

DOI: <https://doi.org/10.36910/6775-2524-0560-2026-63-29>

УДК 004.92

Ратушняк Тетяна Володимирівна, к.ф.-м.н., доцент

<https://orcid.org/0000-0002-9353-4268>

Гладченко Оксана Вікторівна, к.пед.н., доцент

<https://orcid.org/0000-0002-1668-633X>

Державний податковий університет, м. Ірпінь, Україна

ПРОГРАМУВАННЯ КОМП'ЮТЕРНОЇ ГРАФІКИ ВБУДОВАНИМИ ЗАСОБАМИ КОМП'ЮТЕРНИХ ПРОГРАМ ПАКЕТУ ADOBE: МЕТОДОЛОГІЯ АВТОМАТИЗАЦІЇ У ГРАФІЧНОМУ ДИЗАЙНІ АБО ВЕБДИЗАЙНІ

Ратушняк Т.В., Гладченко О.В. Програмування комп'ютерної графіки вбудованими засобами комп'ютерних програм пакету Adobe: методологія автоматизації у графічному дизайні або вебдизайні. У статті досліджено методологію програмування комп'ютерної графіки вбудованими засобами програмного пакету Adobe як стратегічний інструмент підвищення продуктивності фахівця в умовах цифрової економіки. Встановлено, що сучасні графічні редактори трансформувалися у складні програмні екосистеми, інтегровані у високоавтоматизовані ланцюги створення контенту для UI/UX-дизайну, веброзробки та медіа-індустрії. Авторами проведено ґрунтовний порівняльний аналіз основних мов сценаріїв: JavaScript (платформи ExtendScript та Unified Extensibility Platform), VBScript та AppleScript. Особливу увагу приділено технологічному переходу до сучасної платформи Unified Extensibility Platform, що базується на стандартах ES2018+ та асинхронних моделях програмування. У роботі враховано актуальний технологічний контекст, зокрема офіційну депрекацію мови VBScript з боку Microsoft, що робить перехід на JavaScript безальтернативним для професійного проєктування. Експериментально обґрунтовано, що впровадження авторських скриптів для автоматизації рутинних операцій забезпечує скорочення часу виконання завдань та знижує когнітивну втому дизайнера. У статті наведено практичні приклади програмних кодів, які демонструють реалізацію математичних та алгоритмічних основ графіки. Наукова новизна дослідження полягає у формуванні парадигми «обчислювального дизайну», де програмування розглядається як фундаментальна компетенція, що перетворює дизайнера на креативного технолога. Визначено перспективи інтеграції вбудованих засобів автоматизації з агентними системами штучного інтелекту для створення автономних дизайнерських рішень та динамічної персоналізації візуального контенту на основі великих даних.

Ключові слова: графічний дизайн, вебдизайн, UI/UX-дизайн, автоматизація, пакет Adobe, Javascript, UXP, VBScript, програмування комп'ютерної графіки.

Ratushnyak T., Hladchenko O. Computer graphics programming using the built-in tools of Adobe software: methodology of automation in graphic design or web design. The article explores the methodology of computer graphics programming using Adobe software suite's built-in tools as a strategic instrument for enhancing professional productivity within the digital economy. It is established that modern graphic editors have evolved into complex software ecosystems integrated into highly automated content creation workflows for UI/UX design, web development, and the media industry. The authors present a comprehensive comparative analysis of the primary scripting languages: JavaScript (ExtendScript and Unified Extensibility Platform), VBScript, and AppleScript. Particular emphasis is placed on the technological transition to the modern Unified Extensibility Platform, which utilizes ES2018+ standards and asynchronous programming models. The research considers the current technological landscape, including the official deprecation of VBScript by Microsoft, rendering the shift to JavaScript imperative for professional engineering. It is experimentally demonstrated that implementing custom scripts for routine task automation significantly reduces execution time and mitigates designer cognitive fatigue. The article provides practical code examples illustrating the implementation of mathematical and algorithmic graphics fundamentals. The scientific novelty of the study lies in the formulation of the «Computational Design» paradigm, wherein programming is treated as a core competency that transforms a designer into a creative technologist. The paper identifies prospects for integrating built-in automation with Agentic AI systems to facilitate autonomous design solutions and dynamic visual content personalization based on Big Data.

Keywords: graphic design, web design, UI/UX design, automation, Adobe suite, JavaScript, UXP, VBScript, computer graphics programming.

