

УДК 004.4:65.011.56

DOI [https://doi.org/10.24144/2616-7700.2025.47\(2\).207-213](https://doi.org/10.24144/2616-7700.2025.47(2).207-213)**М. М. Повідайчик¹, А. Л. Скляр², Р. М. Годя³**

¹ ДВНЗ «Ужгородський національний університет»,
професор кафедри кібернетики і прикладної математики,
доктор педагогічних наук

mykhailo.povidaichyk@uzhnu.edu.ua

ORCID: <https://orcid.org/0000-0003-1554-2067>

² ДВНЗ «Ужгородський національний університет»,
аспірант 3-го року навчання

artur.skliar@uzhnu.edu.ua

ORCID: <https://orcid.org/0000-0002-7714-7938>

³ ДВНЗ «Ужгородський національний університет»,
аспірант 3-го року навчання

roman.hodia@uzhnu.edu.ua

ORCID: <https://orcid.org/0009-0008-7546-5263>

РОЗРОБЛЕННЯ ДЕЯКИХ МОДУЛІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ «ЦИФРОВИЙ АСИСТЕНТ ВИКЛАДАЧА ЗВО»

Описано розроблений модуль інформаційної системи «Цифровий асистент викладача ЗВО», який призначено для автоматизації процесу створення робочої програми навчальної дисципліни. Подано загальну схему функціонування системи та описано основні процедури її роботи. Розглянуто можливості автоматизації на основі даних освітньо-професійної програми, навчального плану та інших нормативних документів. Визначено методику обчислення числових параметрів робочої програми, які можуть бути отримані автоматизовано з наявних файлів. Описано використання об'єктів Word, що забезпечують керування діапазонами, закладками та таблицями. Зазначено перспективи подальшого розвитку системи як інструменту автоматизованого аналізу розроблених робочих програм і виявлення їх невідповідностей вимогам освітньо-професійної програми та навчального плану.

Ключові слова: інформаційна система, автоматизація, файл-шаблон, VBA.

1. Вступ. Серед задач, які вирішує викладач ЗВО є як творчі, що стосуються планування та проведення занять, досліджень, публікації результатів, керівництва науковою роботою здобувачів, а також рутинні, які вимагають здебільшого механічної діяльності, пов'язаної із заповненням різноманітних документів заданого шаблону. На нашу думку, останні потрібно максимально автоматизувати, щоб вивільняти час саме для розв'язання творчих задач.

Авторська система «Цифровий асистент викладача» передбачає розроблення модулів, які власне беруть на себе значну частину рутинних завдань, які підлягають автоматизації. Прикладом задачі, яка вимагає багато часових ресурсів, може бути розроблення різноманітних тестових завдань для перевірки знань здобувачів, їхньої самопідготовки [1, 2].

Ще одна задача розглядається у даній статті — це автоматизація процесу наповнення робочої програми з дисципліни даними, які зберігаються у різноманітних документах: освітньо-професійній програмі (ОПП), навчальному плані (НП) та ін. Розроблений модуль не може створити повноцінну робочу програму, оскільки її зміст, перелік джерел та ін. вимагає інтелектуальної роботи викладача. Але наповнення документа стандартними даними, як показує досвід, значно прискорює процес його формування.

2. Аналіз останніх досліджень і публікацій. Зазначені проблеми досліджують як вітчизняні, так і зарубіжні науковці. Так, у [3] досліджено підходи й інструменти автоматизації створення і наповнення курсів у Moodle; у [4] розглянуто автоматизовану систему для динамічного формування навчальних компонент (робочих планів/модулів); у [5] описано модуль для автоматизованого формування навчальних планів з урахуванням вибору студентів. Проте, недостатньо дослідженою залишається проблема розроблення цифрових інструментів для автоматизованої роботи з документацією, що стосується освітнього процесу.

3. Постановка задачі. Розробити модуль інформаційної системи «Цифровий асистент викладача ЗВО» для автоматизованого наповнення робочої програми з дисципліни на основі шаблонів MS Word з використанням середовища Visual Basic for Applications.

4. Описання загальної схеми роботи системи. При розробленні інформаційної системи «Цифровий асистент викладача ЗВО» за основу було обрано шаблони MS Word, оскільки вони дозволяють формувати документи заданої структури, а вбудована мова програмування Visual Basic for Applications дозволяє ефективно керувати всіма об'єктами MS Office [1].

Розглянемо загальну схему роботи системи (лістинг 1).

