



# СУЧАСНІ ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ

Матеріали

VI Всеукраїнської

науково-практичної інтернет-конференції

студентів, аспірантів та молодих вчених

за тематикою:  
«Сучасні комп'ютерні системи  
та мережі в управлінні»

30 листопада 2023 р.  
Хмельницький

Міністерство освіти і науки України  
Херсонський національний технічний університет  
Вінницький національний технічний університет  
Криворізький національний університет  
Кременчуцький національний університет ім. М. Остроградського  
Державний університет інтелектуальних технологій і зв'язку  
Львівський національний аграрний університет

*Матеріали*  
*VI Всеукраїнської*  
*науково-практичної інтернет-конференції*  
*студентів, аспірантів та молодих вчених*

**«Сучасні інформаційні системи та технології»**

за тематикою:  
**«Сучасні комп'ютерні системи та мережі в управлінні»**

30 листопада 2023 року

*Хмельницький*

УДК 330.111.66:005.8  
С 91

**С 91 Сучасні комп'ютерні системи та мережі в управлінні:** матеріали VI Всеукраїнської наук.-практ. Інтернет-конф. здобувачів вищої освіти та молодих вчених (30 листопада 2023 р., м. Хмельницький, м. Херсон) / за ред. А.А. Григорової. – Херсон: Книжкове видавництво ФОП Вишемирський В. С., 2023. – 260 с.

**ISBN 978-617-8187-04-0 (електронне видання)**

Доповіді наукової конференції містять результати наступних досліджень: сучасні тенденції розвитку інформаційних технологій; впровадження інновацій та сучасних технологій; моделювання та оптимізація систем управління; інформаційні технології в науці, освіті, економіці, логістиці, туристичній сфері, транспорті; новітні технології в енергетичних системах та в галузі енергозбереження.

Роботи друкуються в авторській редакції, в збірці максимально зменшено втручання в обсяг та структуру відібраних до друку матеріалів. Редакційна колегія не несе відповідальність за достовірність статистичної та іншої інформації, що надано в рукописах, та залишає за собою право не розподіляти поглядів деяких авторів на ті чи інші питання.

Збірник становить інтерес для студентів, аспірантів, викладачів та наукових працівників.

### **ПРОГРАМНИЙ КОМІТЕТ**

**Голова:** Григорова А.А. – к.т.н., доцент, в.о. завідувача кафедри КСтаМ ХНТУ.

**Заступник голови:** Козел В.М. – к.т.н., доцент, декан факультету Інформаційних технологій та дизайну ХНТУ.

#### **Члени комітету:**

Бісікало О.В. – д.т.н., професор, завідувач кафедри автоматизації та інтелектуальних інформаційних технологій ВНТУ.

Купін А. І. – д.т.н., професор, завідувач кафедри комп'ютерних систем та мереж Криворізький національний університет

Тригуба А.М. – д.т.н., професор, завідувач кафедри інформаційних систем та технологій ЛНАУ.

Конох І.С. – к.т.н., доцент кафедри ІУС КрНУ ім. М.Остроградського.

Данілець Є.В. – к.т.н., доцент кафедри інформаційних технологій

Міжнародний гуманітарний університет, м. Одеса.

Сидорук М.В. – к.т.н., доцент кафедри КСтаМ ХНТУ.

Карамушка М.В. – к.т.н., доцент кафедри КСтаМ ХНТУ.

Дідик О.О. – к.т.н., доцент кафедри КСтаМ ХНТУ.

Веселовська Г.В. – к.т.н., доцент кафедри КСтаМ ХНТУ.

Дроздова Є.А. – ст. викладач кафедри КСтаМ ХНТУ.

