

УДК 004.048+004.021

К.С. ЮЩЕНКО*

ПІДХІД ДО ФОРМУВАННЯ МНОЖИНИ ТЕСТОВИХ ЗАВДАНЬ ДЛЯ ОЦІНЮВАННЯ ПЕРСОНАЛУ З ВИКОРИСТАННЯМ ЧАТ-БОТІВ

*Інститут телекомунікацій і глобального інформаційного простору НАН України, м. Київ, Україна

Анотація. У статті запропоновано підхід до формування множини тестових завдань для оцінювання персоналу з використанням чат-бота OpenAI ChatGPT-5.2 Plus. Розроблено механізм добору питань із банку завдань, який забезпечує одночасне досягнення трьох ключових вимог: варіативності тестових сесій, відсутності повторів і збереження заданої структури оцінювання. Показано, що використання ймовірного розподілу для вибору інтервалів складності дозволяє керувати формуванням послідовності завдань, орієнтуючи генератор на переважання питань середнього рівня складності за зменшення частоти крайніх значень. Проведене моделювання підтвердило статистичну узгодженість сформованих вибірок, що свідчить про стабільність і відтворюваність підходу. Розглянуто два альтернативні методи реалізації для генерації номерів тестових завдань: зворотної функції та прийняття-відкидання. Експериментальні результати засвідчили, що обидва підходи формують однорідні вибірки та можуть ефективно застосовуватися у практичних системах тестування. Обґрунтовано доцільність інтеграції модуля генерації завдань у структуру чат-бота, який виконує функції керування тестовою сесією: визначає допустиму множину питань, запобігає повторенням, реалізує діалогову взаємодію з користувачем, фіксує відповіді та адаптує подальший перебіг тестування. Частка правильних відповідей у середньому становила 0,71, що вказує на збалансованість між складністю запропонованих завдань і траєкторією тестування. Для профілів із вищим рівнем компетентності спостерігалось підвищення середньої складності завдань до 1,17–1,21. У менш підготовлених сценаріях цей показник знижувався до 0,79–0,81. Середня тривалість тестування складала 418,8 с. Підсумкові бали варіювалися в діапазоні від 63,7 до 88,4. Імітаційні експерименти показали, що в межах кожної сесії досягається повна унікальність завдань.

Ключові слова: кортеж параметрів, генерація випадкового числа, абсолютна частота, допустима множина, траєкторія тестування.

Abstract. The article proposes an approach to forming a set of test tasks for personnel assessment using the OpenAI ChatGPT-5.2 Plus chatbot. A mechanism for selecting questions from a task bank has been developed, ensuring the simultaneous achievement of three key requirements: the variability of test sessions, the absence of repetitions, and the preservation of the predefined assessment structure. It is shown that the use of a probability distribution for selecting difficulty intervals makes it possible to control the sequence of tasks, guiding the generator toward a higher share of medium-level questions while reducing the frequency of extreme values. The conducted simulation confirmed the statistical consistency of the generated samples, which indicates the stability and reproducibility of the approach. Two alternative methods for generating task numbers are considered: the inverse transform method and the acceptance–rejection method. Experimental results show that both approaches produce statistically homogeneous samples and can be effectively used in practical testing systems. The study also justifies integrating the task generation module into the chatbot structure. In this case, the chatbot manages the test session by defining the allowed set of questions, preventing repetition, supporting dialogue interaction with the user, recording answers, and adapting the further testing process. The average share of correct answers was 0.71, indicating a balance between task difficulty and the testing trajectory. For profiles with a higher level of competence, the average task difficulty increased to 1.17–1.21, while in less prepared scenarios it decreased to 0.79–0.81. The average duration of a test session was 418.8 seconds. Final scores ranged from 63.7 to 88.4. Simulation experiments showed that full uniqueness of tasks is ensured within each session.

Keywords: *parameter tuple, random number generation, absolute frequency, admissible set, testing trajectory.*

DOI: 10.34121/1028-9763-2026-2-58-69

1. Вступ

У сучасних умовах цифрової трансформації управлінських і кадрових процесів особливої актуальності набувають питання автоматизації оцінювання персоналу [1]. Безпосередньо процедури тестування персоналу стають важливим інструментом ухвалення кадрових рішень, зокрема під час добору кадрів, атестації, формування індивідуальних траєкторій навчання та моніторингу професійного розвитку [2].