Постановка проблеми. У сучасній цифровій економіці індустрія графічного дизайну, вебдизайну та проєктування інтерфейсів (UI/UX-дизайну) переживає фундаментальну трансформацію. Сучасні графічні інструменти, такі як комп'ютерні програми пакету Adobe, перетворилися з інструментів для маніпуляції пікселями на складні програмні екосистеми, інтегровані у високоавтоматизовані ланцюги створення контенту. Проблема програмування комп'ютерної графіки вбудованими засобами набуває критичного значення для проєктування цифрових продуктів (зокрема, вебсайтів, електронних видань, іммерсивного контенту, медіа контенту), де швидкість ітерацій та персоналізація контенту визначають успіх на ринку.

Практична значущість цієї проблеми підкріплюється даними 2025–2026 років [1]: близько 88% організацій у креативному секторі вже інтегрували штучний інтелект та засоби автоматизації у свої робочі процеси. У сфері UI/UX-дизайну автоматизація рутинних завдань, таких як підготовка асетів для різних роздільних здатностей екранів чи генерація варіантів інтерфейсних елементів,

дозволяє скоротити час виведення продукту на ринок на 30–50%. Скриптування дозволяє реалізувати концепцію безконтактної обробки («zero-touch»), де графічні файли адаптуються під потреби веброзробки автоматично, без участі дизайнера у кожній окремій операції.

Науковий аспект проблеми полягає у формуванні парадигми «обчислювального дизайну» (Computational Design), де програмування стає фундаментальною компетенцією фахівця. Для фахівців із графічного проєктування це означає перехід від пасивного використання інтерфейсу до активного керування алгоритмами, що лежать в основі графічних інструментів. Такий підхід забезпечує інтеграцію знань з інформатики та дизайну, формуючи глибоке розуміння архітектури цифрових медіасистем для розв'язання складних професійних завдань.

Аналіз останніх досліджень і публікацій. Ґрунтовною працею з автоматизації проєктування у програмних продуктах пакету Adobe є навчальний посібник Р. Ралстона [2], присвячений розробці сценаріїв для оптимізації професійної роботи дизайнерів у Photoshop, Illustrator та InDesign.

Проблеми автоматизації графічних процесів активно досліджуються у контексті дизайну, зокрема: у видавничій справі – авторами І. М. Стельмах, О. В. Зоренко [3], О. І. Пушкар [4], А. В. Бізюк [5], В. Є. Климнюк [6]; у дизайні одягу та аксесуарів – В. І. Чупринка [7], С. Є. Каменець [8] та інших напрямках дизайну [9, 10].

Питання автоматизації у контексті вебдизайну розглянуто у роботі Ю. В. Єфімова [9] та колективній монографії [10] за авторством науковців: О. В. Вовк, І. Б. Чеботарьова, Ж. В. Дейнеко. Технічні аспекти застосування мов програмування, зокрема JavaScript (ExtendScript), для автоматизації обробки великих масивів візуальних даних висвітлено у роботах А. В. Бізюка [5] та Н. С. Дідик [11]. Дослідження останніх років підкреслюють зростання використання сучасних платформ, таких як UXP (Unified Extensibility Platform), які дозволяють інтегрувати графічні редактори з вебтехнологіями через асинхронні моделі програмування. Концепція «обчислювального мислення» (Computational Thinking) розглядається як «місток між технічною точністю програмування та художньою свободою дизайнера [5]».

Виділення не вирішених раніше частин загальної проблеми. Попри наявність технічних інструкцій [12, 13, 14], залишається розрив між знаннями класичного програмування та їх практичним застосуванням у творчих індустріях. Графічні дизайнери або вебдизайнери часто не використовують повний потенціал вбудованих мов через відсутність методики адаптації коду під конкретні естетичні або функціональні завдання дизайну. Часто це пов'язано з вибором конкретної мови програмування залежно від операційної системи (Windows, Linux чи macOS) та специфіки професійних завдань, оскільки технологічна зміна платформ усередині продуктів Adobe створила певний методичний вакуум для фахівців. Тому вибір мови програмування потребує детального аналізу.

Формулювання мети дослідження. Метою статті є обґрунтування методики використання вбудованих засобів програмування як ключового інструменту підвищення продуктивності фахівця з графічного проєктування. Основним завданням є порівняльний аналіз мов програмування, що підтримуються в комп'ютерних програмах пакету Adobe (JavaScript, VBScript, AppleScript), та демонстрація їхнього практичного застосування для автоматизації рутинних завдань.

