УДК 004

О.В. МАЗУРЕЦЬ, О.В. КОВАЛЬЧУК, В.О. СЛОБОДЗЯН
Хмельницький національний університет

ВИКОРИСТАННЯ СПЕЦІАЛІЗОВАНИХ ПРОГРАМНИХ РОЗШИРЕНЬ ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ З ЦИФРОВИМИ ДОКУМЕНТАМИ НАВЧАЛЬНИХ МАТЕРІАЛІВ

В статті досліджено проблему автоматизації роботи з електронними документами навчальних матеріалів у задачах визначення структури змістовних блоків у електронному документі навчального матеріалу та пошуку ключових термінів у контенті навчального матеріалу. Встановлено, що для реалізації програмної обробки електронних документів є доцільним використання спеціалізованих програмних комплексів, що надають необхідний інструментарій для програмної роботи з контентом відповідних файлів. З метою визначення найбільш ефективного програмного розширення проведено аналіз сучасних існуючих спеціалізованих програмних розширень для автоматизації обробки електронних документів: Microsoft.Office.Interap.Word.dll, DocumentFormat.OpenXml.dll та Spire.Doc.dll. За результатами аналізу встановлено, що бібліотека Microsoft.Office.Interap.Word.dll надає доступ до всіх функцій MS Office, так як працює безпосередньо з PIA, для її роботи потрібна ліцензія для MS Office на кожному клієнтському комп'ютері, MS Office завантажується у фоновому режимі, внаслідок чого займає значний обсяг оперативної пам'яті. При роботі з docx-файлом за допомогою DocumentFormat.OpenXml.dll не потрібна наявність MS Office та запуск в фоновому режимі, проте зростає складність програмного використання бібліотеки внаслідок необхідності регулярного співставлення ідентифікатора стиля з контейнером властивостей стилів. Перевагою Spire.Doc.dll визначено реалізацію функцій автоматичного співставлення стилів текстових блоків їх властивостям на рівні розширення. В результаті аналізу розглянутих бібліотек було визначено Spire.Doc.dll оптимальним варіантом для використання в автоматизації обробки електронних документів. Встановлено, що перенесення функцій автоматичного співставлення стилів текстових блоків їх властивостям з рівня функціоналу програмного коду застосунка на рівень функціоналу бібліотеки дозволяє спростити як роботу застосунка з електронним документом, так і процес програмування. Розглянуто практичні особливості використання розширення Spire.Doc.dll при роботі з електронними документами .DOCX. З метою визначення можливості застосування обраного інструменту в задачах роботи з електронними документами було проведено його практичну апробацію в складі програмної системи для вирішення прикладної задачі побудови структури електронного документу та пошуку ключових слів у його контенті. В результаті встановлено можливість та достатню ефективність використання розширення Spire.Doc.dll для роботи з електронними документами навчальних матеріалів.

Ключові слова: електронний документ, цифровий документ, навчальні матеріали, ключові терміни, Microsoft.Office.Interap.Word, DocumentFormat.OpenXml, Spire.Doc.

O.V. MAZURETS, O.V. KOVALCHYK, V.O. SLOBODZIAN
Khmelnytsky National University

USING SPECIALIZED SOFTWARE PACKEGES FOR AUTOMATION OF WORK WITH DIGITAL DOCUMENTS OF EDUCATIONAL MATERIALS

At article investigated a problem of the automation work with electronic documents of educational materials in the tasks for determining the structure of content blocks at electronic documents educational material and search of key terms in content of educational materials. It is established that for software processing of electronic documents need to use specialized program components that provide the necessary tools for work with content of files. Microsoft.Office.Interop.Word.dll, DocumentFormat.OpenXml.dll and Spire.Doc.dll were using for choose the most effective program components for automation work with electronic documents. In process audit established that Microsoft.Office.Interap.Word.dll provide access to all function of MS Office because it works with PIA. This library needs a license for MS Office at every computers of client and MS Office loaded in the background, in result it occupies many RAM. DocumentFormat.OpenXml.dll uses for work with docx file. This component does not need to install MS Office and work in the background but the complexity of use this library is increasing to the need to regularly match the style ID with the styles properties container. The advantage of Spire Doc.dll is the implementation of automatic functions for matching of style of text blocks to their properties. In result audit this libraries there was chosen Spire.Doc.dll for use to automation of processing of electronic documents. It is established that in result transfer of functions for automatic matching styles of text block to their properties from the level of the functional code of the application to the level of the functional library simplify work of the application with electronic document and programming process. There was considered practical features of Spire.Doc.dll for working with electric documents ".DOCX". There was practical use made this tool in program system for determine its the possibilities at using in the tasks for work with electronic documents. The system is designed to build the structure of an electronic document and search keywords in its content. Explored the possibility and effectiveness of using the Spire.Doc.dll library to work with electronic educational materials documents.

Keyword: digital document, electronic document, key terms, educational materials, Microsoft.Office.Interap.Word, DocumentFormat.OpenXml, Spire.Doc.

Постановка проблеми в загальному вигляді

Автоматизація вирішення ряду задач у галузі сучасної вищої освіти (оцінка відповідності навчальних матеріалів вимогам, оцінка відповідності наборів тестових завдань навчальним матеріалам, автоматизована генерація прототипів тестових завдань, реалізація гнучких алгоритмів тестування, допомога та контроль якості при формуванні навчальних матеріалів, допомога та контроль якості при формуванні тестів до навчальних матеріалів, автоматизація формування рефератів та анотацій до елементів навчальних матеріалів тощо) забезпечується через застосування онтології як методу формального опису знань, що містяться в навчальних матеріалах. Модель онтології навчального матеріалу може складатися з ключових

слів, ключових термінів, структури навчального матеріалу, атрибутів ключових слів та ключових термінів, що визначають їх властивості та забезпечують прив'язку до елементів структури навчального матеріалу [1]. За такої моделі, онтологія навчального матеріалу є методом виявлення сенсу навчального матеріалу та засобом вирішення наведеного ряду практичних задач.

Основними з етапів побудови онтології навчального матеріалу є пошук ключових термінів у контенті навчального матеріалу та побудова його логічної структури. Вхідними даними є електронний документ навчального матеріалу. Для автоматизації виконання обох наведених етапів потрібна програмна обробка відповідних цифрових файлів (зазвичай формату .DOCX). Для ефективної реалізації програмної обробки цифрових документів є доцільним використання спеціалізованих програмних комплексів, що надають об'єктно-орієнтований інструментарій для програмної роботи з контентом відповідних файлів.