Традиційні підходи до тестування персоналу здебільшого базуються на використанні фіксованих наборів завдань або на вибірці питань без достатнього врахування професійного профілю працівника, структури компетентностей, рівня складності і цілей конкретної процедури оцінювання [3, 4]. Такий підхід часто призводить до зниження релевантності тестування, нерівномірного покриття предметних областей, дублювання однотипних завдань, а також до недостатньої гнучкості в умовах різних сценаріїв кадрової оцінки. У результаті зменшується точність оцінювання та ускладнюється інтерпретація отриманих результатів.

Одним із перспективних напрямів удосконалення цифрових систем тестування є використання чат-ботів як інтелектуального інтерфейсу взаємодії з користувачем [5]. Застосування чат-ботів у процесах оцінювання персоналу відкриває нові можливості для організації діалогового сценарію тестування, адаптації порядку подання завдань, автоматизованої фіксації відповідей, обліку часу виконання, а також підвищення доступності й зручності проходження тесту. На відміну від традиційних вебформ, чат-бот забезпечує більш природний формат взаємодії, що може позитивно впливати на залученість користувача та ефективність проходження процедури оцінювання.

Водночас результативність застосування чат-ботів у таких системах значною мірою визначається не лише якістю діалогового інтерфейсу, а насамперед обґрунтованістю формування множини тестових завдань. Саме від складу, структури та параметрів цієї множини залежить повнота перевірки компетентностей, збалансованість тесту за рівнями складності, відповідність часовим обмеженням та здатність системи забезпечувати адаптивне й об'єктивне оцінювання персоналу. Незважаючи на наявність досліджень, присвячених електронному тестуванню [3, 6], адаптивним системам оцінювання [7] та використанню чат-ботів в освітньому й кадровому середовищі [8–9], питання формалізованого підходу до формування множини тестових завдань саме для оцінювання персоналу з використанням чат-ботів потребує подальшого опрацювання.

Отже, актуальність дослідження зумовлена необхідністю розроблення такого підходу до формування множини тестових завдань, який би поєднував можливості сучасних інформаційних технологій, принципи компетентнісного оцінювання та переваги діалогової взаємодії, що реалізується за допомогою чат-ботів. Розв'язання цього завдання дозволить підвищити об'єктивність, адаптивність і технологічність оцінювання персоналу, а також створити підґрунтя для побудови інтелектуальних систем підтримки кадрових рішень.

Метою статті є розроблення підходу до формування множини тестових завдань для оцінювання персоналу з використанням чат-ботів, який забезпечує релевантність, збалансованість, адаптивність і ефективність процедури тестування в цифровому середовищі.

Задачі роботи:

- запропонувати модель формування послідовності тестових завдань у чат-боті;
- представити підхід до генерації номера тестового завдання за заданою аналітичною функцією розподілу;

– сформулювати методологію поєднання моделі вибору завдань із чат-ботом.

2. Постановка задачі дослідження

Для побудови підходу до формування множини тестових завдань із метою оцінювання персоналу доцільно подати процес тестування у формалізованому вигляді. Це дає змогу перейти від інтуїтивного або випадкового добору завдань до керованої процедури, заснованої на чітко визначених параметрах, критеріях та обмеженнях.

Приймається, що існує база тестових завдань

$$Z = \{z_1, z_2, \dots, z_n\}, \quad (1)$$

де кожен елемент z_i є окремим завданням, доступним для включення до тесту. Таке подання є природним, оскільки в реальних інформаційних системах оцінювання персоналу використовується не одне завдання, а деякий наперед сформований банк питань або тестових елементів [3, 10]. Саме з цього банку надалі відбирається підмножина, яка найкраще відповідає цілям конкретного оцінювання.

Кожне тестове завдання з моделі (1) описується кортежем параметрів

$$z_i = \langle c_i, d_i, t_i, w_i, \tau_i, r_i \rangle. \quad (2)$$

Таке подання потрібне тому, що одне й те саме завдання не можна характеризувати лише текстом питання. Для автоматизованого формування тесту система повинна враховувати не лише зміст завдання, а й його функціональну роль у процедурі оцінювання [3].

Параметр c_i визначає компетентність, що перевіряється завданням. Його введення обґрунтоване компетентнісним підходом до оцінювання персоналу: тест не повинен бути набором випадкових питань, а має забезпечувати перевірку конкретних професійно значущих компетентностей. Без такого параметра неможливо оцінити, наскільки сформована множина завдань відповідає вимогам до посади або профілю працівника.

Параметр d_i задає рівень складності завдання. Його врахування необхідне для того, щоб тест був диференційованим і дозволяв відокремлювати працівників із різним рівнем підготовки. Якщо всі завдання будуть надто легкими або, навпаки, надто складними, результати оцінювання втратять об'єктивність. Тому рівень складності є однією з базових характеристик при конструюванні збалансованого тесту.