Виклад основного матеріалу дослідження. Для автоматизації завдань у середовищі Adobe Photoshop фахівець може використовувати три основні мови програмування, кожна з яких має свою сферу застосування:

— JavaScript (ExtendScript та UXP) є найбільш універсальним вибором. ExtendScript (.jsx) базується на стандарті ES3 і є кросплатформною мовою (підтримується для Windows та macOS). Сучасна платформа UXP (.psjs) пропонує асинхронне програмування на базі ES2018+. JavaScript має найбільш розвинену об'єктну модель DOM, що дозволяє маніпулювати шарами, масками та фільтрами. Це пріоритетний напрямок розвитку Adobe;

— VBScript використовується виключно в ОС Windows. Взаємодія з Adobe Photoshop відбувається через COM-об'єкти. Ця мова приваблює фахівців своєю простотою та інтеграцією з іншими Windows-додатками (наприклад, для передачі даних з MS Excel безпосередньо у Adobe Photoshop). Проте вона обмежена однією платформою. Іншою проблемою є те, що через офіційну депрекацію з боку Microsoft у 2026 році, використання VBScript для нових систем не рекомендується, оскільки підтримку мови буде поступово видалено з ОС Windows;

— AppleScript застосовується лише для операційної системи macOS. Вона використовує природні мовні метафори (наприклад, tell application «Photoshop» to quit), що робить її легкою для

опанування дизайнерами без глибокого технічного бекграунду, але менш гнучкою порівняно з JavaScript.

Приклад 1. Розглянемо практичний приклад автоматизації: масштабування та поворот документа. Завдання реалізується з використанням математичних і алгоритмічних основ двовимірної графіки (зокрема, афінні перетворення) [15] двома різними мовами програмування.

Розглянемо код мовою JavaScript (ExtendScript) (табл. 1). Цей варіант забезпечує професійну кросплатформну автоматизацію.

Таблиця 1. Код автоматичного масштабування та повороту документа мовою JavaScript

```
// Отримання активного документа
var doc = app.activeDocument;
// Параметри: ширина 800px, висота 600px, поворот 90 градусів
var newWidth = 800;
var newHeight = 600;
var rotationAngle = 90;
// Виконання операцій через методи об'єкта Document
doc.resizeImage(UnitValue(newWidth, "px"), UnitValue(newHeight, "px"));
doc.rotateCanvas(rotationAngle);
```

Розглянемо код мовою VBScript (табл.2). Цей варіант корисний для фахівців, які працюють в операційній системі Windows та потребують швидкої інтеграції з системними скриптами.

Таблиця 2. Код автоматичного масштабування та повороту документа мовою VBScript

```
' Створення об'єкта додатка Photoshop
Dim appRef, docRef
Set appRef = CreateObject("Photoshop.Application")
' Отримання посилання на активний документ
Set docRef = appRef.ActiveDocument
' Масштабування зображення (параметри: ширина, висота)
docRef.ResizeImage 800, 600
' Поворот полотна на 90 градусів
docRef.RotateCanvas 90
' Зміна колірного режиму на RGB (2 відповідає режиму RGB)
docRef.ChangeMode 2
```

Приклад 2. Розглянемо ще один практичний приклад автоматизації: автоматичний поворот зображення за годинниковою стрілкою.

Розглянемо код мовою JavaScript (ExtendScript) (табл. 3). Цей скрипт можна зберегти з розширенням .jsx і запускати безпосередньо у програмі Adobe Photoshop (через File -> Scripts -> Browse) або через Adobe ExtendScript Toolkit.

Таблиця 3. Код для повороту зображення за годинниковою стрілкою мовою JavaScript

```
// Перевіряємо, чи є відкриті документи
if (app.documents.length > 0) {
    var doc = app.activeDocument;
    // Поворот на 90 градусів за годинниковою стрілкою
    // Доступні варіанти: Direction.CLOCKWISE, Direction.COUNTERCLOCKWISE,
    Direction.ONEHUNDREDEIGHTY
    doc.rotateCanvas(Direction.CLOCKWISE);
    alert("Зображення успішно повернуто на 90°!");
} else {
    alert("Немає відкритих документів для повороту.");
}
```

Розглянемо код мовою VBScript (табл. 4). Цей код можна зберегти у файл з розширенням .vbs і запустити в операційній системі Windows подвійним кліком. Він самостійно підключиться до запущеного файлу формату .psd (Photoshop) або відкриє його, — і виконає поворот.

