

DOI: [https://doi.org/10.25140/2411-5363-2025-2\(40\)-221-233](https://doi.org/10.25140/2411-5363-2025-2(40)-221-233)

УДК 37.091.3:004.9:378.018.43

Хижняк Андрій Васильович¹, Пріла Ольга Анатоліївна²¹аспірант, старший викладач кафедри інформаційних та комп'ютерних систем
Національний університет «Чернігівська політехніка» (Чернігів, Україна)
E-mail: alf.andrey@gmail.com, ORCID: <http://orcid.org/0009-0008-8655-3736>²кандидат технічних наук, доцент, доцент кафедри інформаційних та комп'ютерних систем
Національний університет «Чернігівська політехніка» (Чернігів, Україна)
E-mail: olga.prila1986@gmail.com, ORCID: <http://orcid.org/0000-0002-0884-6516>

РОЗРОБКА СИСТЕМИ АВТОМАТИЗОВАНОЇ ГЕНЕРАЦІЇ ТА ПЕРЕВІРКИ ПАРАМЕТРИЗОВАНИХ ПРАКТИЧНИХ ЗАВДАНЬ

У роботі представлена модель системи автоматизованої генерації унікальних (параметризованих) практичних завдань та їх подальшої автоматичної перевірки з підтримкою функції автоматичного розгортання ізольованих та контрольованих індивідуальних навчальних середовищ для виконання цих завдань, що реалізує розроблену авторами інформаційну технологію. Для визначення компонентів, їх структурування і об'єднання в систему, а також для зниження складності системи шляхом абстракції і розмежування повноважень розроблена архітектура системи та описане функціональне призначення кожного модуля системи. Сформовані пропозиції щодо впровадження і використання розробленої системи в освітню діяльність.

Ключові слова: параметризовані практичні завдання; автоматична генерація; автоматичне оцінювання; дистанційна навчальна екосистема; віртуальні середовища.

Рис.: 12. Табл.: 1. Бібл.: 12.

Актуальність дослідження. Дистанційний формат навчання породжує нові виклики для викладачів: проблема академічної доброчесності набуває особливої гостроти, оскільки контроль за самостійністю виконання завдань істотно послаблюється. Багато дослідників (Dr. Robert Stevens, Dr. Lawrence Silver, Dr. Rhonda Richards, Dr. Kitty Campbell) підкреслюють той факт, що випадки академічного шахрайства в онлайн-освіті є складнішими для виявлення [1].

Серед форм академічного шахрайства можна виділити наступні: консультування з третіми особами під час онлайн-іспитів та передавання облікових даних для проходження оцінювання іншими особами [2]. Також студенти активно застосовують технічні засоби для обміну відповідями, отримання сторонньої допомоги під час іспитів та використання генеративних інструментів штучного інтелекту для отримання відповідей [3].

Власні дослідження автора підтверджують ці тенденції: у ході експерименту з дисципліни «Операційні системи Unix» у 2023/2024 навчальному році було виявлено, що 6,7 % студентів використовували ШІ для фальсифікації відповідей, а 16,7 % — графічні редактори. Це лише виявлені випадки, реальні масштаби проблеми, імовірно, є значно більшими: згідно з опитуванням, проведеним Маликом [4], 60 % респондентів зізналися в регулярному списуванні під час онлайн-іспитів.

Для запобігання списуванню застосовуються різноманітні методики, зокрема спеціалізовані програмні рішення для аналізу рухів, сканування очей, відстеження натискань клавіш, моніторингу середовища виконання завдань (так зване proctoring software). Серед популярних рішень можна виділити AIProctor, ProctorU, Proctortrack, Proctorio та Smowl [5].

Значний прогрес спостерігається також у застосуванні штучного інтелекту для виявлення підозрілої поведінки студентів. Так, наприклад, AIProctor інтегрується з LMS, фіксує обличчя та ідентифікатори студентів, здійснює 360-градусне сканування аудиторії, а також аналізує положення та рухи голови для виявлення можливих порушень [6]. Незважаючи на застосування алгоритмів ШІ, для повноцінної перевірки відеозаписів все ще потрібен ручний перегляд, що ускладнює масштабування процесу і викликає занепокоєння щодо конфіденційності.

Постановка проблеми. Одним із ефективних способів зменшення випадків шахрайства є впровадження такого формату навчання, коли кожному студенту пропонується власне завдання на основі унікального набору параметрів або початкових умов. Цей підхід можна визначити термінами «parameterized personalization» або «variant-

based personalization». Такі завдання зберігають загальну методичну структуру, але забезпечують індивідуальність кожного варіанту. Сучасні технології надають поштовху для пошуку інструментів, які забезпечать автоматизовану генерацію процесів створення та оцінювання персоналізованих практичних завдань, а також для автоматичного розгортання і налаштування ізольованих та контрольованих середовищ, в яких студенти зможуть виконувати лабораторні практичні завдання, при цьому підтримуючи асинхронний дистанційний формат навчання.

Аналіз останніх досліджень та публікацій. Розробка інструментів для автоматизованої генерації та оцінювання персоналізованих завдань є предметом активного наукового пошуку. У статті «Designing Programming Assignments to Reduce the Likelihood of Cheating» [7] розглядаються методи створення унікальних завдань для кожного студента як ефективний захід боротьби зі списуванням.

Фреймворк SERA [8] забезпечує гнучку платформу для проведення лабораторних вправ для курсів з безпеки та автоматизації їхнього оцінювання. Система складається з кількох компонентів (SERA Engine = Генератор+Валідатор+Монітор), модулів та утиліт для створення персоналізованих завдань та їхньої перевірки.