Лістинг 1: Головна процедура системи.

```

1 Sub Main()
2     Dim doc As Document
3     Call Отримати_теми
4     Call Отримати_години
5     Call Отримати_ІК
6     Call Отримати_ЗК
7     Call Отримати_ФК
8     Call Отримати_РН
9     Call Матриця_Компетентності
10    Call Матриця_РН
11
12    Call Замінити_маркери
13    Call Закладка_компетентності
14    Call Закладка_зміст
15    Call Таблиці_РН_ОРН
16    Call Таблиці_Модуль12
17    Call Таблиця_Структура_НД
18    Call Семінари_Самостійна_робота
19
20    Set doc = Application.Documents.Add
21    ThisDocument.Range.Copy
22    doc.Range.Paste
23    doc.SaveAs2 fileName:=ThisDocument.Path & "{\}" & назва_дисципліни
        & ".docx"
24    doc.Close
25 End Sub

```

Головну процедуру системи Main можна умовно розділити на три частини. У першій частині знаходимо необхідні дані із трьох файлів:

- «Зміст дисципліни.docx» — це файл, який містить назву дисципліни, перелік модулів та тем до кожного модуля;

- Excel-файл, назва якого відповідає шаблону «НП*.xls?», — це навчальний план ОПП, з якого отримуємо як загальну інформацію (рівень вищої освіти, галузь знань, спеціальність, предметну спеціальність, назву ОПП), так і конкретну для вказаної дисципліни (шифр, години та ін.);
- Word-файл, назва якого відповідає шаблону «ОПП*.doc?», — це освітньо-професійна програма, з якого отримуємо інтегральну компетентність, перелік загальних та фахових компетентностей (на основі матриці відповідності програмних компетентностей освітнім компонентам), а також перелік програмних результатів навчання (на основі матриці забезпечення програмних результатів навчання відповідними освітніми компонентами).

У другій частині на основі отриманих даних формуємо файл-шаблон робочої програми. Тут заміняємо раніше створені маркери даними з НП, редагуємо раніше створені закладки даними з ОПП, формуємо нові таблиці та наповнюємо їх розрахунковими даними. Детальніше це описано нижче.

Третя частина зберігає наповнений даними файл-шаблон у форматі файлу .docx у поточній папці. Далі цей файл викладач має змогу редагувати – додати дані, які не можна отримати з НП і ОПП (очікувані програмні результати, список джерел та ін.).

5. Розроблення шаблону для автоматизованого створення робочої програми. Розглянемо можливості автоматизації процесу розроблення робочої програми дисципліни. Так, у документі стандартної робочої програми більше одного разу зустрічаються:

- галузь знань — 5 разів;
- спеціальність — 2 рази;
- предметна спеціальність — 2 ризи (для педагогічних спеціальностей);
- освітня програма — 3 рази;
- вид підсумкового контролю — 3 рази.

Всі ці дані приведені у навчальному плані освітньо-професійної програми. Крім цього, із вказаного документа можна автоматизовано отримати:

- рівень вищої освіти;
- кількість кредитів ЄКТС;
- загальну кількість годин;
- кількість аудиторних годин на тиждень (для денної форми навчання);
- кількість годин самостійної роботи студента на тиждень (для денної форми навчання);
- рік підготовки;
- семестр;
- кількість лекційних годин;
- кількість практичних (семінарських) годин;
- кількість лабораторних годин.

Отримавши ці дані нам необхідно їх записати у файл-шаблон робочої програми. Для цього використовуємо маркери у форматі «[маркер]», де у квадратних дужках записано параметр для заміни. У лістингу 2 наведено фрагмент коду, що виконує описану дію.

Лістинг 2: Заміна маркерів.

```

1 For i = LBound(markers) To UBound(markers)
2   With ActiveDocument.Content.Find
3     .Text = markers(i)
4     .Replacement.Text = replacements(i)
5     .Execute Replace:=wdReplaceAll
6   End With
7 Next i

```

Далі з освітньо-професійної програми автоматизовано отримуємо:

- інтегральну компетентність;
- загальні компетентності та їхні шифри;
- фахові компетентності та їхні шифри;
- програмні результати навчання та їхні шифри.

Для запису компетентностей у файлі-шаблоні робочої програми створюємо закладку «компетентності» (Вставлення | Посилання | Закладка). Це дозволяє автоматизовано змінювати текст закладки згідно з вимогами ОПП (лістинг 3). Зауважимо, що вибіркові дисципліни не відображаються у матриці відповідності програмних компетентностей освітнім компонентам та матриці забезпечення програмних результатів навчання відповідними освітніми компонентами ОПП, тому доцільно перенести всі компетентності та програмні результати навчання з подальшим редагуванням. Аналогічно до компетентностей через закладку «зміст» формуємо власне зміст дисципліни.

