

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKEJ A POTRAVINÁRSKEJ TECHNOLOGIE
ÚSTAV INFORMATIZÁCIE, AUTOMATIZÁCIE A MATEMATIKY
ODDELENIE INFORMATIZÁCIE A RIADENIA PROCESOV



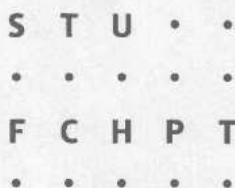
Dynamické generovanie prechodových charakteristík pomocou PHP

Vypracovala: Lýdia Pavúková

Školiteľ: Ing. Tomáš Hirmajer, PhD.

Konzultant: Ing. Ľuboš Čirka, PhD.

Bratislava, 2008



ZADANIE BAKALÁRSKEJ PRÁCE

Autorka práce: **Lýdia Pavúková (21507)**
Odbor: potravinárska technológia a biotechnológia
Študijný odbor:
Vedúci práce: Ing. Tomáš Hirmajer, PhD.
Konzultant: Ing. Ľuboš Čirka, PhD.
Miesto vypracovania: Bratislava

Názov témy: **Dynamické generovanie prechodových charakteristík pomocou PHP**

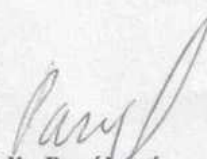
Špecifikácia zadania:

Cieľom práce je vytvoriť funkciu – program v PHP, pomocou ktorého by bolo možné jednoducho generovať prechodové charakteristiky vo formáte XHTML na základe zadaných vstupných údajov od užívateľa.

Rozsah práce: 30 strán

Riešenie zadania práce od: 07. 03. 2008

Dátum odovzdania: 23. 05. 2008


Lýdia Pavúková

riešiteľka bakalárskej práce




prof. Ing. Dr. Miroslav Fikar
vedúci pracoviska


prof. Ing. Fedor Malík, DrSc.
garant študijného programu

Pod'akovanie

Na tomto mieste by som sa chcela poďakovať
predovšetkým môjmu vedúcemu bakalárskej
práce Ing. Tomášovi Hirmajerovi, PhD za jeho
hodnotné rady a pripomienky k vypracovaniu tejto
bakalárskej práce, ktoré mi pomohli zdokonaľiť sa.

Čestné prehlásenie

Podpísaná Lýdia Pavúková týmto prehlasujem, že som bakalársku prácu na tému: Dynamické generovanie prechodových charakteristík pomocou PHP vypracovala samostatne s použitím uvedenej literatúry.

Som si vedomá zákonných dôsledkov v prípade, ak hore uvedené údaje nie sú pravdivé.

V Bratislave 23.5.2008

Abstrakt

Cieľom tejto práce bola demonštrácia využitia dynamického generovania obsahu webovou aplikáciou napísanou v jazyku PHP pre vykresľovanie grafu prechodovej charakteristiky prvého a druhého rádu na základe vstupných údajov zadaných užívateľom cez webový formulár. Práca najskôr sumarizuje použité technológie. V ďalšej časti sa zaoberá vytvorením konkrétnych súčastí webovej aplikácie: rozvrhnutím stránky, vytvorením užívateľského rozhrania s formulárom pre zadanie vstupných údajov prechodovej funkcie a nakoniec popisom fungovania samotného PHP skriptu, ktorý vykresľuje graf.

Kľúčové slová: prechodová charakteristika, prechodová funkcia, PHP

Abstract

This work demonstrates the utilization of dynamic content generation using a web application written in PHP language for rendering of the first and second grade transformation characteristics based on user data input through a web form. At first my work summarizes used technologies. Next part focuses on creation of the specific parts of the web application as follows: the page layout, creation of the user interface with a form for user input of parameters for the transformation function and finally a description of operation of the PHP script drawing the graph.

Keywords: transformation characteristics, transformation function, PHP

OBSAH

ZOZNAM SKRATIEK.....	VIII
ZOZNAM PRÍLOH	IX
ZOZNAM OBRÁZKOV.....	X
ÚVOD	11
1 POUŽITÉ TECHNOLOGIE	12
1.1. ČO JE HTML	12
1.1.1. POPIS JAZYKA.....	12
1.1. XHTML.....	14
1.2.1. ROZDIELY V ZÁPISĚ MEDZI HTML A XHTML	15
1.3. PHP.....	16
1.3.1. TYPICKÉ VLASTNOSTI JAZYKA PHP	17
1.3.2. HISTORICKÝ VÝVOJ PHP	17
1.4. CSS	18
1.5. JAVASCRIPT	19
2 PRECHODOVÉ CHARAKTERISTIKY SYSTÉMOV.....	19
2.1. PRECHODOVÁ FUNKCIA A PRECHODOVÁ CHARAKTERISTIKA	19
2.2. SYSTÉM PRVÉHO RÁDU	20
2.3. SYSTÉM DRUHÉHO RÁDU	21
3 PRAKTICKÁ ČASŤ.....	24
3.1. VYTVORENIE CSS.....	24
3.2. VYTVORENIE ÚVODNEJ STRÁNKY - INDEX.PHP.....	25
3.3. VYTVORENIE STRÁNKY FORMULÁR - VSTUP.HTML.....	25
3.4. VYTVORENIE ADRESÁRA INCLUDE ..	26
3.4.1. VYTVORENIE STRÁNKY FUNKCIA - PCH.PHP	27
3.4.1. VYTVORENIE STRÁNKY GRAF - GRAPH.PHP	27
ZÁVER.....	32
LITERATÚRA	33
PRÍLOHY	34

ZOZNAM SKRATIEK

HTML	HyperText Markup Language	Hypertextový značkový jazyk
XHTML	eXtensible Hypertext Markup Language	Rozšíriteľný hypertextový značkový jazyk
XML	eXtensible Markup Language	Rozšíriteľný značkový jazyk
PHP	Hypertext Preprocessor	Hypertextový preprocesor
CSS	Cascading Style Sheets	Tabuľky kaskádových štýlov
WWW	World Wide Web	Celosvetová sieť
HTTP	HyperText Transfer Protocol	Hypertextový protokol na prenos súborov
W3C	World Wide Web Consortium	Konzorcium pre celosvetovú sieť
SGML	Standard Generalized Markup Language	Štandardný generalizovaný značkový jazyk
CERN	Conseil Européen pour la Recherche Nucléaire	Európska organizácia pre jadrový výskum
DTD	Document Type Definition	Definícia typu dokumentu
GUI	Graphical User Interface	Grafické užívateľské rozhranie
WML	Wireless Markup Language	Bezdrôtový značkový jazyk

ZOZNAM PRÍLOH

PRÍLOHA 1. ZDROJOVÝ KÓD STRÁNKY PCH.PHP (str.34)

PRÍLOHA 2. ZDROJOVÝ KÓD STRÁNKY INDEX.PHP (str.35)

ZOZNAM OBRÁZKOV

- Obr. 1. Prechodové charakteristiky systému prvého rádu s dopravným oneskorením (str. 23)
- Obr. 2. Prechodové charakteristiky vlastného systému druhého rádu pri rôznych hodnotách koeficientu tlmenia (str. 23)
- Obr. 3. Formulár pre zadanie údajov (str. 26)
- Obr. 4. Prechodová charakteristika prvého rádu (str. 28)
- Obr. 5. Varovné hlásenie pri nesprávne zadanej hodnote zosilnenia (str. 29)
- Obr. 6. Prechodové charakteristiky prvého a druhého rádu pri rôznom čase, dopravnom oneskorení a tlmení (1) (str. 30)
- Obr. 7. Prechodové charakteristiky prvého a druhého rádu pri rôznom čase, dopravnom oneskorení a tlmení (2) (str. 31)

ÚVOD

V roku 1989 spolupracovali Tim Berners-Lee a Robert Caillau na prepojenom informačnom systéme pre CERN, fyzikálne výskumné centrum blízko Ženevy vo Švajčiarsku. V tej dobe sa pre tvorbu dokumentov používali obvykle jazyky TeX, PostScript a tiež SGML. Berners-Lee si uvedomoval, že potrebujú niečo jednoduchšieho a v roku 1990 bol teda navrhnutý jazyk HTML a protokol pre jeho prenos v počítačovej sieti – HTTP (HyperText Transfer Protocol – prenosový protokol hypertextu). Zároveň Tim Berners-Lee napísal prvý webový prehliadač, ktorý nazval WorldWideWeb. V roku 1991 CERN spustil svoj web. Nasledoval rýchly rozvoj webu.

