příklad je znovu řešen krok za krokem. viz. obr. 7. Kapitola je opět doplněna o program vytvořený pomocí programovacího jazyka PHP a JavaScript, který řeší ověření správnosti přenosu kódového slova pomocí rozšířeného Hammingova kódu. Student si zde může ověřit správnost svých výpočtů nebo se jenom podívat jak kód funguje. Odkaz na program student najde na konci kapitoly Rozšířený Hammingův kód. Po kliknutí na tento odkaz se program spustí v hlavním okně a student s ním může pracovat. Viz. obr. 8.

Příklad:

Sestrojte kontrolní matici rozšířeného Hammingova kódu $K(8,4)$. Které z přijatých kódových slov 11001011, 00000011, 00011110 je chybné? V případě jedné chyby určete místo výskytu chyby. Pokuste se sestavit celý kód.

Pro kód $K(8,4)$
dostáváme kontrolní $H =$
matici ve tvaru:

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Syndrom přijatého kódového slova je pak dán:

$$s = p \cdot H^T = [p_1 \ p_2 \ p_3 \ p_4 \ p_5 \ p_6 \ p_7 \ p_8] \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = 0$$

Pro syndrom jednotlivých přijatých kódových slov pak platí:

$$s(11001011) = [1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1] H^T = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad \text{Chybný přenos (první místo)}$$

$$s(00000011) = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1] H^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad \text{Chybný přenos s více jak jednou chybou}$$

$$s(00011110) = [0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0] H^T = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{Bezchybný přenos}$$

Obr. 7 – Rozbor příkladu na Hammingův rozšířený kód

 Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Rozšířený Hammingův kód

Zde vložte přijaté kódové slovo pro kontrolu správnosti přenosu:

Rozšířený Hammingův kód

K(4,1)

Kontrolní matice

```
00011110
01100110
10101010
11111111
```

Kódové slovo	Matice H^T	Syndrom přijatého slova
10100011	0011	
	0101	
	0111	1
	1001	0
	1011	1
	1101	0
	1111	
	0001	

Přenos je s více jak jednou chybou.

Obr. 8 – Program pro kontrolu správnosti přenosu pomocí rozšířeného Hammingova kódu

2.2.1.2.1 Popis programu

U Hammingova rozšířeného kódu je o jeden více kontrolních bitů než u Hammingova kódu. Do textového pole student vloží kódové slovo u kterého chce provést kontrolu přenosu a poté vybere typ kódu, který chce použít. V programu jsou naprogramované tyto typy:

- K (4, 1)
- K (8, 4)
- K (16, 11)
- K (32, 26)
- K (64, 57)

Po stisknutí tlačítka „Odeslat“, se nejprve pomocí JavaScriptu provede kontrola odesílaných dat.

Parametry pro správné odeslání a zpracování požadavku:

- pole pro kódové slovo musí být vyplněno
- kódové slovo nesmí obsahovat písmena
- kódové slovo může obsahovat jenom čísla 1 a 0
- kódové slovo musí mít délku odpovídající zvolenému typu rozšířeného Hammingova kódu, tzn. když student vybere typ $K(8, 4)$ tak kódové slovo musí obsahovat 8 znaků (bitů) a podobně pro jiné typy

Po splnění všech předcházejících požadavků se začne provádět samotný program.

Nejprve se načte do proměnné hodnota kódového slova a poté, podle zvoleného kódu se přiřadí do proměnných m , k a r hodnoty odpovídající danému typu kódu.

Při výběru $K(8, 4)$ je $m = 8$, $k = 4$, $r = m - k = 3$.

Poté se vytvoří a vykreslí kontrolní matice délky $m(8)$. A to tak že se vytvoří kontrolní matice pro Hammingův kód, která má pro zvolený typ kódu délku 7 a k ní se přidá jeden nulový sloupec a poté ještě jedničkový řádek. Viz Rovnice 3.

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Rovnice 3 – Kontrolní matice Hammingova rozšířeného kódu $K(8, 4)$

Ta se dále překlopí a vznikne matice H^T viz Rovnice 4, kterou použijeme při samotném výpočtu syndromu slova.

$$H^T = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rovnice 4 – Matice HT pro rozšířený Hammingův kód $K(8, 4)$

K určení správnosti či chyby přenosu je potřeba vypočítat syndrom přijatého kódového slova. Podle níže uvedeného vzorce.

