

The background is a dark blue gradient with a perspective view of a digital landscape. It features a grid of glowing white lines and dots, resembling a network or data flow. Scattered throughout are various sizes of white binary digits (0s and 1s), some appearing to float in the air. The overall effect is a sense of depth and digital connectivity.

# **ЕЛЕКТРОННІ ІНФОРМАЦІЙНІ РЕСУРСИ: СТВОРЕННЯ, ВИКОРИСТАННЯ, ДОСТУП**

**ЗБІРНИК МАТЕРІАЛІВ**

**Міжнародної науково-практичної Інтернет-конференції**

**9-10 листопада 2020 р.**

**Міністерство освіти і науки України**

**Вінницький національний технічний університет**

**Національна академія Державної прикордонної служби України**

**ім. Богдана Хмельницького**

**Вінницький національний медичний університет ім. М.І. Пирогова**

**Комунальний заклад вищої освіти «Вінницька академія безперервної освіти»**

**Комунальний заклад «Сумський обласний інститут післядипломної педагогічної освіти»**

**Люблінська політехніка (Польща)**

**Новий університет Лісабону (Португалія)**

# **«ЕЛЕКТРОННІ ІНФОРМАЦІЙНІ РЕСУРСИ: СТВОРЕННЯ, ВИКОРИСТАННЯ, ДОСТУП»**

**ЗБІРНИК МАТЕРІАЛІВ**

**Міжнародної науково-практичної Інтернет-конференції**

**9-10 листопада 2020 р.**

**Суми/Вінниця  
НІКО/ВНТУ  
2020**

**УДК 004**  
**ББК 32.97**  
**Е50**

Рекомендовано до видання Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 5 від 26.11.2020 р.)

**Електронні інформаційні ресурси: створення, використання, доступ:**  
Збірник матеріалів Міжнародної науково-практичної Інтернет конференції 9-10 листопада 2020 р. – Суми/Вінниця : НІКО/ВНТУ, 2020. – 280 с.

**ISBN 978-617-7422-13-5**

Збірник містить матеріали Міжнародної науково-практичної Інтернет-конференції «Електронні інформаційні ресурси: створення, використання, доступ».

Матеріали збірника подано у авторській редакції. Автори опублікованих матеріалів несуть повну відповідальність за підбір, точність наведених фактів, цитат, статистичних даних, власних імен та інших відомостей, Матеріали відтворюються зі збереженням змісту, орфографії та синтаксису текстів, наданих авторами.

**УДК 004**  
**ISBN 978-617-7422-13-5**

**© Вінницький національний технічний університет, 2020**

**© Вид-во Суми, НІКО, 2020**

Кащенко Н.В.

**ВИКОРИСТАННЯ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ  
ТЕХНОЛОГІЙ ВЧИТЕЛЯ З МИСТЕЦЬКИХ ДИСЦИПЛІН В  
ОСВІТНЬОМУ ПРОЦЕСІ..... 109**

Коваленко О.О., Корягіна Д.О.

**ВИКОРИСТАННЯ АЛГОРИТМІВ У БІБЛІОТЕКАХ МОВ  
ПРОГРАМУВАННЯ..... 119**

Коваленко О.В., Марущак А.В., Шмалюх В.А.

**РОЗВИТОК СТАНДАРТІВ СТВОРЕННЯ ПРОГРАМНИХ ПРОДУКТІВ В  
СВІТІ ТА В УКРАЇНІ..... 126**

Коваленко О.В., Сагайдак Л.Л.

**УДОСКОНАЛЕННЯ МЕТОДІВ ТА МОДЕЛЕЙ КОМПЛЕКСНОЇ  
КОМУНІКАЦІЙНОЇ СИСТЕМИ ДЛЯ МІСЬКОГО ТУРИЗМУ..... 131**

Коваленко О.В., Черначук Н.В.

**РОЗРОБКА СИСТЕМИ МОНИТОРИНГУ ТА АВТОМАТИЗОВАНОГО  
УПРАВЛІННЯ УЧАСНИКАМИ “BLOCKCHAIN” МЕРЕЖІ..... 133**

Козловський А.Ю., Хошаба О.М.

**ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА СИНТАКСИЧНИХ  
АНАЛІЗАТОРІВ HTML ТА XML ФОРМАТІВ ДОКУМЕНТІВ..... 136**

Козьмін В.О., Майданюк В.П.

