DOI: https://doi.org/10.15276/aait.06.2023.21
UDC 004.01

Algorithms and software for verification of scientific and technical text documents

Valerii S. Hlukhov[1)]
ORCID: https://orcid.org/0000-0002-0542-7447; Valerii.S.Hlukhov@lpnu.ua. Scopus Author ID: 56979360900
Dmytro S. Sydorko[1)]
ORCID: https://orcid.org/0009-0006-0965-1506; dmytro.sydorko.ki.2019@lpnu.ua
[1)] Lviv Polytechnic National University. 12, St. Stepan Bandera. Lviv, 79013, Ukraine

ABSTRACT

The work provides a solution to the problem of verifying the design (formatting) of scientific and technical documents for compliance with the requirements of regulatory documents (the problem of document verification). The basis of the check is the analysis of the styles of the Word text editor, which are used to design the paragraphs of the document under study. For each element of the document (headings, annotations, main text, figures, signatures under figures, list of references and others) a reference style of their design was developed. Together, these styles form the set of allowed styles. There can be many sets of allowed styles, each edition has its own set of styles. Only the administrator has access to each of the sets, which can create new styles, new sets, and edit both individual styles and individual sets. Due to the peculiarities of style parsing, the document is treated as a combination of headers and footers and the body of the document. Algorithms for its verification were developed for this structure of the document: an algorithm for analyzing headers and footers, an algorithm for analyzing paragraphs of the main text, and an algorithm for updating style settings by the administrator. .Net, WPF, DocumentFormat.OpenXml technologies were used to implement the algorithms by software. Using DocumentFormat.OpenXml allows you to analyze styles in .doc/.docx format documents; the developed program accepts .doc or .docx format files as input and analyzes them for compliance with specified styles. The result of the analysis is returned in .txt or .doc/.docx format, indicating the detected deviations from the standards. The .txt format file is a list of found deviations, and in the .doc/.docx format files, the deviations are recorded in the form of comments to the original text. Using the program simplifies the process of checking documents, it allows you to identify all deviations from standards and reduce the time and resources spent on checking. .Net and WPF technologies were used to develop the user interface. The developed program was checked in the process of checking the explanatory notes of real bachelor's and master's qualification theses. The style analysis time was determined; the time does not exceed 3 seconds. The developed program can be useful for automating the process of checking documents, ensuring quality and compliance with the design standards of scientific and technical documentation, scientific and technical publications, and, first of all, in the educational process for checking the design of bachelor's and master's qualification works, as well as various reports.

Keywords: MS Word style; text analysis; document analysis; verification of documents; .doc; .docx

***For citation*: Hlukhov V. S., Sydorko D. S. "Algorithmes and software for verification of scientific and technical text documents". *Applied Aspects of Information Technology*. 2023; Vol. 6 No. 3: 304–317. DOI: https://doi.org/10.15276/aait.06.2023.21**

INTRODUCTION

Modern regulatory documents set high requirements for style, formatting and compliance with specified characteristics that can be a challenge for many organizations. Deficiencies in the design of documents can lead to the impossibility of processing the information provided in them.

This work offers a solution to the urgent problem of the complex and costly process of checking the compliance of documents with the requirements of regulatory documents and identifying inconsistencies in reports (the problem of document verification).

To analyze the design of technical and scientific texts, algorithms for their analysis have been developed and implemented using .Net, WPF, DocumentFormat.OpenXml technology. Based on algorithms, software has been developed that automates the process of checking documents and detecting inconsistencies in them. Using this approach allows you to reduce the time and resources spent on document verification, as well as to ensure their quality.

The use of an automated approach to document verification will help to ensure compliance by executors of the standards for the design of technical and scientific documentation, technical and scientific publications, reporting documentation of students in the educational process, reducing time spent on documenting design results, which will contribute to the further development of technical specialties, such as computer engineering.

REVIEW OF LITERARY SOURCES

Attempts to normalize text documents, in particular technical and scientific documents in

order to facilitate their further verification and use, refinement and extension of various approaches to normalizing free-form names to map against a predefined standard taxonomy. In [2], the problem of classifying the political color of a text resource is solved.

One of the important tasks in the educational process is the creation of reporting text documents by students, which must comply with regulatory documents of various levels on formatting and style. Approbation of research results and publication of their results in the form of abstracts of reports at scientific and technical conferences and articles in scientific journals is also important for scientists.

To ensure the quality of the publication, editorial boards prepare recommendations for authors [3], as a rule, without summarizing these recommendations with the concept of "style". But introducing the concept of style significantly facilitates the processing of messages of various types: hand movements [4], determination of authorship [5], and recognition of natural languages [6].

An important and costly stage of processing reporting documents, theses and articles is their verification for compliance with established requirements. This stage of work is very expensive both for students and scientists, as well as for teachers and editorial boards of journals that carry out verification, it becomes critical in case of an increase in the number of students and an increase in the number of articles submitted to the journal.

Non-compliance with the established requirements can complicate the processing and analysis of the information provided in the documents.

There are many ways to analyze the file structure and its components.

In order to simplify and automate the checking of various types of files, namely: office documents such as text, spreadsheets, and presentations, the work [7] describes the structure of document classification and analysis.

For a better assessment of the quality of documents, work [8] proposed a solution where each sentence or paragraph is considered as a node.

The work [9] analyzed the methods of identifying objects in the text (tables, formulas, figures, etc.) and evaluated the effectiveness of each method of checking documents.

In work [10] for the analysis of .docx files, a methodology for extracting structural features is proposed, a framework is considered, which is are constantly being made. Paper [1] describes the performed statically with the help of meta-functions obtained from .docx files.

In [11], a method for shortening large text is analyzed, in which the length of the original text is reduced to a predetermined size, which improves the performance of text analysis and keeps computational costs low.

The paper [12] presents a comparative analysis of textual and model architecture when processing XML documents. In particular, it is shown that that text-based architecture outperformed in storage operation.

The vulnerability of the MS-Word file structure and the analysis of how data can be hidden in an MS-Word file are investigated in the paper [13].

There are known solutions for making it impossible to make changes to Word styles [14] and other parts of the document [15], but they are focused on document authors and do not guarantee that styles with the same names but different parameters will not be created in another document. Therefore, it is important to teach users how to work with styles and to observe safety rules when working with styles [16].