$s = p \cdot HT$ kde p je kontrolované kódové slovo

Jedná se vlastně o násobení jednořádkové matice (kódové slovo) se sloupcovou maticí H^T

Výsledkem bude sloupcový vektor o velikosti r ; v našem případě při použití kódu $K(8, 4)$ je $r = 4$.

Pokud nám syndrom slova vyjde nulový, tak jsme provedli bezchybný přenos. Ale pokud nám vyjde v syndromu přijatého slova nějaká jednička, tak už není tak jednoznačné určit místo chyby, jak tomu bylo u Hammingova kódu. U rozšířeného Hammingova kódu může totiž nastat případ přenosu s více jak jednou chybou, které nelze opravit. Pokud kód najde jenom jednu chybu, tak je automaticky opravitelná.

Určení místa chyby provádí program následovně:

- Vezme výsledný syndrom přenosu kódového slova a celý ho porovnává se sloupci kontrolní matice.
- Pokud najde stejnou hodnotu v kontrolní matici tak místo chyby nám udává číslo sloupce v kontrolní matici kde došlo ke shodě.
- Pokud ale proběhne porovnání a žádný ze sloupců kontrolní matice se nerovná výsledku syndromu přijatého slova znamená to, že přenos byl uskutečněn s více jak jednou chybou.

Zdrojový kód PHP rozšířeného Hammingova kódu viz. Příloha 1.

Zdrojový kód zabezpečení pomocí Java Scriptu pro rozšířený Hammingův kód viz. Příloha 2.

2.2.1.3 Cyklické kódy

Kapitola Cyklické kódy obsahuje jak základní popis těchto kódu, podrobně vysvětlený princip (algoritmus) pro kódování a dekódování a podle vysvětleného principu je zde vyřešen ukázkový příklad, který je řešen krok za krokem. Viz. obr. 9

Příklad:

U Hammingova kódu K(7,4) zakódujte kódové slovo **1010** pomocí cyklického kódu. Proveďte kontrolu přijetí zabezpečeného kódového slova chybně i bezchybně.

- 1) Musíme najít generující polynom $g(z)$, který je $(m - k)$ stupně a dělí polynom $z^m - 1$.
tzn.: $g(z)$ - 3 stupně a dělí $z^7 - 1$
takový je jeden: $g(z) = z^3 + z + 1$
- 2) Vytvoříme nezabezpečený polynom $u(z) = u_0z^{m-1} + u_1z^{m-2} + \dots + u_{k-1}z^{m-k}$
tzn.: $u(z) = 1 \cdot z^6 + 0 \cdot z^5 + 1 \cdot z^4 + 0 \cdot z^3$
- 3) Tento polynom dělíme $g(z)$

$$\begin{array}{r} z^6 + z^4 : z^3 + z + 1 = z^3 - 1 \\ -z^6 - z^4 - z^3 \\ \hline -z^3 \\ +z + z + 1 \\ \hline z + 1 \text{ — zbytek } r(z) \end{array}$$

Zabezpečené kódové slovo $\Rightarrow u(z) - r(z) = z^6 + z^4 + z + 1 = (1010011)$

- 4) Kontrola

$$z^6 + z^4 + z + 1 : z^3 + z + 1 = z^3 + 1$$
 zbytek **0** \Rightarrow správně

$$z^6 + z^4 + z^2 + z + 1 : z^3 + z + 1 = z^3 + 1$$
 zbytek **z^2** \Rightarrow chyba na pozici z^2

Obr. 9 – Příklad na cyklické kódy

2.2.2 Nesystematické kódy

Další kapitolou Bezpečnostních kódů jsou Nesystematické kódy, které v systému dostaly taky své místo pro vysvětlení základních pojmů a principů. Kapitola obsahuje jednak

obecný popis Nesystematických kódu, tak jejich rozdělení na Izokódy a Ekvidistantní kódy. Oba dva kódy jsou na webových stránkách popsány i s příklady.