PHP bolo pôvodne navrhnuté ako niekoľko skriptov v jazyku Perl, neskôr prepísaných do jazyka C. Autorom bol Rasmus Lerdorf v roku 1994. O rok neskôr svoje skripty zverejnil pod názvom Personal Home Page Tools. Kombináciou s ďalším jeho programom Form Interpreter vzniklo PHP/FI. Zeev Suraski a Andi Gutmans, dvaja izraelskí vývojári prepísali syntaktický analyzátor (parser) v roku 1997 na novšiu verziu, ktorá sa stala základom PHP 3. Krátko po zverejnení verzie PHP 3 bola spustená aj oficiálna stránka. V roku 1999 sa Suraski a Gutmans opäť pustili do prepísania jadra, už pod názvom Zend engine. Založili spoločnosť Zend Technologies, ktorá sa odvtedy podieľa na ďalšom vývoji PHP.

Cieľom tejto práce je vytvoriť program v PHP, pomocou ktorého by bolo možné jednoducho generovať prechodové charakteristiky na základe vstupných údajov od užívateľa.

Prvá časť práce sa zaoberá teóriou, ktorá bola nevyhnutná k úspešnému dokončeniu tejto práce, ako napr. čo je to HTML, základná syntax, druhy značiek, alebo ako má vyzeráť štruktúra dokumentu. Podobne bola rozobraná problematika PHP, kaskádových štýlov a JavaScriptu. V ďalšej časti práce je stručne charakterizovaná prechodová funkcia a prechodová charakteristika. V poslednej tretej časti je podrobne vysvetlený a znázornený zdrojový kód v XHTML, PHP a CSS.

1. POUŽITÉ TECHNOLOGIE

1.1 ČO JE HTML

HyperText Markup Language (HTML) [1], je značkový jazyk pre hypertext. Je jedným z jazykov pre vytváranie stránok v systéme WWW (World Wide Web), ktorý umožňuje publikáciu dokumentu na internete.

Jazyk je aplikáciou skôr vyvinutého rozsiahleho univerzálneho značkového jazyka SGML (Standard Generalized Markup Language). Vývoj HTML bol ovplyvnený vývojom webových prehliadačov, ktoré spätne ovplyvňovali definíciu jazyka.

1.1.1 POPIS JAZYKA

- **Koncepcia**

Jazyk HTML je charakterizovaný množinou značiek a ich atribútov definovaných pre daný riadok. Medzi značkami sa uzatvárajú časti textu dokumentu a tým sa určuje význam (sémantika) obsahovaného textu. Názvy jednotlivých značiek sa uzatvárajú medzi uhlové zátvorky (< a >). Časť dokumentu tvorená otváracou značkou, nejakým obsahom a zodpovedajúcou ukončovacou značkou tvorí tzv. element (prvok) dokumentu. Napríklad je otváracia značka pre zvýraznenie textu a STU je element obsahujúci zvýraznený text. Súčasťou obsahu elementu môžu byť ďalšie vnorené elementy. Atribúty sú doplnujúce informácie, ktoré upresňujú vlastnosti elementu.

Značky (zvané tagy) sú obvykle párové (v XHTML sú párové všetky), pričom koncová značka je zhodná so značkou počiatočnou, len má pred názvom znak „/“.

Niektoré značky sú nepárové – nemajú žiadny obsah a nepoužívajú koncovú značku. Príklad pre vykreslenie vodorovnej čiary:

```
<hr>
```

Tagy môžu obsahovať atribúty, ktoré popisujú ich vlastnosti alebo nie sú inou informáciou. Príkladom môže byť odkaz (tag a), ktorého atribút href hovorí, kam sa užívateľ po kliknutí naň dostane (v tomto príklade na stránku <http://stuba.sk>):

```
<a href="http://stuba.sk">text odkazu</a>
```

Od SGML zdedil jazyk HTML aj iné, menej známe konštrukcie pre tvorenie elementov. Jedná sa o tzv. skrátené HTML zápisy:

- <element/Obsah/ je totiž ako <element>Obsah</element>

- `<element>Obsah</>` je to isté ako `<element>Obsah</element>`
- `<el<el2>Obsah` je to isté ako `<el><el2>Obsah`
- `<ul><li>Adam<>Božena</ul>` je to isté ako
`<ul><li>Adam</li><li>Božena</li></ul>`

Všetky tieto zápisy sú síce v súlade s platným štandardom a sú úplne ekvivalentné, ale žiadny zo známych prehliadačov skrátené verzie nepodporuje, takže sa neodporúča ich používať.

Pre každú verziu existuje definícia pravidiel DTD (Document Type Definition). Od verzie 4.01 musí byť odkaz na deklaráciu DTD v dokumente uvedený pomocou kľúčového slova DOCTYPE. DTD definuje pre určitú verziu elementy a atribúty, ktoré je možné používať.

Dokument môže mimo značkovania obsahovať ďalšie prvky:

- Direktívy – začínajú znakmi `<!`, sú určené pre spracovateľa dokumentu (prehliadač).
- Komentáre – pomocné texty pre programátora, nie sú súčasťou obsahu dokumentu a nezobrazujú sa (prehliadač ich ignoruje).
- Kód skriptovacích jazykov.
- Definícia udalostí a kód pre ich obsluhu.

• Štruktúra dokumentu

Dokument v jazyku HTML má predpísanú štruktúru:

- Deklarácia DTD – je uvedená direktívou `<!DOCTYPE`.
- Koreňový element – element `html` (značky `<html>` a `</html>`) reprezentuje celý dokument. Koreňový element je povinný, ale otváracia a ukončovacia značka samotná povinná nie je (pokiaľ tieto značky nebudú v tele dokumentu uvedené, prehliadač si ich sám doplní podľa kontextu).
- Hlavička elementu – obsahuje metadata, ktoré sa vzťahujú k celému dokumentu. Definujú napr. názov dokumentu, jazyk, kódovanie, kľúčové slová, popis, použitý štýl zobrazenia. Hlavička je uzatvorená medzi značkami `<head>` a `</head>`. Element `head` je opäť povinný, ale jeho otváracia a koncová značka povinná nie je, prehliadač ju sám doplní podľa kontextu.
- Telo dokumentu – obsahuje vlastný text dokumentu. Vymedzuje sa značkami `<body>` a `</body>`. Element `body` je povinný, ale jeho otváracia a koncová značka povinná nie je, prehliadač ju sám doplní podľa kontextu

- **Druhy značiek**

Značky možno z hľadiska významu rozdeliť na tri základné skupiny:

- **Štrukturálne značky**

- rozvrhujú štruktúru dokumentu, príkladom sú odstavce (`<p>`), nadpisy (`<h1>`, `<h2>`). Dodávajú dokumentu formu.

- **Popisné (sémantické) značky**

- popisujú povahu obsahu elementu, príkladom je nadpis (`<title>`) alebo adresa (`<address>`).

- **Štylistické značky**

- určujú vzhľad elementu pri zobrazení, typickým príkladom je značka pre tučné písmo (`<b>`). Od tohto druhu značiek sa postupne upúšťa, trendom je používanie kaskádových štýlov, ktoré vzhľad popisujú oddelene od obsahu dokumentu.

- **Textové editory**

Editory HTML sú programy pre ľahkú tvorbu webových stránok. Editorom HTML môže byť vo svojej podstate akýkoľvek program pracujúci s textom. V praxi sa však používajú omnoho sofistikovanejšie programy. Bežný editor HTML zvláda farebnú syntax (farebne rozlišuje jednotlivé časti kódu ako napríklad HTML značky či atribúty a prostý text), dokáže napovedať značky, pozná šikovné tabulátory alebo zvláda overovať dokument podľa predpísaného DTD. Medzi takéto najrozšírenejšie editory HTML u nás určite patrí PSPad [2].