Таблиця 4. Код для повороту зображення за годинниковою стрілкою мовою VBScript

```
Dim appRef, docRef
' Намагаємося підключитися до Photoshop
On Error Resume Next
Set appRef = CreateObject("Photoshop.Application")
On Error GoTo 0
If Not appRef Is Nothing Then
' Перевіряємо, чи є відкриті документи
  If appRef.Documents.Count > 0 Then
    Set docRef = appRef.ActiveDocument
' Поворот на 90 градусів за годинниковою стрілкою
' 1 = Direction.CLOCKWISE (за годинниковою)
' 2 = Direction.COUNTERCLOCKWISE (проти годинникової)
    docRef.RotateCanvas 1
    MsgBox "Зображення успішно повернуто на 90°!", vbInformation, "Готово"
  Else
    MsgBox "Немає відкритих документів у Photoshop.", vbExclamation, "Помилка"
  End If
Else
  MsgBox "Не вдалося запустити або знайти Photoshop.", vbCritical, "Помилка"
End If
```

Впровадження таких мікропрограм (скриптів) у практику графічного дизайнера або вебдизайнера дає кілька стратегічних переваг:

— масштабованість, яку можуть забезпечити цикли (for у JS або For Each у VBS), — фахівець може обробити сотні файлів за лічені хвилини. Час виконання рутинних завдань скорочується в середньому на 56%;

— точність, яку забезпечує алгоритмічний підхід і виключає помилки ручного введення параметрів роздільної здатності чи розмірів, що критично для додрукарської підготовки;

— зниження когнітивного навантаження, оскільки автоматизація mundane-завдань (рутинних дій) дозволяє дизайнеру зосередитися на творчій складовій проекту. Дослідження показують [1], що використання скриптів знижує когнітивну втому персоналу на 22%.

У таблиці 5 наведено переваги та недоліки мов програмування комп'ютерної графіки за критеріями: кросплатформність (операційна система), складність, гнучкість.

Таблиця 5. Переваги та недоліки мов програмування комп'ютерної графіки

Критерій	JavaScript	VBScript
Операційна система	Windows / macOS	Windows
Складність	Середня (ООП)	Низька (процедурна)
Гнучкість DOM	Найвища	Обмежена

Висновки. Програмування комп'ютерної графіки вбудованими засобами цифрових графічних інструментів є необхідною умовою професійного зростання графічного дизайнера або вебдизайнера, яке проявляється у таких набутих компетентностях:

— вміння здійснювати вибір мови програмування, яке залежить від екосистеми: JavaScript залишається професійним стандартом завдяки кросплатформності, тоді як VBScript та AppleScript є ефективними інструментами для специфічних завдань у межах Windows або macOS;

— вміння використовувати скрипти, які дозволяють фахівцю з графічного проектування автоматизувати понад 90% повторюваних дій, значно підвищуючи ROI творчих операцій;

— знання вбудованих засобів програмування, яке перетворює дизайнера на «креативного технолога», здатного проектувати складні автоматизовані системи обробки візуального контенту.

Перспективи подальших досліджень. Подальші наукові дослідження будуть спрямовані на вивчення інтеграції штучного інтелекту в архітектуру UXР-плагінів. Перспективним є створення гібридних систем, де сценарії мовою JavaScript керують не лише функціями редактора, а й взаємодіють із зовнішніми неймережами для автоматичної генерації макетів та персоналізації контенту на основі великих даних. Планується дослідити можливості інтеграції хмарних API Photoshop із генеративними неймережами для створення «автономних дизайнерських агентів».