У цьому дослідженні [9] представлено систему, яка автоматично генерує персоналізовані практичні завдання для студентів та інструментарій для автоматизованого налаштування лабораторного середовища з унікальною конфігурацією для кожного студента. Використання таких індивідуалізованих завдань сприяє зменшенню випадків академічної нечесності та несанкціонованої співпраці та підвищенню залученості студентів у навчальний процес.

Наукова робота «The Design and Implementation of PAGE: Personalised Assessment Generative Engine» презентує PAGE – інструмент, розроблений для спрощення створення та адміністрування персоналізованих цифрових оцінювань у системі управління навчанням Canvas. [10]. Автори демонструють, що автоматизоване створення унікальних завдань для кожного студента є ефективним підходом для зменшення списування та підвищення залученості в освітньому процесі.

Автори роботи [11] також працюють над інструментом оцінювання, що використовує параметризацію задач та автоматизовану перевірку рішень у курсі з основ програмування.

Останні дослідження демонструють також зацікавленість у використанні генеративного штучного інтелекту для створення персоналізованих завдань. Зокрема, у роботі E.Logacheva та ін. «Evaluating Contextually Personalized Programming Exercises Created with Generative AI» аналізується використання GPT-4 для автоматичного генерування індивідуалізованих завдань у курсах з програмування [12].

Ретельний аналіз публікацій та літератури виявив необхідність формування методів, інформаційної технології та інших інструментів, які б забезпечили автоматизовану генерацію та оцінювання персоналізованих практичних завдань, включаючи автоматичне розгортання і налаштування середовищ, у яких студенти можуть виконувати лабораторні практичні завдання, та підтверджує актуальність дослідження.

Виділення недосліджених частин загальної проблеми. Головним недоліком впровадження індивідуальних завдань залишається значне навантаження на викладачів через необхідність ручного створення та перевірки великої кількості унікальних варіантів, що зумовлює актуальність автоматизації цього процесу.

Мета і задачі дослідження. Метою цього дослідження (науковим завданням) є підвищення рівня академічної доброчесності та якості контролю знань зі зменшенням витрат часу викладачами на створення та перевірку практичних завдань студентів. Завдання дослідження:

- проаналізувати поточний стан створення і видачі персональних (варіативних або параметризованих) завдань на кафедрі;
- розробити систему, що реалізує інформаційну технологію автоматизованої генерації унікальних (параметризованих) практичних завдань, автоматичного розгортання індивідуальних навчальних середовищ для виконання цих завдань та їх подальшої автоматичної перевірки
- оцінити рівень ефективності таких завдань порівняно з неунікальними (не параметризованими завданнями),
- сформулювати пропозиції щодо впровадження і використання розробленої інформаційної технології, направленої виконання поставленої мети дослідження.

Виклад основного матеріалу. З метою проведення аналізу поточного стану створення і видачі персональних (варіативних або параметризованих) завдань в НУ “Чернігівська Політехніка” було проведено опитування викладачів інженерних кафедр. Згідно результатів опитування і спілкування з викладачами абсолютна більшість викладачів зазначають, що вважають доцільним використання індивідуальних (варіативних або параметризованих) практичних завдань і намагаються використовувати їх в своїй викладацькій роботі, хоч і в обмеженому вигляді. Викладачі, які не використовують індивідуальні завдання, відмічають, що хотіли б застосовувати їх в освітній діяльності. Серед причин НЕ використання таких завдань насамперед називають великі часові витрати, які необхідні для підготовки та перевірки таких завдань, а також важкість процесу ручної генерації завдань.

Розрахуємо середньостатистичні часові витрати викладача для створення унікальних завдань в рамках курсу “Операційні системи Unix”, спираючись на дані по кількості студентів за останні 5 років (дисципліна викладається студентам на третьому курсі). Дані отримані від деканату Навчально-наукового інституту електронних та інформаційних технологій Національного університету “Чернігівська Політехніка”.

Таблиця 1 – Кількість студентів на 3 курсі спеціальності 123 за 2021-2025 рр.

Роки навчання	Кількість студентів на 3 курсі
2024-2025	92
2023-2024	105
2022-2023	131
2021-2022	95

Джерело: розроблено авторами.

У середньому в потоці маємо 105 студентів. Кількість лабораторних робіт у курсі «Операційні системи Unix» становить 6 (шість). Для підготовки 105 унікальних варіантів завдань для кожної лабораторної роботи, включаючи заповнення завдання в LMS та його подальшу ручну перевірку, потрібно не менше 15 хвилин. Отже, тільки на підготовку та перевірку виконаних практичних завдань для однієї дисципліни викладач має витратити $105 \cdot 15 \cdot 6 = 9540$ хв (157 годин), що в 6 разів перевищує обсяг годин, виділених на цей вид робіт і займає 75 % всіх виділених на дисципліну годин (за навантаженням викладача на дисципліну виділяється 207 годин, з них лише 25,6 годин безпосередньо на перевірку лабораторних робіт).

Інформаційна технологія для вирішення поставленої задачі включає в себе систему автоматизованої генерації унікальних (параметризованих) практичних завдань, автоматичного розгортання індивідуальних навчальних середовищ для виконання цих завдань та їх подальшої автоматичної перевірки.

Для визначення компонентів, їх структурування і об'єднання в систему, а також для зниження складності системи шляхом абстракції та розмежування повноважень розроблена архітектура системи, представлена на рис. 1.

