### [UNIVERSITATEA „ALEXANDRU IOAN CUZA” IAȘI](https://www.google.ro/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwj_kcHhr5bNAhXI7RQKHbA_Ai0QFggbMAA&url=http%3A%2F%2Fwww.uaic.ro%2F&usg=AFQjCNHADpfeOWX8GgXs9G-e862ljqfj2Q)

### FACULTATEA DE INFORMATICĂ

**CONVERTIREA FIȘIERELOR HTML**

**ÎN XHTML/XML ȘI EDITAREA LOR**

**Dina-Priscilia Gavriliuc**

**Sesiunea**: Iunie**-**Iulie, 2016

Coordonator științific

**Lect. Univ. Dr. Alex Moruz**

DECLARAȚIE PRIVIND ORIGINALITATE ȘI RESPECTAREA DREPTURILOR DE AUTOR

Prin prezenta declar că Lucrarea de licență cu titltul ”Modelarea fișierelor HTML și returnarea lor în XML” este scrisă de mine și nu a mai fost prezentată niciodată la altă facultate sau instituție de învățământ superior din țară sau străinătate. De asemenea, declar că toate sursele utilizate, inclusiv cele preluate de pe Internet, sunt indicate în lucrare, cu respectarea regulilor de evitare a plagiatului:

-toate fragmentele de text reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și dețin referința precisă a sursei;

-reformularea în cuvinte proprii a textelor scrise de către alți autori deține referința precisă;

-codul sursă, imaginile etc. preluate din proiecte *open-source* sau alte surse sunt utilizate cu respectarea drepturilor de autor și dețin referințe precise;

-rezumarea ideilor altor autori precizează referința precisă la textul original.

Iași, 22.06.2016

Absolvent *Dina-Priscilia Gavriliuc*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul ”Modelarea fișierelor HTML și returnarea lor în XML”, codul sursă al programlelor și celelalte conținuturi (grafice, multimedia, date de test etc) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea ”Alexandru Ioan Cuza” Iași să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, 22.06.2016

Absolvent *Dina-Priscilia Gavriliuc*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CUPRINS**

INTRODUCERE ...................................................................................... 5

1. PREZENTAREA PIEȚEI......................................................................... 6
   1. Serviciul de validare W3C.................................................................. 6
   2. Serviciul de validare Dr. Watson........................................................ 9
   3. HTML Validator Firefox ADD-on..................................................... 9
   4. Validatorul WDG HTML................................................................... 11
   5. HTML Power Tools............................................................................ 12
   6. CSE HTML Validator........................................................................ 12
   7. HTML Toolbox................................................................................... 13
2. D ESCRIEREA SOLUȚIEI...................................................................... 15
   1. JTidy API.......................................................................................... 15
   2. HTMLCleaner................................................................................... 19
   3. Soluția............................................................................................... 21
   4. Fișierul de configurare...................................................................... 24
   5. Editarea tag-urilor............................................................................. 26
3. TESTE ȘI EVALUARE............................................................................ 29
   1. Evaluarea fișierului de configurare................................................... 32

CONCLUZII ȘI DEZVOLTĂRI............................................................. 33

BIBLIOGRAFIE....................................................................................... 37

ANEXE...................................................................................................... 38

**Introducere**

Lucrarea propune o analiză a fișierelor html nevalide cât și editarea tag-urilor. Validarea documentelor Web este un pas important care poate contribui în mod dramatic la îmbunătățirea și asigurarea calității acestora, și poate salva timp și bani. Printre cei mai semnificativi factori amintim: tag-uri deschise și închise corect, eliminarea tag-urilor nefolositoare, înlocuirea anumitor tag-uri pentru eliminarea ambiguității, etc.

Lucrarea este structurată în trei capitole. Primul capitol prezintă starea actuală a pieței. Cel de-al doilea capitol dezvăluie ideea centrală a tezei și anume nevoia de creare a unui serviciu de corectare și editare a fișierelor html, profitabil dar și deplin satisfăcător pentru consumatori prin calitatea înaltă, descrierea soluției și a tehnologiilor folosite. Capitolul trei ilustrează testarea și evaluarea serviciului iar ultima parte a lucrării prezintă concluziile și dezvoltările.

Cele mai multe documente Web sunt scrise folosind limbaje de marcaje, cum ar fi HTML sau XHTML. Aceste limbaje sunt definite de specificații tehnice, care includ de obicei o gramatică formală care poate fi citită automat. Actul de verificare a unui document împotriva acestor constrângeri este numit de validare , iar asta este o parte a ceea ce face acest serviciu.

Tehnologia avansează tot mai mult cât și nevoia de ea. Întrucât tot mai mulți oameni se implică în dezvoltarea tehnologiilor web, este mare nevoie de acest serviciu de corectare și editare a fișierelor html. Am ales să abordez această temă deoarece în urma experienței mele de încercare a depistării și corectării unor erori sintactice, cât și ștergerea sau înlocuirea anumitor tag-uri a fost dificilă, consider că va fi de mare ajutor programatorilor amatori sau profesioniști.

Majoritatea dezvoltatorilor de aplicații Web, când sunt întrebați dacă își validează paginile Web, răspunsul lor este unul simplu, și anume ”Site-ul meu arată bine și lucrează corect”. Verificarea existenței unei pagini de acest gen, în mai multe browsere contemporane poate fi o asigurare rezonabilă că pagina va ”lucra” astăzi , dar nu garantează că va funcționa mâine . În practică, diferite browsere pot face și afișa aceeași pagină foarte diferit. Acest lucru este în mod deliberat, și nu implică nici un fel de eroare de browser. Un termen folosit uneori pentru acest lucru este WYSINWOG –”What You See Is Not What Others Get”- Ce vezi nu este ceea ce alții primesc ( cu excepția cazului de coincidență ). Este într-adevăr, una dintre principalele forțe ale web-ului, care (de exemplu) un utilizator cu deficiență de vedere poate selecta dimensiuni foarte mari de vizualizare a textului, fără a fi nevoie să se scrie o pagină Web specială pentru așa ceva. Această aplicație ajută la evitarea problemelor de felul acesta.

Facultatea de Informatică ne-a învățat să ne dorim să fim primii, să credem în forțele noastre, să fim creatori de valoare în sfera oricărui domeniu de activitate și mai ales să ne cunoaștem mai bine pentru a ști ceea ce să ne dorim.

Mulțumiri admirabililor profesori care ne-au îndrumat pe căi inteligente și mai ales părinților care ne-au susținut cu încredere la fiecare pas din drumul vieții.

**Cap1. PREZENTAREA PIEȚEI**

Nevoia tehnologiei este tot mai mare deoarece oamenii se confruntă cu diferite probleme, de aceea și dezvoltarea ei este atât de rapidă.

În momentul de față sunt multe servicii care se ocupă de această mare problemă, și anume validarea fișierelor html. Eu voi dezvolta doar primele 7 servicii (după o statistică făcută de [*http://webdesign.about.com*](http://webdesign.about.com/od/htmlvalidators/tp/best_html_validators.htm)) disponibile la momentul de față.

* 1. **Serviciul de validare W3C**

Acest serviciu este cel mai căutat deoarece este la disponibilitatea tuturor care vor să verifice documentul împotriva DTD sau a schemei enumerate în document. De asemenea, este un sistem HTML de validare în conformitate cu standardul internațional ISO / IEC 15445 - HyperText Markup Language , și standardul internațional ISO 8879 - Standard Generalized Markup Language ( SGML ).

Validatorul W3C Markup prevede Perl / CGI / SGML / XML -bazate pe validarea DTD a unei varietăți mari de tipuri de documente. SGML și DTD-urile sunt tehnologii mai vechi care nu sunt folosite pe o scară largă, astfel încât pentru verificarea documentelor HTML, folosind tehnologii moderne, se poate utiliza în schimb W3C HTML Checker.

W3C pune la dispoziție următoarele:

*- Rularea propriului exemplu de W3C Markup Validator*

*- Folosirea API-ul W3C Markup Validator*

*- Controlul asupra comportamentului W3C Markup Validator*

**Rularea propriului exemplu de W3C Markup Validator**

Pentru a oferi propriul serviciu de verificare a documentelor HTML, se execută o instanță a W3C HTML Checker. Instrucțiuni cu privire la modul de a face acest lucru sunt disponibile mai jos :

Nu-HTML-Checker: acesta încapsulează back-end-ul pentru checker.html5.validator.nu, și validator.w3.org/nun. Acesta este lansat sub forma a două pachete: -vnu.jar  este o versiune portabilă de sine stătătoare pentru documente batch-checking din linia de comandă cât și din alte aplicații sau scripturi, sau pentru implementarea corectorului ca un serviciu autonom.

Pentru a verifica unul sau mai multe documente de la linia de comandă:

|  |
| --- |
| java -jar ~/vnu.jar FILE.html FILE2.html FILE3.HTML FILE4.html... |

Pentru a verifica toate documentele dintr-un directoriu:

|  |
| --- |
| java -jar ~/vnu.jar some-directory-name/ |

Pentru a verifica cu un input standard:

|  |
| --- |
| java -jar ~/vnu.jar  exemplu: echo '<!doctype html><title>...' | java -jar ~/vnu.jar |

Pentru a verifica un document Web:

|  |
| --- |
| *java -jar ~/vnu.jar URL*  exemplu: java -jar ~/vnu.jar <http://example.com/foo> |

-vnu.war este pentru implementarea serviciului de verificare printr-un container de servlet cum ar fi Tomcat

**Folosirea API-ului W3C Markup Validator**

Pentru verificarea programatică a documentelor moderne HTML, se poate folosi API-ul W3C Markup Validator. Cea mai simplă metodă de a folosi acest API este de a apela [*https://validator.w3.org/check*](https://validator.w3.org/check) cu un șir de interogare care să includă parametrul ”uri” si URL-ul documentului de verificat, și cu parametrul ”output=json” atașat.

De exemplu: [*https://validator.w3.org/check?uri=https://example.com/&output=json*](https://validator.w3.org/check?uri=https://example.com/&output=json). W3C Markup Validator va returna un fișier JSON.

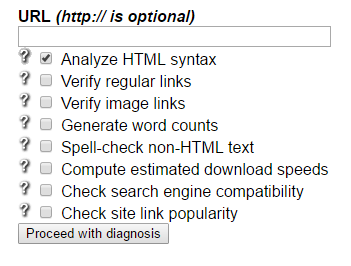
Se pot specifica alte opțiuni pentru acest API. Numele acestor parametri sunt exemplificați mai jos, în secțiunea ”Controlul asupra comportamentului W3C Markup Validator”.