Список бібліографічного опису

1. PwC's 2026 Digital Trends in Operations: How AI Reinvents Enterprise Performance. URL: <https://www.pwc.com/us/en/services/consulting/supply-chain-operations/library/digital-trends-operations-survey.html> (дата звернення: 17.05.2026).
2. Ralston R. *The Designer's Apprentice: Automating Photoshop, Illustrator, and InDesign in Adobe Creative Suite 3*. Berkeley : Adobe Press, 2007. 256 p.
3. Стельмах І. М., Зоренко О. В., Осипова Т. Г. Апаратні та програмні засоби виготовлення електронних оригінал-макетів журнальної продукції. *Технологія і техніка друкарства*. 2011. Вип. 1. С. 14–24. URL: http://nbuv.gov.ua/UJRN/Titd_2011_1_6 (дата звернення: 17.05.2026).
4. Вирішення завдань поліграфічного виробництва в умовах концептуальної невизначеності : монографія / А. С. Гордєєв, Є. М. Грабовський, С. О. Назарова та ін. ; за заг. ред. О. І. Пушкаря. Харків : ХНЕУ ім. С. Кузнеця, 2024. 213 с.
5. Бізюк А. В. Використання скриптів у Photoshop в межах навчальної дисципліни «ПЗКВС» // Поліграфічні, мультимедійні та web-технології. *Сучасні тренди : колективна монографія*. Харків : ТОВ «Друкарня Мадрид», 2025. Т. 1. С. 44–71. DOI: <https://doi.org/10.30837/PMW.2025.T1.044>.
6. Технології комп'ютерного дизайну : методичні рекомендації до самостійної роботи студентів спеціальності 186 "Видавництво та поліграфія" першого (бакалаврського) рівня / уклад. В. Є. Климиук. Харків : ХНЕУ ім. С. Кузнеця, 2018. 47 с.
7. Чупринка В. І., Чупринка Н. В., Корогод Г. О., Науменко Б. В., Упіров І. С. Автоматизоване проєктування деталей виробів дрібної шкіргалантереї. *Наука і техніка сьогодні*. 2026. № 3 (57). С. 3042–3055. DOI: [https://doi.org/10.52058/2786-6025-2026-3\(57\)-3042-3055](https://doi.org/10.52058/2786-6025-2026-3(57)-3042-3055)
8. Каменець С. Є., Васютинська В. В. Використання сучасних методів візуального дизайну та просторового моделювання для створення шкіргалантерейних виробів. *Вісник Хмельницького національного університету*. 2020. № 4. С. 199–205.
9. Єфімов Ю. В. Комп'ютерна графіка: Adobe двома руками : навч. посіб. Київ : Київ. ун-т ім. Б. Грінченка, 2018. 120 с.
10. Поліграфічні, мультимедійні та web-технології. Сучасний стан : монографія / редкол.: О. В. Вовк, І. Б. Чеботарьова, Ж. В. Дейнеко. Харків : ТОВ «Друкарня Мадрид», 2023. 304 с. DOI: <https://doi.org/10.30837/978-617-8254-16-2>
11. Дідик Н. С., Бізюк А. В. Використання Java Script в програмі Adobe Photoshop. *Інформаційні системи та технології*. 2014. С. 188–189. URL: <http://openarchive.nure.ua/handle/document/6460> (дата звернення: 17.05.2026).
12. Adobe-CEP/CEP-Resources. Photoshop Scripting. GitHub. URL: <https://www.adobe.com/content/dam/acom/en/devnet/photoshop/pdfs/photoshop-cc-javascript-ref-2020.pdf> (дата звернення: 17.05.2026).
13. Ратушняк Т. В., Гладченко О. В., Омельчук А. А., Вишемірська Я. С. Технологія створення віртуальної сферичної 3d-панорами для соціальної мережі Facebook. *Прикладні питання математичного моделювання*. 2023. Т. 6, № 1. С. 124–133. DOI: <https://doi.org/10.32782/mathematical-modelling/2023-6-1-14>
14. Комлев В. Вчимося писати скрипти для Photoshop. Komliev.Studio. URL: <https://komliev.studio/p/practice/photoshop-scripting.html> (дата звернення: 17.05.2026).
15. Пічугін М. Ф., Канкін І. О., Воротніков В. В. Комп'ютерна графіка : навч. посіб. Київ : Центр учбової літератури, 2013. 346 с.

References

1. PwC. (2026). *PwC's 2026 Digital Trends in Operations: How AI Reinvents Enterprise Performance*. Retrieved May 17, 2026, from <https://www.pwc.com/us/en/services/consulting/supply-chain-operations/library/digital-trends-operations-survey.html>
2. Ralston, R. (2007). *The Designer's Apprentice: Automating Photoshop, Illustrator, and InDesign in Adobe Creative Suite 3*. Adobe Press. 256 p.
3. Stelmakh, I. M., Zorenko, O. V., & Osypova, T. H. (2011). Aparatni ta programni zasoby vyhotovlennia elektronnykh oryiginal-maketiv zhurnalnoi produktsii [Hardware and software tools for the production of electronic master layouts of magazine products]. *Tekhnolohiia i Tekhnika Drukarstva [Technology and Technique of Typography]*, (1), 14–24. http://nbuv.gov.ua/UJRN/Titd_2011_1_6
4. Hordiciev, A. S., Hrabovskyi, Ye. M., & Nazarova, S. O. (2024). *Vyryshennia zavdan polihrafichnogo vyrobnytstva v umovakh kontseptualnoi nevyznachenosti* [Solving problems of printing production in conditions of conceptual uncertainty] (O. I. Pushkar, Ed.). Simon Kuznets Kharkiv National University of Economics. 213 p.
5. Biziuk, A. V. (2025). Vykorystannia skryptiv u Photoshop v mezhakh navchalnoi dystsypliny "PZKVS" [The use of scripts in Photoshop within the academic discipline "PZKVS"]. In *Polihrafichni, multymediini ta web-tekhnologii. Suchasni trendy [Printing, multimedia and web technologies. Modern trends]* (Vol. 1, pp. 44–71). Madrid Printing House LLC. <https://doi.org/10.30837/PMW.2025.T1.044>
6. Klymniuk, V. Ye. (Comp.). (2018). *Tekhnolohii komp'uternoho dyzainu: metodychni rekomendatsii do samostiinoi*