A number of works are devoted to limiting access to confidential information using hashing, which can be used during file analysis. The use of a cryptographic scheme to ensure the addition of only authorized and valid records to an existing block of records marked with time stamps is proposed in [17]. This was achieved by applying blockchain technology using the SHA256 hashing algorithm to hash accounts, thus protecting sample records that cannot be easily altered.

A hybrid solution to ensure data integrity in case of any deliberate attempt or malicious intent to change or delete data from the database, which can be verified at any subsequent stage, is proposed in [18].

Equivalence between the partial Boolean functions of two encryption algorithms was proved in [19], a common encryption scheme was also developed that can also use the SHA256 hashing algorithm.

A common software development tool for Windows is .NET, a free, open-source, cross-platform developer platform for building many different types of applications [20].

With .NET, you can use multiple languages, editors, and libraries to build websites, mobile devices, desktops, games, the Internet of Things, and more.

Windows Presentation Foundation framework (WPF) creates desktop client applications [21]. The

WPF development platform supports a broad set of application development features, including the application model, resources, controls, graphics, layout, data binding, documents, and security. The platform is also used to create educational materials [22].

The Open XML SDK provides tools for working with Office Word, Excel, and PowerPoint documents. It supports such scenarios as high-performance generation of text documents, spreadsheets and presentations; modification of the document [23]. DocumentFormat.OpenXml.dll is a library that provides a set of classes for working with Open XML documents, such as those created by Microsoft Office. Open XML is an open standard file format used to represent word processor documents, spreadsheets, presentations, and other types of documents.

The DocumentFormat.OpenXml.dll library is used by developers to read, write and process Open XML documents in their programs [24].

The Word text editor offers a set of styles that can be used to quickly format the entire document. If the user needs formatting options that aren't available in Word's built-in styles, you can modify an existing style and customize it to your needs. You can change the formatting (such as font size, color, and text indentation) in styles applied to headings, subheadings, paragraphs, lists, and more. You can also select formatted text in a document to create a new style in the Styles collection [25].

The analysis showed that there are many methods that can be used to solve the given problem, namely, use the free cross-platform .NET for programming, use the DocumentFormat.OpenXml library for processing text elements; it is advisable to develop the user interface based on the structure of the WPF user interface.

At the same time, the possibility of using user-created Word style sets for describing and subsequent checking of the structure of the created documents, as well as for formatting the document, was overlooked by the developers. It is advisable to organize the protection of the created style sets using the hashing function.

OBJECTIVES

The goal of this research is to develop secure user Word style sets that can be used to describe the structure of a user-designed document and to format the document.

Also, the task is the development of algorithms for checking the style of created documents and the development of software based on .Net, WPF and DocumentFormat.OpenXml technologies, which implements the created algorithms and provides automated verification of compliance of technical and scientific texts with the requirements of regulatory documents, as well as detection and documentation of inconsistencies with these requirements.

STYLES OF TEXT DOCUMENTS

For the uniform design of the reporting documents of the educational process by the students at the Computers Department of the Lviv Polytechnic National University, the authors developed a basic set of styles for the Word text editor – styles of the Computers Department. Styles define both an element of the document structure and the formatting of this element.

The name of the style consists of a prefix “CD:” (Computers Department) and its name, which indicates an element of the document structure.

The transition to work with styles forces you to abandon the habit of formatting the text of the document during its preparation. You can format only the styles, not the text of the document.

Adherence to the proposed styles helps to ensure the correct structure and format of documents, compliance with requirements for content and design, and speeds up the verification of documents.

During document verification, the following is checked:

– use of allowed styles (according to the value of the corresponding prefix, which is stored in the “key word” variable);

– saving the parameters of each of the styles. This check is necessary because the user has free access to edit Word styles.

ANALYSIS OF STYLES OF SCIENTIFIC AND TECHNICAL TEXTS

To solve the task of automating the verification of scientific and technical documentation, algorithms have been developed that check the style-based structure of documents. Separate algorithms have been developed for the analysis of headers and footers and the main text of the document.

Header or Footer analysis begins with checking the presence of unanalyzed paragraphs (Fig. 1). If they are, the program checks whether there is text in the paragraph. If the paragraph contains text, the program additionally checks whether the paragraph style is in the list of allowed styles (Table 1) of Computers Department. If so, the program proceeds to the analysis of the next paragraph.

Table 1. **Basic styles of the Computer department**

No.	Text element (document structure element)	The style of the document structure element and its parameters	No.	Text element (document structure element)	The style of the document structure element and its parameters
1	Header, page number	PC: Page number Alignment is to the right Times New Roman 12	8	Content items with 1 number or no number	Contents 1 Alignment is left Times New Roman 14 All are capitalized Line spacing is single
2	Name of the document	PC: Title Alignment is in the center Times New Roman 14 bold All are capitalized Interval before is 0 pt Interval after is 0 pt Interlinear is one and a half	9	Content items with 2 numbers	Contents 2 Alignment is left Times New Roman 14 Line spacing is single Left indent is 0.5 cm
3	Author	PC: Author Alignment is in the center Times New Roman 14 Lower case Bold font Interval before 0 pt Interval after 0 pt Interlinear is one and a half	10	Title 1 (with 1 number)	Header 1 Times New Roman 14 Bold font Interlinear is one and a half All are capitalized Alignment is in the center Numbering is on the left edge Numbering level is 1
4	Specialty	PC: Specialty Alignment is in the center Times New Roman 14 italics Interval after is 0 pt Interlinear is one and a half	11	Conventional abbreviations	PC: Left Alignment is left Times New Roman 14 Interlinear is one and a half
5	Type of work	PC: Center Alignment is in the center Times New Roman 14 Interlinear is one and a half	12	Introduction title	PC: title from a new page Alignment is by width Times New Roman 14 All are large Bold font Interlinear is one and a half
6	Supervisor	PC: Head Alignment is to the right Times New Roman 10 Interval after is 4 p.m Interline is Single	13	Title of the list of conventional abbreviations	PC: title from a new page Alignment is by width Times New Roman 14 All are capitalized Bold font Interlinear is one and a half
7	City and year of document creation	PC: Center Alignment is in the center Times New Roman 14 Interlinear is one and a half	14	Title 1 without number	PC: Title 1 unnumbered Times New Roman 14 Bold font Interlinear is one and a half All are capitalized Alignment is in the center