Аналіз останніх досліджень

Проблему автоматизації побудови логічної структури навчального матеріалу (наприклад: Дисципліна / Розділ / Тема) пропонується вирішувати шляхом визначення ієрархії змістовних блоків у цифровому документі [1], таким чином формуючи верхній рівень вертикальної онтології відповідної навчальної дисципліни. Проблему пошуку ключових термінів у контенті навчального матеріалу пропонується вирішувати шляхом використання відповідної інформаційної технології на основі дисперсійної оцінки [2], таким чином формуючи нижній рівень онтології навчальної дисципліни.

Визначено, що онтологія навчального матеріалу розглянутої моделі може бути методом виявлення сенсу навчального матеріалу й ефективно використана в розробці інформаційних технологій для роботи з навчальними матеріалами [1]. Зокрема, з застосуванням такої моделі онтології навчального матеріалу було розв'язано ряд практичних задач: оцінка відповідності навчальних матеріалів вимогам [3], оцінка відповідності наборів тестових завдань навчальним матеріалам [4], автоматизована генерація прототипів тестових завдань [5], реалізація гнучких алгоритмів тестування [6], автоматизація формування рефератів та анотацій до елементів навчальних матеріалів [7] та інші.

Для реалізації програмної обробки цифрових документів є доцільним використання розглянутих в даній статті спеціалізованих програмних комплексів, що надають необхідний інструментарій для програмної роботи з контентом відповідних файлів, зокрема: Microsoft.Office.Interap.Word.dll [8], DocumentFormat.OpenXml.dll [9], Spire.Doc.dll [10].

Постановка задачі

Метою статті є аналіз сучасних існуючих спеціалізованих програмних розширень для автоматизації обробки цифрових документів і визначення найбільш ефективного програмного розширення. Методом визначення можливості застосування обраного інструменту в задачах роботи з цифровими документами є його практична апробація в складі програмної системи для вирішення відповідних прикладних задач.

Викладення основних матеріалів дослідження

На сучасному етапі для зберігання електронних документів навчальних матеріалів використовується файл з розширенням .DOCX (або створених на його основі .HTML, .PDF тощо). На відміну від файлів .DOC, які зберігають дані документа в одному бінарному файлі, файли .DOCX створюються за допомогою відкритого формату XML, в якому зберігаються документи як колекції окремих файлів і папок в стиснутому пакеті. .DOCX-файли містять .XML-файли і три папки, docProps, Word, і _rels (Рис. 1а), які містять властивості документа, його зміст (Рис. 1б) і відношення між файлами, тему (Рис. 1в) та включені файли (Рис. 1г). .DOCX-файли розроблені, щоб зробити вміст документів доступним і відкритим – так, текстовий документ чи зображення зберігаються як окремі прості файли в такій колекції у складі одного файлу .DOCX.

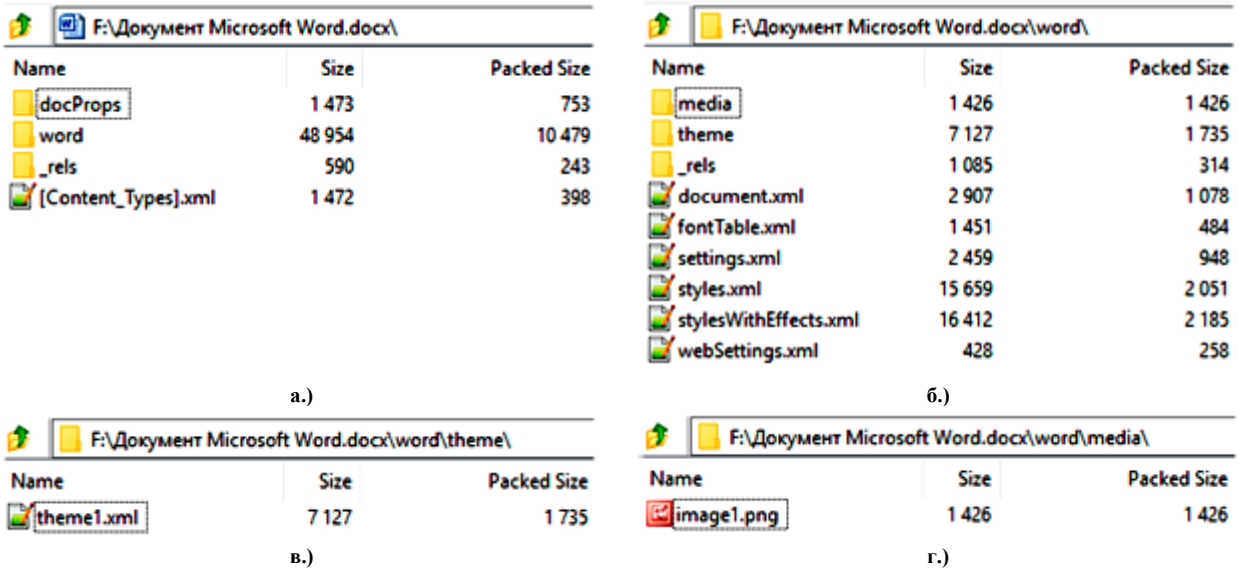


Рис. 1. Структура .DOCX-файлу

Office OpenXML є форматом електронних офісних документів, таблиць, презентацій та діаграм, розроблений компанією Microsoft. Даний формат є міжнародним стандартом, який затверджений в Міжнародній організації зі стандартизації (ISO) [11]. Він включає в себе такі первинні мови розмітки:

- WordprocessingML – для обробки текстових документів;
- SpeadsheetXL – електронних таблиць;
- PresentationML – для презентацій;
- DrawingML – для векторної графіки, діаграм.

Хоча Word реалізує свою функціональність через об'єктну модель, збереження даних відбувається у наборі XML-файлів, що ускладнює можливості для автоматизації прямого програмного парсингу [12]. Оскільки файл стилів та текст знаходяться в різних файлах у складі .XML, потрібно зчитувати одразу ряд файлів та на базі їх взаємозв'язків програмно формувати об'єктну модель документа, що є не найкращим рішенням. Тому для реалізації програмної обробки цифрових документів є доцільним використання спеціалізованих програмних комплексів, що надають об'єктно-орієнтований інструментарій для програмної роботи з контентом відповідних файлів. Найбільш відомими з таких спеціалізованих програмних комплексів є розширення Microsoft.Office.Interap.Word.dll [8], DocumentFormat.OpenXml.dll [9] та Spire.Doc.dll [10].