**УДК 330.111.66:005.8**

ISBN 978–617–8187–04–0 (електронне видання)

© Кафедра КСтаМ ХНТУ, 2023  
© ФОП Вишемирський В. С., 2023

## ЗМІСТ

|                                                                                                                                                         |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>СЕКЦІЯ 1. СУЧАСНІ ТЕНДЕНЦІЇ РОЗВИТКУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ</b> .....                                                                              | 9  |
| Алексеева Г.М., Горбатюк Л.В., Овсянніков О.С. Впровадження дистанційних технологій навчання: проблеми та рішення.....                                  | 10 |
| Глоба А. Р., Дмитрієв В.Г., Романюк О.Н. Аналіз векторних графічних редакторів .....                                                                    | 13 |
| Дергач А.Є., Сидорова М.Г. Аналіз підходів до вирішення задачі зменшення шуму у аудіо .                                                                 | 15 |
| Vladyslav Dzinziura, Andrii Kopp Towards the automation of project documentation maintenance using large language models .....                          | 17 |
| Капітон А.М., Земський Н.В. Використання інформаційних технологій в сфері музичного мистецтва .....                                                     | 19 |
| Kokidko B.S., Shushura, O.M. Application of fuzzy logic for analysis graph databases based on social networks.....                                      | 21 |
| Корніловська Н.В., Андрієвський І.І., Вишемирська С.В. Аналіз універсальних програм консолідації інформації з результатів автоматичного тестування..... | 23 |
| Корніловська Н.В., Дубонос І.О., Крупнов А.С. Використання сучасних комп'ютерних систем та інформаційних технологій в туристичній сфері України .....   | 26 |
| Мазур В.В. Глоба А.Р., Романюк О. Н. Формування зображення рельєфної поверхні тривимірних об'єктів .....                                                | 29 |
| Москалик Д.О., Антонюк Д.С. Аналіз розподілу складності задач при розробці програмного забезпечення з відкритим вихідним кодом .....                    | 31 |
| Нікітін Є.О., Каіров В.О. Розробка месенджера на мові програмування JavaScript .....                                                                    | 33 |
| Погорелова К.І., Єфімов Д.В. Професійна підготовка вчителів сучасні тенденції та проблеми .....                                                         | 35 |
| Позур М.Ю., Войтко В.В. Метапрограмування в .NET з використанням Expression.....                                                                        | 37 |
| Прус О.В., Майданюк В.П. Багатопроектні середовища та спільна розробка інтерактивних веб-інтерфейсів.....                                               | 39 |
| Разказов М.П., Павловський В.І. Аналіз методологій розробки програмного забезпечення TDD .....                                                          | 44 |
| Романюк О. Н., Станіславенко Є.Г., Шевченко О.О. Шейдерна технологія зафорбування ....                                                                  | 46 |
| Романюк О.Н., Форостяний А.Б. Використання інформаційних технологій у системах штучного інтелекту блокчейну та інтернет-речей.....                      | 48 |
| Романюк О. Н., Чехмestruc Р. Ю. Мазур В.В. Вибір методу зафарбовування для задач рендерингу .....                                                       | 51 |
| Скидан Д.К., Дідик О.О. Розробка комп'ютерної системи діагностики клавіатури Genesis THOR 300 TKL на основі Arduino.....                                | 52 |
| Складанюк О.О., Майданюк В.П. Редагування та розробка відеоігор у сучасній тенденції розвитку інформаційних технологій .....                            | 54 |
| Forkert P.P, Sydorova M.G. Challenges of using Golang as a foundation for new programming languages.....                                                | 55 |
| Чернюк С.В., Гайдаєнко О.В. Роль інформаційних систем в управлінні проектами.....                                                                       | 56 |

компіляції виразу в лямбда функцію є доволі затратним. Неправильне використання такого підходу може навпаки призвести до погіршення продуктивності роботи додатку.

#### **Перелік джерел посилання**

1. Ingebrigtsen E. Metaprogramming in C#: Automate your .NET development and simplify overcomplicated code / Einar Ingebrigtsen. – Birmingham, 2023. – 352 с. – (Packt Publishing).
2. Hazzard K. Metaprogramming in .NET / K. Hazzard, J. Brock. – New York, 2013. – 360 с. – (Manning).
3. Expression trees [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/dotnet/csharp/advanced-topics/expression-trees/>.
4. Warren M. Why is reflection slow? [Електронний ресурс] / Matt Warren. – 2016. – Режим доступу до ресурсу: <https://mattwarren.org/2016/12/14/Why-is-Reflection-slow/>.