**Controlul asupra comportamentului W3C Markup Validator**

Această secțiune descrie comportamentul fiecărei opțiuni expuse în interfața W3C Markup Validator API.

* Encoding(charset) – ne dă posibilitatea de a suprascrie informația codificării caracterelor despre acel document. Se poate folosi această opțiune pentru teste, dar în cele din urmă va trebui să se seteze documentului codificarea caracterelor corectă.
* Folosirea fallback-ului în locul suprascrierii(fbc) – folosește mecanismul suprascrierii codificării caracterelor descris mai sus, dar doar dacă documentul nu este servit cu informația codificării caracterelor.
* Tipul documentului (doctype) – pentru a suprascrie declarația DOCTYPE a documentului.
* Folosirea fallback-ului în locul suprascrierii(fbc) – folosește mecanismul suprascrierii DOCTYPE descris mai sus, dar doar dacă documentul nu are o declarație doctype.
* Arată sursa(ss) – afișează sursa documentului HTML validat și leagă mesajele de eroare direct la liniile din acest output. În acest mod, este mai ușor de a vedea erorile.
* Arată schiță(outline) – va genera o schiță a documentului de la elementele H1 la H6. Pentru un document formatat corect, asta va genera o structură de arbori imbricați. Vizualizarea structurii documentului în acest mod, va ușura descoperirea anumitor erori, de exemplu sărirea peste un heading.
* Validarea paginilor de erori (No200) – atunci când se va încerca validarea unei pagini care nu a putut fi regăsită (de exemplu, dacă serverul va afișa eroarea 404-not found ), această opțiune ne dă posibilitatea de a valida asfel de pagini.
* Verbose output(verbose) – verbose output adaugă explicații și sugestii rezultatului în urma validării, și dă mai multe informații despre resursa validată.
  1. **Serviciul de validare Dr. Watson**

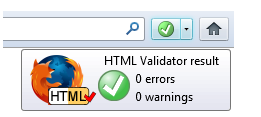
Dr. Watson este un serviciu la disponibilitatea tuturor pentru a analiza o pagină web. Prin tastarea URL-ului paginii, Watson va lua o copie a paginii direct de pe server. Watson poate verifica și alte aspecte ale paginii web, incluzând validitatea link-ului, viteza de download, compatibilitatea motor de căutare, și link-ul de popularitate.



**Fig. 1.1 Corectarea cu Dr. Watson**

* 1. **HTML Validator Firefox ADD-on**

Acest validator este inclus în browserul Firefox pe Windows. Validează paginile HTML în momentul accesării. Numărul erorilor este văzut în partea de sus a paginii, în bara de status, sub formă de icon.



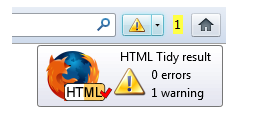
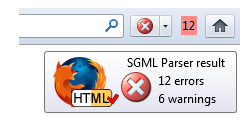
**Fig. 1.2 Icon HTML Validator Result**

Această extensie se bazează pe Tidy și OpenSP (SGML parser). Inițial, acești algoritmi au fost dezvoltați de Web Consortium W3C, acum folosiți de mai multe servicii. Amândoi algoritmi sunt incluși în Mozilla/Firefox și fac validarea local, fără să mai trimită documentul HTML la un al treilea server.

1. HTML Tidy este un program util, care încearcă să ajute în corectarea erorilor HTML. Ele sunt clasificate în 3 categorii:
2. Erori: erori HTML pe care Tidy nu le poate repara sau înțelege.
3. Avertismente: erori HTML pe care Tidy le poate corecta automat.
4. (opțional) avertismente de accesibilitate: avertismente HTML pentru nivelurile 3 prioritare definite în W3C WAI.
5. OpenSP este un parser SGML. Este același program care rulează în interiorul validatorului W3C.

Cum funcționează:

* Câteva fișiere XUL și javascript pentru reprezentarea rezultatului
* Câteva pagini HTML pentru a ajuta user-ul să înțeleagă erorile și avertismentele
* O librărie dinamică C, DLL sau SO, care conține:
* Codul sursa pentru Tidy
* Codul sursă pentru OpenSP
* Cod C pentru a face Tidy conform XPCOM și de a da posibilitatea apelării acestuia din javascript sau XUL

**Fig. 1.3 Icon HTML Tidy Result Fig. 1.4 Icon SGML Parser Result**

* 1. **Validatorul WDG HTML**

Validatorul WDG HTML este ușor de folosit și poate da informații despre paginile respective, direct. Acesta este asemănător cu validatorul oferit de W3C. Erorile raportate sunt aceleași în aproape toate cazurile. Cu toate acestea, pot apărea unele diferențe minore :

* Returnează avertismente pentru HTML valid: start-tag-uri neînchise (de exemplu, *<p> <img src=foo alt=bar </p>*), end-tag-uri neînchise (de exemplu, *<p> foo</>*). Alte validatoare nu dau astfel de avertismente.
* Returnează avertismente pentru referințe nedefinite, cum ar fi &#147; și &#153; . Serviciul de validare W3C raportează referințele nedefinite ca și erori pentru documentele XML, dar nu raportează nimic pentru documentele HTML.
* Spre deosebire de alte validatoare, acesta folosește o declarație specială SGML atunci când documentul se referă la un DTD personalizat. Rezultatul este că multe DTD-uri personalizate, în special acelea construite cu HTML 4.0 Transitional DTD, lucrează corect cu validatorul WDG HTML dar nu și cu alte validatoare.

Exemplu de validare a unui document HTML prin tastarea URL-ului:

|  |
| --- |
| <A  HREF="http://www.htmlhelp.com/cgi-  bin/validate.cgi?**url=referer**">Validate  me</A> |

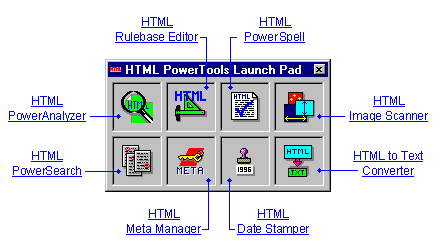
Pentru a include sursa documentului se folosește:

|  |
| --- |
| <A  HREF="http://www.htmlhelp.com/cgi-bin/validate.cgi?**url=referer&amp;input=yes**">Validate  me</A> |

* 1. **HTML Power Tools**

HTML Power Tools este o suită de aplicații Windows care ajută în managerarea website-urilor.

Acest serviciu va valida întregul site offline, va verifica link-urile și ortografia.



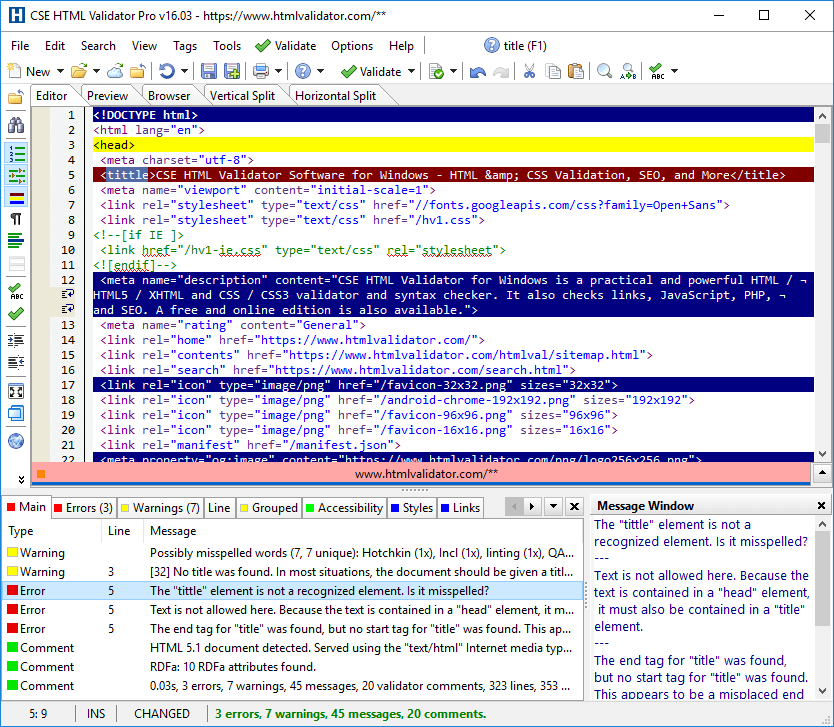
**Fig. 1.5 WDG Validator**

HTML Power Analyzer este un serviciu de scanare a fișierelor HTML și de a anunța user-ul de toate erorile descoperite. În plus față de toate tipurile de erori de sintaxă, HTML Power Analyzer va descoperi caracterele nevalide, codurile entitate, caracterele non-text, referințe lipsă / nevalide (fișiere locale și pagini web pe alte servere), ancore lipsă, precum și trimiterile care conțin legături de litere majuscule (care pot fi cauza problemelor pe serverele case-sensitive Unix ).

* 1. **CSE HTML Validator**

Acest validator este pentru profesioniști, webmasteri, corporații, agenții guvernamentale, persoane fizice, profesori, studenți, organizații non-profit, sau orice persoană cu un site web.

Spre deosebire de alte validatoare, acesta dispune de o verificare puternică de sintaxă a documentelor HTML funcționând fără a încărca documentele pe Internet, astfel încât datele fiind protejate.



**Fig. 1.6 CSE Validator**

* 1. **HTML Toolbox**

HTML Toolbox este un serviciu de subscripție online care va valida pagina web, va verifica link-urile și altele descrise mai jos. Nu este atât de folosit deoarece reprezintă un serviciu de subscripție.

Detalii despre acest serviciu:

* Verificarea link-ului: peste 5% din link-uri sunt distruse. Acest serviciu va căuta site-ul tău și va testa fiecare link pe care îl va găsi. Va returna un status pentru fiecare link verificat.
* Verificarea HTML și repararea: utilizarea tag-urilor într-un mod greșit, erori de sintaxă de cod HTML pot face dificil pentru browser de a procesa documentul și pentru vizitatori de a citi site-ul. Validatorul găsește tag-urile scrise greșit și va genera automat o pagină corectată pentru a fi încărcată pe site.
* Compatibilitatea cu browser-ul
* Verificarea timpului de încărcare
* Corectitudinea codului din punct de vedere sintactic.