Microsoft.Office.Interap.Word.dll

Бібліотека Microsoft.Office.Interap.Word.dll [8] дозволяє керованому коду взаємодіяти з MS Office та об'єктною моделлю на базі COM.

За допомогою процесів автоматизації Office Automation програми, написані мовами, такими як Visual C# .NET, отримують можливість програмно керувати іншими програмами. Автоматизація Microsoft Word (далі Word) дозволяє виконувати відповідні дії, такі як створення нового документу, додавання даних у документ або створення таблиць. З програмами Word та іншими програмами Microsoft Office практично всі дії, які користувач може виконувати вручну за допомогою користувацького інтерфейсу, також можуть бути виконані програмним шляхом за допомогою автоматизації Office Automation.

Word реалізує цю програмну функціональність через об'єктну модель [13]. Об'єктна модель є набором класів і методів, які служать аналогами логічних компонентів Word. Наприклад, існує об'єкт Application, об'єкт Document та об'єкт Paragraph, кожен з яких містить функціональність цих частин Word (рис. 2).

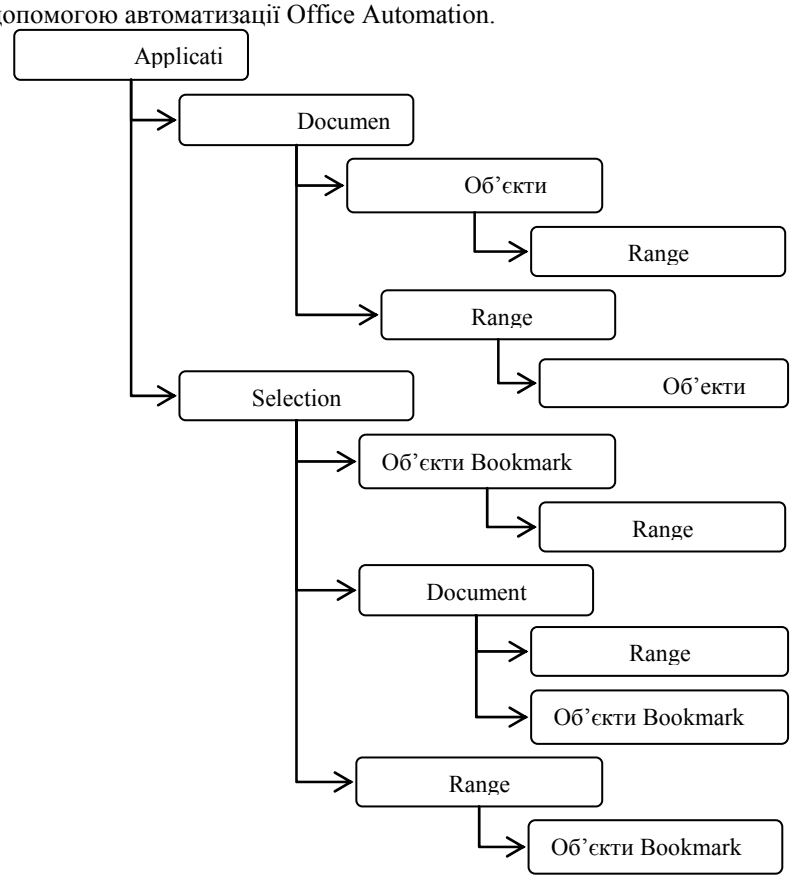


Рис. 2. Об'єктна модель Word

Щоб використовувати функції програми MS Office в проекті, можна використовувати первинний взаємозв'язок PIA (Primary Interop Assembly). PIA дозволяє керованому коду взаємодіяти з MS Office та об'єктною моделлю на базі COM. При створенні нового проекту Office, Visual Studio додає посилання на PIA, необхідний для побудови проекту. При цьому, у деяких сценаріях доводиться додавати посилання на додаткові PIA (наприклад, якщо необхідно використовувати функцію MS Office Word в проекті для MS Office Excel).

Для роботи з інтерфейсом Microsoft.Office.Interop.Word.dll необхідно встановити відповідну збірку, для чого в меню Tools потрібно обрати пункт NuGet Package Manager та запустити Package Manager Console та ввести наступну команду:

```
PM> Install-Package Microsoft.Office.Interop.Word
```

Після того як менеджер пакетів повідомить про успішне завершення встановлення, можна починати працювати з необхідними інтерфейсами.

DocumentFormat.OpenXml.dll

Бібліотека класів .NET DocumentFormat.OpenXml.dll (версії 2.0 / 2.5 / 2.8.1) [9] дає можливість розробникам програмного забезпечення працювати з пакетом Microsoft Office. Компанія надає дану бібліотеку безкоштовно в повному доступі, й її можна завантажити з офіційного сайту Microsoft або

засобами Visual Studio.

Для роботи з файлами формату .DOCX, який має зображену на рисунку 1 структуру, використовується мова розмітки WordprocessingML [14]. На рисунку 3 зображено приклад структури цієї мови розмітки.

```
<?xml version="1.0" encoding="utf-8"?>
<w:document xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main">
<w:body>
<w:p>
<w:r>
<w:t> Текст </w:t>
</w:r>
```

Рис. 3. Приклад WordprocessingML-розмітки .DOCX-документу

Теги, наведені на рис. 3, мають такі властивості:

- *document* – є основою частиною документа
- *body* – контейнер для збору структур рівня блоку, які складають основну історію;
- *p* – абзац (paragraph);
- *r* – контейнер який містить в собі текст з однаковими властивостями (run);
- *t* – власне текстовий елемент (text).

Всі інші теги, які можуть використовуватися, є допоміжними до вищенаведених і додають їм стильових властивостей.

Spire.Doc.dll

Spire.Doc.dll [10] для .NET є повністю незалежною бібліотекою класів .NET Word, спеціально створеною для розробників, й дозволяє швидко генерувати, відкривати, редагувати та зберігати документи Word різних версій. Конвертація функціональних можливостей дозволяє розробникам здійснювати перетворення Word в інші актуальні формати документів (.PDF, .EPub, .HTML, .RTF, .Image, .XML тощо). Можна створювати та обробляти вже існуючі документи Word динамічно.

Бібліотека Spire.Doc.dll забезпечує роботу з майже всіма елементами документа Word, а саме: сторінки, розділи, заголовки, колонтитули, абзаци, переліки, таблиці, текст, виноски, поля, гіперпосилання, закладки, коментарі, зображення, стилі, фонові параметри, функції друку, налаштування документа та захисту. Крім того, підтримуються графічні об'єкти, включаючи форми, текстові поля, зображення, OLE-об'єкти та елементи керування. Бібліотека Spire.Doc.dll підтримує функцію пошуку та заміни, вирівнювання, перерву сторінки, поле заповнення, об'єднання документів, копіювання документів, друк тощо.