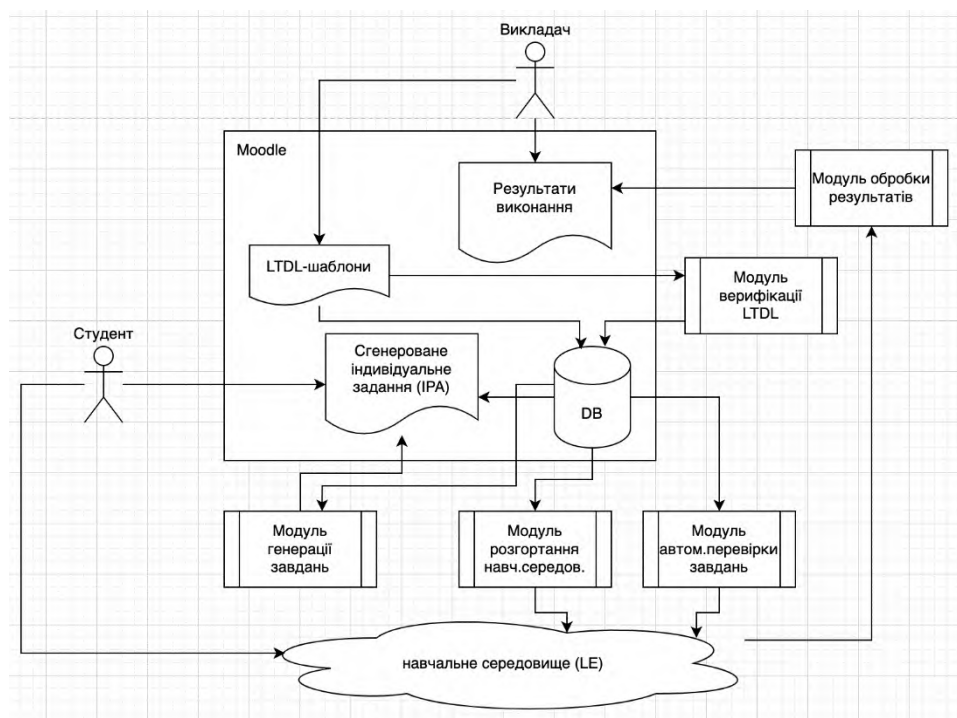


Рис. 1. Архітектура системи

Джерело: розроблено авторами.

Система складається із 5 модулів: модулю генерації завдань, модулю верифікації LTDL, модулю розгортання навчального середовища, модулю автоматичної перевірки завдань та модулю обробки результатів виконання. Функціональне призначення модулів представлено на наступному рис. 2.

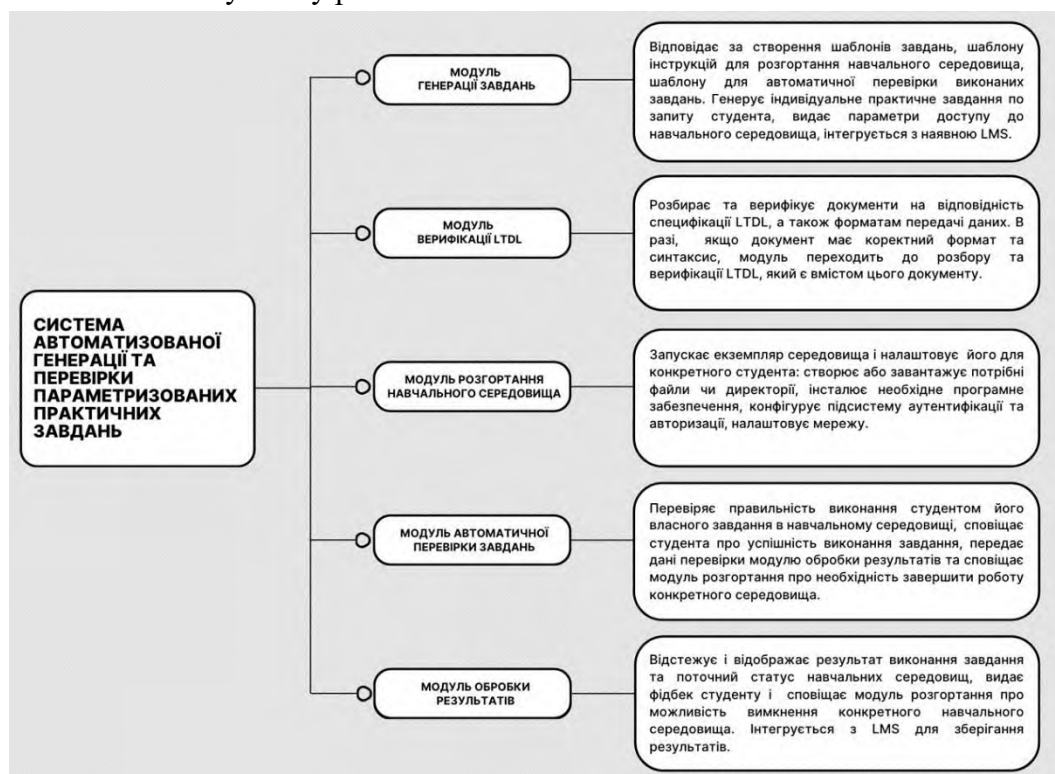


Рис. 2. Функціональне призначення модулів системи

Джерело: розроблено авторами.

Розглянемо детальніше кожен модуль, його функціональне призначення, місце в архітектурі та взаємозв'язки з іншими компонентами системи.

Модуль генерації завдань.