**Cap 2. DESCRIEREA SOLUȚIEI**

Limbajul folosit în scrierea acestui serviciu este JAVA deoarece la ora actuală este cel mai folosit limbaj de programare, fiind independent de platformă. Poate fi utilizat în dezvoltarea conținutului web, aplicațiilor mobile sau a diferitelor programe.

Pentru validarea documentelor HTML m-am folosit de librăria JTIDY și HTMLCleaner.

JTIDY este un JAVA port de HTML Tidy, un corector HTML la nivel sintactic cu un output frumos organiza. JTidy poate fi folosit ca un instrument de curățare a HTML-ului malformat. De asemena, JTidy conține o interfață DOM pentru documentul care este procesat, dându-ți posibilitatea de a folosi JTidy ca un parser DOM pentru real-world HTML.

JTidy a fost scris de Andy Quick, care mai târziu a renunțat de a mai îngriji de el. Acum JTidy este îngrjit de un grup de voluntari.

Erorile găsite le clasifică în 3 categorii:

1. Erori: erori HTML pe care JTIDY nu le poate repara și nu le înțelege.
2. Avertismente: erori HTML pe care JTIDY le poate repara automatic.
3. (optional)Avertismente de accesibilitate: avertismente HTML pentru nivelurile 3 prioritare definite în w3C WAI.

Pentru ca JTidy să poată fi folosit, trebuie downloadată o distribuție JTidy. În ea se găsește o *jtidy-{version}.jar*, unde *version* este numărul de lansare al JTidy: acesta este jar-ul care conține toate clasele JTidy de care este nevoie. Nu este nevoie de nici o altă librărie.

JTidy poate fi folosit în mai multe moduri, cum ar fi: JTidy executabil, ant task, JTidy API. Modul folosit de mine este JTidy API.

**2.1 JTidy API**

Pentru a accesa funcționalitățile lui JTidy este nevoie de clasa org.w3c.tidy.Tidy . Acesta este un exemplu simplu de a fi folosit:

|  |
| --- |
| Tidy tidy = new Tidy(); // creează o instanță JTidy  tidy.setXHTML(boolean xhtml); // setează configurările dorite folosind  setters // (echivalent cu opțiunile de la linia de comandă)  tidy.parse(inputStream, System.out); // rulează tidy, obținând un input și un output stream |

Outputul poate fi un fișier XML sau XHTML. Am ales ca output strucura XHTML, deoarece este o structură mai nouă, având elemente de HTML combinate cu cele de XML, ceea ce îmi dă posibilitatea de a folosi acel fișier mai ușor, în mai multe moduri.

Cum lucrează JTidy:

* Sunt detectate și corectate tag-urile finale lipsă sau nepotrivite

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | <h1>heading  <h2>subheading</h3> | <h1>heading</h1>  <h2>subheading</h2> | |

* Tag-urile închise în ordinea greșită sunt corectate

|  |  |
| --- | --- |
| <p>here is a para <b>bold  <i>bold italic</b>  bold?</i> normal? | <p>here is a para <b>bold <i>bold italic</i> bold?</b> normal? |

* Repară problemele cu heading emphasis

|  |  |
| --- | --- |
| <h1><i>italic heading</h1>  <p>new paragraph | <h1><i>italic heading</i></h1>  <p>new paragraph |

* Repară tag-urile amestecate

|  |  |
| --- | --- |
| <i><h1>heading</h1></i>  <p>new paragraph <b>bold text  <p>some more bold text | <h1><i>heading</i></h1>  <p>new paragraph <b>bold text</b>  <p><b>some more bold text</b> |

* Aduce tag-ul <hr> în locul potrivit

|  |  |
| --- | --- |
| <h1><hr>heading</h1>  <h2>sub<hr>heading</h2> | <hr>  <h1>heading</h1>  <h2>sub</h2>  <hr>  <h2>heading</h2> |

* Adaugă ”/” în tag-urile finale

|  |  |
| --- | --- |
| <a href="#refs">References<a | <a href="#refs">References</a> |

* Perfecționează listele prin adăugarea tag-urilor lipsă

|  |  |
| --- | --- |
| <body>  <li>1st list item  <li>2nd list item | <body>  <ul>  <li>1st list item</li>  <li>2nd list item</li>  </ul> |

* Ghilimelele lipsă din valorile atributelor sunt adăugate
* Atributele necunoscute / de proprietate sunt raportate

Tidy are o cunoaștere cuprinzătoare a atributelor definite în HTML 4.0, recomandarea de la W3C . Acest lucru permite de multe ori să descoperiți locul unde ați scris greșit un atribut sau o valoare .

* Elementele de proprietate sunt recunoscute și raportate ca atare

Tidy va funcționa chiar pe ce versiune de HTML utilizați, înserând elementul DOCTYPE , conform recomandărilor W3C.

* Tag-urile lipsite de ' > ' sunt reparate .

Acest lucru trebuie reparat de user pentru că Tidy este nesigur de locul unde ar trebui să fie introdus.

**Așezarea în pagină**

Puteți alege stilul pe care doriți ca Tidy să îl utilizeze când generează marcajul curățat: de exemplu, puteți alege ca elementele să indenteze conținutul lor. Păstrarea așezării în pagină originale nu este posibilă datorită modului cum lucrează Tidy. Tidy începe prin construirea unui arbore de parsare Tidy curat din fișierul sursă. Arborele de parsare nu conține nici o informație cu privire la aspectul original. Tidy apoi creează aspectul dorit folosind opțiunile curente. Încercarea de a păstra aspectul original ar interacționa prost cu operațiunile de reparații necesare pentru a construi un arbore de parsare curat și complică în mod considerabil codul.

Unele browsere pot strica alinierea dreaptă a textului în funcție de modul în care poziționați acea rubrică. Ca un exemplu, luați în considerare :

|  |
| --- |
| <h1 align="right">  Heading  </h1>  <h1 align="right">Heading</h1> |

Ambele ar trebui să fie așezate la fel. Din păcate, din cauza unui bug comun, browser-ul nu reușește să șteargă spațiul gol aliniind greșit primul aliniat. HTML Tidy vă va proteja de acest bug , cu excepția cazului când setați opțiunea la indent cu " da ".

Setarea opțiunii de indentare poate provoca, de asemenea, probleme în aspectul unui tabel pentru unele browsere:

|  |
| --- |
| <td><img src="foo.gif"></td>  <td><img src="foo.gif"></td>  -va arăta ușor diferit de la:  <td>  <img src="foo.gif">  </td>  <td>  <img src="foo.gif">  </td> |

Puteți evita astfel de modificări prin utilizarea identării nesetată sau auto în fișierul de configurare.

**2.2 HTML Cleaner**

HTMLCleaner este un parser HTML open-source scris în JAVA. Ca orice alt proiect open-source, se bazează pe contribuțiile comunității. Cel care îngrijește acest proiect este Scott Wilson.

HTMLCleaner poate fi folosit în codul java, la linia de comandă sau ca un ant task. Este proiectat pentru a fi independent (nu are dependențe la rulare – decât JRE 1.5+), rapid și flexibil (comportamentul lui este configurabil printr-un număr de parametri).

Acesta are ca input un fișier HTML și generează o structură de tip arbore perfectă pentru manipulare programatică. Serializările sunt responsabile pentru a salva structura DOM înr-un fișier XML, HTML, DOM sau Jdom.

HTMLCleaner este thread safe, însemnând că o singură instanță poate curăța mai multe surse html în același timp.

Un mic exemplu despre folosirea acestui serviciu în Java:

|  |
| --- |
| *// create an instance of HtmlCleaner*  HtmlCleaner cleaner = **new** HtmlCleaner();  *// take default cleaner properties*  CleanerProperties props = cleaner.getProperties();  *// customize cleaner's behaviour with property setters*  props.setXXX(...);  *// Clean HTML taken from simple string, file, URL, input stream,*  *// input source or reader. Result is root node of created*  *// tree-like structure. Single cleaner instance may be safely used*  *// multiple times.*  TagNode node = cleaner.clean(...);  *// optionally find parts of the DOM or modify some nodes*  TagNode[] myNodes = node.getElementsByXXX(...);  *// and/or*  Object[] myNodes = node.evaluateXPath(xPathExpression);  *// and/or*  aNode.removeFromTree();  *// and/or*  aNode.addAttribute(attName, attValue);  *// and/or*  aNode.removeAttribute(attName, attValue);  *// and/or*  cleaner.setInnerHtml(aNode, htmlContent);  *// and/or do some other tree manipulation/traversal*  *// serialize a node to a file, output stream, DOM, JDom...*  **new** XXXSerializer(props).writeXmlXXX(aNode, ...);  myJDom = **new** JDomSerializer(props, **true**).createJDom(aNode);  myDom = **new** DomSerializer(props, **true**).createDOM(aNode); |

Un exemplu despre cum funcționează acest serviciu:

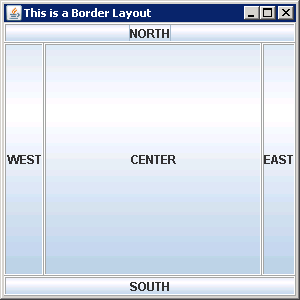
**Tabel 2.1 Exemplu de text convertit**

|  |  |
| --- | --- |
| Textul Original | Textul corectat |
| <table id=table1 cellspacing=2px  <h1>CONTENT</h1>  <td><a href=index.html>1 -> Home Page</a>  <td><a href=intro.html> 2-> Introduction</a> | **<h1>**CONTENT**</h1>**  **<table** id="table1" cellspacing="2px"**>**  **<tbody>**  **<tr>**  **<td>**  **<a** href="index.html"**>**1 -&gt; Home Page**</a>**  **</td>**  **<td>**  **<a** href="intro.html"**>**2 -&gt; Introduction**</a>**  **</td>**  **</tr>**  **</tbody>**  **</table>** |

* 1. **Soluția**

Aplicația dispune de o interfață grafică pentru ca user-ul să poată folosi cât mai ușor acest serviciu și pentru a ascunde implementarea acestuia.

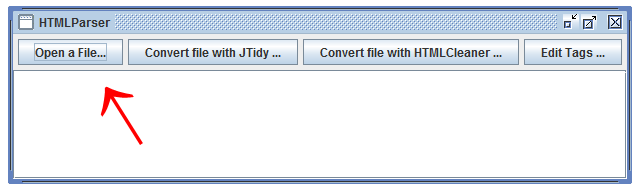
Interfața grafică este formată dintr-un Border Layout.