1.2. XHTML

Jazyk podobný HTML ale s prísnejšou syntaxou je XHTML (eXtensible Hypertext Markup Language) [3]. Je to rozšíriteľný hypertextový značkový jazyk. Zatiaľ čo HTML je aplikáciou SGML, veľmi pružného značkového jazyka, XHTML je aplikáciou XML, obmedzenej podmnožiny SGML.

Jazyk XHTML bol navrhnutý s ohľadom na kompatibilitu všetkých možných editorov a prehliadačov. Vďaka novým mechanizmom profilácie dokumentov budú servery a proxy servery schopné rýchlejšie a lepšie spracovať obsah dokumentu.

XHTML je oproti jazyku HTML jednoduchší. Sú odstránené niektoré atribúty prvkov, ktoré upravovali vzhľad rozmery a pozíciu týchto prvkov. Tieto vlastnosti boli nahradené pravidlami jazyka CSS.

1.2.1 ROZDIELY V ZÁPÍSE MEDZI HTML A XHTML

Keďže XHTML je implementáciou XML, oproti HTML má niektoré dodatočné požiadavky, ktoré zjednodušujú jeho parsovanie a uľahčuje zapamätanie pravidiel človekom. Značky musia byť písané malými písmenami. Párové značky musia byť ukončené. (Značka "option" sa v XHTML považuje za párovú.)

HTML:

```
<p>Toto je odsek.
```

```
<tr><td>Prvá bunka<td>Druhá bunka
```

```
<option>Možnosť 1
```

XHTML:

```
<p>Toto je odstavec.</p>
```

```
<tr><td>Prvá bunka</td><td>Druhá bunka</td></tr>
```

```
<option>Možnosť 1</option>
```

Nepárové značky musia obsahovať koncovú lomku.

HTML:

```
<br>
```

```
<input type="text" name="text1">
```

XHTML:

```
<br />
```

```
<input type="text" name="text1" />
```

Vlastnosti musia byť uvedené v úvodzovkách alebo apostrofoch.

HTML:

```
<td colspan=3>
```

XHTML:

```
<td colspan="3">
```

Vlastnosti musia mať vždy uvedenú hodnotu.

HTML:

```
<textarea readonly>Nejaký text</textarea>
```

XHTML:

```
<textarea readonly="readonly">Nejaký text</textarea>
```

Prvok "img" musí mať určenú vlastnosť "alt".

HTML:

```

```

XHTML:

```

```

Párové tagy sa nesmú krížiť.

HTML:

```
<p><a href="text.html">Text v odstavci.</p></a>
```

XHTML:

```
<p><a href="text.html">Text v odstavci.</a></p>
```

1.3. PHP

PHP (Hypertext Preprocessor) [4] je hypertextový preprocesor, pôvodne Personal Home Page. Je to skriptovací programovací jazyk, určený predovšetkým pre programovanie dynamických internetových stránok. Najčastejšie sa začleňuje priamo do štruktúry jazyka HTML, XHTML či WML, čo možno využiť pri tvorbe webových aplikácií. PHP možno použiť aj k tvorbe konzolových a desktopových aplikácií.

PHP skripty sú prevádzkované na strane servera, k užívateľovi je prenášaný až výsledok ich činnosti. Syntax jazyka je inšpirovaná niekoľkými programovacími jazykmi (Perl, C, Pascal a Java). PHP je nezávislý na platforme, skripty fungujú bez väčších úprav na mnohých rôznych operačných systémoch. Podporuje mnoho knižníc pre rôzne účely - napr. spracovanie textu, grafiky, prácu so súborami, prístup k väčšine databázových systémov, podporu celého radu internetových protokolov.

PHP sa stalo veľmi obľúbeným predovšetkým vďaka jednoduchosti použitia a preto, že kombinuje vlastnosti viacerých programovacích jazykov a necháva tak vývojárom čiastočnú slobodu v syntaxi. V kombinácii s operačným systémom Linux, databázovým systémom (obvykle MySQL[5] alebo PostgreSQL) a webovým serverom Apache [6] je často využívaný k tvorbe webových aplikácií. Pre túto kombináciu sa vžila skratka LAMP – teda spojenie Linux, Apache, MySQL a PHP alebo Perl.

1.3.1 TYPICKÉ VLASTNOSTI JAZYKA PHP

- Jazyk PHP je dynamicky typový, tzn. že dátový typ premennej sa určí v okamihu priradenia hodnoty.
- Vďaka tomu má PHP dva typy porovnania, '=' rovnaký ako v C, a '===' ktorý platí len keď sú obidva výrazy rovnakého typu.
- Polia sú heterogénne, môžu teda obsahovať akékoľvek údaje, rovnako tak ako ich indexy.
- Reťazce možno uzatvárať ako do úvodzoviek (obsah je parsovaný), tak do apostrofov (obsah nie je parsovaný).

1.3.2 HISTORICKÝ VÝVOJ PHP

Tvorcom PHP bol Rasmus Lerdorf. Vytvoril ho v roku 1994 pre svoju osobnú potrebu prepísaním niekoľkých skriptov z Perlu do jazyka C. Sada skriptov bola vydaná ešte v tom istom roku pod názvom Personal Home Page Tools, skrátene PHP. Od tej doby sa technológia PHP stala jednou z najpoužívanějších technológií pre tvorbu dynamicky generovaných WWW stránok.

V polovici roku sa systém PHP spojil s programom Form Interpreter toho istého autora. Tak vzniklo PHP/FI 2.0. Zeev Suraski a Andi Gutmans v roku 1997 prepísali parser a sformovali tak základ PHP3. Súčasne bol názov zmenený na dnešnú podobu PHP Hypertext Preprocessor. PHP verzia 3 vyšla v roku 1998, bola rýchlejšia, obsahovala viac funkcií.

1.4 CSS

CSS (Cascading Style Sheets) [7] je skratka pre slovenský názov tabuľky kaskádových štýlov, ktoré predstavujú jazyk popisujúci spôsob zobrazenia stránok napísaných napr. v jazykoch HTML, XHTML, XML. Autorom prvotného návrhu bol Håkon Wium Lie.

Hlavným cieľom CSS je umožniť oddelenie vzhľadu dokumentu od jeho štruktúry a obsahu.

Výhody používania CSS v porovnaní s HTML v praxi

- rozsiahlejšie možnosti formátovania na rozdiel od samotného HTML
- konzistentný štýl
- oddelenie štruktúry a štýlu
- dynamická práca so štýlmi
- formátovanie XML dokumentov
- väčšia kompatibilita alternatívnych webových prehliadačov
- kratšia doba načítania stránky

Najväčšou výhodou CSS oproti starému formátovaniu v HTML je to, že kód a obsah webu je uložený v súbore .html a akýkoľvek dizajn a formátovanie sa načíta z jedného súboru s príponou .css, ktorý býva väčšinou spoločný pre celý web. Teda ak máme v pláne zmenu dizajnu webu, stačí zmeniť jeden súbor .css a zmena sa aplikuje na celý web. Ak sa súbor CSS uloží do medzipamäte prehliadača a nezmení sa, načíta sa iba raz a zobrazenie stránok sa veľmi urýchli.

Špecifikácia CSS umožňuje dokonca napísať štýly pre hlasový syntetizátor alebo pre hmatovú čítačku Braillovho písma.

Hlavnou nevýhodou CSS je miestami nedostatočná podpora vo väčšine prehliadačov. Niekedy nie je ľahké napísať kód tak, aby sa výsledok vo všetkých prehliadačoch zobrazil rovnako.

Štýlový predpis sa skladá z postupnosti pravidiel. Každé pravidlo určuje vzhľad niektorého elementu dokumentu, prípadne skupiny elementov. Pravidlo začína tzv. selektorom, ktorý špecifikuje skupinu elementov. Za selektorom nasleduje deklarácia určujúca vzhľad vybranej skupiny elementov. Celý zoznam je uzavretý v zložených zátvorkách a jednotlivé deklarácie sú oddelené bodkočiarkou.

1.5 JAVASCRIPT

JavaScript [8] je skriptovací jazyk, ktorého autorom je Brendan Eich z vtedajšej firmy Netscape.