**ВИКОРИСТАННЯ МЕХАНІЗМУ WEB WORKERS API У СУЧАСНИХ  
БРАУЗЕРАХ..... 140**

Конфедрат Ю.Ю.

**ДИСТАНЦІЙНЕ НАВЧАННЯ НА GOOGLE MEET..... 142**

Коробейнікова Т.І., Мичуда Л.З., Савицька Л.А.

**ЗАСІБ СТВОРЕННЯ КАТАЛОГУ ЕЛЕКТРОННИХ ПОСИЛАНЬ НА  
ОСНОВІ КОРИСТУВАЦЬКОЇ СИСТЕМИ КАТЕГОРИЗАЦІЇ..... 144**

Лебідь О.В.

**ІКТ-КОМПЕТЕНТНІСТЬ ПЕДАГОГА В СИСТЕМІ НЕПЕРЕРВНОЇ  
ОСВІТИ ПОЛЬЩІ..... 151**

**ЕЛЕКТРОННІ ІНФОРМАЦІЙНІ РЕСУРСИ:  
СТВОРЕННЯ, ВИКОРИСТАННЯ, ДОСТУП:**

Збірник матеріалів

Міжнародної науково-практичної Інтернет-конференції  
9-10 листопада 2020 р.

Редактор С.А.Пойда, Н.А. Ніколаєнко  
Комп'ютерне верстання С.А.Пойда, М.С. Ніколаєнко

Підписано до друку 01.11.2020 Гарнітура Times New Roman  
Формат 60x84/16 Папір офсетний  
Друк цифровий Ум. друк. арк. 16,3  
Тираж 300 пр. Зам. № 2/20

Видавництво НІКО  
м.Суми, вул.Харківська, 54  
Свідоцтво про внесення до Державного реєстру  
суб'єктів видавничої справи України  
серія СМв № 044  
від 15.10.2012  
E-mail: [ms.niko@i.ua](mailto:ms.niko@i.ua)  
Телефон для замовлень: +38(066) 270-64-68

### Список використаної літератури

1. Усе про біткойни та криптовалюти [Електронний ресурс] Режим доступу до ресурсу: [https://krc.ua/blog/use\\_pro\\_bitkoini\\_ta\\_kriptovalyuti.html](https://krc.ua/blog/use_pro_bitkoini_ta_kriptovalyuti.html)
2. Криптовалюта [Електронний ресурс] Режим доступу до ресурсу: <https://bit.ly/3qruX1E>
3. Метод порівняння [Електронний ресурс] Режим доступу до ресурсу: [studopedia.com.ua/1\\_27319\\_metod-porivnyannya.html](http://studopedia.com.ua/1_27319_metod-porivnyannya.html)

*Козловський А.Ю.,  
студент, Вінницький національний технічний університет  
Хошаба Олександр Мирославович,  
кандидат технічних наук,  
доцент кафедри програмного забезпечення,  
Вінницький національний технічний університет*

### ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА СИНТАКСИЧНИХ АНАЛІЗАТОРІВ HTML ТА XML ФОРМАТІВ ДОКУМЕНТІВ

*В роботі виконано розгляд синтаксичних аналізаторів HTML та XML форматів документів, де надані особливості роботи деяких поширених програмних засобів. Показані приклади їх використання та визначено найбільш функціонально розвинутого аналізатора серед розглянутих програмних засобів.*

В даній роботі порівняльна характеристика синтаксичних аналізаторів HTML та XML форматів документів виконується з метою розробки програмних систем визначення якості веб-сайтів за критеріями користувачів з вадами зору.

HTML документи складають серцевину Інтернету. Всі сторінки, які користувач використовує в Інтернеті, є HTML кодом, незалежно від того, динамічно вони генеруються за допомогою JavaScript, JSP, PHP, ASP або будь-якої іншої веб-технології.

Браузер за своїм призначенням аналізує HTML код і робить його інформативним для користувача. Для цього існують певні прикладні бібліотеки, що використовуються мовами високого рівня, які аналізують HTML код. При цьому, виконується аналіз всього HTML документ де знаходяться існуючі елементи, теги та атрибути [1].