**Fig. 2.1 Border Layout**

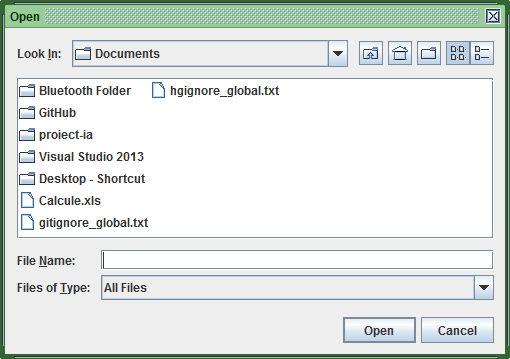
Acesta este împărțit în mai multe zone: zona nordică, centrală, estică, vestică și sudică. În zona nordică sunt așezate butoanele: ”Open a File”, ”Convert file with JTidy”, ”Convert file with HTMLCleaner”, ”Edit Tags”, iar în zona centrală este scroll panel în care se află un Text Area pentru a scrie diferite mesaje.

Din interfața grafică se poate alege fișierul pentru a fi dat mai departe la corectat, convertit și editat prin apăsarea butonului ”Open a File...”.



**Fig. 2.2 Open a File**

Prin apăsarea acestui buton se va declanșa un File Chooser din librăria Java Swing ce va permite utilizatorului să își aleagă fișierul dorit.



**Fig. 2.3 File Chooser**

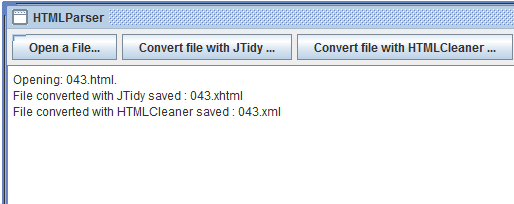
După alegerea fișierului de corectat, se poate alege varianta de convertire. Poate fi una singură sau pot fi amândouă pentru un singur fișier. Fiecare variantă aleasă va crea fișiere diferite la output, cu același nume ca și al fișierului original, însă cu extensie diferită (.xhtml pentru JTidy și .xml pentru HTMLCleaner).



**Fig. 2.4 Extensie fișiere**

Programul va prelua calea fișierului ales și va fi trimis ca parametru, în funcție de varianta de convertire aleasă, funcției care se ocupă cu asta.

Mesaje corespunzătoare vor fi afișate în dialog.

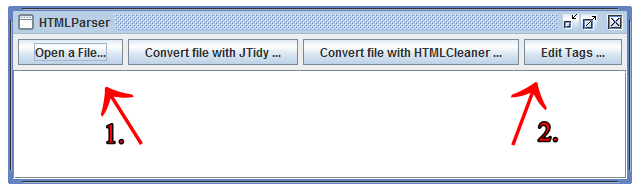


**Fig. 2.5 Exemplu de dialog**

Ceea ce acest proiect aduce în plus este editarea unui fișier XHTML sau XML din punct de vedere sintactic. Editarea se face la nivel de tag-uri și atribute, ele putând fi înlocuite în funcție de un atribut, șterse cu text, șterse fără text. Aceste tag-uri vor fi scrise într-un fișier de configurare cu o structură strictă ce trebuie respectată pentru funcționarea corectă a serviciului.

**2.4 Fișier de configurare**

User-ul construiește un fișier XML de configurare unde va scrie toate tag-urile ce trebuie a fi editate.



**Fig. 2.6 Pași pentru editare fișier**

Utilizatorul va deschide fișierul de configurare prin apăsarea butonului ”Open a file...” apoi va trimite fișierul ( calea fișierului este trimisă ca parametru funcției ) la editare prin apăsarea butonului ”Edit Tags...”.

Structura fișierului de configurare este următoarea:

|  |
| --- |
| <?xml version="1.0" encoding="utf-8" standalone="no"?>  <tags>  <filename>replace.xhtml</filename>  <replaceTags>  <oldTag>span  <oldAttr>class</oldAttr> <attrValue>c25</attrValue> </oldTag>  <newTag>i</newTag>  </replaceTags>  <deleteTagWithoutText>span <attribute>class</attribute> <attrValue>c18</attrValue> </deleteTagWithoutText>  <deleteTagWithText>p <attribute>class</attribute> <attrValue>c6</attrValue> </deleteTagWithText>  <deleteTagWithText>i <attribute>class</attribute> <attrValue>c3</attrValue> </deleteTagWithText>  </tags> |

Vom lua fiecare rând din fișierul de configurare pentru explicarea lui.

* *<filename>replace.xhtml</filename>* : numele fișierului de editat
* *<replaceTags> </replaceTags>* : tag-urile care vor fi înlocuite.
* *<oldTag>span <oldAttr>class</oldAttr> <attrValue>c25</attrValue> </oldTag>* : este important ca aceste tag-uri să fie scrise pe același rând. Între tag-urile ”oldTag” va fi scris tag-ul care vrem să fie înlocuit, urmat de tag-ul ”oldAttr” însemnând atributul după care căutăm acel tag, și ”attrValue”, valoarea atributului.
* *<newTag>i</newTag>* : tag-ul nou, cu care va fi înlocuit cel vechi. Dacă avem mai multe tag-uri de înlocuit, vom avea nevoie de o pereche nouă de tag-uri ”replaceTags”, în care vom completa datele necesare.
* *<deleteTagWithoutText>span <attribute>class</attribute> <attrValue>c18</attrValue> </deleteTagWithoutText>* : este important ca aceste tag-uri să fie scrise pe un singur rând. Între tag-urile ”deleteTagWithoutText” va fi trecut tag-ul care vrem să fie șters, însă textul va fi păstrat. Dacă avem mai multe tag-uri de șters, tot în cadrul tag-ului ”deleteTagWithoutText”, însă unul nou, scris pe un rând nou (nu neapărat sub celălalt, ordinea nu este importantă), vom trece datele necesare.
* *<deleteTagWithText>i <attribute>class</attribute> <attrValue>c3</attrValue> </deleteTagWithText>* : este important ca aceste tag-uri să fie scrise pe un singur rând. Între tag-urile ”deleteTagWithText” va fi trecut tag-ul pentru ștergere, inclusiv textul. Dacă vrem ca mai multe tag-uri să fie șterse, respectăm aceeasi regulă ca la ștergerea fără text.

Pentru fiecare fișier ce se dorește a fi modificat se creează câte un fișier de configurare.

**2.5 Editarea tag-urilor**

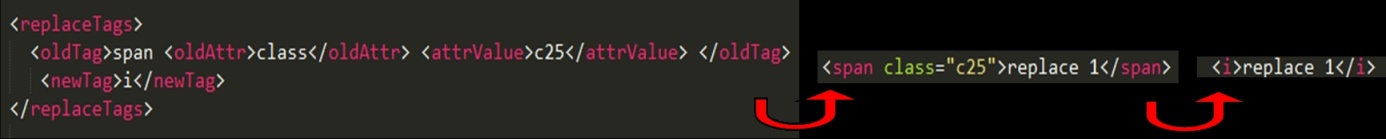
Fișierul este accesat și manipulat cu ajutorul lui DOM (Document Object Model). Acesta definește o structură logică și standardizată a documentelor, o ordine prin care putem parcurge și edita elemente și conținut din documente HTML sau XML. Structura DOM, asemenea unui arbore, este o ierarhie de obiecte, compatibilă cu a limbajului JavaScript sau cu a oricărui limbaj bazat pe obiecte.

Fiecare pereche de tag-uri este salvată ca un nod în arbore, și poate fi accesat în diferite moduri, depinde de nevoia noastră. Cel mai simplu mod de a accesa un nod este prin folosirea funcției *getElementsByTagName(”tagName”)*.

Prima dată este accesat nodul ”filename” pentru a ști în ce fișier vor fi editate tag-urile, cu ajutorul expresiei Xpath ”//filename”.

XPath folosește expresii de cale pentru a selecta noduri sau seturi de noduri într- un document XML. Nodul este selectat urmând o cale sau pași .

Apoi sunt accesate și salvate nodurile ”replaceTags”. Acestea sunt trimise ca parametru unei funcții replaceTag care mai are ca parametru numele fișierului. Pentru accesarea tututor nodurilor ce vrem a fi înlocuite, folosesc expresiile Xpath. Expresiile sunt alcătuite automat de o altă funcție numită constructXPathExp, având ca input un string în care este salvat numele nodului, numele atributului și valoarea atributului, returnând un string de forma: //numeTag[@numeAtribut=”valoareAtribut”]. Expresia XPath îmi va selecta nodurile cu numele tag-ului specificat în expresie dar doar care au un anumit atribut cu o anumită valoare (specificate tot în expresie). Nodurile accesate în urma parsării expresiei Xpath sunt modificate. Mai întâi vor fi șterse toate atributele existente pentru acel nod, apoi va fi redenumit nodul, tot cu ajutorul lui DOM, apelând funcția *renameNode(node,null,newName)*, textul din cadrul nodului rămânând neschimbat. După terminarea acestei acțiuni, mesajul ”REPLACE succeeded ! Verify filename.xhtml file. ” va fi afișat în consolă.

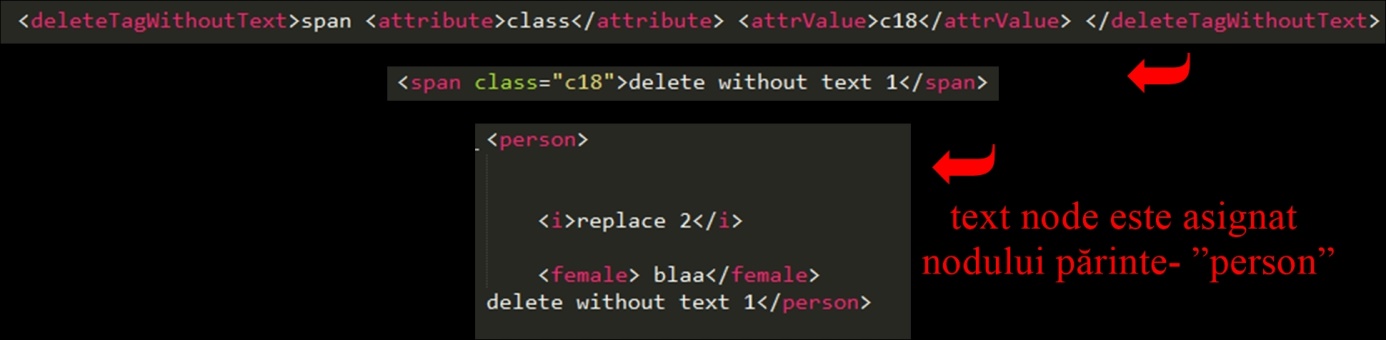


**Fig. 2.7 Înlocuire tag**

Urmând aceeași pași, vor fi citite nodurile ”deleteTagsWithoutText” și ”deleteTagsWithText”, după care vor fi trimise ca parametru funcțiilor deleteTagWithoutText() și deleteTagWithText().