Používa sa ako interpretovaný programovací jazyk pre WWW stránky. Výhodou je možnosť vnorenia priamo do HTML kódu stránky. Obyčajne ním bývajú ovládané rôzne interaktívne prvky GUI (tlačidlá, textové políčka) alebo tvorené animácie a efekty obrázkov.

Na základe jeho názvu je pravdepodobné, že syntax Javascriptu sa podobá Jave, v skutočnosti sa jeho tvorca najviac inšpiroval jazykom Self. Slovo Java je súčasťou jeho názvu iba z marketingových dôvodov, hlavne pre vtedajšiu popularitu jazyka Java. JavaScript bol štandardizovaný asociáciou ECMA (European Computer Manufacturers Association) a neskôr ISO (International Organization for Standardisation). Štandardizovaná verzia JavaScriptu je ECMAScript a z nej boli odvodené ďalšie implementácie napr. ActionScript.

Program sa najprv vyvíjal pod názvom Mocha, neskôr LiveScript. Pre verziu firmy Microsoft je používaný názov JScript, ktorý ale nedodržuje špecifikáciu ECMAScript, preto vznikajú rozdiely v implementácii, ktoré majú za následok čiastočnú neprenosnosť kódu medzi internetovými prehliadačmi používajúcimi JScript (Microsoft Internet Explorer) a JavaScript (Mozilla Firefox).

Program v JavaScripte sa obvykle spúšťa až po stiahnutí WWW stránky z internetu (tzv. na strane klienta) z čoho plynú isté bezpečnostné obmedzenia - napr. nemôže pracovať so súbormi, aby tým neohrozil súkromie užívateľa.

JavaScript je možné použiť aj na strane servera, podobne ako napr. program PHP. Prvá implementácia JavaScriptu na strane servera bol LiveWire firmy Netscape. Dnes už však existuje niekoľko ďalších možností.

2. PRECHODOVÉ CHARAKTERISTIKY SYSTÉMOV

2.1 PRECHODOVÁ FUNKCIA A PRECHODOVÁ CHARAKTERISTIKA

Prechodová funkcia [9] je odpoveď dynamického systému na jednotkový skok pri nulových počiatočných podmienkach. Jej výpočet sa vykonáva spätnou Laplaceovou transformáciou obrazu výstupnej veličiny zo systému pri obraze vstupu $1/s$.

Jednotková skoková funkcia:

$$u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (2.1)$$

Laplaceov obraz jednotkovej skokovej funkcie:

$$L\{1\} = \frac{1}{s} \quad (2.2)$$

Prechodová charakteristika [10] je grafické znázornenie prechodovej funkcie.

Obrazový prenos $G(s)$ sa definuje ako pomer Laplaceovej transformácie výstupného signálu $Y(s)$ k Laplaceovej transformácii odpovedajúceho vstupného signálu $U(s)$ pri nulových počiatočných podmienkach,

$$G(s) = \frac{Y(s)}{U(s)} = \frac{Y(s)}{s} \quad (2.3)$$

Obrazový prenos [11] a nedá stanoviť priamo experimentálne, získava sa z diferenciálnej rovnice alebo z ostatných úplných dynamických charakteristík príslušným prepočtom.

2.2 SYSTÉM PRVÉHO RÁDU

Typický tvar diferenciálnej rovnice prvého rádu:

$$Ty'(t) + y(t) = Ku(t) \quad (2.4)$$

kde

$u(t)$ je vstupná veličina

$y(t)$ je výstupná veličina

K je zosilnenie

T je časová konštanta

Zosilnenie systému K je dané hodnotou prechodovej charakteristiky v nekonečne $K = y(\infty)$.

Obrazový prenos prvého rádu opísaného diferenciálnou rovnicou:

$$G(s) = \frac{K}{Ts + 1} \quad (2.5)$$

Prechodová funkcia k systému:

$$Y(s) = \frac{K}{s} - \frac{K}{s + \frac{1}{T}} \quad (2.6)$$

Laplaceov obraz prechodovej funkcie (Obr. 1):

$$y(t) = K \left(1 - e^{-\frac{t-T_D}{T}} \right) \quad (2.7)$$

kde

T_D je dopravné oneskorenie

2.3 SYSTÉM DRUHÉHO RÁDU

Typický tvar diferenciálnej rovnice druhého rádu:

$$T^2 y''(t) + 2\xi T y'(t) + y(t) = Ku(t) \quad (2.8)$$

Obrazový prenos druhého rádu opísaného diferenciálnou rovnicou je vo všeobecnosti:

$$G(s) = \frac{K}{T^2 s^2 + 2\xi Ts + 1} \quad (2.9)$$

kde

ξ je koeficient tlmenia, čo je bezrozmerná veličina

Charakter prechodovej charakteristiky závisí od koreňov charakteristickej rovnice (Obr. 2).

- Obrazový prenos druhého rádu pre $\xi > 1$, korene charakteristickej rovnice sú reálne a rôzne:

$$G(s) = \frac{K}{T^2 s^2 + 2\xi Ts + 1} = \frac{K}{(T_1 s + 1)(T_2 s + 1)} \quad (2.10)$$

Prechodová funkcia k systému:

$$Y(s) = K \frac{1}{s} - \frac{KT_1}{(T_1 - T_2)} \frac{1}{\left(s + \frac{1}{T_1}\right)} + \frac{KT_2}{(T_1 - T_2)} \frac{1}{\left(s + \frac{1}{T_2}\right)} \quad (2.11)$$

Laplaceov obraz prechodovej funkcie:

$$y(t) = K \left(1 - \frac{T_1}{(T_1 - T_2)} e^{-\frac{t-T_D}{T_1}} + \frac{T_2}{(T_1 - T_2)} (t - T_D) e^{-\frac{t-T_D}{T_2}} \right) \quad (2.12)$$

- Obrazový prenos druhého rádu pre $\xi = 1$, korene charakteristickej rovnice sú reálne a rovnaké:

$$G(s) = \frac{K}{T^2 s^2 + 2\xi Ts + 1} = \frac{K}{(Ts + 1)^2} \quad (2.13)$$

Prechodová funkcia k systému:

$$Y(s) = K \frac{1}{s} - K \frac{1}{s + \frac{1}{T}} - \frac{K}{T} \frac{1}{\left(s + \frac{1}{T}\right)^2} \quad (2.14)$$

Laplaceov obraz prechodovej funkcie:

$$y(t) = K \left(1 - e^{-\frac{t-T_D}{T}} - \frac{t-T_D}{T} e^{-\frac{t-T_D}{T}} \right) \quad (2.15)$$

- Obrazový prenos druhého rádu pre $0 < \xi < 1$, korene charakteristickej rovnice sú dva komplexne združené s nenulovou reálnou časťou:

$$G(s) = \frac{K}{T^2 s^2 + 2\xi Ts + 1} \quad (2.16)$$

Prechodová funkcia:

$$Y(s) = K \frac{1}{s} - K \frac{s + \xi \frac{1}{T}}{\left(s + \frac{\xi}{T}\right)^2 + \left(\frac{\sqrt{1-\xi^2}}{T}\right)^2} - K \frac{\xi}{\sqrt{1-\xi^2}} \frac{\frac{\sqrt{1-\xi^2}}{T}}{\left(s + \frac{\xi}{T}\right)^2 + \left(\frac{\sqrt{1-\xi^2}}{T}\right)^2} \quad (2.17)$$

Laplaceov obraz prechodovej funkcie:

$$y(t) = K \left[1 - e^{-\frac{\xi(t-T_D)}{T}} \left(\cos \left(\frac{\sqrt{1-\xi^2}}{T} (t-T_D) \right) + \frac{\xi}{\sqrt{1-\xi^2}} \sin \left(\frac{\sqrt{1-\xi^2}}{T} (t-T_D) \right) \right) \right] \quad (2.18)$$

- Obrazový prenos druhého rádu pre $\xi = 0$, korene charakteristickej rovnice sú dva komplexne združené s nulovou reálnou časťou:

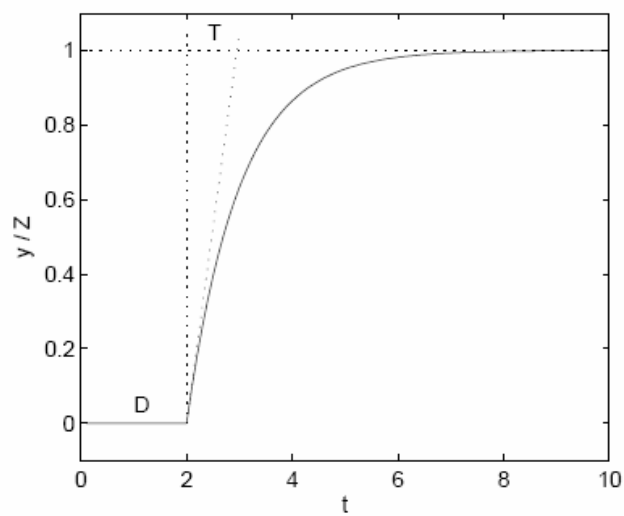
$$G(s) = \frac{K}{T^2 s^2 + 2\xi Ts + 1} = \frac{K}{T^2 s^2 + 1} = \frac{\frac{K}{T^2}}{s^2 + \frac{1}{T^2}} \quad (2.19)$$

Prechodová funkcia:

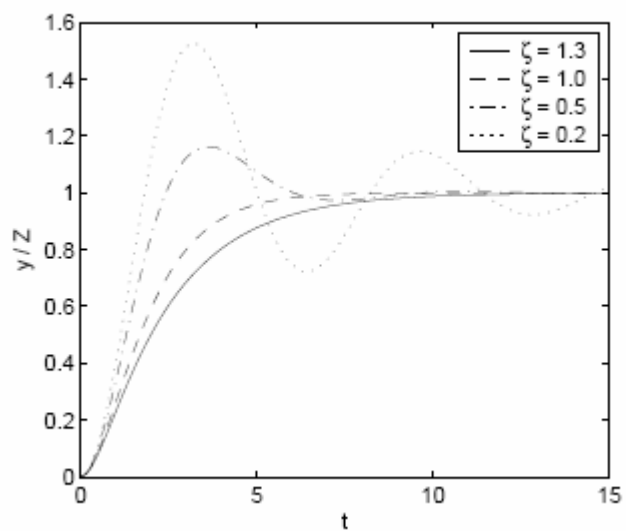
$$Y(s) = K \left(\frac{1}{s} - \frac{s}{s^2 + \frac{1}{T^2}} \right) \quad (2.20)$$

Laplaceov obraz prechodovej funkcie:

$$y(t) = K \left[1 - \cos \left(\frac{1}{T} t \right) \right] \quad (2.21)$$



Obr. 1. Prechodová charakteristika systému prvého rádu s dopravným oneskorením.



Obr. 2. Prechodové charakteristiky vlastného systému druhého rádu pri rôznych hodnotách koeficientu tlmenia.

3. PRAKTICKÁ ČASŤ

3.1 VYTVORENIE CSS

Pomocou programu CSS bol naprogramovaný vzhľad stránok ako napr. či má byť stránka orámovaná.

```
.frejm
{
border-style: 0
}
```

Ďalej bola definovaná farba pozadia.

```
body
{
background-color: #5E5A80
}
```

Farba, duktus a font písma.

```
.pismo
{
color: #E9CFEC;
font-weight:bold;
font-family:arial
}
```

Potom bolo navrhnuté orámovanie odosielacieho tlačítka a jeho farba, pozadie a farba pozadia, šírka a výška tlačítka a zarovnanie na výšku.

```
.tlacitko
{
border-style:double;
border-color:#E9CFEC;
background:none;
background-color:#5E5A80;
width:150px;
height:25px;
vertical-align:middle;
}
```


Nakoniec bol definovaný vzhľad políčka pre zapisovanie vstupných údajov ako rám a farba rámu, pozadie a farba pozadia, šírka a výška políčka a farba, duktus a font písma.

```
.policko
{
    border-style:none;
    border-color:#5E5A80;
    background:none;
    background-color:#E9CFEC;
    width:150px;
    height:20px;
    color:#5E5A80;
    font-weight:bold;
    font-family:arial
}
```

3.2. VYTvorenie úvodnej stránky - INDEX.PHP

Najprv bola vytvorená úvodná stránka index.php (príloha 1). Prvé, čo musí byť v dokumente definované je DOCTYPE (DOCument TYPE definition), ktorý nám bude určovať o aký typ XHTML dokumentu ide a akú normu bude využívať. XHTML kód každej webstránky je umiestnený medzi tagmi <html> a </html>. XHTML stránka je zložená z hlavičky medzi tagmi <head> a </head> a tela medzi tagmi <body> a </body>.

V hlavičke je uvedený názov hlavnej stránky vo vnútri tagu <title> a </title>, ktorý sa zobrazí v prehliadači. V hlavičke je tiež zadefinovaný riadok, podľa ktorého sa aktivuje slovenská diakritika a odkaz na súbor s kaskádovými štýlmi.

3.3. VYTvorenie stránky formulár - VSTUP.HTML

Stránka obsahuje hlavný formulár vstup.html (príloha 2). V tomto formulári (Obr. 3.) si užívateľ stránky môže vybrať, či chce vykresliť graf prechodovej charakteristiky prvého alebo druhého rádu. Na to slúži vstupný tag `input type="radio"`, ktorý má v tomto prípade funkciu prepínača. V ďalších riadkoch sú parametre rovnice prechodovej charakteristiky pre prvý rád, ktorých hodnotu by mal užívateľ zvoliť na základe povolených hodnôt uvedených za konštantou v zátvorke. V opačnom prípade sa graf nebude dať zobrazíť. Ak si užívateľ vyberie prechodovú charakteristiku druhého rádu, pod konštantou času sa

zobrazí ďalšia konštanta „Tlmenie“. Nakoniec si užívateľ môže vybrať opäť pomocou prepínača, či chce mať graf vykreslený samostatne, alebo niekoľko v tom istom obrázku. Maximálny počet zobrazených grafov je štyri, pričom každý nový graf bude znamenať vymazanie najstaršieho zadaného. Napravo od formulára sa zobrazuje aktuálny vzorec pre prvý rád a druhý rád v závislosti od tlmenia. Pod formulárom je umiestnený obrázok grafu.

Rád ☒ 1. ☐ 2.

K (povolené hodnoty: kladné čísla)

T (povolené hodnoty: kladné čísla)

Čas (povolené hodnoty: kladné čísla 1-9999)

Dopravné oneskorenie (povolené hodnoty: 0-999)

☒ Do nového obrázka
 ☐ Do toho istého obrázka

VYKRESLIŤ

Laplaceov obraz prechodovej funkcie prvého rádu

$$y(t) = K \left(1 - e^{-\frac{t-T_0}{T}} \right)$$

y(t)- výstupná veličina

K- zosilnenie

T- časová konštanta

t- čas

Td- dopravné oneskorenie

Obr. 3. Formulár pre zadanie údajov.

3.4. VYTVORENIE ADRESÁRA INCLUDE

Adresár include obsahuje stránky pch.php (FUNKCIA) a graph.php (GRAF) . Je to externý súbor, ktorý sa zahrňuje pomocou funkcie `include ( )`, ktorej primárnym cieľom je čerpanie informácií z externých skriptov.