Модуль відповідає за створення індивідуальних практичних завдань студентів, інтегрується з наявною LMS і дозволяє створити individualized practical assignment (далі по тексту IPA) в рамках вивчення якоїсь теми або групи тем обраної дисципліни. В LMS Moodle, з якою автори інтегрували систему, створюється новий тип питання, яке можна використовувати для стандартних moodle-активностей типу Quizz, Lesson etc.

За допомогою цього модулю і спеціально розробленої мови Learning Task Definition Language викладач задає такі шаблони:

a) student assignment template - шаблон завдання, яке буде використане для генерації параметризованого завдання для конкретного студента з певним набором унікальних змінних параметрів.

Для ілюстрації роботи модулю візьмемо приклад з дисципліни “Операційні системи” по темі вивчення роботи з архівами і створимо IPA. Мета завдання - перевірити навички студента користуватися командами роботи з архівами. Ціллю виконання завдання буде створення архіву із відповідними файлами всередині. Таким чином у нас буде 2 етапи: пошук потрібних файлів та створення архіву.

Для завдання з автоматичною перевіркою student assignment template має такий вигляд:

```
Зайдіть на віддалену систему, використовуючи наступні дані:  
ім'я користувача {{login}}  
пароль користувача {{password}}  
host {{IP}}  
  
Знайдіть в директорії з іменем "{{dir_name}}" файли, які містять в своїй назві символи "{{symbols}}".  
Заархівуйте знайдені файли в архів з іменем {{arch_name}}.tgz і помістіть його в директорію з іменем "{{dir_name}}".
```

Рис. 3. Вигляд шаблону “student assignment template”

Джерело: розроблено авторами.

б) learning environment creation instructions template - набір інструкцій для модуля №3, який відповідає за розгортання learning environment (LE). У цьому шаблоні вказується тип обраного LE та перелік дій або команд, які повинні бути виконані модулем для правильного розгортання і налаштування цього навчального середовища.

Оскільки для виконання параметризованого завдання студент повинен отримати не тільки опис завдання, але і параметри доступу до навчального середовища, то перед видачею завдання студенту треба його персональне навчальне середовище запустити. Розглянемо варіанти інструкцій розгортання для різних бекендів:

Для запуску навчального середовища на основі віртуальної машини з ОС Linux Ubuntu 24.04 і гіпервізора VirtualBox learning environment creation instructions template повинен мати такі інструкції:

```
LE type: Virtual Machine  
LE provider: virtualbox  
LE version: Linux Ubuntu 24.04  
LE parameters: host: {{IP}}, username: {{username}}, password: {{password}}  
LE create instructions: [VBoxManage startvm "Linux Ubuntu 24.04" --type headless ]  
LE provision instructions: [  
    create user {{username}},  
    echo {{password}} | passwd {{username}},  
    ip a add {{IP}}/24 dev enp0s3  
]
```

Рис. 4. Приклад інструкцій для запуску LE на основі ВМ з ОС Linux Ubuntu 24.04 і гіпервізора VirtualBox

Джерело: розроблено авторами.

Для запуску навчального середовища на основі контейнера з ОС Linux Ubuntu 24.04 і платформи Docker потрібно задати наступні інструкції у відповідних секціях learning environment creation instructions template. Ці інструкції будуть пізніше використані модулем розгортання для запуску конкретного екземпляру LE відповідного бекенду.

```
LE type: container
LE provider: docker
LE version: ubuntu24.04
LE parameters: host: {{IP}}, username: {{username}}, password: {{password}}, filename:{{file_name}}, token:{{token}}
LE create instructions: [docker run --net base --ip 172.18.0.22 -it ubuntu24.04 bash]
LE provision instructions: [
docker exec -it my_container sh -c "create user {{username}}",
docker exec -it my_container sh -c "echo {{password}} | passwd {{username}}",
docker exec -it my_container sh -c "ip a add {{IP}}/24 dev enp0s3"
docker exec -it my_container sh -c "echo {{token}} > /usr/{{file_name}}"
]
```

Рис. 5. Приклад інструкцій для запуску LE на основі контейнера з ОС Linux Ubuntu 24.04 і платформи Docker

Джерело: розроблено авторами.

в) IPA verification instructions template - набір інструкцій для модуля №4 автоматичної перевірки IPA, згідно яких фіксується успішність виконання завдання в навчальному середовищі.

Для наведеного прикладу із створенням архіву в шаблоні повинні бути вказані такі інструкції:

```
LE parameters: host: {{IP}}, username: {{username}}, password: {{password}}
LE connect protocol: [ssh]
LE check instructions: [tar -tvzf /home/{{username}}/{{dir_name}}/{{arch_name}} | grep "{{symbols}}"]
```

Рис. 6. Приклад IPA verification instructions template

Джерело: розроблено авторами.

При отриманні запиту від студента на виконання завдання модуль генерації завдань по шаблону “student assignment template” сформує унікальне параметризоване завдання і надасть параметри доступу до навчального середовища, де це завдання повинно бути виконано.

```
Зайдіть на віддалену систему, використовуючи наступні дані:
ім'я користувача RM634oi
пароль користувача Jtu725430
host 10.10.10.10

Знайдіть в директорії з іменем "Wjh381" файли, які містять в своїй назві символи "Soho".
Заархівуйте знайдені файли в архів з іменем Arch87.tgz і помістіть його в директорію з іменем " Wjh381".
```

Рис. 7. Унікальне параметризоване завдання і параметри доступу до навчального середовища, які безпосередньо отримує студент.

Джерело: розроблено авторами.

Модуль верифікації LTDL.