În cadrul funcției ”deleteTagWithoutText”, într-o variabilă de tip Node este salvat părintele nodului curent și textul acelui nod într-un string, apoi acesta este șters. Pentru a rămâne textul în fișier, creem un nou nod text, asignat nodului părinte, salvat anterior, în care punem textul salvat. Astfel, nodul curent este șters, însă textul rămâne neschimbat asignat nodului părinte:

|  |
| --- |
| parent.removeChild(nodes.item(i));  parent.appendChild(doc.createTextNode(textNode)); |



**Fig. 2.8 Ștergere fără text a unui tag**

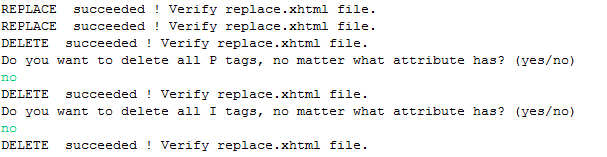
Funcția ”deleteTagWithText” șterge nodul curent complet, apelând funcția din DOM *removeChild(node).*

De asemenea, înainte ca un nod citit din fișier să fie trimis spre editare, user-ul are la dispoziție opțiunea de a șterge toate aparițiile acelui nod în fișier indiferent de atribut și valoarea lui. Pentru aceasta, este apelată funcția *userWish()* care are ca parametri numele nodului, numele atributului, tag-ul nou (în caz că se dorește înlocuirea) și opțiunea (0: pentru ștergerea tag-ului indiferent de atribut). Utilizatorul este întrebat (în consolă) dacă dorește ștergerea tutoror tag-urilor cu numele respectiv, indiferent de atribut și valoarea acestuia, în schimbul comenzii scrise în fișierul de configurare. Dacă utilizatorul răspunde pozitiv (funcția va returna 1), atunci comanda curentă din fișierul de configurare este ignorată și se execută acea opțiune. Am pus la dispoziția utilizatorului și această opțiune deoarece de multe ori el nu își dă seama concret de ce anume are nevoie, sau uită să treacă și această comandă în fișier. Dacă răspunsul este negativ (funcția va returna 0), se continuă cu comanda curentă.

Acest serviciu are suport și pentru expresii regulate. Utilizatorul poate introduce în fișierul de configurare o comandă de genul: ” *<oldTag>span <oldAttr>class</oldAttr> <attrValue>z\*</attrValue> </oldTag>*”. Programul va citi și va verifica dacă la sfârșitul valorii atributului se află ”\*”. Dacă răspunsul este pozitiv, va fi apelată funcția *editWithRegularExpressions()*, unde comanda va fi interpretată astfel: caut în fișier un nod cu numele ”span” care are ca atribut ”class”. Va lua fiecare nod pe rând și va verifica dacă valoarea atributului respectă expresia regulată. Simbolul ”\*” ține locul oricărui string posibil și valid (însemnând, de asemenea, că cifrele stau doar la sfârșitul stringului), de exemplu: ”z25”, ”zz25”, ”zClass34”.

Expresia regulată pentru exemplul de mai sus, are următoarea formă: ”*z[a-z]\*[0-9\**”. După ”z” poate să urmeze alte litere sau cifre.

După terminarea acestor acțiuni, consola va conține următoarele mesaje:



**Fig. 2.8 Exemplu de consolă**

Se poate continua cu încărcarea altui fișier de configurare pentru editarea altui fișier, sau relurea corectării/convertirii altor fișiere.

**Cap 3. TESTE ȘI EVALUARE**

În urma testelor făcute pe 4 fișiere, am ajuns la concluzia că serviciul JTidy este mai eficient din mai multe puncte de vedere decât HTMLCleaner.

Inițial, HTMLCleaner căzuse orice test deoarece nu avea setată codificarea pe UTF-8, și atunci diacritecele sau spațiile sau alte simboluri nu erau intepretate corect, apărând în locul lor un string de numere și simboluri.

**Tabel 3.1 Test1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| File  name | Original Text | Converted Text with JTIDY | Converted text with HTMLCleaner | Best Choice |
| 043 | <p><b>     PIULÚG</b> s. n. v. <b>pilug.</b></p>  <p><br/>  *(****PIULÚG****s. n. v.****pilug.***  *)* | <p><b>&nbsp; &nbsp; &nbsp;PIULÚG</b> s. n. v. <b>pilug.</b></p>  <p><br />  *(****PIULÚG****s. n. v.****pilug.****)* | <p>  <b>&#194;&#160; &#194;&#160; &#194;&#160;PIULĂ&#353;G</b>  s. n. v.  <b>pilug.</b>  </p>  *(****Â  Â  Â PIUL�šG****s. n. v.****pilug.****)* | JTIDY |
| 043 | <b>PIUNÉZĂ</b> s. f. Cui special cu capul  rotund şi plat<br/>  *(****PIUNÉZĂ****s. f. Cui special cu capul rotund şi plat)* | &nbsp; &nbsp; &nbsp;<b>PIUNÉZĂ</b> s. f. Cui special cu capul rotund şi plat<br />  *(****PIUNÉZĂ****s. f. Cui special cu capul rotund şi plat)* | &#194;&#160; &#194;&#160; &#194;&#160;  <b>PIUNĂ&#8240;Z&#196;&#8218;</b>  s. f. Cui special cu capul  rotund Ĺźi plat  <br/>  *(Â  Â  Â****PIUN�‰ZÄ‚****s. f. Cui special cu capul rotund şi plat )* | JTIDY |
| 043 | NR randuri:758 | NR randuri: 654 | NR randuri: 2821 | JTIDY |

*Sursa: Anexa1. Tabel 1.*

Apoi fișierele au fost testate și din punctul de vedere al dimensiunii. Serviciul HTMLCleaner adaugă foarte multe rânduri noi, practic fiecare tag este scris pe câte un rând. Asta a condus la o dimensiune, în medie, de 4 ori mai mare față de fișierul original. JTidy în schimb, reduce dimensiunea fișierului, în medie, cu aproximativ 200 de rânduri.

După efectuarea acestor teste, serviciul JTidy este mai eficient.

Alte teste au fost făcute după setarea lui HTMLCleaner pe UTF-8. În urma testelor, au fost detectate alte probleme.

**Tabel 3.2 Test2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| File  name | Original Text | Converted text with JTIDY | Converted text with HTMLCleaner | Best Choice |
| 043 | <span class="c1">C. PETRESCU,  C. V. 42</span>. <i>Pereţii scorojiţi<br/>  abia suportau afişele prinse cu pioneze.</i>  C. PETRESCU, C. V. 42.Pereţii scorojiţi | <span class="c1">C. PETRESCU, C. V. 42</span>  <p>. <i>Pereţii scorojiţi<br />  abia suportau afişele prinse cu pioneze.</i>  C. PETRESCU, C. V. 42  *. Pereţii scorojiţi* | <span class="c1">  C. PETRESCU,  C. V. 42  </span>  .  <i>  Pereţii scorojiţi  <br />  abia suportau afişele prinse cu pioneze.  </i>  C. PETRESCU, C. V. 42 .*Pereţii scorojiţi* | **HTML**  **Cleaner** |
| 043 | <br />p</i><i>ioneză .</i>  *pioneză .* | <br />p</i><i>ioneză .</i>  *pioneză .* | <br />  p  </i>  <i>ioneză .</i>  *p ioneză .* | **JTIDY** |
| 043 | NR randuri: 759 | NR randuri: 656 | NR randuri: 2822 | **JTIDY** |

*Sursa: Anexa 2. Tabel 2.*

Deoarece HTMLCleaner scrie fiecare tag pe câte un rând, unele propoziții le taie în două, prima parte pe un rând, a doua pe alt rând, ca în exempul de mai sus descris în Tabelul 3.2. Aceeași problemă a apărut nu în cadrul unei propoziții, ci în cadrul unui cuvânt. Să luăm ca exemplu cazul 2 din tabelul 3.2 de mai sus : ”pioneză” -> ”p ioneză” . Litera ”p” este cuprinsă în tag-urile ”i”, dar putem observa și tag-ul ”br” înainte de litera ”p”. Tag-ul ”br” este scris pe o linie nouă, urmat de linia ce conține litera ”p”+” ”, după care este scrisă restul propoziției. Din această cauză cuvântul ”pioneză” nu este scris corect în fișierul creat de HTMLCleaner.

O problemă apărută la JTidy este cazul 1 din tabelul 3.2 de mai sus. Fiecare caracter care nu este inclus într-un tag, Jtidy îl include într-un paragraf (”p”). Putem observa că ”.”, în textul original nu este in cadrul nici unui tag, de aceea JTidy l-a inclus în tag-ul ”p”, paragrafele fiind scrise pe un rând nou, atunci când pagina HTML este deschisă cu un browser.

O altă problemă a serviciului JTidy este că nu păstrează ghilimele românești, ci le convertește la cele englezești, spre deosebire de HTMLCleaner care le păstrează pe cele originale (deoarece HTMLCleaner le convertește la date unicode).

Din punct de vedere al dimensiunii, testele au returnat aceleași valori, JTidy fiind cel favorabil în acest caz.

În urma acestor teste, JTidy a rezultat ca fiind mai favorabil față de HTMLCleaner.

**3.1 Evaluarea fișierului de configurare**

Dacă în fișierul de configurare nu avem de scris tag-uri pentru a fi înlocuite, nu lăsăm tag-urile fără text, de exemplu: ”<newTag> </newTag>”, ci ștergem tot ce conțin tag-urile ”replaceTags”, inclusiv pe ele, altfel, programul va returna eroare. Eroarea returnată în cazul exemplului de mai sus este *” org.w3c.dom.DOMException: INVALID\_CHARACTER\_ERR: An invalid or illegal XML character is specified.”,* însemnând că DOM nu poate înlocui un tag vechi cu spațiu (spațiu fiind un caracter invalid în acest caz).