Параметр t_i відображає тематичну належність завдання. Його введення потрібне для забезпечення змістової повноти тестування. У межах однієї компетентності можуть існувати різні тематичні блоки, й система має контролювати, щоб сформована множина завдань не була зміщена лише в одну вузьку тему.

Параметр w_i є ваговим коефіцієнтом завдання. Він потрібний тому, що не всі завдання мають однакову діагностичну цінність. Частина завдань може перевіряти базові знання, а частина — критично важливі професійні вміння, тому їхній внесок у підсумковий результат повинен бути різним. Ваговий коефіцієнт також дає змогу гнучко налаштувати систему оцінювання залежно від цілей тесту.

Параметр τ_i задає очікуваний час виконання завдання. Його введення обґрунтовується тим, що будь-яка процедура тестування має часові межі. На практиці важливо сформулювати не просто якісний, а ще й реалістичний набір завдань, який працівник здатен виконати за відведений час. Тому при доборі завдань потрібно враховувати не лише їхню змістову користь, а й часову вартість.

Параметр r_i характеризує релевантність завдання конкретному профілю працівника. Саме цей параметр пов'язує банк завдань із персоналізованим оцінюванням. Його введення дає змогу відійти від універсального тесту для всіх і перейти до більш точного добору завдань, що найбільше відповідають посаді, функціональним обов'язкам, досвіду чи поточним цілям перевірки конкретного працівника.

Для врахування індивідуальних особливостей працівника його профіль доцільно подати у вигляді вектора

$$P = \langle p_1, p_2, \dots, p_m \rangle, \quad (3)$$

де p_j відображає значущість або рівень контролю j -ї компетентності. Таке подання є виправданим [11], оскільки в реальних задачах оцінювання різні компетентності мають неоднакову вагу. Векторний опис профілю (3) дозволяє кількісно пов'язати вимоги до працівника зі структурою тесту, а саму задачу розглянути як багатокритеріальний відбір. Саме це створює підґрунтя для побудови алгоритму автоматизованого формування тесту в системах оцінювання персоналу з використанням чат-ботів.

У межах дослідження для розрахунків використовувалася персональна робоча станція з такими характеристиками: процесор Intel Core i9 (Intel Corporation, США), оперативна пам'ять 32 GB DDR4, накопичувач SSD 1 TB. У роботі як основне середовище для аналітичних і статистичних розрахунків використовувалася операційна система Microsoft Windows 11 Pro (Microsoft Corporation, США). Як середовище програмування обрано Python (Python Software Foundation, США). Програмування та представлення результатів виконано в Google Colab (Google LLC, США). Для перевірки підходу використовувався чат-бот OpenAI ChatGPT-5.2 Plus.

3. Модель формування послідовності тестових завдань у чат-боті

Кінцева мета побудови вибірки тестових питань полягає не у використанні всіх наявних завдань (1), а у відборі найбільш доцільних із визначеним кортежем (2). У цьому випадку ставиться задача сформувати підмножину

$$Z^* \subseteq Z. \quad (4)$$

Подання результату саме як (4) є принципово важливим, бо тест завжди є вибіркою з більшої бази завдань. При цьому ця вибірка повинна бути не випадковою, а оптимальною щодо (2). Якість сформованої множини завдань можна оцінювати цільовою функцією

$$F(Z^*) = \alpha C(Z^*) + \beta R(Z^*) + \gamma B(Z^*) - \delta T(Z^*). \quad (5)$$

Представлення (5) є обґрунтованим, оскільки задача формування тесту має багатокритеріальний характер із врахуванням відношень (2) та (4). Коефіцієнти $\alpha, \beta, \gamma, \delta$ вводяться для того, щоб керувати пріоритетністю окремих критеріїв. Сама цільова функція відображає компроміс між повнотою, персоналізацією, збалансованістю та економією часу.

Для забезпечення практичної реалізованості моделі (5) вводяться обмеження.

Перше обмеження

$$\sum_{z_i \in Z^*} \tau_i \leq T_{max}$$

означає, що сумарний очікуваний час виконання всіх включених до тесту завдань не повинен перевищувати максимально допустимий час тестування. Це безпосередньо відповідає реальним організаційним умовам проведення оцінювання.