Якщо використовувати мову програмуванням Java, то існує можливість

синтаксичних аналізу HTML та XML документів за допомогою аналізаторів, таких як DOM і SAX [2].

Разом з тим, необхідно зазначити, що у базовій бібліотеки JDK немає бібліотеки з використання синтаксичних аналізаторів HTTP або HTML. Тому, у випадку необхідності використання програмних засобів синтаксичний аналізу файлу HTML, то багатьом розробникам доводиться виконувати пошук у Google, щоб дізнатись, як отримати значення тегів HTML у мові програмування Java.

Також, існують декілька прикладних бібліотек з вирішення даної задачі. До найпоширеніших відноситься JSoup. У порівнянні з іншими прикладними бібліотеками, JSoup не тільки забезпечує підтримку читання та синтаксичного аналізу HTML-документів, але й дозволяє визначати будь-який елемент HTML-файлу, їх атрибут, клас CSS у стилі JQuery, та виконувати їх модифікації.

Розглянемо синтаксичні аналізатори HTML документів більш детально.

NekoHTML - це простий HTML-сканер та балансир тегів, який дозволяє програмістам аналізувати документи HTML та отримувати доступ до інформації за допомогою стандартних інтерфейсів XML [3,4]. Синтаксичний аналізатор може сканувати HTML-файли та "виправляти" багато типових помилок, які користувачі припускають при написанні документів HTML. NekoHTML може додавати відсутні батьківські елементи, автоматично закривати елементи за допомогою необов'язкових кінцевих тегів та обробляти невідповідні вбудовані теги елементів.

Але під час використання NekoHTML існують певні обмеження, які полягають в наступному. Є документи HTML, для яких NekoHTML не може належним чином обробити добре сформований потік подій XML-документів. Наприклад, документи з декількома тегами `<html>` за своєю суттю неправильно сформовані, оскільки документи XML можуть мати лише один кореневий елемент.

Програмний код, що доданий до базової реалізації DOM в Xerces-J 2.0.1, вносить помилку в реалізацію HTML DOM на його основі. Ця помилка призводить до того, що вузли елементів у результуючому об'єкті документа HTML мають тип `org.apache.xerces.dom.ElementNSImpl` замість відповідних об'єктів HTML DOM. Така проблема зачіпає користувачів NekoHTML, які використовують синтаксичний аналізатор із Xerces-J 2.0.1 та всіх, хто використовує реалізацію HTML DOM у Xerces-J 2.0.1.

Інших основних обмежень у цьому програмному продукту немає. У подальшому, розробники виконують додаткову роботу, щоб покращити продуктивність, виправити помилки та додати функціональність.

Приклад використання NekoHTML може бути наступним:

```

final DOMParser parser = new DOMParser();
try {
    parser.parse(new InputSource(urlIS));
    document = parser.getDocument();
} catch (SAXException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

```

TagSoup - це безкоштовне програмне забезпечення з відкритим кодом [5,6]. Починаючи з версії 1.2, він ліцензований за ліцензією Apache, версія 2.0, яка дозволяє повторне використання, а також використання з проектами GPL 3.0 або GPL 2.0 або новішої версії.

Синтаксичний аналізатор TagSoup вважається SAX-сумісним, що написаний на мові високого рівня Java. За допомогою інтерфейсу SAX, він дозволяє застосовувати стандартні інструменти XML до складного HTML коду. TagSoup також використовує процесор командного рядка, який читає файли HTML і може генерувати чистий HTML код або добре сформований XML код, що є близьким наближенням до XHTML.

Приклад використання NekoHTML може бути наступним:

```

final Parser parser = new Parser();
SAX2DOM sax2dom = null;
try {
    sax2dom = new SAX2DOM();
    parser.setContentHandler(sax2dom);
    parser.setFeature(Parser.namespacesFeature, false);
    parser.parse(new InputSource(urlIS));
} catch (Exception e) {
    e.printStackTrace();
}
document = sax2dom.getDOM();

```

HTML Tidy існує у версіях комп'ютерної програми і бібліотека [7,8]. Метою бібліотеки jTidy є виправлення аналіз HTML коду та у разі необхідності, поліпшення зовнішнього вигляду і стилю відступів отриманої розмітки HTML документу.