De asemenea, dacă lăsăm spațiu între textul nodului și tag-ul acestuia, de exemplu: ” <newTag> i </newTag>”, programul va returna aceeași eroare ca mai sus, deoarece îl ia ca atare, cu spațiu, deci interpretat ca un simbol invalid.

Între tag-urile ”filename” nu este nevoie a fi precizată calea fișierului, deoarece căutarea se va face la nivel global.

De obicei fișierele de configurare sunt fișiere text, însă eu am optat pentru XML, deoarece, în acest caz, este mai ușor de parsat.

Dacă, în interfața grafică, se apasă întâi pe unul din butoanele de convertire, în consolă se va afișa o eroare, însă nu va afecta cu nimic rularea aplicației, deci se poate continua cu deschiderea fișierului. Fereastra se poate maximiza sau minimiza.

**CONCLUZII ȘI DEZVOLTĂRI**

Întrucât browserele web contemporane fac o treabă tot mai bună în parsarea chiar și a celui mai rău HTML ”tag soup”, unele erori nu sunt întotdeauna prinse cu grație. Foarte des, un anumit software pe diferite platforme nu se va ocupa de erori într-un mod similar, ceea ce face extrem de dificil de aplicat un stil sau un aspect consecvent.

Folosind standard-ul, marcajele interoperabile și stylesheet-urile, pe de altă parte, oferă o șansă mult mai mare de a avea o singură pagină manipulată în mod constant pe mai multe platforme și user-agenți. Într-adevăr, majoritatea dezvoltatorilor care crează aplicații Web bogate știu că scripting-ul de încredere are nevoie ca documentul să fie analizat de către user-agenți, fără nici o eroare neașteptată, și se va asigura că marcajele lor și CSS-ul este validat înainte de a crea un layer interactiv bogat .

Atunci când s-au realizat studii (studiul din 2009, realizat de W3C) , marea majoritate a profesioniștilor Web au afirmat că erorile de validare este primul lucru pe care îl verifică de fiecare dată când intră în contact cu stilizarea paginilor Web sau bug scripting.

Majoritatea dezvoltatorilor de aplicații Web, când sunt întrebați dacă își validează paginile Web, răspunsul lor este unul simplu, și anume ”Site-ul meu arată bine și lucrează corect”. Verificarea existenței unei pagini de acest gen, în mai multe browsere contemporane poate fi o asigurare rezonabilă că pagina va ”lucra” astăzi , dar nu garantează că va funcționa mâine .

Este rezonabil să se considere că standarde, precum HTML și CSS sunt o formă a unui ”stil de scriere a unui cod”, care este definit global. Crearea de pagini web sau aplicații în conformitate cu un stil acceptat pe scară largă le face mai ușor de întreținut, chiar dacă menținerea și evoluția este realizată de altcineva.

Mulți profesioniști au fost întreținători Web cu HTML și CSS pentru ani și cunosc aceste tehnologii pe de rost. Începătorii și studenții, pe de altă parte, vor găsi instrumente automate de verificare foarte bune în descoperirea greșelilor. De asemenea, unii profesori subliniază faptul că testele automate de validare sunt o bună introducere la mai multe concepte de calitate mai complexe, cum ar fi accesibilitatea.

În ziua de astăzi, există puține sau nici o certificare pentru profesioniștii Web, ci doar câteva universități care predau tehnologii Web, lăsând ca cei mai mulți doritori să învețe de la sine, cu un succes variat. Profesioniștii capabili se mândresc cu crearea de conținut web folosind marcaje semantice bine formatate, separarea de stil și de conținut etc. Validarea poate fi apoi folosită ca o verificare rapidă pentru a determina dacă codul este scris curat, sau scris rapid cu greșeli multe.

Validarea, ca orice proces de cod de depanare, este uneori dificil, iar îmbunătățirile vaste de corectare automată a erorilor a făcut browser-ul modern să facă față foarte bine cu erorile din HTML sau CSS. Acest lucru face ca validarea să pară inutilă sau costisitoare pentru mulți oameni, și următoarele întrebări ( sau declarații) sunt larg răspândite (preluate de pe site-ul oficial W3C) :

1. ”Site-ul meu arată bine și funcționează bine - nu este de ajuns ?”- în practică, diferite browsere pot face și afișa aceeași pagină foarte diferit. Acest lucru este în mod deliberat, și nu implică nici un fel de eroare de browser. Un termen folosit uneori pentru acest lucru este WYSINWOG –”What You See Is Not What Others Get”- Ce vezi nu este ceea ce alții primesc ( cu excepția cazului de coincidență ). Este într-adevăr, una dintre principalele forțe ale web-ului, care (de exemplu) un utilizator cu deficiență de vedere poate selecta dimensiuni foarte mari de vizualizare a textului, fără a fi nevoie să se scrie o pagină Web specială pentru așa ceva.
2. ”O mulțime de site-uri web nu sunt validate - inclusiv societățile cu nume de uz casnic”- companiile cu nume de uz casnic se așteaptă ca oamenii să viziteze paginile din cauza numelui, în ciuda site-urilor web îngrozitoare. Există riscul de a fi pe partea greșită a unui proces judiciar în cazul în care site-ul se dovedește a fi inaccesibil - de exemplu - o persoană cu handicap care nu poate utiliza un browser ”convențional”. Accesibilitatea este legea în multe țări. În timp ce validarea nu garantează accesibilitatea, ar trebui să fie o componentă importantă de exercitare a verificării antecedentelor. Pentru prima dată o instanță a acordat daune unui utilizator orb împotriva proprietarului unui site web găsit inaccesibil ( Maguire vs SOCOG , august 2000) .
3. ”Validarea înseamnă site-uri plictisitoare, și înfrânge creativitatea”- S-ar putea să fi fost cazul cu un deceniu în urmă, când validarea a fost instrumentul de alegere a persoanelor mai interesate de valorificarea puterii limbajelor de marcaje decât crearea de modele frumoase pentru conținutul lor, când mulți designeri nu au fost învățați elementele de bază ale tehnologiei Web și ar fi creat site-uri web frumoase, dar fragile și nesigure. Acest argument este complet discutabil astăzi. În ultimul deceniu, cele mai uimitoare site-uri Web, cu conținut și design bogat au fost construite cu standardul ( X ) HTML , CSS și scripting.

După cum este descris și mai sus, sunt multe motive pentru care validarea documentelor Web este atât de importantă și folositoare. Însă serviciul creat de mine, pune la dispoziția utilizatorului, în primul rând o comparare între 2 servicii de validare foarte bune și foarte folosite, și în al doilea rând, editarea tag-urilor.

Editarea tag-urilor automată, nu este prea întalnită în ziua de astăzi print-un serviciu care să primească ca input un anumit tag cu anumite proprietăți, iar acel serviciu să execute comanda dorită de utilizator, cum ar fi ștergerea tuturor acelor tag-uri în întregime, redenumirea lor, ștergerea lor fără text. Serviciul DOM poate fi folosit în această situație, însă utilizatorul trebuie să fie cunoscător de HTML și CSS, și a unui limbaj de programare. Pe când, serviciul creat de mine, poate fi folosit de orice persoană, cu cunoștințe în domeniul programării, sau fără cunoștințe. Tot ce trebuie să facă utilizatorul este să ruleze aplicația, și să încarce fișierul de configurare (din interfața grafică), care are o structură strictă, inclus în pachetul de instalare a serviciului, deci el nu trebuie decât să completeze fișierul de configurare cu datele necesare. Interfața grafică, scrisă cât mai simplu și cât mai deductibilă, oferă utilizatorului un suport enorm în ușurința folosirii lui.

Editarea tag-urilor are suport pentru expresii regulate, însemnând că oferă utilizatorului editarea tag-ului care, de exemplu, începe cu o anumită literă. O proprietate extrem de importantă a acestui serviciu, oferind utilizatorului șansa de a generaliza căutarea anumitor tag-uri și editarea lor.

Obiectivele principale ale acestui serviciu sunt, în primul rând, de a conștientiza programatorii amatori, și chiar și cei profesioniști, că validarea documentelor HTML este o parte importantă din crearea de pagini curate și rulabile în orice mediu, cât și în a le răspunde nevoii de a își schimba fișierul XML sau XHTML în orice formă dorită de ei. Poate că utilizatorul consideră că un anumit tag cu un atribut și o valoare greu de reținut, trebuie redenumit cu o valoare simplă și scurtă.

Acest serviciu poate fi, de asemenea, extins în mai multe moduri. Acestă nevoie de editare și transformare spre mai bine a unui fișier XML este mare, de aceea se poate exinde cât mai mult spre această parte. În primul rând, fișierul de configurare poate fi extins către mai multe comenzi ce pot fi introduse de utilizator, de exemplu, redenumirea sau ștergerea după numele și valoarea atributului (vreau să redenumesc toate tag-urile care au ”class=c25”). Acest lucru este nevoie atunci când vreau să elimin din stiluri o anumită proprietate.

De asemenea, se poate automatiza alegerea făcută între JTidy și HTMLCleaner, în funcție de deosebirile găsite în acele fișiere. Se poate folosi un clasificator care să învețe din testele făcute anterior, și să aleagă varianta optimă (clasificator de la Învațare Automată).

Programatorii de multe ori, după terminarea scrierii unui document HTML de dimensiuni mari, dorește eliminarea anumitor elemente, tag-uri din acel fișier pentru a crește viteza de procesare și parsare a acestuia. Unii programatori nici nu se gândesc la așa ceva, deoarece nu știu că este posibil acest lucru printr-un serviciu.