Для створення вказаних шаблонів пропонується використовувати спеціально розроблений для цієї мети Learning Task Definition Language (LTDL). Документи LTDL в подальшому будуть перетворені на опис конкретного унікального завдання, конфігураційні файли систем розгортання для хмарних сервісів або конфігураційних файлів Docker-контейнерів чи віртуалізаторів на кшталт VirtualBox, VMWare, Vagrant.

Створені за допомогою web-інтерфейсу шаблони зберігаються в існуючій базі даних LMS та можуть бути в подальшому змінені, оновлені чи експортовані в один з форматів зберігання даних, наприклад JSON. Також модуль дозволяє імпорт шаблонів з підтримуваних форматів.

Для функціонування системи модуль генерації завдань повинен створити коректний LTDL-документ, який в подальшому буде використовуватися іншими модулями: модулем розгортання LE та модулем автоматичної перевірки IPA. Для цього всі модулі системи будуть використовувати модуль верифікації LTDL, задача якого:

- спочатку провести розбір та верифікацію документу на відповідність специфікації LTDL, а також форматам передачі даних JSON, YAML тощо.
- якщо документ має коректний формат та синтаксис, модуль переходить до розбору та верифікації LTDL, який є вмістом цього документу.

Модуль розгортання LE.

Задача модуля - розгорнути індивідуальне середовище для виконання IPA конкретним студентом згідно інструкцій. Ці інструкції, зокрема, визначають, який тип середовища треба розгорнути: окрему віртуальну машину, LXC\docker контейнер, хмарний сервіс, JupiterLab тощо.

В залежності від обраного типу середовища модуль запускає екземпляр LE і виконує перелік дій, які налаштують конкретний екземпляр LE для конкретного студента: створить або завантажить потрібні файли чи директорії, інсталує необхідне програмне забезпечення, сконфігурує підсистему аутентифікації та авторизації, налаштує мережу.

Для виконання цих задач модуль повинен отримати інструкції від LMS у вигляді LTDL-документу і провести верифікацію цього документу за допомогою відповідного модулю. Коли конкретний студент створить запит на виконання свого завдання, модуль розгортання отримає LTDL-документ з необхідними для створення LE параметрами. Розглянемо кілька прикладів передачі даних.

```
{
  "LE_type": "Virtual Machine",
  "LE_provider": "virtualbox",
  "LE_version": "Linux Ubuntu 24.04",
  "LE_parameters": {
    "host": "10.10.10.10",
    "username": "PM634oi",
    "password": "Jru725430"
  },
  "LE_create_instructions": [
    "VBoxManage startvm \"Linux Ubuntu 24.04\" --type headless"
  ],
  "LE_provision_instructions": [
    "create user PM634oi",
    "echo Jru725430 | passwd PM634oi",
    "ip a add 10.10.10.10/24 dev enp0s3"
  ]
}
```

Рис. 8. Інструкції для бекенду VirtualBox у вигляді JSON-файлу

Джерело: розроблено авторами.

```
LE_type: container
LE_provider: docker
LE_version: ubuntu24.04
LE_parameters:
  host: "{{IP}}"
  username: "{{username}}"
  password: "{{password}}"
  filename: "{{file_name}}"
  token: "{{token}}"
LE_create_instructions:
  - "docker run --net base --ip 172.18.0.22 -it ubuntu24.04 bash"
LE_provision_instructions:
  - "docker exec -it my_container sh -c \"create user {{username}}\""
  - "docker exec -it my_container sh -c \"echo {{password}} | passwd {{username}}\""
  - "docker exec -it my_container sh -c \"ip a add {{IP}}/24 dev enp0s3\""
  - "docker exec -it my_container sh -c \"echo {{token}} > /usr/{{file_name}}\""
```

Рис. 9. Інструкції для Docker-бекенду у вигляді YAML-файлу

Джерело: розроблено авторами.

Результат розгортання LE модуль повертає у вигляді відповідного статусу і в подальшому виконує його контроль. Конфігураційні параметри навчального середовища будуть використані модулем верифікації при підключенні до екземпляру LE для проведення верифікації і формування звіту про прогрес виконання завдання та його фінальні результати. Модуль також виконує функції згортання або знищення непотрібних LE, для цього він взаємодіє з модулем автоматичної перевірки завдань та з модулем обробки результатів, від яких в певний момент отримує сигнал на завершення роботи відповідного LE.

Модуль автоматичної перевірки IPA.

Задача модуля - перевірити правильність виконання студентом його власного завдання в окремому індивідуальному навчальному середовищі згідно отриманих інструкцій. Спочатку цей модуль також використовує модуль верифікації LTDL для перевірки коректності отриманих інструкцій.

Згідно з отриманими інструкціями модуль виконує підключення до конкретного навчального середовища з використанням відповідних параметрів для аутентифікації та авторизації та перевіряє чи виконане завдання. Підключення та перевірка виконується з певною, заданою наперед в LTDL-інструкціях, періодичністю, наприклад, кожні 5 або 10 хвилин. У разі успішного виконання завдання (чи переліку завдань) модуль виконує наступні дії: сповіщає студента про успішність виконання завдання, передає дані перевірки модулю обробки результатів та сповіщає модуль розгортання про необхідність завершити роботу конкретного середовища. Якщо модуль не зміг підключитися до навчального середовища, він сповіщає про це модуль обробки результатів.