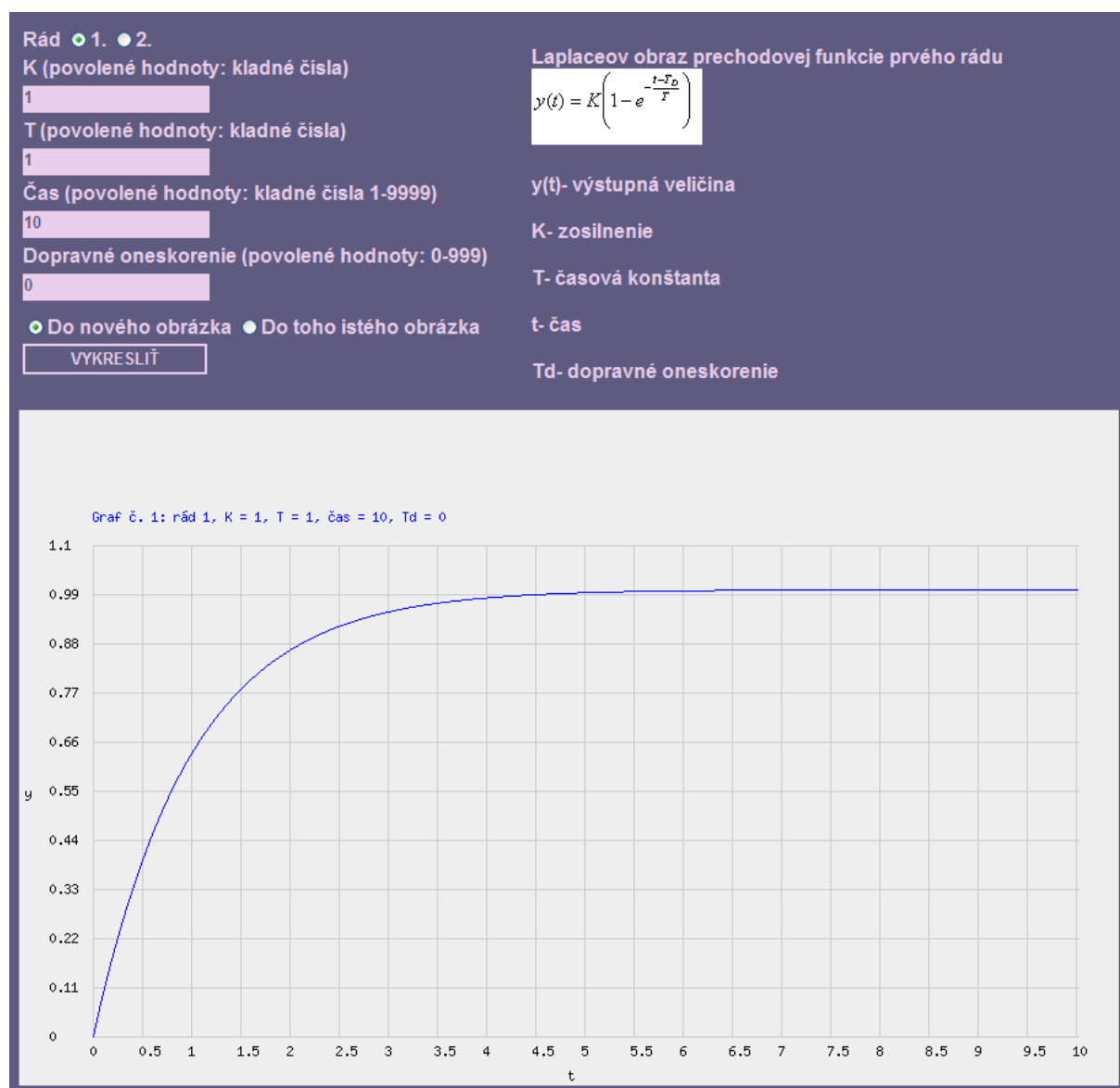
3.4.1. VYTVORENIE STRÁNKY FUNKCIA - PCH.PHP

Stránka pch.php obsahuje funkciu s dvoma vetvami kódu. Jedna sa vykoná vtedy, keď sa jedná o prvý rád a druhá sa vykoná pre prípad druhého rádu. Ak sa jedná o druhý rád, kód sa ďalej rozvetvuje na základe tlmenia. Tento kód má tri samostatné vetvy, jedna sa vykoná v prípade, že hodnota tlmenia sa nachádza v intervale $<0;1$), druhá pre prípad hodnoty tlmenia rovnej 1, a tretia vetva slúži pre výpočet hodnoty funkcie s tlmením vyšším ako 1. Zápornú hodnotu tlmenia sme neuvažovali. Vstupom tejto funkcie sú všetky parametre pre výpočet hodnoty prechodovej funkcie:

\$x	predstavuje čas zobrazený na x-ovej osi grafu.
\$K	predstavuje zosilnenie
\$D	je dopravné oneskorenie
\$T	je časová konštanta
\$t _{tlmenie}	predstavuje koeficient tlmenia
\$rad	určuje, či sa jedná o prechodovú charakteristiku prvého alebo druhého rádu

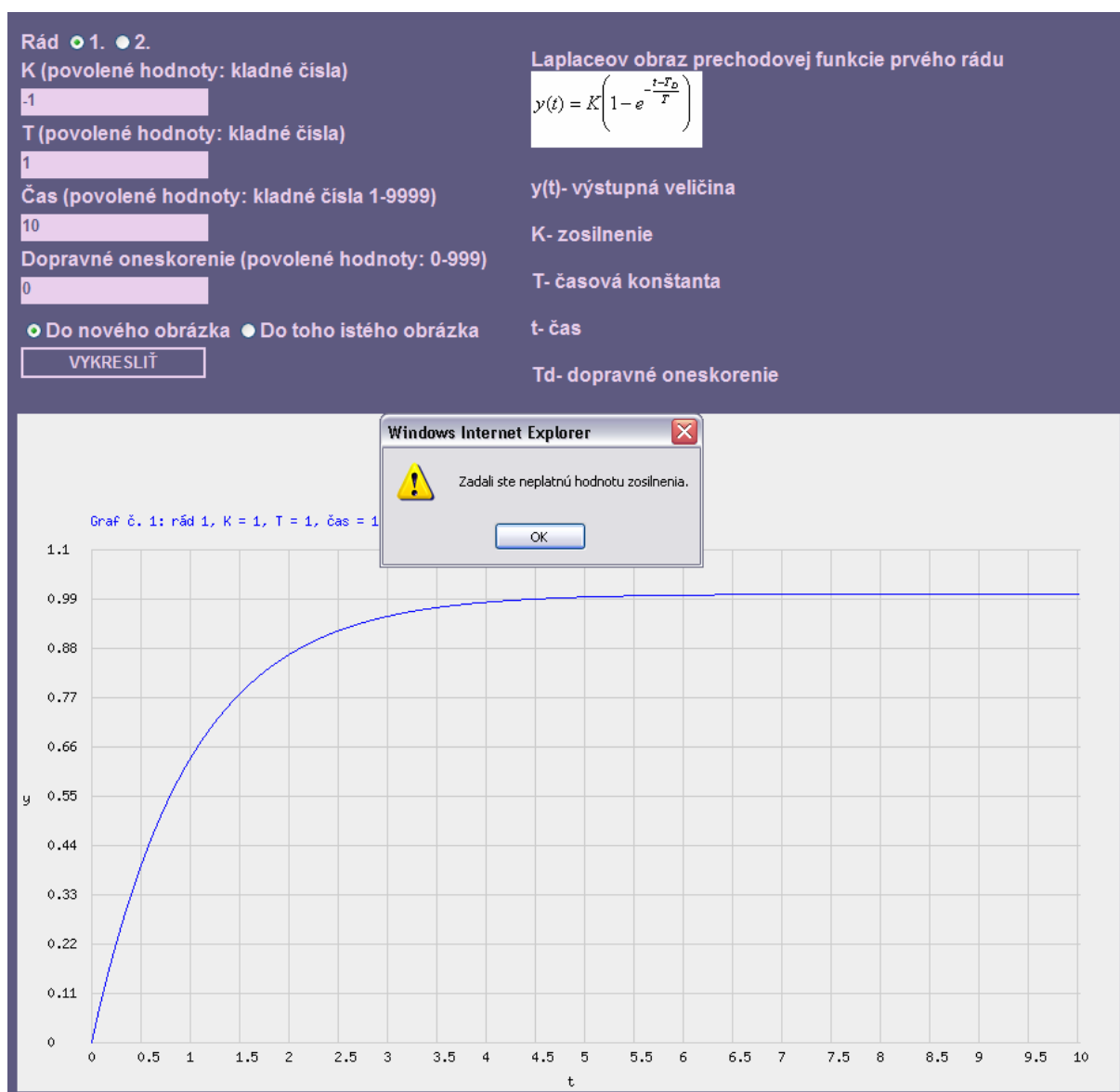
3.4.2. VYTVORENIE STRÁNKY GRAF - GRAPH.PHP

Súbor graph.php obsahuje kód, ktorý zo vstupných údajov zadaných užívateľom cez formulár vstup.html vykreslí obrázok s grafom (Obr. 4.).