**BIBLIOGRAFIE**

1. <http://www.benmccann.com/java-html-parsing-library-comparison/>
2. <http://webdesign.about.com/od/htmlvalidators/tp/best_html_validators.htm>
3. <https://www.w3.org/People/Raggett/tidy/>
4. <http://validator.w3.org/about.html>
5. <http://watson.addy.com/>
6. <http://users.skynet.be/mgueury/mozilla/>
7. <http://htmlhelp.com/tools/validator/differences.html.en>
8. <http://htmlhelp.com/tools/validator/tips.html.en>
9. <http://www.tali.com/indexo.html>
10. <https://www.htmlvalidator.com/>
11. <http://www.fatcow.com/product/html_toolbox.bml>
12. <https://docs.oracle.com/javase/tutorial/jaxp/dom/>
13. <https://docs.oracle.com/javase/7/docs/api/javax/xml/xpath/package-summary.html>
14. Studiul ”De este importantă validarea?”(2009):

<https://validator.w3.org/docs/why.html>

1. ”Maguire vs SOCOG , august 2000”:

https://www.w3.org/WAI/bcase/socog-case-study

1. <https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>
2. <http://www.tiobe.com/tiobe_index>
3. <http://www.java2s.com/Tutorial/Java/0240__Swing/WhatistheBorderLayout.htm>
4. <http://www.java2s.com/Tutorial/Java/0240__Swing/UsingaBorderLayoutManager.htm>
5. <http://webdesign.about.com/cs/html40/a/aa030303a.htm>
6. http://webdesign.about.com/od/htmlvalidators/a/aa092799.htm

Anexă 1

Tabel 1. Test 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filename | Original Text | Converted Text with JTIDY | Converted text with HTMLCleaner | Best Choice |
| 043.html | <p><b>     PIULÚG</b> s. n. v. <b>pilug.</b></p>  <p><br/>  *(****PIULÚG****s. n. v.****pilug.***  *)* | <p><b>&nbsp; &nbsp; &nbsp;PIULÚG</b> s. n. v. <b>pilug.</b></p>  <p><br />  *(****PIULÚG****s. n. v.****pilug.****)* | <p>  <b>&#194;&#160; &#194;&#160; &#194;&#160;PIULĂ&#353;G</b>  s. n. v.  <b>pilug.</b>  </p>  *(****Â  Â  Â PIUL�šG****s. n. v.****pilug.****)* | JTIDY |
| 043.html | <b>PIUNÉZĂ</b> s. f. Cui special cu capul  rotund şi plat<br/>  *(****PIUNÉZĂ****s. f. Cui special cu capul rotund şi plat)* | &nbsp; &nbsp; &nbsp;<b>PIUNÉZĂ</b> s. f. Cui special cu capul rotund şi plat<br />  *(****PIUNÉZĂ****s. f. Cui special cu capul rotund şi plat)* | &#194;&#160; &#194;&#160; &#194;&#160;  <b>PIUNĂ&#8240;Z&#196;&#8218;</b>  s. f. Cui special cu capul  rotund Ĺźi plat  <br/>  *(Â  Â  Â****PIUN�‰ZÄ‚****s. f. Cui special cu capul rotund şi plat )* | JTIDY |
| 0.43.html | <i>Piureul de cartofi</i></p>  <span class="user">rmarian</span><span class="seq">51918</span>  <p><i>﻿à la Parmentier, un fel de ipsos de lipit  geamurile.<br/></i><span class="c1">I. BOTEZ, B. I, 191</span>.  (*Piureul de cartofi*  *﻿à la Parmentier, un fel de ipsos de lipit geamurile.* I. BOTEZ, B. I, 191.) | <i>Piureul de cartofi</i></p>  <span class="user">rmarian</span><span class="seq">51918</span>  <p><i>﻿à la Parmentier, un fel de ipsos de lipit geamurile.<br />  </i><span class="c1">I. BOTEZ, B. I, 191</span>.  ( *Piureul de cartofi*  *﻿à la Parmentier, un fel de ipsos de lipit geamurile.* I. BOTEZ, B. I, 191.) | <i>Piureul de cartofi</i>  </p>  <span class="user">rmarian</span>  <span class="seq">51918</span>  <p>  <i>  ď&#187;żĂ&#160; la Parmentier, un fel de ipsos de lipit  geamurile.  <br />  </i>  <span class="c1">I. BOTEZ, B. I, 191</span>  .  ( *Piureul de cartofi*  *�»��  la Parmentier, un fel de ipsos de lipit geamurile.* I. BOTEZ, B. I, 191 .) | JTIDY |
| 043.html | <p><b>     2.</b> Dreptul (luat în arendă) de a  încasa pivniceritul<br/>  *(****2.****Dreptul (luat în arendă) de a încasa pivniceritul)* | <p><b>&nbsp;&nbsp; &nbsp; 2.</b> Dreptul (luat în arendă) de a încasa pivniceritul<br />  *(****2.****Dreptul (luat în arendă) de a încasa pivniceritul)* | <p>  <b>&#194;&#160;&#194;&#160; &#194;&#160; 2.</b>  Dreptul (luat Ă&#174;n arend&#196;�) de a  Ă&#174;ncasa pivniceritul  <br />  *(****Â Â  Â  2.****Dreptul (luat �®n arendÄ?) de a �®ncasa pivniceritul )* | JTIDY |
| 043.html | <i>Fumăritul asemenea era o  dajdie ce plăteau<br/></i></p>  <span class="user">rmarian</span><span class="seq">53838</span><i>stăpînii de pivniţe şi cîrciumi.</i>  <span class="c1">FILIMON, O. I, 121</span>.<br/>  <i>Ea ţine cheile de la cămară şi de la pivniţă.</i> <span class="c1">ALECSANDRI, T. I, 45</span>, <span class="c1">cf. CIHAC, II,  260</span>  ( *Fumăritul asemenea era o dajdie ce plăteau*  *stăpînii de pivniţe şi cîrciumi.* FILIMON, O. I, 121. *Ea ţine cheile de la cămară şi de la pivniţă.* ALECSANDRI, T. I, 45, cf. CIHAC, II, 260) | <i>Fumăritul asemenea era o dajdie ce plăteau<br />  </i></p>  <span class="user">rmarian</span><span class="seq">53838</span><i>stăpînii de pivniţe şi cîrciumi.</i> <span class="c1">FILIMON, O. I, 121</span>  <p>.<br />  <i>Ea ţine cheile de la cămară şi de la pivniţă.</i> <span class="c1">ALECSANDRI, T. I, 45</span>, <span class="c1">cf. CIHAC, II, 260</span>  (*Fumăritul asemenea era o dajdie ce plăteau*  *stăpînii de pivniţe şi cîrciumi.* FILIMON, O. I, 121  **.** ---------------------------------X *Ea ţine cheile de la cămară şi de la pivniţă.* ALECSANDRI, T. I, 45, cf. CIHAC, II, 260  ) | <i>  Fum&#196;�ritul asemenea era o  dajdie ce pl&#196;�teau  <br />  </i>  </p>  <span class="user">rmarian</span>  <span class="seq">53838</span>  <i>st&#196;�pĂ&#174;nii de pivniĹŁe Ĺźi cĂ&#174;rciumi.</i>  <span class="c1">FILIMON, O. I, 121</span>  .  <br />  <i>Ea ĹŁine cheile de la c&#196;�mar&#196;� Ĺźi de la pivniĹŁ&#196;�.</i>  <span class="c1">ALECSANDRI, T. I, 45</span>  ,  <span class="c1">  cf. CIHAC, II,  260  </span>  ( *FumÄ?ritul asemenea era o dajdie ce plÄ?teau*  *stÄ?p�®nii de pivniţe şi c�®rciumi.* FILIMON, O. I, 121 .  *Ea ţine cheile de la cÄ?marÄ? şi de la pivniţÄ?.* ALECSANDRI, T. I, 45 , cf. CIHAC, II, 260 ) | JTIDY |
| 043.html | <p> </p>  *( )* | <p>&nbsp;</p>  *( )* | <p>&#194;&#160;</p>  *(Â )* | JTIDY |
| 043.html | NR randuri: 758 | NR randuri: 654 | NR randuri: 2821 | JTIDY |
| 044.html | <p><b>    PÍXUS</b> s. n. v. <b>pixis</b>.</p>  <p class="c14">    </p>  *(****PÍXUS****s. n. v.****pixis****.*  *)* | <p><b>&nbsp;&nbsp;&nbsp;&nbsp;PÍXUS</b> s. n. v. <b>pixis</b>.</p>  <p class="c14">&nbsp;&nbsp;&nbsp;&nbsp;</p>  *(****PÍXUS****s. n. v.****pixis****.*  *)* | <p>  <b>&#194;&#160;&#194;&#160;&#194;&#160;&#194;&#160;PĂŤXUS</b>  s. n. v.  <b>pixis</b>  .  </p>  <p class="c14">&#194;&#160;&#194;&#160;&#194;&#160;&#194;&#160;</p>  *(****Â Â Â Â PĂŤXUS****s. n. v.****pixis****.*  ***Â Â Â Â***  *)* | JTIDY |
| 044.html | <span class="user">marius.clim</span><span class="seq">53538</span><i>de mult.</i> <span class="c1">MAT. FOLK.  240.</span> <i>Şi cară bani cu caru, Şî ţîne<br/>  pizma cu anu</i>. <span class="c1">VÎRCOL, V. 72</span>. ◊  <span class="c3">Expr</span>. (Învechit şi<br/>  *(MAT. FOLK. 240. Şi cară bani cu caru, Şî ţîne pizma cu anu. VÎRCOL, V. 72. ◊ Expr. (Învechit şi)* | <span class="user">marius.clim</span><span class="seq">53538</span><i>de mult.</i> <span class="c1">MAT. FOLK. 240.</span> <i>Şi cară bani cu caru, Şî ţîne<br />  pizma cu anu</i>  <p>. <span class="c1">VÎRCOL, V. 72</span>. ◊ <span class="c3">Expr</span>. (Învechit şi<br />  *( MAT. FOLK. 240. Şi cară bani cu caru, Şî ţîne pizma cu anu -------------X*  *. VÎRCOL, V. 72. ◊ Expr. (Învechit şi*  ) | <span class="c1">  MAT. FOLK.  240.  </span>  <i>  Ĺži car&#196;� bani cu caru, ĹžĂ&#174; ĹŁĂ&#174;ne  <br />  pizma cu anu  </i>  .  <span class="c1">VĂŽRCOL, V. 72</span>  . &#226;&#8212;&#352;  <span class="c3">Expr</span>  . (ĂŽnvechit Ĺźi  <br />  *(MAT. FOLK. 240. Ĺži carÄ� bani cu caru, ĹžĂ® ĹŁĂ®ne  pizma cu anu . VĂŽRCOL, V. 72 . â—Š Expr . (ĂŽnvechit Ĺźi )* | JTIDY |
|  | <p><span class="c1">TEAHA, C. N. 253.</span><br/>  − Pl.: <i>pizghiriţi,  -te.<br/>  *(TEAHA, C. N. 253.       − Pl.: pizghiriţi, -te.)* | <p><span class="c1">TEAHA, C. N. 253.</span><br />  &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;− Pl.: <i>pizghiriţi, -te.<br />  *(TEAHA, C. N. 253.       − Pl.: pizghiriţi, -te.)* | <p>  <span class="c1">TEAHA, C. N. 253.</span>  <br />  &#194;&#160;&#194;&#160;&#194;&#160;&#194;&#160;&#194;&#160;&#194;&#160;&#226;�&#8217; Pl.:  <i>  pizghiriĹŁi,  -te.  <br />  *(TEAHA, C. N. 253.  Â Â Â Â Â Â â�’ Pl.: pizghiriĹŁi, -te. )* | JTIDY |
| 044.html | NR randuri: 845 | NR randuri: 746 | NR randuri: 3077 | JTIDY |
| 045.html | <span class="c1">CF</span><span class="c1">. SCRIBAN, D.,  DM.</span> <small>♦</small> (Substantivat; regional; în<br/>  <p>forma <i>pîca</i>) Lulea. <span class="c1">CF</span><span class="c1">. ŞEZ. VII, 182</span>.</p>  *(CF. SCRIBAN, D., DM. ♦ (Substantivat; regional; în*  *forma pîca) Lulea. CF. ŞEZ. VII, 182.*  *)* | <span class="c1">CF</span><span class="c1">. SCRIBAN, D., DM.</span> <small>♦</small>  <p>(Substantivat; regional; în<br />  </p>  <p>forma <i>pîca</i>) Lulea. <span class="c1">CF</span><span class="c1">. ŞEZ. VII, 182</span>.</p>  *(CF. SCRIBAN, D., DM. ♦*  *(Substantivat; regional; în*  *forma pîca) Lulea. CF. ŞEZ. VII, 182.*  *)* | <span class="c1">CF</span>  <span class="c1">  . SCRIBAN, D.,  DM.  </span>  <small>&#226;&#8482;&#166;</small>  (Substantivat; regional; Ă&#174;n  <br />  <p>  forma  <i>pĂ&#174;ca</i>  ) Lulea.  <span class="c1">CF</span>  <span class="c1">. ĹžEZ. VII, 182</span>  .  </p>  *(CF . SCRIBAN, D., DM. â™¦ (Substantivat; regional; Ă®n*  *forma pĂ®ca ) Lulea. CF . ĹžEZ. VII, 182 .*  *)* | JTIDY |
| 045.html | <span class="missing">text lipsa</span><br/>  <span class="user">elenacarmen</span><span class="seq">53787</span><i>şi în jos e popuşoi.</i> <span class="c1">ib.  IX, 139</span>. <i>Strînsul pînei albe (a<br/>  *(text lipsa şi în jos e popuşoi. ib. IX, 139. Strînsul pînei albe (a)* | <span class="missing">text lipsa</span><br />  <span class="user">elenacarmen</span><span class="seq">53787</span><i>şi în jos e popuşoi.</i> <span class="c1">ib. IX, 139</span>  <p>. <i>Strînsul pînei albe (a<br />  *(text lipsa şi în jos e popuşoi. ib. IX, 139*  *. Strînsul pînei albe (a*  *)* | <span class="missing">text lipsa</span>  <br />  <span class="user">elenacarmen</span>  <span class="seq">53787</span>  <i>Ĺźi Ă&#174;n jos e popuĹźoi.</i>  <span class="c1">  ib.  IX, 139  </span>  .  <i>  StrĂ&#174;nsul pĂ&#174;nei albe (a  <br />  *(text lipsa  Ĺźi Ă®n jos e popuĹźoi. ib. IX, 139 . StrĂ®nsul pĂ®nei albe (a )* | JTIDY |
| 045.html | NR randuri: 1273 | NR randuri: 1078 | NR randuri: 4813 | JTIDY |
| 046.html | NR randuri:685 | NR randuri:609 | NR randuri:2418 | JTIDY |
| Exemplu | <html>  <body>  <p> AAA  AAA  </body> | <html>  <head>  <meta name="generator" content="HTML Tidy for Java (vers. 2009-12-01), see jtidy.sourceforge.net" />  <title></title>  </head>  <body>  <p>AAA AAA</p>  </body>  </html> | <?xml version="1.0" encoding="utf-8"?>  <html>  <head />  <body>  <p>  AAA  AAA  </p>  </body>  </html> | HTMLCLEANER |
| Exemplu | <body>  <p> <i> AAA <b> AAA  </body> | <body>  <p><i>AAA <b>AAA</b></i></p>  </body> | <body>  <p>  <i>  AAA  <b>  AAA  </b>  </i>  </p>  </body> | = |
| exemplu | <p> <i> <b>BBBBBBB<i> AAA <b> AAA </i>  EXPECTATION:  <p> <i> <b>BBBBBBB </b></i> <i> AAA <b> AAA </i> | <p><i><b>BBBBBBB <i>AAA <b>AAA</b></i></b></i></p> | <p>  <i>  <b>  BBBBBBB  <i>  AAA  <b>AAA</b>  </i>  <b>  </b>  </b>  </i>  </p> | JTIDY |
| Exemplu | <p> <i> <b>BBBBBBB </i><i> AAA <b> AAA </i> | <p><i><b>BBBBBBB</b></i> AAA <b>AAA</b></p> | <p>  <i>  <b>BBBBBBB</b>  </i>  <b>  <i>  AAA <b>AAA</b>  </i>  <b>  </b>  </b>  </p> | JTIDY |