Continuation of Table 1

No.	Text element (document structure element)	The style of the document structure element and its parameters	No	Text element (document structure element)	The style of the document structure element and its parameters
15	Content title	PC: title from a new page Alignment is by width Times New Roman 14 All are capitalized Bold font Interlinear is one and a half	21	Text	PC: main text Indent is 1.25 cm Times New Roman 14 Interlinear is one and a half Alignment is by width
16	Heading 2 (with two numbers)	Header 2 Times New Roman 14 Interlinear is one and a half Alignment is by width Numbering level 2 Indent is 1.25 cm Interval before is 6 p.m Interval after is 6 p.m The position of the number is 1.25 cm from the left edge Do not take away from the next	22	The title of the conclusions	PC: title from a new page Alignment is by width Times New Roman 14 All are large Bold font Interlinear is one and a half
17	Text	PC: main text Indent is 1.25 cm Times New Roman 14 Interlinear is one and a half Alignment is by width	23	Table number	PC: Table number Alignment is to the right Times New Roman 14 italics Interlinear is one and a half Do not take away from the next The name for numbering is Table
18	Text in a table, aligned to the left	PC: Table_left Times New Roman 12 Bold font (title) Interline is Single Alignment is left	24	Name of the table	PC: Name of the table Alignment is in the center Times New Roman 14 bold italic font Interlinear is one and a half Do not take away from the next
19	Number and name of the figure	PC: Name of the figure Alignment is in the center Times New Roman 12 italics Interlinear is one and a half The name for numbering is Fig.	25	Title of the list of used sources	PC: title from a new page Alignment is by width Times New Roman 14 All are large Bold font Interlinear is one and a half
20	Figure	PC: Figure Alignment is in the center Times New Roman 12 Interlinear is one and a half Do not take away from the next	26	References	PC: Literature Times New Roman 14 Interlinear is semi-torque Title for numbering is Literature

Source: **compiled by the authors**

If the style is not in the allowed list, the program checks whether the first letters of the style name are equal to the value stored in the "key word" variable (only the administrator can change this variable). If they are, the program checks whether the style has been modified from the original. If yes, then a note with the parameter "edited" is created, if not, the program proceeds to the analysis of the next paragraph, as well as after the condition "Yes" is met.

When the first letters of the style name do not equal the "key word" value, the program generates a remark indicating that the paragraph does not meet the required criteria.

If the paragraph does not contain text, the program proceeds to the analysis of the next paragraph in the footer.

The program continues this process in a loop until there are no paragraphs left in the header and footer to analyze.

After the completion of the cycle, the program proceeds to further analysis of the document.

As a result of further analysis, an array is created from all paragraphs of the main part of the document (Body, Fig. 2). And for each element of the array, it is checked whether the current paragraph is part of the table of contents (TOC), for this its style is checked. If so, the program checks to see if the style is valid. If so, the program proceeds to the analysis of the next paragraph. If the style is invalid, the program handles the mismatch of the paragraph from the table of contents.

If the paragraph is not part of the table of contents, the program checks whether the paragraph is part of the table. If so, the row number, cell number, and table number associated with the paragraph are determined. The program then checks to see if the style is valid. If so, the program proceeds to the analysis of the next paragraph. If the style is invalid, the program handles the mismatch.

If a paragraph is not part of a table, it is considered to be a paragraph of the main text. The program checks whether the style in this paragraph is valid. If so, the program proceeds to the analysis of the next paragraph. If the style is invalid, the program handles the paragraph style mismatch.

The program continues to go through all the paragraphs in the body of the document until it has processed them all.

After completing the checks, the program completes the analysis by closing the files used and created.

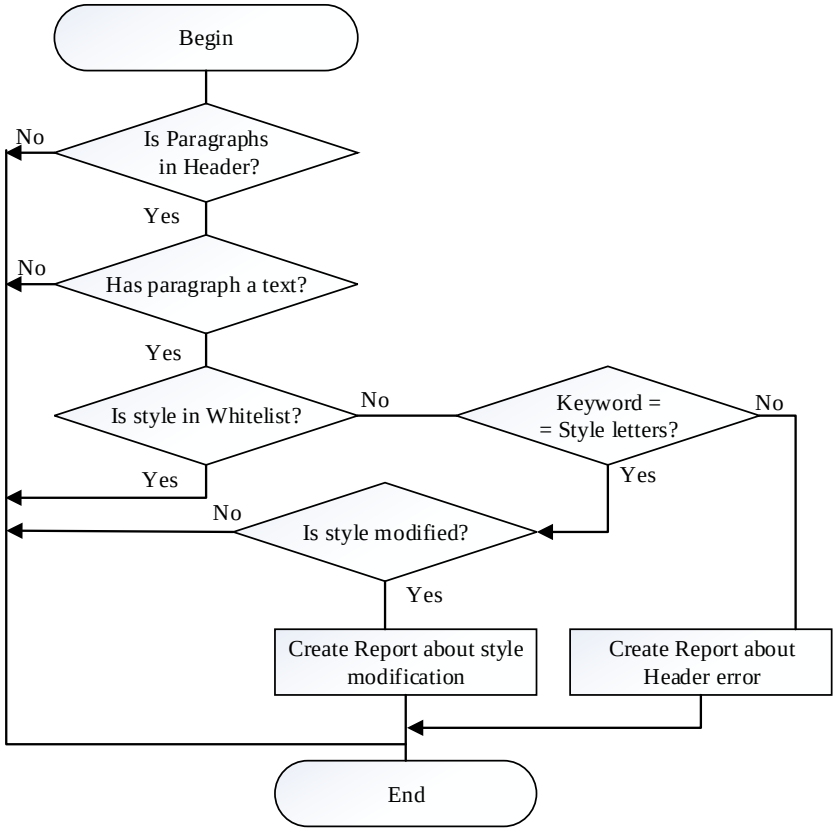


Fig. 1. **Header and Footer analysis algorithm**
Source: **compiled by the authors**