Лістинг 3: Фрагмент роботи із закладкою «компетентності».

```

1 Set bm = ThisDocument.Bookmarks("компетентності")
2 Set rng = bm.Range
3   ' Видаляємо попередній текст закладки
4 rng.Text = ""
5   ' Додаємо нові абзаци з текстом
6 rng.InsertAfter "Загальні компетентності:" & vbCrLf
7 For i = 1 To кількість_зк
8   If зк(i).наявна Then
9     rng.InsertAfter Trim(зк(i).код) & ". " & Trim(зк(i).текст) &
        vbCrLf
10  End If
11 Next i

```

Програмні результати навчання та їхні шифри записуємо у відповідну таблицю, попередньо видаливши застарілі дані (лістинг 4). Для таблиці з очікуваними результатами навчання з дисципліни готуємо порожню таблицю та вписуємо тільки шифри, які відповідають програмним результатам.

Лістинг 4: Фрагмент роботи із таблицею «Програмні результати навчання».

```

1   ' -- Отримуємо посилання на таблицю --
2 Set tbl1 = ThisDocument.Tables(tblIndex1) 'РН
3   ' -- Видаляємо всі рядки, крім першого --
4 For i = tbl1.Rows.Count To 2 Step -1
5   tbl1.Rows(i).Delete
6 Next i
7   ' -- Додаємо нові рядки після першого --
8 For i = 1 To кількість_рн
9   If рн(i).наявна Then

```

```

10      tbl1.Rows.Add
11      tbl1.cell(tbl1.Rows.Count, 1).Range.Text = Trim(рн(i).текст)
12      tbl1.cell(tbl1.Rows.Count, 2).Range.Text = Trim(рн(i).код)
13  End If
14  Next i

```

Ми не можемо автоматизовано наповнити таблицю «Очікувані результати навчання», оскільки це вимагає інтелектуальної роботи викладача, але аналогічно до лістингу 4 можемо підготувати порожню таблицю на записати шифри програмних результатів навчання. Крім цього, програмно здійснюємо форматування тексту (вирівнювання, напівжирний та ін.), що також економить час при завершальному оформленні документа.

Наступною можливістю для автоматизації є формування та наповнення таблиці, яка відображає розподіл балів за темами кожного модуля (табл. 1).

Таблиця 1.

Розподіл балів, які отримують здобувачі вищої освіти

Поточне оцінювання та самостійна (індивідуальна) робота				Модульна контрольна робота	Сума
T_1	T_2	...	T_n	50	100
k_1	k_2	...	k_n		

Тут n — кількість тем певного модуля, $k_1, k_2, \dots, k_n$ — бали, які можуть отримати здобувачі вищої освіти за вказану тему. Отже, ми можемо створити таблицю, врахувавши кількість тем, а також задати рівномірний розподіл балів за темами із можливістю подальшого редагування викладачем. Оскільки структура таблиці досить складна, то ми спочатку об'єднуємо всі її комірки у одну, видаляємо застарілий текст, а потім формуємо нову таблицю (лістинг 5).

Лістинг 5: Фрагмент роботи із таблицею «Розподіл балів».

```

1  Set tbl = ThisDocument.Tables(tblIndex) 'Оновлюємо посилання
2  ' -- Крок 1. Об'єднати всі клітинки в одну --
3  tbl.cell(1, 1).Merge tbl.cell(tbl.Rows.Count, tbl.Columns.Count)
4  ' -- Крок 2. Видалити весь вміст --
5  tbl.cell(1, 1).Range.Text = ""
6  ' -- Крок 3. Розбити таблицю заново --
7  tbl.Rows(1).Cells(1).Split NumRows:=3, NumColumns:=модулі(n).
   кількість_тем + 2

```

Розглянемо ще одну підзадачу — формування таблиці «Структура навчальної дисципліни» (табл. 2).

Оскільки табл. 2 має об'єднання по вертикалі, то видалення рядків аналогічно до лістингу 4 викликає помилку виконання. Тому ми розділимо її на таблицю із заголовками та іншу частину, яку видаляємо. Далі додаємо необхідну кількість рядків, у залежності від кількості тем, та обчислюємо попередній рівномірний розподіл годин за темами (лістинг 6).

Таблиця 2.

Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
		лек.	пр.	лаб.	інд.	с.р.
1	2	3	4	5	6	7
Модуль 1.						
Тема 1.1.						
Тема 1.2.						
...						
Разом за модулем						
Модуль 2.						
Тема 2.1.						
Тема 2.2.						
...						
Разом за модулем						
Разом						