Для роботи з бібліотекою Spire.Doc необхідно встановити відповідну збірку, для чого потрібно запустити Package Manager Console та виконати наступну команду:

```
PM> Install-Package Spire.Doc
```

Після завершення встановлення можна починати працювати з бібліотекою.

Дискусія

Хоч бібліотека Microsoft.Office.Interap.Word.dll надає доступ до всіх функцій MS Office, так як працює безпосередньо з PIA, для її роботи потрібна ліцензія для MS Office на кожному клієнтському комп'ютері. Крім того, при використанні Automation, MS Office завантажується у фоновому режимі, внаслідок чого займає деякий обсяг оперативної пам'яті та завантажує велику кількість файлів і DLL. Додатки MS Office були розроблені як додатки для користувацького інтерфейсу, і через це Microsoft.Office.Interop.Word.dll працює повільно. Microsoft не рекомендує використовувати Office Automation (або будь-який Office Interop) на сервері.

На відміну від Microsoft.Office.Interap.Word.dll, DocumentFormat.OpenXml.dll та Spire.Doc.dll не вимагають наявності MS Office на машині користувача та його запуску в фоновому режимі.

При роботі з docx-файлом за допомогою DocumentFormat.OpenXml.dll потрібно дотримуватися тієї ж ієрархії, що і сама структура розмітки тобто: *document-paragraph-run-text*. Також відповідно до розглянутих вище особливостей взаємозв'язків всередині документу, при присвоєнні параграфу деякого стилю, у властивостях параграфа надається лише ідентифікатор (ID) на даний стиль, натомість сам стиль описується окремо у файлі style.xml. Внаслідок необхідності регулярного співставлення ID стиля з контейнером style.xml для одержання характеристик стилів абзаців, зростає складність програмного використання бібліотеки.

Spire.Doc.dll не вимагає встановлення в систему кожного користувача MS Office, тобто є можливість повністю незалежної від нього роботи. Повна версія Spire.Doc.dll є платною, а безкоштовна версія FreeSpire.Doc має ряд обмежень (наприклад обробка не більше 500 абзаців і 25 таблиць) [15]. Перевагою Spire.Doc.dll визначено відсутність необхідності співставлення ID стиля з контейнером style.xml, оскільки дана функція реалізована на рівні бібліотеки.

Таким чином, в результаті аналізу сучасних існуючих спеціалізованих програмних розширень для автоматизації обробки цифрових документів, було визначено Spire.Doc.dll оптимальним варіантом для використання в автоматизації обробки цифрових документів. Основними перевагами Spire.Doc.dll встановлено відсутність необхідності наявності MS Office на машині користувача та запуску в фоновому режимі, а також реалізацію функцій автоматичного співставлення стилів текстових блоків їх властивостям на рівні розширення.

Перенесення функцій автоматичного співставлення стилів текстових блоків їх властивостям з рівня функціоналу програмного коду застосунка на рівень функціоналу бібліотеки дозволяє спростити як роботу застосунка з цифровим документом, так і процес програмування.

Програмна реалізація

При зчитуванні файлу Spire.Doc.dll перетворює документ в об'єктну модель. Ця модель має структуру, яка починається документом і закінчується об'єктом *TextRange* (рис. 4). Приклад використання такої об'єктної моделі документу MS Office показаний на рис. 5.