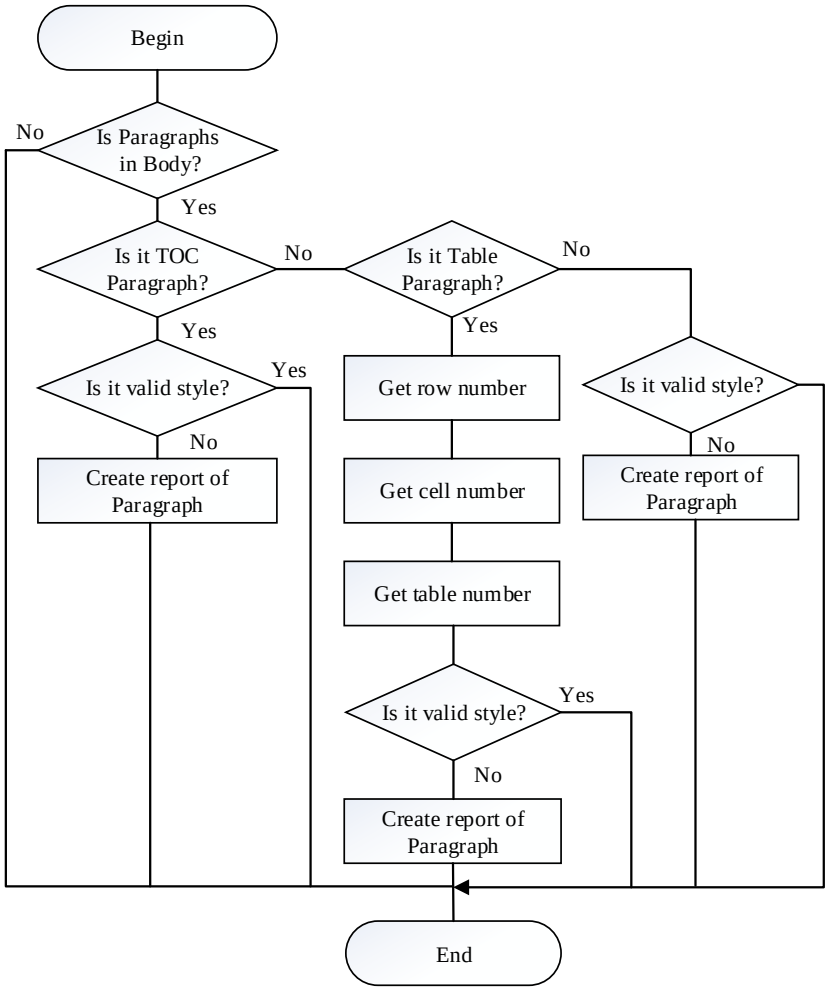


Fig. 2. **Algorithm for analysis of paragraphs located in Body**
Source: **compiled by the authors**

ADMINISTRATION OF ALLOWED STYLES

To ensure the change of design styles of technical and scientific documents, the descriptions and parameters of the styles are stored in a separate initialization file. This file is protected by its hash. Only an administrator can change the initialization file.

The administration algorithm (updating styles by the administrator, Fig. 3) begins with determining the hash of the initialization file and comparing it with the control value from the additional service file. This allows you to detect unauthorized modification of style settings stored in the initialization file.

If the hash does not match the checksum, it means that the data in the initialization file has been tampered with or corrupted. Then an initialization file error message is generated.

If the hash matches the control value, the program switches to editing style parameters mode.

After finishing the editing, the program allows the administrator to cancel or save the changes.

If the administrator chooses to save the changes, the application saves the style settings in the initialization file, hashes it using the SHA256 algorithm, and stores the newly calculated hash value in an additional service file to verify the integrity of the initialization file in the future. After that, the algorithm completes its work.

IMPLEMENTATION AND TESTING OF THE TEXT DOCUMENT VERIFICATION SYSTEM

After starting the program, a window is displayed on the user's screen (Fig. 4) in which the “Upload” button is available, intended for uploading a file that needs to be checked. After uploading, the document is processed and the upload status is displayed (Fig. 5), where you can see whether the execution was successful or an error occurred.

During the analysis, two files are created – “Report.txt” (analysis log) and the initial file with

program comments “ANALYSED.docx” (Fig. 6), but the user can cancel the creation of the “ANALYSED” file by selecting the appropriate option (Fig. 4).

To change the style checking setting, you need to run the executable file with administrator rights (Fig. 7).

The interface has almost not changed, except that an additional authorization button has appeared (Fig. 8), when clicked, a new modal window for authorization opens (Fig. 9).

The default password is “admin”.

After successful confirmation of the password, the administrator has two additional options: “Change the password” and “Setting criteria for checking program styles” (Fig. 10).

In the configuration mode, the administrator will open a model window where he can change the parameters or add new criteria to the style parameters (Fig. 11). He can also change the “key word”, which corresponds to which characters the name of a valid style (prefix) should start with.