Obr. 4. Prechodová charakteristika prvého rádu.

Spracovanie vstupných údajov sa vykoná ešte na strane klienta a po nesprávnom vyplnení formulára sa zobrazí varovné okno (Obr. 5.), ktoré upozorňuje na chybné alebo žiadne vyplnenie niektorého z blokov. Pre správnosť sa zadané vstupné hodnoty testujú aj logickou podmienkou. Pri nesplnení podmienky skript nepokračuje ďalej a ukončí svoju činnosť vypísaním chybového hlásenia.



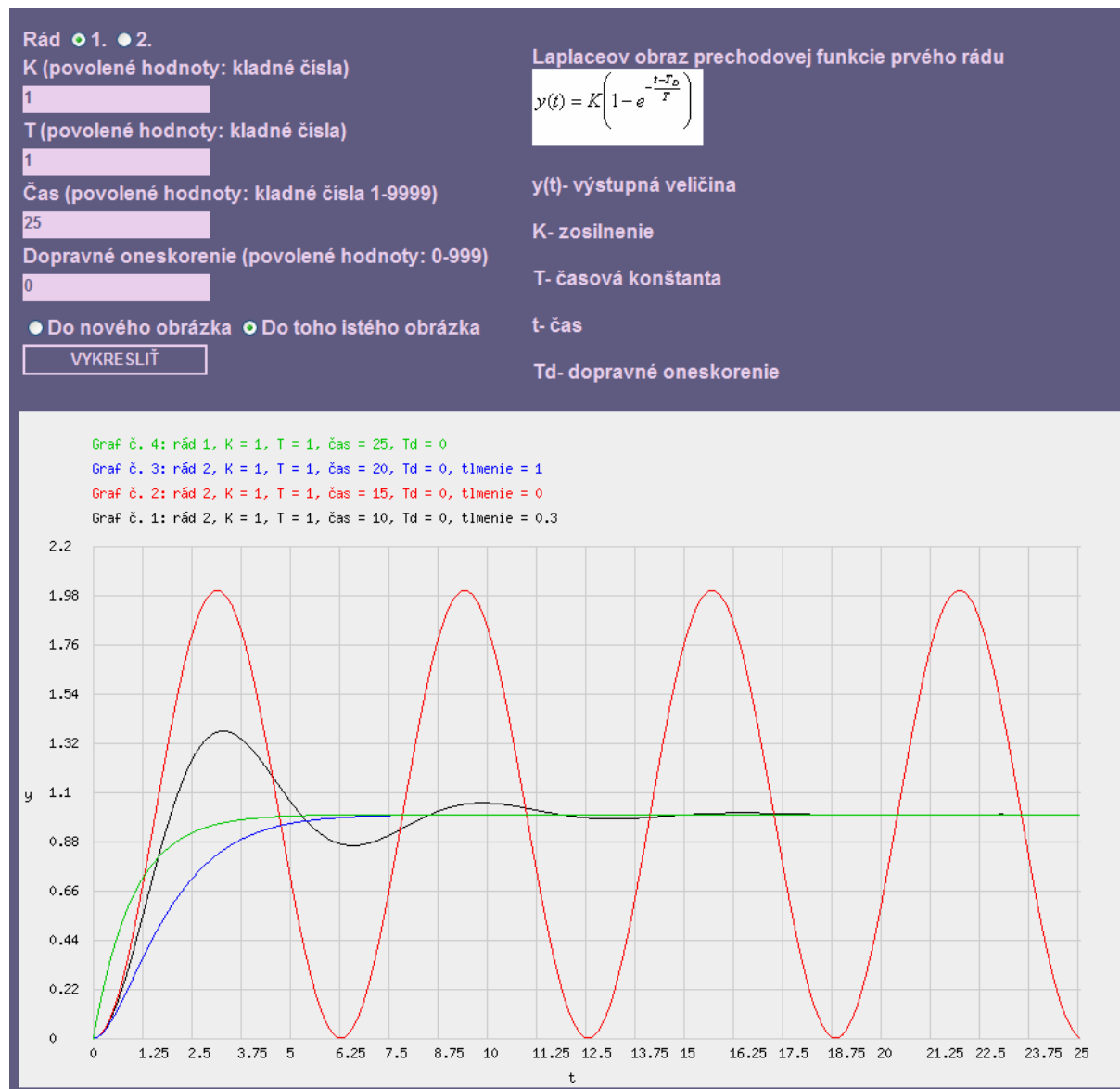
Obr. 5. Varovné hlásenie pri nesprávne zadanej hodnote zesilnenia.

Jednou zo základných premenných, ktoré boli zadefinované a ďalej použité na vykreslenie obrázka bola $\$jemnost$. Tento parameter určuje „kotrbatosť“ grafu t. z. odstup jednotlivých čiarok, z ktorých je graf zložený. Pomerne kvalitné vykreslenie sa dosiahne nastavením tohto parametra na hodnotu z intervalu 0,001-0,01.

Ak boli vstupné údaje zadane správne, spustí sa hlavná časť programu. Kód sa rozdelí na dve vetvy na základe toho, či sa má vykresliť graf do nového obrázka, alebo sa má vykresliť viacero grafov do toho istého obrázka.

Aby sa mohlo vykresľovať viacero grafov do toho istého obrázka (Obr. 6.), parametre práve vykresleného grafu sa uložia do poľa, ktoré je následne uložené do deklarovaného poľa polí parametrov. Ak užívateľ zadal pre rôzne grafy odlišnú hodnotu času, do úvahy sa bude

brať len ten najvyšší zadaný čas z poľa grafov. Z takto získanej hodnoty času sa určí mierka osi x. Rozsah hodnôt na osi y sa určí tak, že sa zistí maximum všetkých vykreslených grafov.



Obr. 6. Prechodové charakteristiky prvého a druhého rádu pri rôznom čase, dopravnom oneskorení a tlmení (1).

Keďže do jedného obrázka sa môžu vykreslili maximálne štyri grafy, prekročením tohto počtu sa prvý graf zmaže, zvyšné tri sa posunú o jedno miesto v poli grafov, a posledný sa nahradí aktuálnym (Obr. 7.).

Rád ☒ 1. ☐ 2.

K (povolené hodnoty: kladné čísla)

1

T (povolené hodnoty: kladné čísla)

1

Čas (povolené hodnoty: kladné čísla 1-9999)

25

Dopravné oneskorenie (povolené hodnoty: 0-999)