УДК 681.5015:007

*Прус О.В., аспірант I курсу спеціальності  
«Інженерія програмного забезпечення»  
Майданюк В.П., канд. техн. наук, доцент  
кафедри програмного забезпечення*

## **БАГАТОПРОЕКТНІ СЕРЕДОВИЩА ТА СПІЛЬНА РОЗРОБКА ІНТЕРАКТИВНИХ ВЕБ-ІНТЕРФЕЙСІВ**

Вінницький національний технічний університет, Україна

### **Вступ**

Сучасне інформаційне суспільство надає величезний акцент на інтернет як найпоширеніше та динамічне середовище для взаємодії, споживання інформації та здійснення операцій. Ця зростаюча залежність від веб-технологій створює підвищену потребу у створенні веб-інтерфейсів, які не тільки ефективно взаємодіють з користувачами, але й можуть швидко адаптуватися до змінних потреб [1].

Однак розробка інтерактивних веб-інтерфейсів у багатопроектному середовищі вимагає від розробників більш відповідального та збалансованого підходу. З одного боку, це означає спільну роботу та координацію між численними учасниками проекту, від яких очікується стабільність та продуктивність. З іншого боку, інтерфейси повинні бути добре продуманими та ефективно реалізованими, щоб забезпечити задоволення різних категорій користувачів [1, 2].

Основною метою цієї статті є дослідження проблеми розробки інтерактивних веб-інтерфейсів у багатопроектних середовищах та пропозиція практичних підходів до її вирішення. Розглянуто модульність і перевикористання коду як ключові поняття у структуруванні розробки інтерфейсів, а також дослідимо особливості використання бібліотек для створення інтерактивних інтерфейсів. Це дослідження базується на аналізі попередніх досліджень і публікацій у сфері веб-розробки та інтерактивних інтерфейсів, а також на практичних випробуваннях та прикладах використання.

Результати дослідження не тільки розкриють потенціал більш ефективної, стабільної та зручної розробки інтерактивних веб-інтерфейсів, але й нададуть корисні рекомендації для розробників та команд, що працюють над покращенням якості своїх інтерфейсів у багатопроектних середовищах.

## Аналіз досліджень та постановка завдання

Розробка інтерактивних веб-інтерфейсів завжди була завданням, що стоїть перед розробниками та командами проектів. Попередні дослідження та публікації у цій галузі надали важливі відомості та вказівки для полегшення цього процесу. Перш ніж ми перейдемо до подальших аспектів нашого дослідження, проведемо аналіз ключових внесків та тенденцій, які знайшли відображення в попередніх дослідженнях [2, 3].

Попередні дослідження в цій галузі були проведені численними авторами та дослідниками. Деякі з важливих імен включають Натан Кертіс, який досліджував аспекти модульності та перевикористання коду у веб-розробці [4], Спінг Зген, чий внесок полягає в аналізі використання бібліотек для створення інтерактивних інтерфейсів [5], і Нікола Дж. Сміт, який досліджував вплив багатопроєктних середовищ на розробку інтерфейсів [6].

Натан Кертіс відзначив, що модульність та перевикористання коду є важливими складовими розробки ефективних інтерактивних інтерфейсів. Його дослідження показали, що правильна організація коду може значно полегшити процес розробки та забезпечити його стабільність [4].

Роботи Спінг Зген вказують на те, що використання бібліотек і фреймворків є дієвим способом спростити розробку інтерактивних інтерфейсів та забезпечити сумісність з різними платформами та браузерами [5].

Нікола Дж. Сміт зосередився на проблемах багатопроєктних середовищ і вказав на необхідність більш ефективної спільної роботи та управління версіями коду у веб-розробці [6].

Незважаючи на досягнені висновки, залишаються невирішені аспекти та відкриті питання. Наприклад, попередні дослідження не завжди враховують особливості конкретних багатопроєктних середовищ та різноманітні потреби користувачів. Також є потреба у більш глибокому розумінні того, як краще організувати спільну розробку інтерфейсів для досягнення оптимальних результатів.

**Постановка завдання.** Після аналізу попередніх досліджень та враховуючи актуальні висновки, ставимо перед собою наступні завдання:

Розглянути модульність та перевикористання коду у контексті інтерактивних веб-інтерфейсів.

Зробити аналіз особливостей використання бібліотек та фреймворків для створення інтерактивних інтерфейсів.

Зробити огляд кращих практик та підходів до спільної розробки в багатопроєктних середовищах.