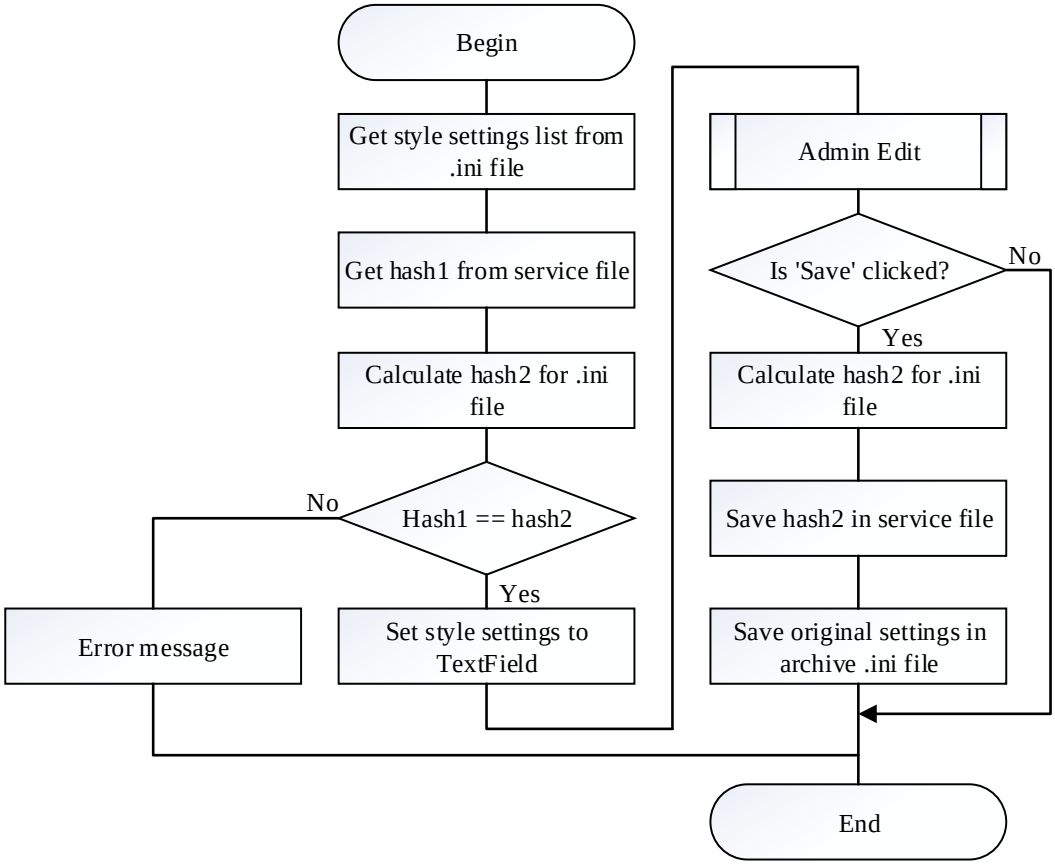


Fig. 3. **The algorithm for updating style settings by the administrator**
Source: **compiled by the authors**

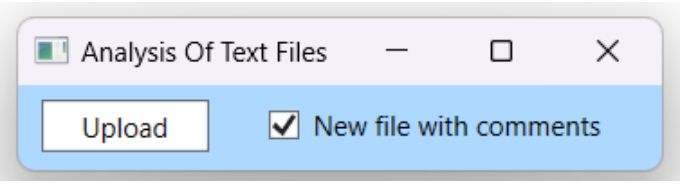
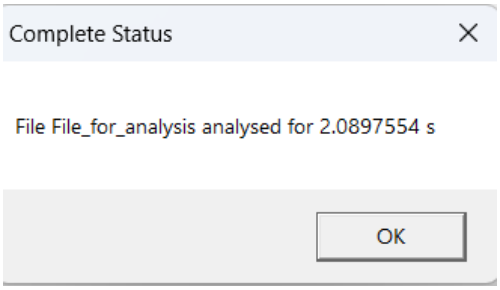
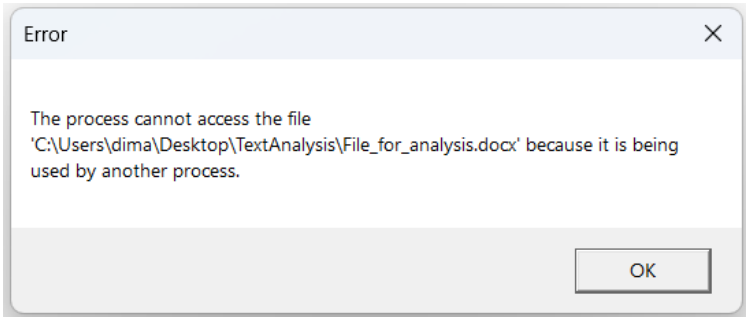


Fig. 4. **File selection interface**
Source: **compiled by the authors**



a



b

Fig. 5. **Display of execution: a - successful; b - with an error**
Source: **compiled by the authors**

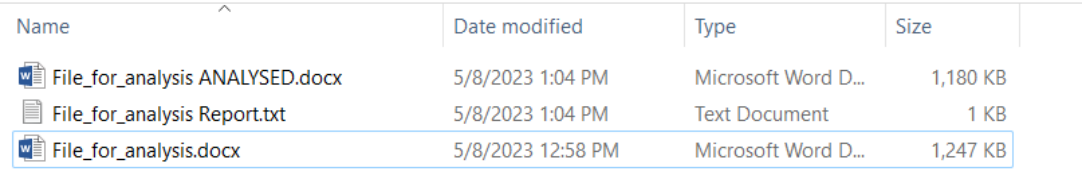


Fig. 6. **Input and output files after processing**
Source: **compiled by the authors**

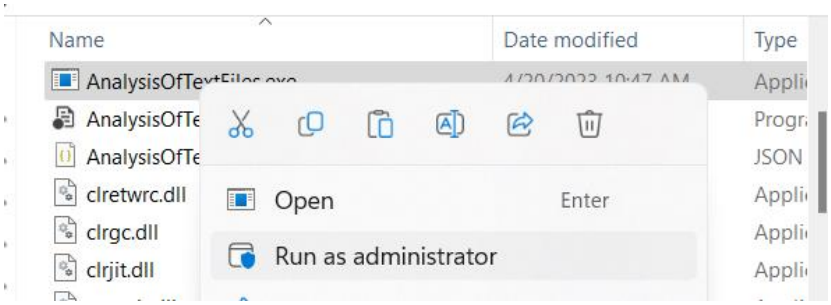


Fig. 7. **Run the program with administrator rights**
Source: **compiled by the authors**

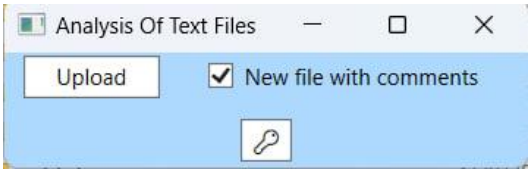


Fig. 8. **Program interface for the administrator**
Source: **compiled by the authors**

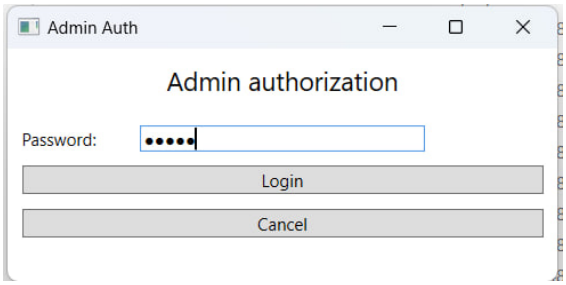


Fig. 9. **Modal administrator authorization window**
Source: **compiled by the authors**

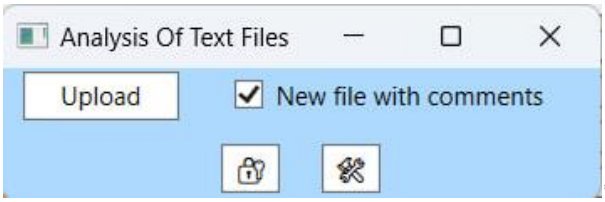


Fig. 10. **Run the program with administrator rights**
Source: **compiled by the authors**