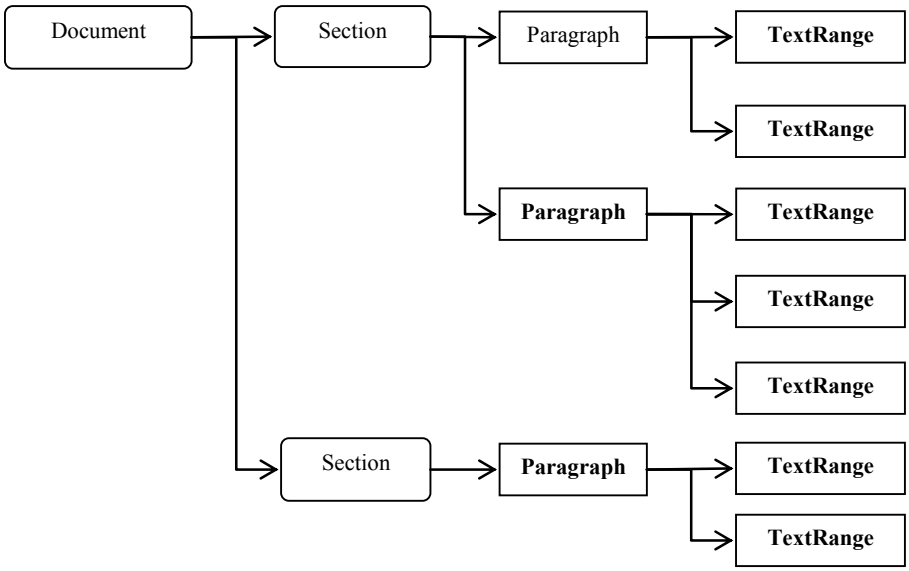


Рис. 4. Загальна структура об'єктної моделі документу

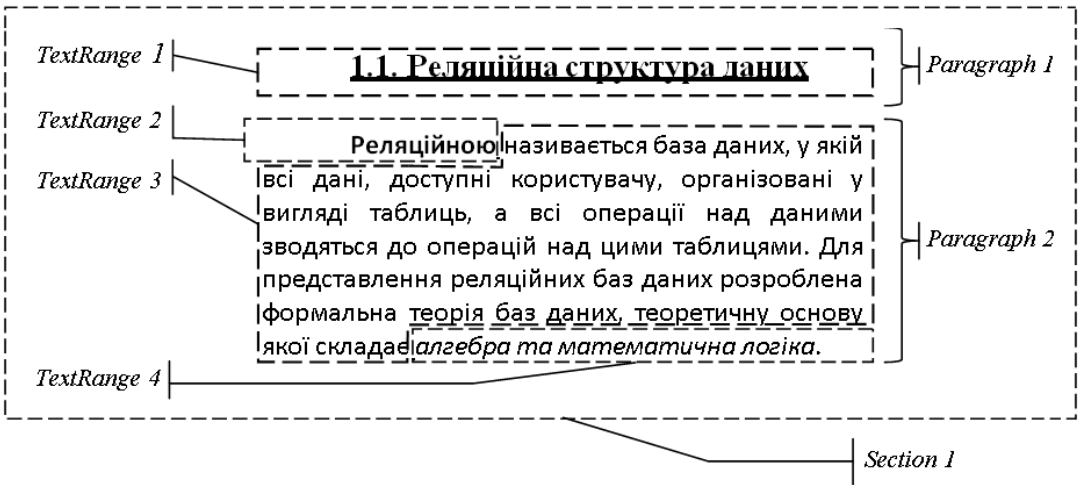


Рис. 5. Приклад використання об'єктної моделі документу MS Office

Відповідно до об'єктної моделі документу, MS Office використовує розділи (*Section*), щоб вказати частини документа, що мають різну орієнтацію сторінки, стовпці або заголовки та нижні колонтитули. Розбиття на Section дозволяє користувачеві вказати місце, де розпочинається і закінчується інше форматування. Тому Section використовуються коли потрібне використання заголовків, колонтитулів, схем, нумерації сторінок, розмірів аркушу, поля чи різні рівні захисту. Об'єкти Section містяться в об'єкті Document (рис. 5), в колекції Selections. А розділи, в свою чергу, містять в собі наступний елемент структури – абзаци (*Paragraph*).

Paragraph в MS Word визначає будь-який текст, який закінчується жорстким поверненням, яке формується, наприклад, при натисканні клавіші Enter. Формат абзацу дозволяє контролювати зовнішній вигляд фрагменту, якщо є окремі абзаци. Наприклад, можна змінити відстань між рядками або вирівнювання тексту по лівому краю на вирівнювання по центру. Можна додати індивідуальні стилі,

відступи для абзаців, або визначити заголовки і списки. В об'єктній моделі *Paragraph* знаходиться в *Document -> Sections -> Section -> Paragraphs* (рис. 6). Одержати відомості про стилі *Paragraph* можна за допомогою методу *GetStyle()* (рис. 7).

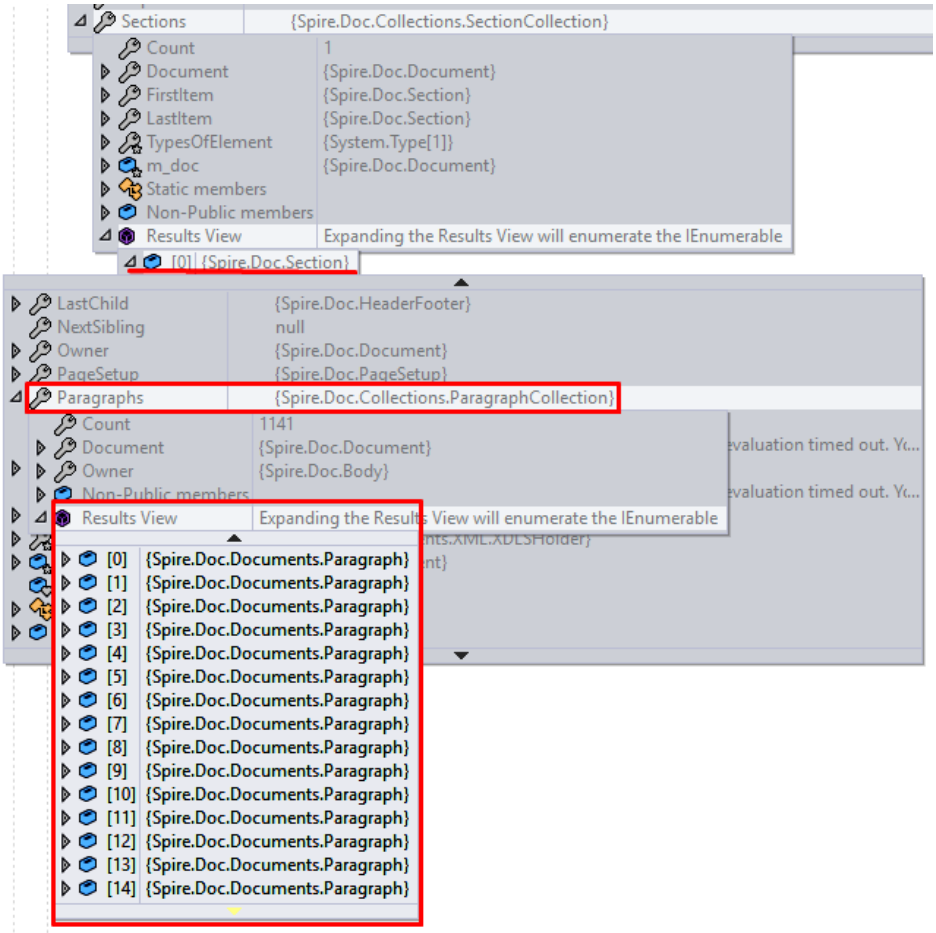


Рис. 6. Позиція визначеного об'єкта Paragraph в об'єктній моделі документа

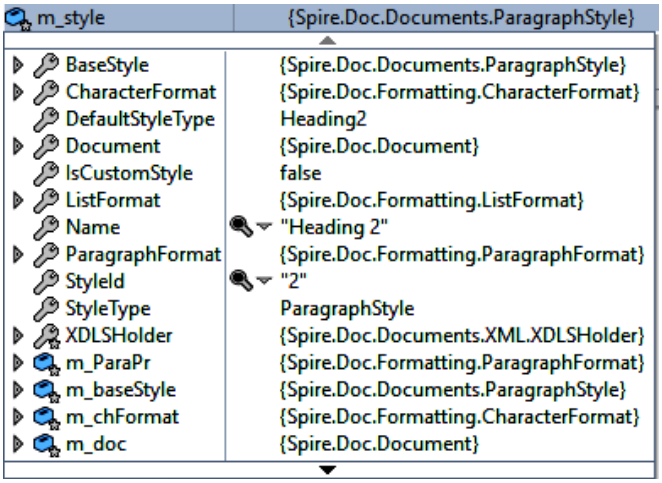


Рис. 7. Одержання відомостей про стилі Paragraph за допомогою методу GetStyle()

TextRange є найнижчим рівнем структури документу, що визначає фрагмент тексту однакового стилю. Тому для того, щоб при розв'язку прикладних задач отримати окремо кожне слово з власними властивостями стилів, потрібна розробка додаткового алгоритму для розбиття цих фрагментів *TextRange* на слова. *TextRange* знаходиться в *Paragraph -> Items*. Через роботу з відповідними властивостями, можна одержати текст (рис. 8) та стилі (рис. 9), що відносяться до визначеного *TextRange*.