3 UMÍSTĚNÍ WEBOVÝCH STRÁNEK

3.1 Kde stránky najdete

Tyto webové stránky budou sloužit jako podpora výuky pro předmětu Základy informatiky na Fakultě aplikované informatiky Univerzity Tomáše Bati ve Zlíně. A budou k dispozici na školním výukovém serveru Moodle na adrese „<http://www.vyuka.fai.utb.cz>“.

Na tento školní systém se dostanou pouze přihlášení a zaregistrovaní studenti této univerzity.

3.2 Co je to školní systém Moodle

Moodle je softwarový balík, který slouží pro tvorbu výukových systému a elektronických kurzů na internetu. Jedná se o neustále se rozvíjející projekt, který je navržen především pro rozsáhlejší vzdělávání.

Moodle je typ „Open Source“ což je software poskytovaný zdarma a spadající pod obecnou veřejnou licenci GNU. To v podstatě znamená, že je chráněn autorskými právy, ale zároveň při tom poskytuje uživatelům tohoto systému značnou svobodu. Moodle se může používat, kopírovat, upravovat, pokud ovšem souhlasíte s tím, že takto upravený zdroj budete poskytovat ostatním. Nesmí se ale měnit ani odstraňovat původní údaje o licencích a autorských právech.

Slovo Moodle bylo původně akronymem pro Modular Object-Oriented Dynamic Learning Enviroment (Modulární objektově orientované dynamické prostředí pro výuku).

Jak sem již výše zmiňoval, stránky budou umístěny právě na tomto výukovém systému Moodle. Viz. obr. 10.

The screenshot displays the Moodle course interface for 'Základy informatiky'. The top navigation bar includes the course name, a user login status ('Jste přihlášení jako Chramcov Bronislav'), and a 'Vypnout režim úprav' button. The main content area is divided into several sections:

- Osoby:** Účastníci, Skupiny, Upravit profil.*
- Činnosti:** Fóra, Studijní materiály, Testy, Úkoly.
- Hledat:** Search bar and 'Prohledat fóra' button.
- Správa:** Vypnout režim úprav, Nastavení..., Učitelé..., Studenti..., Záloha..., Obnovit..., Škály..., Body..., Protokoly..., Soubory..., Nápopověda..., Učitelé..., Studenti..., Záloha..., Obnovit..., Škály..., Body..., Protokoly..., Soubory..., Nápopověda..., Učitelé..., Studenti..., Záloha..., Obnovit..., Škály..., Body..., Protokoly..., Soubory..., Nápopověda...

The central 'Osnova týdnů' section shows a list of activities for the week of 6. únor - 12. únor:

- ODKAZ NA WWW stránky předmětu "ZÁKLADY INFORMATIKY" (highlighted with a red box)
- Novinky
- Podmínky pro zápočet, zkoušku
- Seminární práce - PROJEKT
- Zápočtový test - Mathematica
- Otázky ke zkoušce
- Test zkouška
- Ústní zkouška

Below the activity list, the course content for the week of 6. únor - 12. únor is shown:

CVIČENÍ:

Seznámení se s prostředím Mathematica

- prostředí, palety, styly, šablony
- nápověda
- základní pravidla
- základní matematické operace

PŘEDNÁŠKA:

Vznik a vývoj teorie informace.

- hlavní představitel vzniku a vývoje teorie informace
- pojem informace a informatika
- kde se můžeme setkat s teorií informace

On the right side, there are several utility blocks:

- Poslední novinky:** Přidat nové téma..., (Boud nebyly vloženy žádné novinky)
- Nadcházející události:** Zápočtový test - Mathematica (Test se uzavírá) - Čtvrtek, 15. červen (20:00)
- Nedávná činnost:** Činnost od, Úplná zpráva o nedávné činnosti...
- Aktualizace kurzu:** Přidáno Studijní materiál: ODKAZ NA WWW stránky předmětu "ZÁKLADY INFORMATIKY"
- Bloky:** Přidat...

Obr. 10 – Výukový systém Moodle

ZÁVĚR

Hlavním cílem této bakalářské práce bylo doplnit existující elektronickou učební pomůcku ve formě WWW stránek pro podporu výuky předmětu Základy Informatiky na Fakultě aplikované informatiky Univerzity Tomáše Bati ve Zlíně, která má sloužit jak studentům kombinované formy studia tak studentům prezenční formy studia. Dále jsem měl za úkol pozměnit stávající strukturu, kvůli lepší orientaci a přehlednosti a doplnit kapitolu Bezpečnostních kódů, do kterých patří Systematické a Nesystematické kódy.