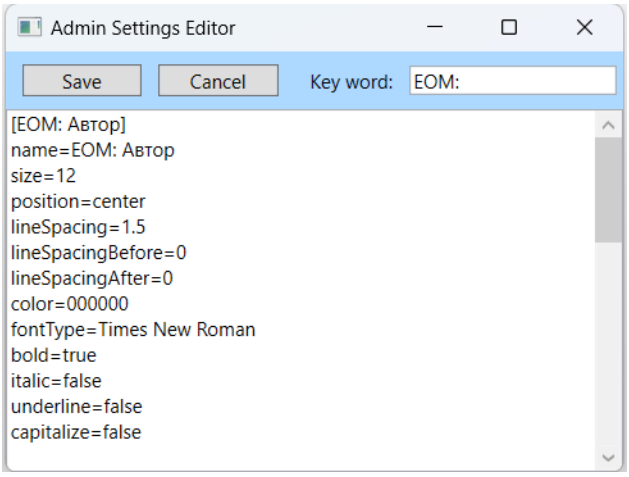


Fig. 11. **Modal window for editing styles**
Source: **compiled by the authors**

If you select the Cancel option after editing, the window will close, leaving the previous changes. And if you choose save, information about the status will be highlighted and, if no errors are detected, all information will be saved in the initialization file styleSettings.ini and additionally a SHA256 key will be generated for future data integrity verification.

The “Report.txt” file contains records of all style inconsistencies, indicates the places in the document where they are detected, as well as the parameters of these inconsistencies and specific types of detected inconsistencies (Fig. 12).

“ANALYSED.docx” is a copy of the input file with comments about the mismatch of styles (Fig. 13). Places in the text with disallowed styles are marked with a colored background.

Allowed style names include “Header 1”, “Header 2”, “Header 3”, “Content 1”, “Content 2” and all that begin with the prefix specified in the “key word” variable.

The test results showed that the analysis of a document with 6800 words takes only 2.089 seconds (Fig. 5), which provides a significant reduction in the time required for a person to review document styles.

The authors do not know programs that can perform a similar verification of documents, so it is impossible to compare the achieved results with similar solutions.

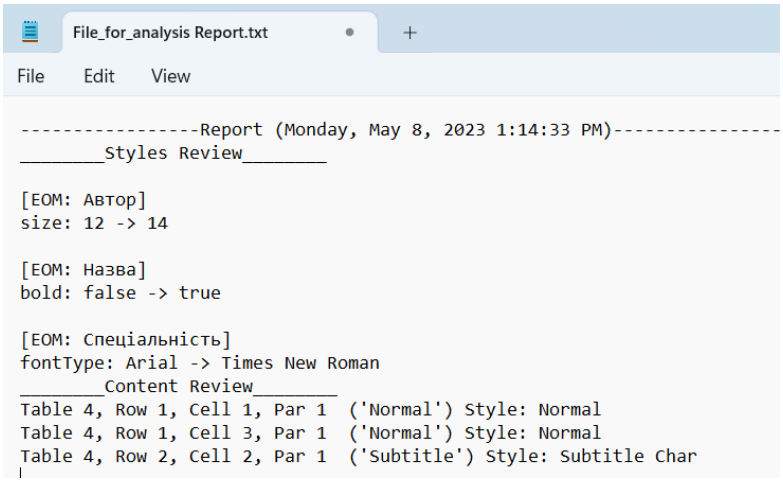


Fig. 12. **The contents of the Report.txt file with detected inconsistencies**
Source: **compiled by the authors**

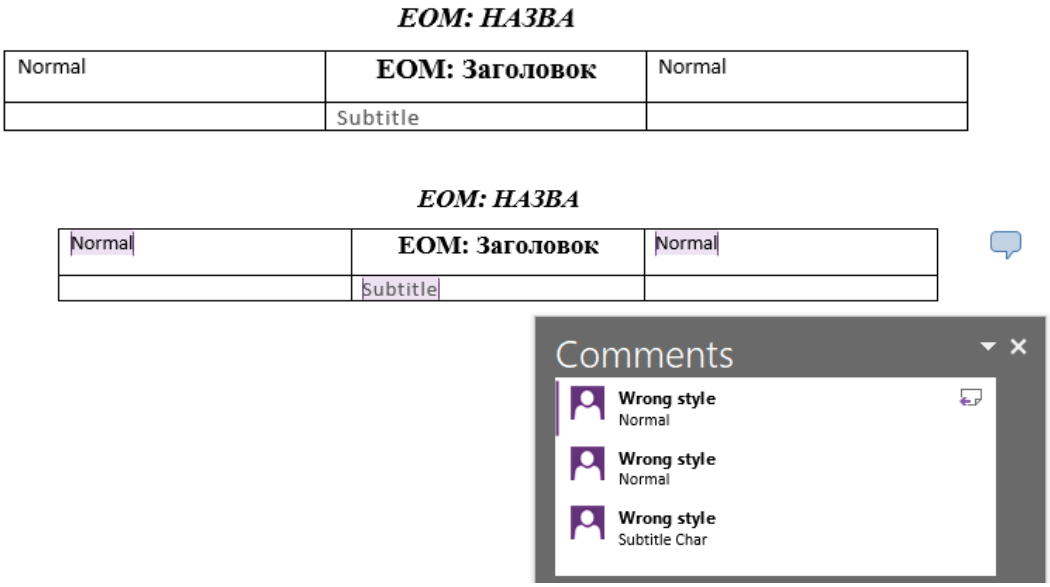


Fig. 13. **Comparing the contents of the ANALYSED.docx file with the input file**
Source: **compiled by the authors**

CONCLUSIONS

The article offers a solution to the problem of reducing time spent on verification of scientific and technical documents. This task arises when checking the compliance of documents with established requirements and identifying inconsistencies with these requirements.

The proposed solution is based on the use of Word document styles to describe the structure of text documents, the creation of a set of permitted styles for each document edition, and the use of only these permitted styles when creating documents. Verification consists in checking the compliance of the styles of the created document with the given set of allowed styles.

On the basis of the conducted analysis, algorithms for checking text documents were developed and a software implementation of these algorithms was created, which ensured the creation of sets of permitted styles and verification of document styles for compliance with permitted styles. The correctness of the created program was also tested. With the help of the created program, the style check of a document with 6800 words took only 2.089 seconds.