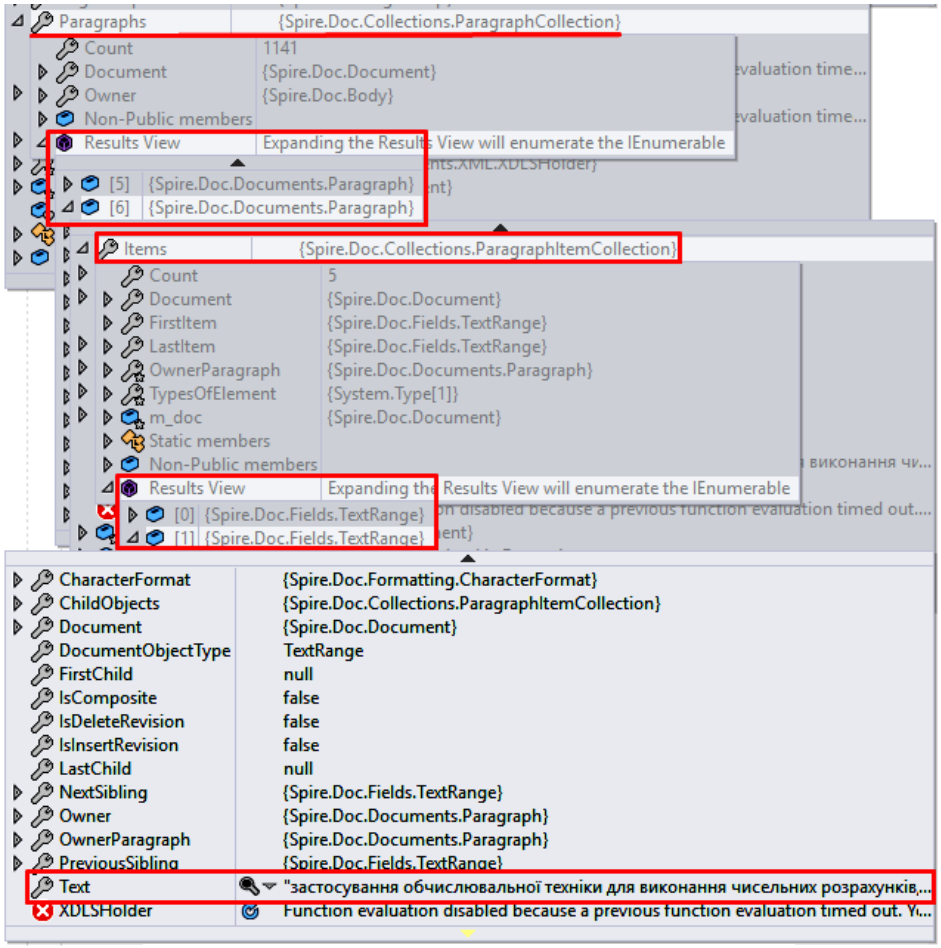


Рис. 8. Позиція визначеного тексту в TextRange у об'єктній моделі

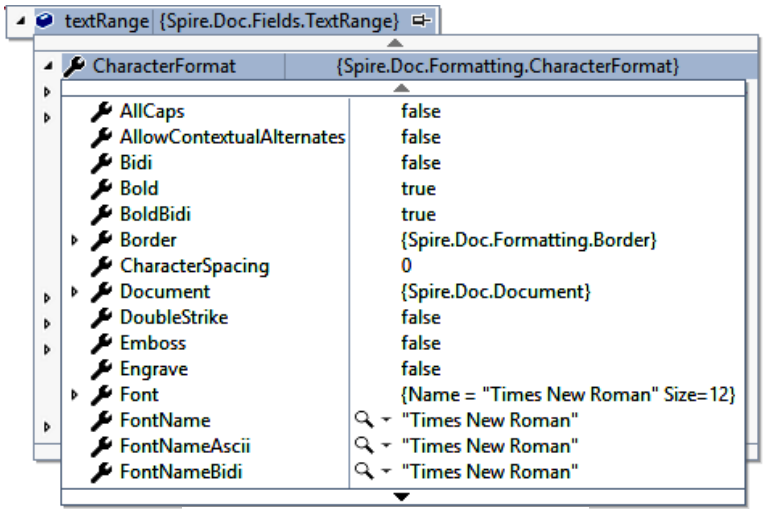
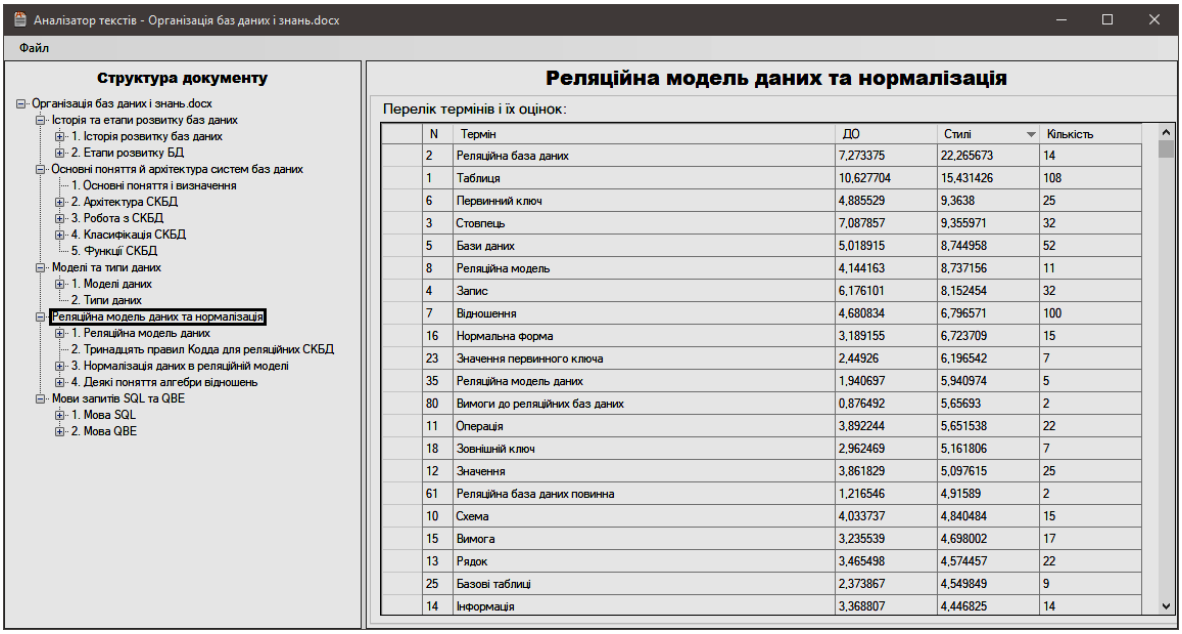