Для проведення автоматичної перевірки в рамках роботи із Модулем №1 було створено IPA verification instructions template, згідно якого модуль автоматичної перевірки отримує наступний JSON-файл:


```

"LE_parameters": {
  "host": "10.10.10.10",
  "username": "PM634oi",
  "password": "Jru725430"
},
"LE_connect_protocol": [ ssh ],
"LE_check_instructions": [tar -tvzf /home/PM634oi/Wjh381/Arch87.tgz | grep "Soho"]
}

```

Рис. 10. JSON-файл для перевірки виконання завдання студентом

Джерело: розроблено авторами.

Модуль обробки результатів.

Задача модуля полягає у відстеженні і відображенні результатів виконання завдання. Загальний вигляд результату виконання завдання залежить від налаштувань конкретної LMS і може мати як фіксований набір значень: “виконане”, “спроба провалена”, “термін виконання сплив” так і прогрес виконання у відсотках.

Також модуль відображає поточний статус навчальних середовищ. Статус навчальних середовищ модуль отримує від модулю розгортання або від модулю автоматичної перевірки завдань. Можливі статуси навчального середовища: “не створене”, “створене”, “помилка створення”, “запущене”, “скасоване”, “завершене”.

Результати виконання завдання даний модуль отримує від модулю автоматичної перевірки завдань та/або студента, якщо система функціонує в напіваавтоматичному режимі, адже не для всіх типів завдань потрібна автоматична перевірка виконання.

Іноді в рамках виконання завдання студенту треба виконати якусь послідовність дій для отримання якогось результату чи об'єкту. Наприклад, є потреба в перевірці навичок пошуку файлів. Система підтримує створення таких завдань в частині генерації параметризованого завдання, розгортання навчального середовища та отримання результату від студента. У такому режимі роботи системи після виконання завдання студент вводить відповідь вручну у відповідне поле в LMS.

Для такого завдання з напіваавтоматичною перевіркою student assignment template має такий вигляд:

```

Зайдіть на віддалену систему, використовуючи наступні дані:
ім'я користувача {{login}}
пароль користувача {{password}}
host {{IP}}

Знайдіть файл з іменем {{file_name}} та введіть токен з цього файлу в поле відповіді:

```

Рис. 11. Вигляд student assignment template в режимі напіваавтоматичної перевірки

Джерело: розроблено авторами.

При отриманні запиту від студента на виконання завдання на основі цього шаблону модуль генерації завдань створить унікальне параметризоване завдання виду:

```

Зайдіть на віддалену систему, використовуючи наступні дані:
ім'я користувача d745
пароль користувача vfby62-fds
host 172.18.0.22

Створіть директорію з іменем JHvbdk38d та введіть токен з цього файлу в поле відповіді:

```

Рис. 12. Вигляд параметризованого завдання студентові в режимі напіваавтоматичної перевірки

Джерело: розроблено авторами.

Для реалізації такої схеми роботи достатньо розгорнути персональне середовище для студента та очікувати від нього результатів. При отриманні результату виконання завдання від студента при перевірці напівавтоматичного типу завдання модуль видає студенту фідбек, наприклад, інформацію про правильність виконання, та сповіщає модуль розгортання про можливість вимкнення конкретного навчального середовища.

Пілотний проєкт був запусканий у межах дисципліни «Операційні системи Unix». Подальше впровадження цієї інформаційної технології в освітню діяльність передбачає:

- визначення відповідальної особи;
- визначення кола дисциплін і викладачів, які бажають впровадити технологію в освітній процес;
- проведення навчання користувачів у форматі навчального семінару для викладачів кафедри та розробку внутрішньої інструкції для користувачів;
- розробку банку шаблонів завдань (створення бібліотеки шаблонів завдань для різних тем та дисциплін);
- налаштування інтеграції системи з LMS, що використовується на кафедрі;
- інтеграцію із корпоративними засобами комунікації;
- підготовку інфраструктури для розгортання навчальних середовищ: визначення бекендів навчальних середовищ, обрахунок їх кількості і виділення необхідних ресурсів для розгортання;
- моніторингу стабільності системи (відповідальна особа відстежує навантаження на сервери, контролює наявність активних середовищ, проводить оновлення системи);
- організацію роботи із захисту персональної інформації;
- організацію регулярного резервного копіювання навчальних середовищ і результатів;
- придбання необхідного периферійного обладнання, наприклад, вебкамер;
- облаштування hardware-лабораторій;
- створення міжвикладацької ініціативної групи для адаптації системи до інших курсів кафедри, для обговорення можливих недоліків системи і її покращення;
- оцінку результатів і масштабування;
- адміністративні кроки, в тому числі формалізація в навчальному процесі (включити використання системи в робочі програми навчальних дисциплін);
- позиціонування системи як інноваційного проєкту кафедри - презентувати систему на конференціях, днях відкритих дверей, виставках EdTech.

Висновки. Проведений аналіз наукової бази і існуючих рішень показав, що одним із перспективних напрямів вирішення проблеми академічної недоброочесності при виконанні практичних лабораторних робіт є використання унікальних завдань. Головним недоліком впровадження таких завдань було і залишається неосяжне навантаження на викладачів, обумовлене необхідністю ручного створення та перевірки великої кількості унікальних варіантів, що зумовлює актуальність автоматизації цього процесу.

В рамках роботи було досліджено поточний стан використання персональних (варіативних або параметризованих) завдань на інженерних кафедрах НУ "Чернігівська Політехніка". У результаті досліджень сформульовано та вирішено актуальне наукове завдання щодо створення інформаційної технології, яка описує процеси автоматизованої генерації унікальних (параметризованих) практичних завдань, автоматичного розгортання індивідуальних навчальних середовищ для виконання цих завдань та їх подальшої автоматичної перевірки. Була обґрунтована необхідність розробки системи, направленої на автоматизацію освітніх процесів, а разом з тим - на зменшення навантаження на викладачів, зменшення кількості випадків списування та підвищення якості навчання. Також у рамках роботи безпосередньо розроблена система автоматизованої генерації та перевірки завдань та її архітектура.