Jelikož má tato elektronická učební pomůcka sloužit také studentům kombinované formy studia, navrhl jsem strukturu menu tak, aby veškeré potřebné informace a materiály k předmětu Základy informatiky byly dostupné na těchto stránkách. Studenti zde naleznou teoretické podklady k daným tematickým okruhům, aktuální otázky ke zkoušce, přednášky k jednotlivým kapitolám ve formě prezentace, mohou se odtud přihlásit na zkoušku ze školního systému STAG nebo mohou odevzdávat své seminární práce do školního výukového systému Moodle. Taky je v menu položka „Odkazy“, na které po kliknutí studenti naleznou další webové odkazy s informacemi k danému předmětu.

Grafický vzhled je zvolen nepříliš komplikovaný. Celá stránka je rozdělena do tří rámců a vzhled stránek je v barvách žluté a bílé což jsou barvy fakulty Aplikované informatiky Univerzity Tomáše Bati ve Zlíně.

Mým hlavním přínosem bylo vytvoření programů na lepší pochopení principu kontroly správnosti přenesení dat, které byly zabezpečeny pomocí Hammingových a rozšířených Hammingových kódů. Dalším úkolem bylo pak obohatit teoretické části o řešené ukázkové příklady z dané problematiky a názorně objasnit postupy při těchto výpočtech.

Závěrem lze konstatovat, že byly splněny všechny požadavky formulované v zadání práce. Elektronická příručka je přehledná a lze v ní nalézt vše potřebné ke studiu. Na tyto webové stránky se dostanou pouze studenti Univerzity Tomáše Bati ve Zlíně, kteří budou přihlášení do předmětu Základy informatiky ve školním výukovém systému Moodle.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

HTML – HyperText Markup Language

XHTML – eXtended HyperText Markup Language

DHTML – Dynamic HyperText Markup Language

PHP – Hypertext Preprocessor

XML - Extensible Markup Language

CSS – Cascade Style Sheet

ASP – Active Server Pages

W3C – World Wide Web Consortium

WWW – World Wide Web

SEZNAM OBRÁZKŮ

OBR. 1 – KLIENSKÝ SKRIPT.....	8
OBR. 2 – ÚVODNÍ STRÁNKA.....	10
OBR. 3 – STRUKTURA MENU BEZPEČNOSTNÍ KÓDY.....	12
OBR. 4 – HAMMINGOVA KOSTKA.....	13
OBR. 5 – ROZBOR PŘÍKLADU NA HAMMINGŮV KÓD.....	17
OBR. 6 – PROGRAM PRO KONTROLU SPRÁVNOSTI PŘENOSU POMOCÍ HAMMINGOVA KÓDU.....	18
OBR. 7 – ROZBOR PŘÍKLADU NA HAMMINGŮV ROZŠÍŘENÝ KÓD.....	19
OBR. 8 – PROGRAM PRO KONTROLU SPRÁVNOSTI PŘENOSU POMOCÍ ROZŠÍŘENÉHO HAMMINGOVA KÓDU.....	20
OBR. 9 – PŘÍKLAD NA CYKlickÉ KÓDY.....	23
OBR. 10 – VÝUKOVÝ SYSTÉM MOODLE.....	26

SEZNAM TABULEK

TAB. 1 – KÓD PRO ZOBRAZOVÁNÍ DATUMU A DNE.....	11
---	-----------

SEZNAM ROVNIC

ROVNICE 1 – KONTROLNÍ MATICE K (7, 4).....	15
ROVNICE 2 – MATICE HT.....	15
ROVNICE 3 – KONTROLNÍ MATICE HAMMINGOVA ROZŠÍŘENÉHO KÓDU K (8, 4) 21	
ROVNICE 4 – MATICE HT PRO ROZŠÍŘENÝ HAMMINGŮV KÓD K (8, 4).....	22