Рис. 9. Деякі стилі визначеного TextRange

Paragraph може містити лише один з кількох визначених форматів: *Level1, Level2, Level3, Level4, Level5, Level6, Level7, Level8, Level9, Body*. Відповідно до цього значення, можна визначити заголовки та їх рівні. Кожен похідний елемент має посилання на батьківський, і таким чином існує можливість визначити прив'язку певного елементу до певного модулю, заголовку чи підзаголовку при визначенні структури змістовних блоків у електронному документі навчального матеріалу.

З метою визначення можливості застосування розширення Spire.Doc.dll в задачах роботи з електронними документами, було проведено його практичну апробацію в рамках розробки програмної системи для вирішення прикладної задачі побудови структури електронного документу та пошуку ключових слів у його контенті. Зокрема, на рисунку 10 показано приклад обробки файлу «*Організація баз даних і знань.docx*» з автоматичним визначенням структури навчального матеріалу за допомогою аналізу системи заголовків та переліків ключових слів для контенту кожного елементу структури.



N	Термін	ДО	Стилі	Кількість
2	Реляційна база даних	7,273375	22,265673	14
1	Таблиця	10,627704	15,431426	108
6	Первинний ключ	4,885529	9,3638	25
3	Стовпець	7,087857	9,355971	32
5	Бази даних	5,018915	8,744958	52
8	Реляційна модель	4,144163	8,737156	11
4	Запис	6,176101	8,152454	32
7	Відношення	4,680834	6,796571	100
16	Нормальна форма	3,189155	6,723709	15
23	Значення первинного ключа	2,44926	6,196542	7
35	Реляційна модель даних	1,940697	5,940974	5
80	Вимоги до реляційних баз даних	0,876492	5,65693	2
11	Операція	3,892244	5,651538	22
18	Зовнішній ключ	2,962469	5,161806	7
12	Значення	3,861829	5,097615	25
61	Реляційна база даних повинна	1,216546	4,91589	2
10	Схема	4,033737	4,840484	15
15	Вимога	3,235539	4,698002	17
13	Рядок	3,465498	4,574457	22
25	Базові таблиці	2,373867	4,549849	9
14	Інформація	3,368807	4,446825	14

Рис. 10. Приклад результату обробки файлу навчального матеріалу з використанням розширення Spire.Doc.dll

Маючи стилі форматування *Paragraph* та *TextRange*, можна визначити заголовки, рівні заголовків, текст для аналізу чи ігнорування, прив'язку певного фрагменту тексту до заголовку.

При цьому, визначення властивостей *Paragraph* надало можливість для формування структури документу, а обробка властивостей елементів *TextRange* визначила необхідний для програмної обробки контент, що дозволило визначити множини ключових слів за допомогою методу дисперсійного оцінювання.

За результатами використання розширення Spire.Doc.dll в розробленому програмному комплексі встановлено, що дана бібліотека надає достатній інструментарій для роботи з цифровими документами навчальних матеріалів у рамках розглянутих актуальних задач, й може бути ефективно використана в реалізації інформаційних технологій для автоматизації роботи з навчальними матеріалами.

Висновки

Отже, у статті було досліджено проблему автоматизації роботи з цифровими документами навчальних матеріалів у задачах визначення структури змістовних блоків у цифровому документі навчального матеріалу та пошуку ключових термінів у контенті навчального матеріалу. Встановлено, що для реалізації програмної обробки цифрових документів є доцільним використання спеціалізованих програмних комплексів, що надають необхідний інструментарій для програмної роботи з контентом відповідних файлів.

З метою визначення найбільш ефективного програмного розширення було проведено аналіз сучасних існуючих спеціалізованих програмних розширень для автоматизації обробки цифрових документів: Microsoft.Office.Interap.Word.dll, DocumentFormat.OpenXml.dll та Spire.Doc.dll. В процесі аналізу встановлено, що бібліотека Spire.Doc.dll є оптимальним варіантом для використання при автоматизації обробки цифрових документів. Встановлено, що перенесення функцій автоматичного співставлення стилів текстових блоків їх властивостям з рівня функціоналу програмного коду застосунка на рівень функціоналу бібліотеки дозволяє спростити як роботу застосунка з цифровим документом, так і процес програмування.

Розглянуто практичні особливості використання розширення Spire.Doc.dll при роботі з цифровими документами .DOCX. З метою визначення можливості застосування обраного інструменту в задачах роботи з цифровими документами було проведено його практичну апробацію в складі програмної системи для вирішення прикладної задачі побудови структури цифрового документу та пошуку ключових слів у його контенті. В результаті встановлено можливість та достатню ефективність використання розширення Spire.Doc.dll для роботи з цифровими документами навчальних матеріалів.

Подальші дослідження спрямовані на впровадження розширення Spire.Doc.dll в розробку автоматизованих систем роботи з цифровими документами навчальних матеріалів для вирішення прикладних задач.

Література

1. Мазурець О. В. Онтологічний підхід до побудови семантичної моделі навчальних матеріалів / О. В. Мазурець // Вісник Хмельницького національного університету. Серія: Технічні науки. – 2017. – № 6. – С. 223–229.

2. Бармак О. В. Інформаційна технологія автоматизованого визначення термінів у навчальних матеріалах / О. В. Бармак, О. В. Мазурець // Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка в технологічних процесах». – Хмельницький, 2015. – № 2. – С. 94–102.

3. Коренчук О. В. Система інтелектуального аналізу відповідності лекційних матеріалів навчальних

дисциплін стандартам освіти / О. В. Коренчук, О. В. Мазурець // Збірник наукових праць за матеріалами восьмої міжнародної науково-технічної конференції «Актуальні проблеми комп'ютерних технологій 2014». – Хмельницький, 2014. – С. 191–201.

4. Поліщук А. О. Інформаційна технологія автоматизації аналізу відповідності тестових завдань лекційним матеріалам навчальних дисциплін / А. О. Поліщук, О. В. Мазурець // Збірник наукових праць за матеріалами дев'ятої міжнародної науково-технічної конференції «Актуальні проблеми комп'ютерних технологій 2015». – Хмельницький, 2015. – С. 207–218.