Викладені в статті наукові результати мають безпосереднє практичне значення. Описана система дозволить вирішити проблему із видачею студентам унікальних (параметризованих) практичних завдань. Завдяки автоматизації ця система стане вигідним і зручним інструментом для навчання студентів інженерних спеціальностей. Для впровадження і використання даної системи в діяльність кафедри були сформовані рекомендації по впровадженню.

Розроблена система — реалістична, інноваційна, добре структурована, з високим потенціалом для трансформації освітнього процесу на кафедрі. Її впровадження вимагає поступової реалізації у 4 фазах: підготовка інфраструктури, навчання викладачів та студентів, оцінка і аналіз результатів впровадження, масштабування на інші дисципліни/кафедри.

Подальші перспективи досліджень полягають в розширенні переліку дисциплін, щодо яких можливе використання розробленої інформаційної технології та системи, інтеграції з іншими LMS-системами та генеративним ШІ для створення шаблонів, підтримці більшої кількості бекендів, а також покращенні системи так, аби початкові параметри враховували навчальні результати або рівень підготовки студента.

Заява про використання генеративного ШІ та технологій на основі ШІ у процесі написання тексту статті

Автори використали ШІ (Chat GPT) для покращення читабельності та виправлення стилістичних і граматичних помилок у цій статті. Після використання цього інструменту автори переглянули та відредагували зміст за потреби та взяли на себе повну відповідальність за зміст публікації.

Список використаних джерел

1. Stevens, R., Silver, L., Richards, R., & Campbell, K. (2022). A comparison of faculty and student perspectives of academic integrity in an online environment: A pilot study. *Journal of Business Administration*. Online, 16(2), 1–13. <https://www.atu.edu/business/jbao/fall2022/202202%2002%20Stevens%20Silver%20Richards%20Campbell.pdf>
2. Valizadeh, M. (2022). Cheating in Online Learning Programs: Learners' Perceptions and Solutions. *Turkish. Online Journal of Distance Education*, 23(1), 195–209. <https://doi.org/10.17718/tojde.1050394>.
3. Naidu, K., & Sevnarayan, K. (2023). ChatGPT: An ever-increasing encroachment of artificial intelligence in online assessment in distance education. *Online Journal of Communication and Media Technologies*, 13(1), e202336. <https://doi.org/10.30935/ojcm/13291>.
4. Malik, A. A., Hassan, M., Rizwan, M., Mushtaque, I., Lak, T. A., & Hussain, M. (2023). Impact of academic cheating and perceived online learning effectiveness on academic performance during the COVID-19 pandemic among Pakistani students. *Frontiers in Psychology*, 14, 1124095. <https://doi.org/10.3389/fpsyg.2023.1124095>.
5. Arnò, S., Galassi, A., Tommasi, M., Saggino, A., & Vittorini, P. (2021). State-of-the-Art of commercial proctoring systems and their use in academic online exams. *International Journal of Distance Education Technologies*, 19(2), 55–76. <https://doi.org/10.4018/ijdet.20210401.0a3>.
6. AI Proctoring System Official Site. <https://ai-proctor.com>.
7. Simon. (2017). Designing programming assignments to reduce the likelihood of cheating. *У The nineteenth australasian computing education conference*. ACM Press. <https://doi.org/10.1145/3013499.3013507>.
8. Agudo, I., Rios, R. & Nieto, A. (2019). Personalized computer security tasks with automatic evaluation and feedback. *Proceedings of the 2019 AIS SIGED International Conference on Information Systems Education and Research*. <https://aisel.aisnet.org/siged2019/1>.
9. Vykopal, J. Švábenský, V., Seda, P. & Čeleda, P. (2022). Preventing Cheating in Hands-on Lab Assignments. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1 (SIGCSE 2022), Vol. 1*. Association for Computing Machinery, New York, NY, USA, 78–84. <https://doi.org/10.1145/3478431.3499420>.

10. Chan, J., & Teng, S. (2024). The Design and Implementation of PAGE: Personalised Assessment Generative Engine. *Pacific Journal of Technology Enhanced Learning*, 6(2), 33-46. <https://doi.org/10.24135/pjtel.v6i2.203>.
11. Zampirolli, F. D. A., Pisani, P. H., Josko, J. M., Kobayashi, G., Fraga, F., Goya, D., & Savegnago, H. R. (2020). Parameterized and automated assessment on an introductory programming course. In *Simpósio brasileiro de informática na educação*. Sociedade Brasileira de Computação. <https://doi.org/10.5753/cbie.sbie.2020.1573>.
12. Logacheva, E., Hellas, A., Prather, J., Sarsa, S., & Leinonen, J. (2024). Evaluating contextually personalized programming exercises created with generative AI. In *ICER 2024: ACM conference on international computing education research* (95–113). ACM. <https://doi.org/10.1145/3632620.3671103>.