As a result of the research and development of the software solution, it was possible to solve one of the important problems that arises during the manual verification of compliance of documents with the

established requirements and when inconsistencies in the styles of these documents are detected.

The developed software solution can be used to automatically check documents for compliance with requirements and identify inconsistencies in styles for student reports during the educational process, when preparing scientific and technical publications, when preparing technical and scientific documentation, which will reduce the time and effort spent on text verification documentation. In addition, this project opens up opportunities for further improvement and expansion of the functionality of the developed solution. In general, the results of this project are an important step in the direction of automating document verification processes and increasing the efficiency of the relevant specialists.

REFERENCES

1. Bocharova, M. Y., Malakhov, E. V. & Mezhuyev, V. I. "VacancySBERT: the method for representation of titles and skills for semantic similarity search in the recruitment domain". *Applied Aspects of Information Technology.* 2023; 6 (1): 52–59. DOI: https://doi.org/10.15276/aait.06.2023.4.

2. Kosiv, Y. A. & Yakovyna, V. S. "Three language political leaning text classification using natural language processing methods". *Applied Aspects of Information Technology*. 2022; 5 (4): 359–370. DOI: https://doi.org/10.15276/aait.05.2022.24.

3. Applied Aspects of Information Technology. Scientific Journal. Article Guadlines. *AAIT.OPU.UA.* – Available from: https://aait.op.edu.ua/?fetch=page&with=aguidelines. – [Accessed: 06-th February, 2023].

4. Xu, W., Tian, J., Cao, Y. & Wang, S. "Challenge-response authentication using in-air handwriting style verification". In: *IEEE Transactions on Dependable and Secure Computing.* 2020; 17 (1): 51–64. [Scopus 8038051]. DOI: https://doi.org/10.1109/TDSC.2017.2752164.

5. Ou, W., Ding, S.H.H., Tian, Y. & Song, L. "SCS-Gan: Learning functionality-agnostic stylometric representations for source code authorship verification". *IEEE Transactions on Software Engineering.* 2023; 49 (4): 1426–1442, https://www.scopus.com/authid/detail.uri?authorId=57712420800. DOI: https://doi.org/10.1109/TSE.2022.3177228.

6. Lagutina, K., Lagutina, N., Boychuk, E. et al. "A survey on stylometric text features". *25th Conference of Open Innovations Association (FRUCT).* Helsinki: Finland. 2019. p. 184–195. [Scopus 8981504]. DOI: https://doi.org/10.23919/FRUCT48121.2019.8981504.

7. Eken, S., Menhour, H. & Koksal, K. "DoCA: A content-based automatic classification system over digital documents". *IEEE Access.* 2019; 7: 97996–98004. [Scopus 8768370]. DOI: https://doi.org/10.1109/access.2019.2930339.

8. Jung, D., Kim, M. & Cho, Y.-S. "Detecting documents with inconsistent context". *IEEE Access,* 2022; 10: 98970–98980. https://www.scopus.com/authid/detail.uri?authorId=57887759300.
DOI: https://doi.org/10.1109/access.2022.3204151.

9. Nguyen, K., Nguyen, A., Vo., N. D. & Nguyen, T. V. "Vietnamese document analysis: Dataset, method and benchmark suite". *In: IEEE Access.* 2022: 10: 108046–108066, https://www.scopus.com/authid/detail.uri?authorId=56537681800].
DOI: https://doi.org/10.1109/access.2022.3211069.

10. Nissim, N., Cohen A. & Elovici, Y. "ALDOCX: Detection of unknown malicious microsoft office documents using designated active learning methods based on new structural feature extraction methodology". *IEEE Transactions on Information Forensics and Security.* 2019; 12 (3): 631–646. [Scopus 7762928]. DOI: https://doi.org/10.1109/TIFS.2016.2631905.

11. Fiok, K., Karwowski, W., Gutierrez-Franco, E. et al. "Text Guide: Improving the quality of long text classification by a text selection method based on feature importance". *IEEE Access.* 2021; 9: 105439–105450. [Scopus 9494560]. DOI: https://doi.org/10.1109/ACCESS.2021.3099758.

12. Chau, K. T., He, Q., Hu, X. & Wu, R. "Comparison on performance of text-based and model-based architecture in open source native XML database". *IEEE 4th International Conference on Signal and Image Processing (ICSIP).* Wuxi: China. 2019. p. 340–344. [Scopus 8868709]. DOI: https://doi.org/10.1109/SIPROCESS.2019.8868709.

13. Lee, H. & Lee, H.-W. “Hidden message detection in MS-Word file by analyzing abnormal file structure”. *International Conference on Green and Human Information Technology (ICGHIT).* Hanoi: Vietnam. 2020. p. 54–57. [Scopus 9058358]. DOI: https://doi.org/10.1109/ICGHIT49656.2020.00021.

14. Wright, E. “How to restrict style changes in Microsoft Word”. *Erin Wright Writing*. – Available from: https://erinwrightwriting.com/how-to-restrict-style-changes-in-microsoft-word-tutorial. – [Accessed: 06-th February, 2023].

15. “Allow changes to parts of a protected document”. *Microsoft.* – Available from: https://support.microsoft.com/en-gb/office/allow-changes-to-parts-of-a-protected-document-187ed01c-8795-43e1-9fd0-c9fca419dadf. – [Accessed: 06-th February, 2023].

16. Wyatt, A. “Preventing changes to styles in documents”. tips.net. – Available from: https://wordribbon.tips.net/T008097_Preventing_Changes_to_Styles_in_Documents.html. – [Accessed: 06-th February, 2023].

17. Xenya, M. C. & Quist-Aphetsi, K. “A cryptographic technique for authentication and validation of forensic account audit using SHA256”. *International Conference on Cyber Security and Internet of Things (ICSIoT).* Accra: Ghana. 2019. p. 11–14. [Scopus 9058345]. DOI: https://doi.org/10.1109/ICSIoT47925.2019.00008.

18. Sarwar, M. I., Iqbal, M. W., Alyas, T. et al. “Data vaults for blockchain-empowered accounting information systems”. *IEEE Access.* 2021; 9: 117306–117324. https://www.scopus.com/authid/detail.uri?authorId=57215290328.
DOI: https://doi.org/10.1109/ACCESS.2021.3107484.