Anexă 2

Tabel 2. Test 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Filename | Original Text | Converted text with JTIDY | Converted text with HTMLCleaner | Best Choice |
| 043.html | <span class="c1">C. PETRESCU,  C. V. 42</span>. <i>Pereţii scorojiţi<br/>  abia suportau afişele prinse cu pioneze.</i>  *C. PETRESCU, C. V. 42. Pereţii scorojiţi* | <span class="c1">C. PETRESCU, C. V. 42</span>  <p>. <i>Pereţii scorojiţi<br />  abia suportau afişele prinse cu pioneze.</i>  *C. PETRESCU, C. V. 42*  *. Pereţii scorojiţi* | <span class="c1">  C. PETRESCU,  C. V. 42  </span>  .  <i>  Pereţii scorojiţi  <br />  abia suportau afişele prinse cu pioneze.  </i>  *C. PETRESCU, C. V. 42 . Pereţii scorojiţi* | **HTML**  **Cleaner** |
| 043.html | <br />p</i><i>ioneză .</i>  *pioneză .* | <br />p</i><i>ioneză .</i>  *pioneză .* | <br />  p  </i>  <i>ioneză .</i>  *p ioneză .* | **JTIDY** |
| 043.html | <p><b>    PÍXIS</b> s. n. (Maghiarism învechit)  Tabacheră.<i>Tabachieriu</i></p>  *Tabacheră.Tabachieriu* | <p><b>&nbsp;&nbsp;&nbsp;&nbsp;PÍXIS</b> s. n. (Maghiarism învechit) Tabacheră.<i>Tabachieriu</i></p>  *Tabacheră.Tabachieriu* | <p>  <b>&#160;&#160;&#160;&#160;P&#205;XIS</b>  s. n. (Maghiarism &#238;nvechit)  Tabacheră.  <i>Tabachieriu</i>  </p>  *Tabacheră. Tabachieriu* | **JTIDY** |
| 043.html | NR randuri: 759 | NR randuri: 656 | NR randuri: 2822 | **JTIDY** |
| 044.html | <p><b>    PIZẮR</b> s<small>.</small> m. v.  <b>pisar</b>.</p>  ***PIZẮR****s. m. v.****pisar****.* | <p><b>&nbsp;&nbsp;&nbsp;&nbsp;PIZẮR</b> s<small>.</small> m. v. <b>pisar</b>.</p>  ***PIZẮR****s. m. v.****pisar****.* | <p>  <b>&#160;&#160;&#160;&#160;PIZẮR</b>  s  <small>.</small>  m. v.  <b>pisar</b>  .  </p>  ***PIZẮR****s . m. v.****pisar****.* | **JTIDY** |
| 044.html | <i>Rîvniţi şi p<span class="eroare">í</span></i><i>zmiţi şi nu veţi putea  dobîndi.<br/></i>  *Rîvniţi şi pízmiţi şi nu veţi putea dobîndi.* | <i>Rîvniţi şi p<span class="eroare">í</span></i><i>zmiţi şi nu veţi putea dobîndi.<br />  </i>  *Rîvniţi şi pízmiţi şi nu veţi putea dobîndi.* | <i>  R&#238;vniţi şi p  <span class="eroare">&#237;</span>  </i>  <i>  zmiţi şi nu veţi putea  dob&#238;ndi.  <br />  </i>  *Rîvniţi şi p í zmiţi şi nu veţi putea dobîndi.* | **JTIDY** |
| 044.html | NR randuri: 846 | NR randuri: 748 | NR randuri: 3078 | **JTIDY** |
| 045.html | <span class="c3">p o  k l o s</span> „a umple<br/>  de lepră”.<br/>  *p o k l o s „a umple de lepră”.* | <span class="c3">p o k l o s</span> "a umple<br />  de lepră".<br />  *p o k l o s "a umple de lepră".* | <span class="c3">  p o  k l o s  </span>  &#8222;a umple  <br />  de lepră&#8221;.  <br />  *p o k l o s „a umple  de lepră”.* | **HTMLCleaner** |
| 045.html | <i>Iată, pe masa divanului,  dovezi făr</i><span class="eroare">ă</span> <i>greş a unor<br/>  *Iată, pe masa divanului, dovezi făr*ă *greş a unor* | <i>Iată, pe masa divanului, dovezi făr</i><span class="eroare">ă</span> <i>greş a unor<br />  *Iată, pe masa divanului, dovezi făr*ă *greş a unor* | <i>  Iată, pe masa divanului,  dovezi făr  </i>  <span class="eroare">ă</span>  <i>  greş a unor  <br />  *Iată, pe masa divanului, dovezi făr*ă *greş a unor* | **JTIDY** |
| 045.html | NR randuri: 1274 | NR randuri: 1080 | NR randuri: 4814 | **JTIDY** |
| 046.html | NR randuri: 686 | NR randuri: 611 | NR randuri: 2419 | **JTIDY** |