References

1. Stevens, R., Silver, L., Richards, R., & Campbell, K. (2022). A comparison of faculty and student perspectives of academic integrity in an online environment: A pilot study. *Journal of Business Administration*. 16(2), 1–13. <https://www.atu.edu/business/jbao/fall2022/202202%2002%20Stevens%20Silver%20Richards%20Campbell.pdf>.
2. Valizadeh, M. (2022). Cheating in Online Learning Programs: Learners' Perceptions and Solutions. Turkish. *Online Journal of Distance Education*, 23(1), 195–209. <https://doi.org/10.17718/tojde.1050394>.
3. Naidu, K., & Sevnarayan, K. (2023). ChatGPT: An ever-increasing encroachment of artificial intelligence in online assessment in distance education. *Online Journal of Communication and Media Technologies*, 13(1), e202336. <https://doi.org/10.30935/ojcm/13291>.
4. Malik, A. A., Hassan, M., Rizwan, M., Mushtaque, I., Lak, T. A., & Hussain, M. (2023). Impact of academic cheating and perceived online learning effectiveness on academic performance during the COVID-19 pandemic among Pakistani students. *Frontiers in Psychology*, 14, 1124095. <https://doi.org/10.3389/fpsyg.2023.1124095>.
5. Arnò, Simone & Galassi, Alessandra & Tommasi, Marco & Saggino, Aristide & Vittorini, Pierpaolo. (2021). State-of-the-Art of Commercial Proctoring Systems and Their Use in Academic Online Exams. *International Journal of Distance Education Technologies*. <http://dx.doi.org/10.4018/IJDET.20210401.0a3>.
6. AI Proctoring System Official Site. <https://ai-proctor.com>.
7. Simon. 2017. Designing Programming Assignments to Reduce the Likelihood of Cheating. In *Proceedings of the Nineteenth Australasian Computing Education Conference (ACE '17)*. Association for Computing Machinery. New York, NY, USA, 42–47. <https://doi.org/10.1145/3013499.3013507>.
8. Agudo, Isaac; Rios, Ruben; and Nieto, Ana, "PERSONALIZED COMPUTER SECURITY TASKS WITH AUTOMATIC EVALUATION AND FEEDBACK" (2019). *Proceedings of the 2019 AIS SIGED International Conference on Information Systems Education and Research*. <https://aisel.aisnet.org/siged2019/1>.
9. Jan Vykopal, Valdemar Švábenský, Pavel Seda, and Pavel Čeleda. 2022. Preventing Cheating in Hands-on Lab Assignments. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1 (SIGCSE 2022)*, Vol. 1. Association for Computing Machinery. New York, NY, USA, 78–84. <https://doi.org/10.1145/3478431.3499420>.
10. Chan, J., & Teng, S. (2024). The Design and Implementation of PAGE: Personalised Assessment Generative Engine. *Pacific Journal of Technology Enhanced Learning*, 6(2), 33-46. <https://doi.org/10.24135/pjtel.v6i2.203>.
11. Zampirolli, F. D. A., Pisani, P. H., Josko, J. M., Kobayashi, G., Fraga, F., Goya, D., & Savegnago, H. R. (2020). Parameterized and automated assessment on an introductory programming course. In *Simpósio brasileiro de informática na educação*. Sociedade Brasileira de Computação. <https://doi.org/10.5753/cbie.sbie.2020.1573>.
12. Logacheva, E., Hellas, A., Prather, J., Sarsa, S., & Leinonen, J. (2024). Evaluating contextually personalized programming exercises created with generative AI. In *ICER 2024: ACM conference on international computing education research* (95–113). ACM. <https://doi.org/10.1145/3632620.3671103>.

Отримано 27.05.2025

Andrii Khyzhniak¹, Olga Prila²

¹PhD student, Senior Lecturer at Information and Computer Systems Department
Chernihiv Polytechnic National University (Chernihiv, Ukraine)

E-mail: alf.andrey@gmail.com. **ORCID:** <http://orcid.org/0009-0008-8655-3736>

²PhD, Associate Professor at Information and Computer Systems Department
Chernihiv Polytechnic National University (Chernihiv, Ukraine)

E-mail: olga.prila1986@gmail.com. **ORCID:** <http://orcid.org/0000-0002-0884-6516>

DESIGNING A SYSTEM FOR AUTOMATED GENERATION AND AUTOMATED ASSESSMENT OF PARAMETERIZED PRACTICAL ASSIGNMENTS

The article addresses the issue of ensuring academic integrity in the context of distance learning, particularly with regard to completion of practical assignments in engineering disciplines. The increasing incidence of unauthorized use of external resources, generative artificial intelligence, technical tools during online assessments, and identity substitution underscores urgent need for task individualization and automated monitoring of task completion. In response to these challenges, the authors propose the information technology and system for the automated generation of parameterized practical assignments (individualized for each student), with subsequent deployment of controlled and isolated learning environments and automatic result verification.

To define the system components, they were organized, and overall complexity was reduced through abstraction and separation of responsibilities, the system architecture was developed and the functional purpose of each module was described.

The architecture comprises five modules, namely: assignment generation, instructions verification module, personal learning environment deployment, automated assessment and result processing. Within the article, the specialized language Learning Task Definition Language (LTDL) that formalizes assignment templates, environment configuration instructions and evaluation criteria was introduced.

The conducted study demonstrates that manual preparation of personalized assignments imposes excessive time demands on instructors, whereas automation significantly reduces workload, enhances transparency of assessment, and mitigates cases of academic dishonesty. The system integrates with the Moodle learning management system and supports different backends: virtual machines, Docker containers, cloud platforms and JupiterLab.

The article also provides specific recommendations for implementing the system within the educational processes of technically oriented academic departments.

Keywords: parameterized practical assignments; automated generation; automated evaluation; automated assessment; distance learning ecosystem; virtual environment.

Fig.: 12. Table: 1. References: 12.