19. Xu, W., Xu, Y., Huo, G., Yang, Y. & Jin, Y. “Optimized dual-mode security encryption chip design based on hash algorithm”. *IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT).* Indore: India. 2022. p. 566–570, https://www.scopus.com/authid/detail.uri?authorId=57771538600.
DOI: https://doi.org/10.1109/CSNT54456.2022.9787655.

20. “What is .NET?” *Microsoft.* – Available from: https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet. – [Accessed 06-th February, 2023].

21. “Tutorial: Create a simple WPF application with C#”. *Microsoft.* – Available from: https://learn.microsoft.com/en-us/visualstudio/get-started/csharp/tutorial-wpf?view=vs-2022. – [Accessed: 06-th February, 2023].

22. Filipova-Petrakieva, S. & Shopov S. “Educational Windows presentation foundation and XAML Application for information protection based on the cryptographic methods – part II”. *13th Electrical Engineering Faculty Conference (BulEF).* Varna: Bulgaria. 2021. p. 1–8, https://www.scopus.com/authid/detail.uri?authorId=57195100645.
DOI: https://doi.org/10.1109/BulEF53491.2021.9690842.

23. “DocumentFormat.OpenXml”. *NuGet*. – Available from: https://www.nuget.org/packages/DocumentFormat.OpenXml. – [Accessed 06-th February, 2023].

24. “DocumentFormat. OpenXml.dll: Free Download”. *DLLme*. – Available from: https://www.dllme.com/dll/files/documentformat_openxml. – [Accessed: 06-th February, 2023].

25. “Customize or create new styles”. *Microsoft.* – Available from: https://support.microsoft.com/en-gb/office/customize-or-create-new-styles-d38d6e47-f6fc-48eb-a607-1eb120dec563#:~:text=Right%2Dclick%20the%20text%20on,appear%20in%20the%20Styles%20gallery. . – [Accessed: 06-th February, 2023].

Conflicts of Interest: The authors declare no conflict of interest

Received 29.04.2023
Received after revision 30.08.2023
Accepted 18.09.2023

DOI: https://doi.org/10.15276/aait.06.2023.21
УДК 004.01

Алгоритми та програмне забезпечення для нормоконтролю наукових та технічних текстових документів

Глухов Валерій Сергійович[1)]
ORCID: https://orcid.org/0000-0002-0542-7447; Valerii.S.Hlukhov@lpnu.ua. Scopus Author ID: 56979360900
Сидорко Дмитро Степанович[1)]
ORCID: https://orcid.org/0009-0006-0965-1506; dmytro.sydorko.ki.2019@lpnu.ua
[1)] Національний університет «Львівська Політехніка», вул. Степана Бандери 12. Львів, 79013, Україна

АНОТАЦІЯ

У роботі надано вирішення задачі перевіряння оформлення (форматування) наукових та технічних документів на дотримання вимог нормативних документів (задачі нормоконтролю документів). В основу перевірки покладено аналіз стилів текстового редактора Word, які використовуються для оформлення абзаців досліджуваного документа. Для кожного елемента документа (заголовків, анотацій, основного тексту, рисунків, підписів під рисунками, списком літератури ті інших) було розроблено еталонний стиль їхнього оформлення. Разом ці стилі утворюють набір дозволених стилів. Наборів дозволених стилів може бути багато, для кожного видання – свій набір стилів. Доступ до кожного з наборів має тільки адміністратор, який може створювати нові стилі, нові набори та редагувати як окремі стилі, так і окремі набори. З огляду на особливості аналізу стилів документ розглядається як об'єднання колонтитулів та основного тексту документу. Для такої структури документу було розроблено алгоритми його нормоконтролю: алгоритм аналізу колонтитулів, алгоритм аналізу абзаців основного тексту, а також алгоритм оновлення налаштувань стилів адміністратором. Для реалізації алгоритмів програмним способом було використано технології .Net, WPF, DocumentFormat.OpenXml. Використання DocumentFormat.OpenXml дозволяє аналізувати стилі в документах формату .doc/.docx, розроблена програма приймає на вхід файли формату .doc чи .docx і аналізує їх на відповідність заданим стилям. Результат аналізу повертається у форматі .txt чи .doc/.docx, із зазначенням виявлених відхилень від еталонів. Файл формату .txt представляє собою перелік знайдених відхилень, а у файлах форматів .doc/.docx відхилення фіксуються у вигляді коментарів до початкового тексту. Використання програми спрощує процес перевірки документів, дозволяє визначити всі відхилення від еталонів та знизити витрати часу та ресурсів на виконання нормоконтролю. Для розробки інтерфейсу користувача було використано технології .Net та WPF. Розроблену програму було перевірено в процесі нормоконтролю пояснювальних записок реальних бакалаврських та магістерських кваліфікаційних робіт. Було визначено час аналізу стилів, час не перевищує 3 с. Розроблена програма може бути корисною для автоматизація процесу нормоконтролю документів, забезпечення якості та дотримання стандартів оформлення наукової та технічної документації, наукових та технічних видань, і, в першу чергу, у навчальному процесі для нормоконтролю бакалаврських та магістерських кваліфікаційних робіт, а також різноманітних студентських звітів.

Ключові слова: стиль MS Word; аналіз тексту; аналіз документу; нормоконтроль документів; doc; docx

ABOUT THE AUTHORS

Valerii S. Hlukhov – Doctor of Engineering Sciences, Professor, Professor of the Department of Electronic Computing. Lviv Polytechnic National University, 12, Stepan Bandera Str. Lviv, 79013, Ukraine
ORCID: https://orcid.org/0000-0002-0542-7447; Valerii.S.Hlukhov@lpnu.ua.
Scopus Author ID: 56979360900
***Research field*:** Algorithms and structures of primary information processing devices; specialized computers for control and navigation systems; cryptoprocessors

Глухов Валерій Сергійович – доктор технічних наук,професор, професор кафедри Електронних обчислювальних машин. Національний університет «Львівська Політехніка», вул. Степана Бандери, 12. Львів, 79013, Україна

Dmytro S. Sydorko – Master of Electronic Computing Department. Lviv Polytechnic National University, 12, Stepan Bandera Str. Lviv, 79013, Ukraine
ORCID: https://orcid.org/0000-0002-0542-7447; dmytrosydorko@gmail.com
***Research field*:** Text analysis; standard control of documents

Сидорко Дмитро Степанович – магістр кафедри Електронних обчислювальних машин. Національний університет «Львівська Політехніка», вул. Степана Бандери, 12. Львів, 79013, Україна