SEZNAM POUŽITÉ LITERATURY

- [1] Co je to PHP? Dostupné z: <http://www.php-o-php.jicinet.cz/cojephphp>
- [2] Úvod do JavaScriptu. Dostupné z: <http://www.jakpsatweb.cz/javascript/javascript-uvod.html>
- [3] Co to je CSS. Dostupné z <http://www.owebu.cz/css/vypis.php?clanek=181>
- [4] Zelinka, I.: Základy informatiky. Skriptum UTB ve Zlíně, Fakulta technologická, Zlín, 2005. ISBN 80-7318-299-8
- [5] Farana, R.: Kapitoly ze základů informatiky. Skriptum VŠB-TU Ostrava, 2003. 108s. ISBN 80-248-0265-1
- [6] Kosek Jiří: PHP – Tvorba interaktivních internetových aplikací – podrobný průvodce. Praha. Grada, 1999. ISBN 80-7169-373-1
- [7] Popelková Kateřina: Současná česká grafická úprava knih, možnosti propagace literatury, elektronická kniha. Bakalářská práce. FT UTB Zlín, 2001.

SEZNAM PŘÍLOH

Příloha 1: Zdrojový kód pro rozšířený Hammingův kód	34
Příloha 2: Zdrojový kód JavaScriptu pro kontrolu odeslání dat	38

Příloha 1: Zdrojový kód PHP pro rozšířený Hammingův kód

```

<form name="formRhk" action="<? $_SERVER['PHP_SELF']; ?>" method="post"
onSubmit="return kontrolaRhk(this)">
Zde vložte přijaté kódové slovo pro kontrolu správnosti přenosu:<br />
<input type="text" name="slovo" size="50"><br />
Rozšířený Hammingův kód<br />
<select name="rhk">
  <option value="1">K( 4,1 )</option>
  <option value="2">K( 8,4 )</option>
  <option value="3">K( 16,11 )</option>
  <option value="4">K( 32,26 )</option>
  <option value="5">K( 64,57 )</option>
</select>&nbsp;&nbsp;&nbsp;&nbsp;<input type="submit" value="odeslat"
name="formRhk">
</form>

<?php
if($_POST['formRhk'])
{
$slovo = $_POST['slovo'];
$hk = $_POST['rhk'];

if($rhk==1) // urceni parametru m, k, h
{ $m=4; $k=1; $r=$m-$k;}

elseif($rhk==2)
{ $m=8; $k=4; $r=$m-$k;}

elseif($rhk==3)
{ $m=16; $k=11; $r=$m-$k;}

elseif($rhk==4)
{ $m=32; $k=26; $r=$m-$k;}

elseif($rhk==5)
{ $m=64; $k=57; $r=$m-$k;}

$dec = 1;
for($j=0; $j<$m-1; $j++)
  ///vytvoreni kontrolni matice podle velikosti kodoveho slova
  {
  $bin = decbin($dec);
  $delkaBin = strlen($bin);

  if($delkaBin < ($r-1))
  {
  $oKolik = ($r-1) - $delkaBin;
  $nuly = "";
  for($i=0; $i<$oKolik; $i++)
  {
  $nuly = $nuly . "0";
  }
  $bin = $nuly . $bin;
  }

  $dec++;

  for($i=0; $i<$r-1; $i++)
  {
  $pole[$i][$j]= $bin{$i};
  }
  }
for($i=0; $i<$r-1; $i++)
{

```

```

        $pole[$i][$m-1]=0;
    }
    for($i=0; $i<$m; $i++)
    {
        $pole[$r-1][$i]=1;
    }

/*****/

$radek = 0;
$delkaSlova = strlen($slovo);

for($i=0; $i<=$delkaSlova; $i++)
{
    $sword[0][$i] = $slovo{$i};
} // vypis kodoveho slova do pole

/*****/
echo "\n <strong>Kontrolní matice</strong><br />\n";
for($i=0; $i<=$r; $i++)
{
    for($j=0; $j<=$m; $j++)
    {
        if($radek < $i)
        {
            echo "<br />" . $pole[$i][$j];
        }
        else
        {
            echo $pole[$i][$j];
        }
        $radek=$i;
    }
}
echo "<p>";

/*****/
/
echo " <table width='700' cellpadding='0' cellspacing='0' class='tab'
border='0'>
    <tr align='center' class='top'>
        <td>Kdov slovo</td>
        <td>Matice H<sup>T</sup></td><td>Syndrom přijatého
slova</td></tr>
    <tr align='center' valing='middle'>
        <td>$slovo</td>
        <td>";

/***** vytvoření matice H t *****/
for($i=0; $i<$r; $i++)
{
    for($j=0; $j<$m; $j++)
    {
        $maticeHt[$j][$i] = $pole[$i][$j];
    }
}

/***** vypis matice H t *****/
$radek = 0;
for($i=0; $i<$m; $i++)
{
    for($j=0; $j<$r; $j++)
    {
        if($radek < $i)
        {
            echo "<br />\n" . $maticeHt[$i][$j];