На основі зібраної інформації дати практичні рекомендації щодо оптимізації процесів спільної розробки інтерактивних веб-інтерфейсів в багатопроєктних середовищах.

### Виклад основного матеріалу

Модульність в розробці інтерактивних веб-інтерфейсів полягає в розділенні коду на окремі компоненти, які можна розглядати як окремі модулі [7]. Це дозволяє створювати незалежні частини веб-інтерфейсу, які можна легко підтримувати і перевикористовувати в інших проєктах. Прикладами модульності можуть бути виділені компоненти для форм, меню, списків тощо. Кожен компонент може мати власну логіку та вигляд [8].

Перевикористання коду включає в себе можливість використовувати певний код, наприклад, компоненти чи модулі, у різних частинах проєкту або навіть у різних проєктах. Це допомагає зменшити дублювання коду, полегшує зміни та оновлення та робить розробку більш ефективною [9].

Використання модульності та перевикористання коду дозволяє забезпечити підтримку та розвиток інтерактивних веб-інтерфейсів у багатопроєктних середовищах, а також полегшує роботу розробників.

Використання бібліотек та фреймворків для створення інтерактивних веб-інтерфейсів включає в себе розгляд різних аспектів:

Архітектура: Кожна бібліотека або фреймворк має власну архітектуру, яка впливає на структуру проекту. Наприклад, бібліотеки React використовують Virtual DOM, Angular має модель даних і контролери, а Vue.js пропонує компонентний підхід. Розуміння архітектури допомагає вибрати підхід, що найкраще підходить до конкретного проекту [7].

Функціональність: Різні бібліотеки та фреймворки надають різні можливості для створення інтерактивних елементів. Деякі з них фокусуються на швидкості та ефективності, інші на зручності розробки. Розгляд функціональності допомагає вибрати інструмент, який відповідає потребам проекту [4].

Можливості розширення та плагіни: Більшість бібліотек дозволяють розширювати їхню функціональність за допомогою плагінів та розширень. Важливо вивчити, як це працює для конкретної бібліотеки та як це може бути використано у проекті [5].

Детальний аналіз цих аспектів допомагає розробникам вибрати найкращі бібліотеки та фреймворки для створення інтерактивних веб-інтерфейсів у своїх проектах.

Проведемо огляд найкращих практик та інноваційних підходів до спільної розробки інтерактивних веб-інтерфейсів в багатопроєктних середовищах:

Використання мікросервісної архітектури:

Мікросервіси - це невеликі, автономні компоненти, які можуть бути розроблені та підтримуватися окремими командами. Вони сприяють модульності та перевикористанню коду.

Контейнеризація (наприклад, Docker) дозволяє упаковувати мікросервіси з їх залежностями, забезпечуючи їх легку розгортання та масштабування.

Оркестрація (наприклад, Kubernetes) допомагає керувати та автоматизувати роботу з мікросервісами у багатопроєктних середовищах [10].

Безперервна інтеграція та доставка (CI/CD):

CI/CD пайплайни дозволяють автоматизувати тестування, збірку та розгортання коду. Це сприяє швидкій ітерації та забезпечує стабільну роботу інтерфейсів.

Автоматизоване тестування включає одиниць тестування, інтеграційні тести та тести на прийомку, що допомагає виявляти помилки на ранніх стадіях розробки [11].

Дизайн-системи та спільна бібліотека компонентів:

Дизайн-системи визначають стандарти дизайну та розмітки, забезпечуючи єдність вигляду та поведінки інтерфейсів.

Спільні бібліотеки компонентів містять готові рішення для створення інтерфейсів, що можуть бути перевикористані в різних проектах [12].

Керування залежностями та версіями:

Використовуйте системи управління версіями (наприклад, Git) для контролю за кодом та його змінами.

Встановіть правила семантичного версіювання для бібліотек та компонентів, щоб забезпечити сумісність та відслідковуваність версій [13].

Спільна комунікація та внутрішні інструменти:

Спільні чати та інструменти для співпраці (наприклад, Slack, Microsoft Teams) допомагають командам зберігати зв'язок та обмінюватися інформацією.

Віртуальні наради та робочі групи сприяють обговоренню технічних питань та прийняттю рішень [14].