HTML Tidy був розроблений Дейвом Реггеттом з Консорціуму Всесвітньої павутини (W3C), потім перенесений до проекту SourceForge. Її вихідний код написаний на ANSI C для максимальної переносимості та скомпільований у виконавчі файли, що доступні для різних платформ, в тому



числі і для Java.

HTML Tidy доступна за ліцензією W3C, нові версії у вигляді вихідного коду, а також скомпільованих пакетів для різних операційних систем доступні через репозиторій GitHub.

Разом з тим, у разі необхідності HTML Tidy може внести такі модифікації в HTML код:

- виправляти змішані теги;
- виправляти пропущені або некоректні теги що закриваються;
- додавати пропущені елементи (деякі теги, лапки та інше);
- повідомляти про пропрієтарні HTML-розширення, що можуть змінити макет розмітки на визначений стиль;
- перетворити символи з деяких кодувань в HTML-послідовності.

Приклад використання jTidy може бути наступним:

```
final Tidy tidy = new Tidy();
tidy.setQuiet(true);
tidy.setShowWarnings(false);
tidy.setForceOutput(true);
document = tidy.parseDOM(urlIS, null);
```

HtmlCleaner - це синтаксичний аналізатор документів Html на мові Java з відкритим кодом [9,10]. HtmlCleaner здатний реорганізувати кожен елемент документа HTML і обробляти добре сформований документ HTML.

За замовчуванням HtmlCleaner дотримується правил, подібних до тих, що використовуються більшістю веб-браузерів для створення об'єктної моделі HTML документа. Однак розробники можуть надавати власні теги та групи правил для фільтрації та аналізу.

Приклад використання HtmlCleaner може бути наступним:

```
final HtmlCleaner cleaner = new HtmlCleaner(urlIS);
try {
    cleaner.clean();
    document = cleaner.createDOM();
} catch (Exception e) {
    e.printStackTrace();
}
```

*Висновки.* Таким чином, серед розглянутих синтаксичних аналізаторів HTML та XML форматів документів перевагу у роботі слід віддати HtmlCleaner аналізатору.

### Список використаної літератури

1. Difference between XML and HTML [Електронний ресурс] – Режим доступу до

- статті: <https://www.guru99.com/xml-vs-html-difference.html>
2. A Lexical Analyzer for HTML and Basic SGML [Електронний ресурс] – Режим доступу до статті: <https://www.w3.org/MarkUp/SGML/sgml-lex/sgml-lex>
  3. CyberNeko HTML Parser [Електронний ресурс] – Режим доступу до статті: <http://nekohtml.sourceforge.net/>
  4. codelibs/nekohtml [Електронний ресурс] – Режим доступу до статті: <https://github.com/codelibs/nekohtml>
  5. TagSoup - Just Keep On Truckin [Електронний ресурс] – Режим доступу до статті: <http://vrici.lojban.org/~cowan/XML/tagsoup/>
  6. jukka/tagsoup [Електронний ресурс] – Режим доступу до статті: <https://github.com/jukka/tagsoup>
  7. ABOUT JTIDY [Електронний ресурс] – Режим доступу до статті: <http://jtidy.sourceforge.net/>
  8. jtidy/jtidy [Електронний ресурс] – Режим доступу до статті: <https://github.com/jtidy/jtidy>
  9. HtmlCleaner [Електронний ресурс] – Режим доступу до статті: <http://htmlcleaner.sourceforge.net/>
  10. How to useHtmlCleaner [Електронний ресурс] – Режим доступу до статті: <https://www.codota.com/code/java/classes/org.htmlcleaner.HtmlCleaner>

**УДК 004.422.83**

***Козьмін Віталій Олександрович,**  
студент факультету інформаційних технологій та комп'ютерної інженерії,  
Вінницький національний технічний університет,  
**Майданюк Володимир Павлович,**  
кандидат технічних наук,  
доцент кафедри програмного забезпечення,  
Вінницький національний технічний університет*

## **ВИКОРИСТАННЯ МЕХАНІЗМУ WEB WORKERS API У СУЧАСНИХ БРАУЗЕРАХ**

*Розглянуто процес виконання «важких» операцій у браузерних переглядачах. Проведено аналіз стану основного потоку виконання під час виконання «важких» операцій на клієнтській частині веб-додатків. Запропоновано винесення «важких» обчислювальних операцій у окремий потік*