2

☒ Do nového obrázka ☐ Do toho istého obrázka

VYKRESLIŤ

Laplaceov obraz prechodovej funkcie prvého rádu

$$Y(s) = K \left(1 - e^{-\frac{t-T_d}{T}} \right)$$

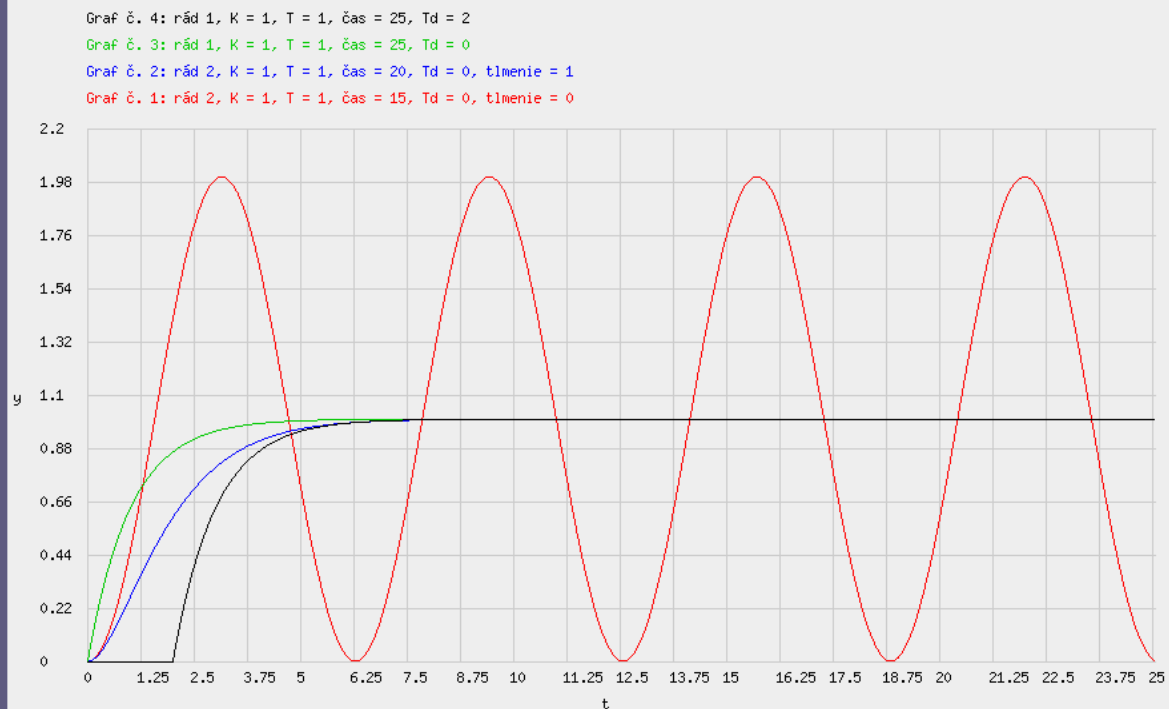
y(t)- výstupná veličina

K- zosilnenie

T- časová konštanta

t- čas

Td- dopravné oneskorenie



Obr. 7. Prechodové charakteristiky prvého a druhého rádu pri rôznom čase, dopravnom oneskorení a tlmení (2).

ZÁVER

Cieľom tejto bakalárskej práce bolo vytvoriť program v jazyku PHP, pomocou ktorého by bolo možné jednoducho generovať prechodové charakteristiky v tvare obrázku použitím XHTML na základe zadaných vstupných údajov od užívateľa.

Najprv bol vypracovaný dizajnový návrh, na základe ktorého bolo spracované užívateľské rozhranie s formulárom pre zadávanie vstupných údajov a následne bol naprogramovaný samotný PHP skript, ktorého výstupom bol obrázok s grafom vo formáte PNG, ktorý sa zobrazí v spodnej časti stránky.

Hotová stránka môže byť použitá na demonštráciu závislosti priebehu prechodovej charakteristiky od zadaných parametrov.

Program je navrhnutý tak, že jednoduchým prepísaním funkcie `returnvalue` v súbore `graph.php` je možné generovať grafy prakticky hocijakej závislosti. Výsledky tejto práce je možné využiť v nejakom väčšom projekte, ktorý by skúmal napríklad simuláciu zložitých dejov vo výrobných procesoch.

LITERATÚRA

- [1] Hypertext markup language. Dostupné na <http://sk.wikipedia.org/wiki/Html>, dňa 15. februára 2008
- [2] <http://www.pspad.com>, PSPad - freeware unicode developer editor
- [3] XHTML. Dostupné na <http://sk.wikipedia.org/wiki/XHTML>, dňa 17. februára 2008
- [4] PHP. Dostupné na <http://cs.wikipedia.org/wiki/Php>, dňa 15. februára 2008
- [5] <http://www.mysql.com>, The World's Most Popular Open Source Database, MySQL AB, © 1995-2006 MySQL AB. All rights reserved.
- [6] <http://www.apache.org>, - The Apache Software Foundation
- [7] Kaskádové štýly. Dostupné na http://sk.wikipedia.org/wiki/Kask%C3%A1dov%C3%A9_%C5%A1t%C3%BDly, dňa 21. februára 2008
- [8] JavaScript. Dostupné na <http://sk.wikipedia.org/wiki/JavaScript>, dňa 17. februára 2008
- [9] Bakošová, M.: Prednášky zo základov automatizácie. Dostupné na <http://www.kirp.chnf.stuba.sk/moodle/mod/resource/view.php?inpopup=true&id=5981> dňa 17. februára 2008
- [10] Fikar, M. - Bakošová, M. - Čirka, Ľ.: Základy automatizácie. Laboratórne cvičenia zo základov automatizácie. Bratislava: STU, 2003, str. 49. ISBN 80-227-1831-9.
- [11] Palenčár, R. et al: Meracie prostriedky. Dostupné na <http://www.kam.sjf.stuba.sk/KATEDRA/publikacie/leonardo/ucebnica/04s.pdf>, dňa 17. februára 2008

PRÍLOHA 1. ZDROJOVÝ KÓD STRÁNKY PCH.PHP

```
<?php
/**
 * rata vlastnu hodnotu prechodovej funkcie v case x, s parametrami:
 *
 * @param float $x
 * @param float $D
 * @param float $K
 * @param float $T
 * @param float $tlmenie
 * @param float $rad
 * @return float
 */
function returnValue($x,$D,$K,$T,$tlmenie, $rad)
{
    if($x < $D)
        return 0;
    else
        $x = $x - $D;

    if ($rad == 1) {
        $value = $K*(1-exp((-1)*($x)/$T));
    } else if (0<=$tlmenie && 1>$tlmenie) {
        $value = $K*(1-exp((-1)*$tlmenie*($x)/$T)*(cos(sqrt(1-($tlmenie*$tlmenie)))*($x)/$T))-
        ($tlmenie/sqrt(1-($tlmenie*$tlmenie)))*exp((-1)*$tlmenie*($x)/$T)*(sin(sqrt(1-
        ($tlmenie*$tlmenie)))*($x)/$T));
    } else if ($tlmenie == 1) {
        $value = $K*(1-exp((-1)*($x)/$T)-((($x)/$T)*exp((-1)*($x)/$T)));
    } else if ($tlmenie>1) {
        $T1 = (-1)*$tlmenie/$T+sqrt($tlmenie*$tlmenie-1)/$T;
        $T1 = (-1)/$T1;
        $T2 = (-1)*$tlmenie/$T-sqrt($tlmenie*$tlmenie-1)/$T;
        $T2 = (-1)/$T2;
        $value = $K*(1-($T1/($T1-$T2))*exp((-1)*($x)/$T1)+($T2/($T1-$T2))*exp((-
        1)*($x)/$T2));
    }
    return $value;
}
?>
```

PRÍLOHA 2. ZDROJOVÝ KÓD STRÁNKY INDEX.PHP

```
<?php
session_start();

head();

require_once 'include/graph.php';

main();

tail();

function main()
{
    global $_POST;

    $chyba = false;

    $error_text = '';

    // default hodnoty

    $where = 0;

    $rad = 1;

    $out = array('K'=>'', 'T'=>'', 'cas'=>'', 'tlmenie'=>'0', 'D'=>'0');

    $graphs = new Graphs();

    if(isset($_POST['odoslane'])) {
        foreach($_POST as $key=>$val)

            $_POST[$key] = floatval(ereg_replace(",", ".", $val));

        if(!$_POST['K'] || !$_POST['T'] || !$_POST['cas'] || $_POST['cas']>9999 ||
(isset($_POST['tlmenie']) && $_POST['tlmenie'] < 0) || $_POST['D'] < 0 ) {

            $error_text = "<div style='color:red; font-family:arial; font-
weight:bold'>Chyba: nesprávne vstupné údaje. Rešpektujte povolené hodnoty.</div>";

            $chyba = true;

        }

        if(!$chyba) {

            if($_POST['where']==0)

                $graphs->clear();

            $graphs->addGraph($_POST['D'], $_POST['K'], $_POST['T'], isset($_POST['tlmenie']) ? $_POST['tlmenie'] :
0, $_POST['rad'], $_POST['cas']);

        }

        $where = $_POST['where'];

        $rad = $_POST['rad'];

        foreach($_POST as $key=>$val)

            $out[$key] = $val;

    } else {
```

```

        $graphs->clear();
    }
    include 'vstup.html';
}
function display_error()
{
}
function head()
{
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="sk">
<head>
    <title>Graf prechodovej charakteristiky</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link rel="StyleSheet" href="style/style.css" TYPE="text/css"/>
</head>
<body>
<?php
}
function tail()
{
    echo '</body></html>';
}

```