Автоматизований моніторинг та аналіз даних:

Використовуйте інструменти моніторингу (наприклад, Prometheus, Grafana) для відстеження стану інтерфейсів та швидкого реагування на проблеми.

Аналіз даних та метрик допомагає зрозуміти, як користувачі взаємодіють з інтерфейсами та як можна їх покращити [15].

Ці практики та підходи спрямовані на поліпшення спільної розробки інтерактивних веб-інтерфейсів у багатопроєктних середовищах, забезпечуючи якість та ефективність роботи команди.

Практичні рекомендації щодо оптимізації процесів спільної розробки інтерактивних веб-інтерфейсів в багатопроектних середовищах:

Впровадження мікросервісної архітектури та контейнеризації:

Рекомендація: Використовуйте мікросервісну архітектуру для розділення функціональності інтерфейсу на невеликі, автономні компоненти. Враховуйте модульність та перевикористання коду при розробці окремих сервісів.

Обґрунтування: Дослідження показують, що мікросервіси сприяють модульності та підтримці перевикористання коду, що є важливими аспектами спільної розробки.

Розгляд CI/CD та автоматизованого тестування:

Рекомендація: Впроваджуйте безперервну інтеграцію та доставку (CI/CD) разом з автоматизованим тестуванням. Обов'язково включайте тести модульності та інтеграційні тести в процес розробки.

Обґрунтування: Висновки з попередніх досліджень показують, що CI/CD та автоматизоване тестування допомагають виявляти та виправляти помилки на ранніх стадіях розробки, зокрема в контексті модульної розробки.

Використання дизайн-систем та спільних бібліотек компонентів:

Рекомендація: Розробляйте та використовуйте дизайн-систему для визначення стандартів дизайну. Впроваджуйте спільні бібліотеки компонентів, які підтримують модульність та перевикористання.

Обґрунтування: Згідно з дослідженнями, використання дизайн-систем та спільних бібліотек допомагає підтримувати єдність вигляду та поведінки інтерфейсів, що сприяє модульності.

Вибір фреймворків та бібліотек:

Рекомендація: При виборі фреймворків та бібліотек для розробки інтерфейсу обов'язково враховуйте їх підтримку модульності та можливість перевикористання компонентів.

Обґрунтування: Наявність гнучкості та підтримки модульності в фреймворках та бібліотеках полегшує спільну розробку та забезпечує більшу реюзабельність коду.

Ці рекомендації базуються на наукових результатах та дослідженнях та спрямовані на поліпшення розробки інтерактивних веб-інтерфейсів в багатопроектних середовищах з урахуванням модульності та ефективного використання коду.

## Висновки

На основі виконаної роботи та аналізу досліджень у галузі спільної розробки інтерактивних веб-інтерфейсів в багатопроектних середовищах, ми надаємо наступні висновки:

Розглядаючи модульність і перевикористання коду в контексті інтерактивних веб-інтерфейсів, було визначено, що ці принципи є критичними для поліпшення ефективності та якості розробки. Розбиття інтерфейсу на компоненти та модулі спрощує спільну роботу команд та забезпечує легку перевикористаність коду.

Провівши аналіз особливостей використання бібліотек та фреймворків для створення інтерактивних інтерфейсів, визначено, що правильний вибір інструментів може значно полегшити розробку. Важливо обирати ті рішення, які підтримують модульність та забезпечують зручну інтеграцію в багатопроектне середовище.

Під час огляду кращих практик та підходів до спільної розробки в багатопроектних середовищах, було виявлено, що ефективна комунікація, автоматизація процесів та збереження єдності у дизайні та компонентах є важливими факторами успіху.

На основі отриманої інформації, було розроблено практичні рекомендації щодо оптимізації процесів спільної розробки інтерактивних веб-інтерфейсів в багатопроектних середовищах. Ці рекомендації включають в себе впровадження мікросервісної архітектури, CI/CD підходу, використання дизайн-систем та спільних бібліотек, а також ретельний вибір фреймворків та бібліотек.



В цілому, робота підкреслює важливість модульності, перевикористання коду, правильного вибору інструментів та дотримання кращих практик у спільній розробці інтерактивних веб-інтерфейсів. Надані рекомендації можуть сприяти поліпшенню ефективності та якості розробки в багатопроектних середовищах, допомагаючи розробникам та командам досягти кращих результатів.