5. Бармак О. В. Інформаційна технологія автоматизованого формування тестових завдань / О. В. Бармак, О. В. Мазурець, В. І. Кліменко // Вісник Хмельницького національного університету. Серія: Технічні науки. – 2017. – № 5. – С. 93–103.

6. Бармак О. В. Застосування інформаційної технології гнучкого тестування рівня знань у середовищі Moodle / О. В. Бармак, О. В. Мазурець, А. О. Матвійчук // Вісник Хмельницького національного університету. Серія: Технічні науки. – 2017. – № 3. – С. 103–115.

7. Бармак О. В. Інформаційна технологія автоматизованого анотування та реферування цифрових текстів / О. В. Бармак, О. В. Мазурець, А. В. Живілік // Вісник Хмельницького національного університету. Серія: Технічні науки. – 2017. – № 4. – С. 147–158.

8. Considerations for server-side Automation of Office [Електронний ресурс]. – Режим доступу : https://support.microsoft.com/en-us/help/257757/considerations-for-server-side-automation-of-office/

9. International Organization for Standardization - Open XML file formatsb [Електронний ресурс]. – Режим доступу : https://www.iso.org/search/x/query/Open%2520XML

10. Spire.Doc for .NET [Електронний ресурс]. – Режим доступу : https://www.nuget.org/packages/Spire.Doc/

11. International Organization for Standardization - Open XML file formatsb [Електронний ресурс]. – Режим доступу : https://www.iso.org/search/x/query/Open%2520XML

12. Office Open XML [Електронний ресурс]. – Режим доступу : https://uk.wikipedia.org/wiki/Office_Open_XML

13. How to automate Microsoft Excel from Microsoft Visual C#.NET [Електронний ресурс]. – Режим доступу : https://support.microsoft.com/en-us/help/302084/how-to-automate-microsoft-excel-from-microsoft-visual-c--net

14. DocX for creates or modifies Microsoft Word files [Електронний ресурс]. – Режим доступу : https://github.com/xceedsoftware/DocX

15. Free Spire.Doc for .NET [Електронний ресурс]. – Режим доступу : https://www.e-iceblue.com/Introduce/free-doc-component.html

References

1. Mazurets O. V. Ontologichiy pidhid do pobudovi semantichnoi modely navchalnih materialiv / O. V. Mazurets // Naukovyy zhurnal „Herald of Khmelnytskyi National University" seriya: Tekhnichni nauky. Khmelnytsky, 2017, № 6. – S. 223-229.

2. Barmak O. V., Mazurets O. V. Informatsiyna tekhnolohiya avtomatyzovanoho vyznachennya terminiv u navchalnykh materialakh / O. V. Barmak, O. V. Mazurets // Mizhnarodnyy naukovo-tekhnichnyy zhurnal «Vymiryuvalna ta obchyslyuvalna tekhnika v tekhnolohichnykh protsesakh» – Khmelnytsky, 2015. – №2. – C.94–102.

3. Korenchuk O. V., Mazurets O. V. Systema intelektualnoho analizu vidpovidnosti lektsiynykh materialiv navchalnykh dystsyplin standartam osvity / O. V. Korenchuk, O. V. Mazurets // Zbirnyk naukovykh prats za materialamy vosmoyi mizhnarodnoyi naukovo-tekhnichnoyi konferentsiyi «Aktualni problemy kompyuternykh tekhnolohiy 2014». Khmelnytsky – 2014. – S.191-201.

4. Polishchuk A. O., Mazurets O. V. Informatsiyna tekhnolohiya avtomatyzatsiyi analizu vidpovidnosti testovykh zavdan lektsiynym materialam navchalnykh dystsyplin / A. O. Polishchuk, O. V. Mazurets // Zbirnyk naukovykh prats za materialamy devyatoyi mizhnarodnoyi naukovo-tekhnichnoyi konferentsiyi «Aktualni problemy kompyuternykh tekhnolohiy 2015». Khmelnytsky – 2015. – S.207-218.

5. Barmak O. V., Mazurets O. V., Klimenko V. I. Informatsiyna tekhnolohiya avtomatyzovanoho formuvannya testovykh zavdan / O. V. Barmak, O. V. Mazurets, V. I. Klimenko // Naukovyy zhurnal „Herald of Khmelnytskyi National University" seriya: Tekhnichni nauky. Khmelnytsky, 2017, № 5. – S.93-103.

6. Barmak O. V., Mazurets O. V., Matviychuk A. O. Zastosuvannya informatsiynoyi tekhnolohiyi hnuchkoho testuvannya rivnya znan u seredovyshchi Moodle / O. V. Barmak, O. V. Mazurets, A. O. Matviychuk // Naukovyy zhurnal „Herald of Khmelnytskyi National University" seriya: Tekhnichni nauky. Khmelnytsky, 2017, №3. – S.103-115.

7. Barmak O. V., Mazurets O. V., Zhyvilik A. V. Informatsiyna tekhnolohiya avtomatyzovanoho anotuvannya ta referuvannya tsyfrovykh tekstiv / O. V. Barmak, O. V. Mazurets, A. V. Zhyvilik // Naukovyy zhurnal „Herald of Khmelnytskyi National University" seriya: Tekhnichni nauky. Khmelnytsky, 2017, № 4. – S. 147-158.

8. Considerations for server-side Automation of Office URL: https://support.microsoft.com/en-us/help/257757/considerations-for-server-side-automation-of-office/

9. International Organization for Standardization - Open XML file formatsb. URL: https://www.iso.org/search/x/query/Open%2520XML

10. Spire.Doc for .NET URL: https://www.nuget.org/packages/Spire.Doc/

11. International Organization for Standardization - Open XML file formatsb. URL: https://www.iso.org/search/x/query/Open%2520XML

12. Office Open XML. URL: https://uk.wikipedia.org/wiki/Office_Open_XML

13. How to automate Microsoft Excel from Microsoft Visual C#.NET. URL: https://support.microsoft.com/en-us/help/302084/how-to-automate-microsoft-excel-from-microsoft-visual-c--net

14. DocX for creates or modifies Microsoft Word files. URL: https://github.com/xceedsoftware/DocX

15. Free Spire.Doc for .NET. URL: https://www.e-iceblue.com/Introduce/free-doc-component.html

Рецензія/Peer review : 19.01.2018 р. Надрукована/Printed :28.01.2018 р.
Рецензент: стаття рецензована редакційною колегією