Лістинг 6: Фрагмент роботи із таблицею «Структура навчальної дисципліни».

```

1 Set tbl = ThisDocument.Tables(tblIndex)
2 tbl.cell(6, 1).Select ' Нижче заголовків
3 Selection.SplitTable
4 ThisDocument.Tables(tblIndex + 1).Delete
5 p = 0
6 For i = 1 To кількість_модулів
7     tbl.Rows.Add 'Модуль i
8     tbl.cell(tbl.Rows.Count, 1).Range.Text = Trim(модулі(i).назва)
9     For j = 1 To модулі(i).кількість_тем
10        p = p + 1
11        tbl.Rows.Add 'Тема j
12        tbl.cell(tbl.Rows.Count, 1).Range.Text = Trim(теми(p).номер) & "
13            " & Trim(теми(p).назва)
14        tbl.cell(tbl.Rows.Count, 2).Range.Text = години_теми(p).усього
15        tbl.cell(tbl.Rows.Count, 3).Range.Text = години_теми(p).лек
16        If наявні_пр Then
17            tbl.cell(tbl.Rows.Count, 4).Range.Text = години_теми(p).пр
18        End If
19        If наявні_лаб Then
20            tbl.cell(tbl.Rows.Count, 5).Range.Text = години_теми(p).лаб
21        End If
22        If наявні_інд Then
23            tbl.cell(tbl.Rows.Count, 6).Range.Text = години_теми(p).інд
24        End If
25        tbl.cell(tbl.Rows.Count, 7).Range.Text = години_теми(p).ср
26    Next j
27    tbl.Rows.Add 'Разом за модулем i
28    tbl.cell(tbl.Rows.Count, 1).Range.Text = "Разом за модулем"
29 Next i

```

Крім описаних підзадач слід відмітити можливості автоматизації роботи викладача, пов'язані з форматуванням тексту, виділенням певних фрагментів та ін. Це вимагає певних затрат часу у процесі розробки, проте потім економить його при експлуатації системи.

6. Висновки. Описана інформаційна система покликана економити час викладача, беручи на себе рутинні завдання при створенні робочої програми. На нашу думку, цю та подібні системи доцільно розробляти, зіставляючи витрати часу на їхнє розроблення та економію часу при їх експлуатації. Описана система містить більше 700 рядків коду, тобто її розроблення вимагає значних часових ресурсів. Тим не менше, при створенні нових освітньо-професійних програм виникає необхідність у розробленні значної кількості робочих програм. У цьому випадку використання інформаційної системи «Цифровий асистент викладача ЗВО» вважаємо за доцільне.

У подальшому систему можна розвивати як інструмент автоматизованого аналізу розроблених робочих програм та виявлення у них невідповідностей ОПП та НП.

Список використаної літератури

1. Povidaichyk, M. M., Sharkadi, M. M., Katsala, R. A., Hodia, R. M., & Yanchii, I. V. (2025). Development of the Framework of the Information System "Computer Simulator for Financial Mathematics". *Scientific Bulletin of Uzhhorod University. Series: Mathematics and Informatics*. 46(1), 247–255. [https://doi.org/10.24144/2616-7700.2025.46\(1\).247-255](https://doi.org/10.24144/2616-7700.2025.46(1).247-255) [in Ukrainian].
2. Povidaichyk, M. M., Povidaichyk, O. S., Herych, M. S., & Popovych, A. O. (2022). *Development of Automated Learning and Knowledge Control Systems for Pupils and Students: A Textbook*. Uzhhorod: UzhNU Publishing House "Hoverla" [in Ukrainian].
3. Shcherbyna, O. A. (2023). Automation of Creation, Filling, and Administration of Course Categories on the MOODLE Website. *Information Technologies and Learning Tools*, 93(1), 178–198. <https://doi.org/10.33407/itlt.v93i1.5117>
4. Panovyyk, U. P., Panovyyk, R. R., Petriv, R. I., & Fedyna, B. I. (2024). Automated Support System for the Educational Process with Dynamic Formation of Elective Components. *Scientific Notes*, 1(68), 94–106.
5. Kosiuk, M. M., Bilovskyi, K. E., & Lysak, V. M. (2023). Automated Information System for Managing a Higher Education Institution "Electronic University". *Information Technologies and Learning Tools*, 93(1), 96–116. <https://doi.org/10.33407/itlt.v93i1.5107>

Povidaichyk M. M., Skliar A. L., Hodia R. M. Development of Certain Modules of the Information System "Digital Assistant for a Higher Education Teacher".

A developed module of the information system "Digital Assistant for a Higher Education Teacher" is described, which allows automating the development of a syllabus (working program of a discipline). The general workflow of the system and its main procedures are presented. The possibilities of automation based on data from the educational and professional program, the curriculum, and other documents are considered. The calculation of those numerical parameters of the syllabus that can be automatically obtained from existing files is described. The work with Word objects that make it possible to manage ranges, bookmarks, and tables is presented. The potential for further system development as a tool for automated analysis of developed syllabuses and identification of their inconsistencies with the educational and professional program and curriculum is indicated.

Keywords: information system, automation, template file, VBA.

Одержано 14.09.2025