#### Перелік джерел посилання

1. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
2. Robert S. Martin. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall. 23-40.
3. David Farley, Jez Hamblin. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley Professional. 172-182.
4. Nathan Curtis. (2010). *Modular Web Design: Creating Reusable Components for User Experience Design and Documentation*. New Riders. 186-210.
5. Yeping Zhen. (2022). The Interactive Design of Library Information Sharing in View of Network Communication Technology. *Advances in Multimedia*. 1-14.
6. N. J Smith, George Hagan, Denise Bower. (2011). Managing complex projects in multi-project environments. *University of Leeds*. 788-796
7. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
8. Модульне програмування в веб-розробці: Переваги використання модулів та компонентів для підтримки коду [Електронний ресурс] – режим доступу: <https://it-rating.ua/modulne-programuvannya-v-veb-rozrobsi-perevagi-vikoristannya-moduliv-ta-komponentiv-dlya-pidtrimki-kodu>. Дата звернення: 23 жов. 2023.
9. Чому SOLID — важлива складова мислення програміста. Розбираємося на прикладах з кодом [Електронний ресурс] – режим доступу: <https://dou.ua/lenta/articles/solid-principles/>. Дата звернення: 23 жов. 2023.
10. Мікросервісна архітектура [Електронний ресурс] – режим доступу: <https://dou.ua/lenta/articles/solid-principles/>. Дата звернення: 23 жов. 2023.
11. Неперервна інтеграція CI/CD: бізнес-переваги [Електронний ресурс] – режим доступу: <https://soft-den.com/continuous-integration-and-deployment/>. Дата звернення: 23 жов. 2023.
12. Design System and Component Library: Approach for efficient FrontEnd development [Електронний ресурс] – режим доступу: <https://www.linkedin.com/pulse/design-system-component-library-approach-efficient-kamatchinathan/>. Дата звернення: 23 жов. 2023.
13. A Guide to Dependency Management in Front-End Development [Електронний ресурс] – режим доступу: <https://webdesign.tutsplus.com/a-guide-to-dependency-management-in-front-end-development--cms-33963t/>. Дата звернення: 23 жов. 2023.
14. Офісне спілкування: 10 заповідей внутрішньої комунікації [Електронний ресурс] – режим доступу: <https://notagroup.com.ua/news/ofisne-spilkuвання-10-zapovidej-vnutrishnoyi-komunikatsiyi/>. Дата звернення: 23 жов. 2023.
15. What are some of the tools and techniques that you use to automate data quality assessment and monitoring? [Електронний ресурс] – режим доступу: <https://www.linkedin.com/advice/1/what-some-tools-techniques-you-use-automate-data/>. Дата звернення: 23 жов. 2023.

*Наукове електронне видання*

**МАТЕРІАЛИ**

**VI Всеукраїнської  
науково-практичної інтернет-конференції  
студентів, аспірантів та молодих вчених**

**СУЧАСНІ КОМП'ЮТЕРНІ СИСТЕМИ  
ТА МЕРЕЖІ В УПРАВЛІННІ**

**ЗБІРКА НАУКОВИХ ПРАЦЬ**

*30 листопада 2023 року  
(м. Херсон, м. Хмельницький)*

**ISBN 978-617-8187-04-0 (електронне видання)**



*Комп'ютерна верстка: к.т.н., доцент Дідик О.О.  
Відповідальний за випуск: к.т.н., доцент Григорова А.А.  
Дизайн обкладинки: к.т.н., доцент Дідик О.О.*

Підписано до видання 04.12.2023 р. Формат 60×84/8.  
Гарнітура Times. Ум. друк. арк. 28,00. Обл.-вид. арк. 30,11. Замовлення № 3087.

Книжкове видавництво ФОП Вишемирський В. С.  
Свідоцтво про внесення до Державного реєстру  
суб'єктів видавничої справи серія ХС № 48 від 14.04.2005 р.  
видано Управлінням у справах преси та інформації  
73000, Україна, м. Херсон, вул. Соборна, 2,  
тел. +38 (050) 133-10-13,  
e-mail: printvvs@gmail.com