- roboty studentiv spetsialnosti 186 "Vydavnytstvo ta polihrafiia" pershoho (bakalabrskoho) rivnia [Computer design technologies: guidelines for independent work of students of specialty 186 "Publishing and Printing" of the first (bachelor's) level]. Simon Kuznets Kharkiv National University of Economics. 47 p.
7. Chuprynka, V. I., Chuprynka, N. V., Korohod, H. O., Naumenko, B. V., & Upirov, I. S. (2026). Avtomatyzovane proiektuvannya detalei vyrobiv drubnoi shkirhalanterei [Automated design of parts for small leather goods]. *Nauka i Tekhnika Sohodni [Science and Technology Today]*, (3), 3042–3055. [https://doi.org/10.52058/2786-6025-2026-3\(57\)-3042-3055](https://doi.org/10.52058/2786-6025-2026-3(57)-3042-3055)
8. Kamenets, S. Ye., & Vasiutynska, V. V. (2020). Vykorystannia suchasnykh metodiv vizualnoho dyzainu ta prostorovoho modeliuvannya dlia stvorennia shkirhalantereynykh vyrobiv [Using modern methods of visual design and spatial modeling to create leather goods]. *Visnyk Khmelnytskoho Natsionalnoho Universytetu [Herald of Khmelnytskyi National University]*, (4), 199–205.
9. Yefimov, Yu. V. (2018). *Komp'uterna hrafiika: Adobe dvoma rukamy* [Computer graphics: Adobe with two hands]. Borys Grinchenko Kyiv University. 120 p.
10. Vovk, O. V., Chebotarova, I. B., & Deineko, Zh. V. (Eds.). (2023). *Polihrafichni, multymediini ta web-tekhnohii. Suchasnyi stan* [Printing, multimedia and web technologies. Current state]. Madrid Printing House LLC. 304 p. <https://doi.org/10.30837/978-617-8254-16-2>
11. Didyk, N. S., & Biziuk, A. V. (2014). Vykorystannia Java Script v prohrami Adobe Photoshop [Using JavaScript in Adobe Photoshop]. *Informatsiini Systemy ta Tekhnolohii [Information Systems and Technologies]*, 188–189. <http://openarchive.nure.ua/handle/document/6460>
12. Adobe Systems. (2020). *Photoshop Scripting*. Adobe Developer Connection. Retrieved May 18, 2026, from <https://www.adobe.com/content/dam/acom/en/devnet/photoshop/pdfs/photoshop-cc-javascript-ref-2020.pdf>
13. Ratushniak, T. V., Hladchenko, O. V., Omelchuk, A. A., & Vyshemirska, Ya. S. (2023). Tekhnolohiia stvorennia virtualnoi sferychnoi 3d-panoramy dlia sotsialnoi merezhi Facebook [Technology of creating a virtual spherical 3D panorama for the social network Facebook]. *Prykladni Pytannia Matematychnoho Modeliuvannya [Applied Issues of Mathematical Modeling]*, 6(1), 124–133. <https://doi.org/10.32782/mathematical-modelling/2023-6-1-14>
14. Komliev, V. (2024). *Vchymosia pysaty skrypty dlia Photoshop* [Learning to write scripts for Photoshop]. Komliev.Studio. Retrieved May 18, 2026, from <https://komliev.studio/p/practice/photoshop-scripting.html>
15. Pichuhin, M. F., Kankin, I. O., & Vorotnikov, V. V. (2013). *Komp'uterna hrafiika* [Computer graphics]. Center of Educational Literature. 346 p.

Історія статті:

Отримано: 08.05.2026 Доопрацьовано: 18.05.2026 Прийнято до друку: 23.05.2026 Опубліковано: 29.05.2026