```

```

        }
        else
        {
            echo $maticeHt[$i][$j];
        }
        $radek=$i;
    }
}
echo "</td><td>";

/***** vypočet a výpis *****/
for($i=0; $i<$r; $i++)
{
    for($j=0; $j<$m; $j++)
    {
        $a[$i][$j] = $word[0][$j] * $maticeHt[$j][$i];
        /// pomocny prepocet matice $word * matice H^t
    }
}
/*****/

for($i=0; $i<$r; $i++)
{
    $x = 0;
    for($j=0; $j<$m; $j++)
    {
        if($a[$i][$j]==1)
        {
            $x++;
        }
    }
    if($x % 2 == 0)
        // vypočet syndromu preneseneho slova
    {
        $vysledek[$i][0]=0;
    }
    else
    {
        $vysledek[$i][0]=1;
    }
}

/*****/
for($i=0; $i<$r; $i++)
{
    echo "<strong>";
    echo $vysledek[$i][0];
    echo "</strong><br />";
}
echo "</td></tr></table>";

/*****/
$chyba = "";
for($j=0; $j<$m; $j++)
{
    $a = 0;
    for($i=0; $i<$r; $i++)
    {
        if( $vysledek[$i][0] == $pole[$i][$j] )
        {
            $a++;
        }
    }
    if($a==$r)
    {

```

```
        $chyba = $j + 1;
        break;
    }
}

$ok=0;
for($i=0; $i<$r; $i++)
{
    if($vysledek[$i][0]!=0)
    {
        $ok++;
    }
}

if($ok==0)
{
    echo "<h2>Bezchybný přenos.</h2>";
}

else if ($chyba=="")
{
    echo "<h2>Přenos je s více jak jednou chybou.</h2>";
}

else
{
    echo "<h2>Přenos s chybou. Chyba je na místě ".$chyba.".";
}
}
?>
```

Příloha 2: Zdrojový kód JavaScriptu pro kontrolu odeslání dat

```
function kontrolaRhk(form)
{
var slovo = document.formRhk.slovo.value;
var delka = slovo.length;
var kod    = document.formRhk.rhk.value;
    if (slovo == "")
    {
        alert("Kódové slovo musíte vypnit!");
        document.formRhk.slovo.focus();
        return false;
    }

    else if(isNaN(slovo))
    {
        alert("Kódové slovo nesmí obsahovat písmena!");
        document.formRhk.slovo.focus();
        return false;
    }

    for(i=0; i<delka-1; i++)
    {
        if (slovo.charAt(i) > 1 )
        {
            alert("Kódové slovo může obsahovat jenom 1 a 0!");
            document.formRhk.slovo.focus();
            return false;
        }
    }
    if (kod == 1 && delka != 4)
    {
        alert("Pro kód K (4,1) musí mít kódové slovo 4 bity!");
        document.formRhk.slovo.focus();
        return false;
    }
    else if (kod == 2 && delka != 8)
    {
        alert("Pro kód K (8,4) musí mít kódové slovo 8 bitů!");
        document.formRhk.slovo.focus();
        return false;
    }
    else if (kod == 3 && delka != 16)
    {
        alert("Pro kód K (16,11) musí mít kódové slovo 16 bitů!");
        document.formRhk.slovo.focus();
        return false;
    }
    else if (kod == 4 && delka != 32)
    {
        alert("Pro kód K (32,26) musí mít kódové slovo 32 bitů!");
        document.formRhk.slovo.focus();
        return false;
    }
    else if (kod == 5 && delka != 64)
    {
        alert("Pro kód K (64,57) musí mít kódové slovo 64 bitů!");
        document.formRhk.slovo.focus();
        return false;
    }
    else
        return true;
}
```