Друге обмеження

$$C_j(Z^*) \geq C_j^{min}, j = 1, \dots, m$$

вказує, що для кожної компетентності має бути забезпечений мінімально допустимий рівень покриття. Воно потрібне для запобігання ситуації, коли тест формально має високий загальний показник якості, але фактично ігнорує окремі важливі компетентності.

Третє обмеження

$$|Z^*| \leq N_{max}$$

формує кількість завдань у тесті. Воно є необхідним з погляду когнітивного навантаження на працівника, зручності проходження тесту через чат-бот та технічної доцільності.

Оскільки тест не статичний, у чат-боті доцільно використовувати адаптивний сценарій. У такому сценарії наступне завдання обирається з урахуванням кількох факторів: складності попередніх завдань, правильності відповідей, часу виконання, належності завдання до певної компетентності, а також наявності або відсутності підказки. Додатково система може використовувати професійний тезаурус для уточнення формулювань, якщо тестування спрямоване на перевірку термінологічної точності або знання предметної області.

Отже, вибір завдань (5) можна представити як логіко-ймовірнісну процедуру, в якій поєднуються логічні обмеження на допустимі переходи між питаннями, контроль неповторюваності завдань та перевірка статистичної однорідності сформованих вибірок. Припускається, що банк завдань уже структуровано за рівнем складності. Для спрощення моделювання за (5) вводиться шкала складності від 0 до 100 балів з діапазоном на 10 інтервалів. Для кожного інтервалу задається накопичена ймовірність вибору завдання (табл. 1). Це дає змогу моделювати не рівномірний, а керований розподіл, за якого найчастіше потрапляють завдання середньої складності, а дуже прості або дуже складні — рідше.

Таблиця 1 — Накопичені відносні частоти для генерації завдань за рівнем складності

№	Ліва межа	Права межа	Накопичена ймовірність P
1	0	10	0,182
2	10	20	0,398
3	20	30	0,612
4	30	40	0,776
5	40	50	0,882
6	50	60	0,943
7	60	70	0,976
8	70	80	0,991
9	80	90	0,997
10	90	100	1,000

Безпосередньо генератор реалізовано на Python на основі функції розподілу.

При реалізації моделі (5) генератор спочатку визначає інтервал складності, а потім випадково обирає конкретне завдання всередині цього інтервалу. Така схема дозволяє підтримувати задану структуру тесту та реалізувати в чат-боті сценарій «від простого до складного» або «від середнього до уточнювального».

Щоб переконатися, що генератор формує вибірки завдань за стабільним законом розподілу, можна порівняти дві незалежно згенеровані вибірки за непараметричним критерієм Пірсона. Нехай:

n_1 — обсяг першої вибірки;

n_2 — обсяг другої вибірки;

m_{1i} — кількість елементів першої вибірки в i -му інтервалі;

m_{2i} — кількість елементів другої вибірки в i -му інтервалі;

l — кількість інтервалів.

Тоді статистика критерію матиме вигляд

$$\chi^2 = n_1 n_2 \sum_{i=1}^l \frac{1}{m_{1i} + m_{2i}} \left(\frac{m_{1i}}{n_1} - \frac{m_{2i}}{n_2} \right)^2. \quad (6)$$

Формула (6) дає змогу оцінити, чи можна вважати дві вибірки такими, що отримані з одного й того самого розподілу. Якщо обчислене значення χ^2 менше за критичне табличне значення, підстав відхиляти гіпотезу про однорідність немає. Для експерименту було згенеровано дві вибірки $n_1 = n_2 = 2000$. Результати перевірки однорідності наведені на рис. 1.

```

import random
from bisect import bisect_left

# -----
# 1. Параметри моделі
# -----
N1 = 2000
N2 = 2000

# Межі інтервалів складності
interval_bounds = [(0, 10), (10, 20), (20, 30), (30, 40), (40, 50), (50, 60), (60, 70), (70, 80), (80, 90), (90, 100)]

# Накопичені ймовірності вибору інтервалу
cum_probs = [0.182, 0.398, 0.612, 0.776, 0.882, 0.943, 0.976, 0.991, 0.997, 1.000]

# Критичне значення хі-квадрат для alpha=0.05 і d
CHI2_CRITICAL = 16.919

# -----
# 2. Генерація одного завдання
# -----
def generate_task_difficulty():
    """
    Генерує складність завдання за заданим накопиченням ймовірностей
    Повертає число з інтервалу [0, 100).
    """
    u = random.random()
    idx = bisect_left(cum_probs, u)
    left, right = interval_bounds[idx]

```

Виконання розпочало
 Initializing environment
 Installing packages
 Running code
 Частоти першої вибірки: [357, 418, 434, 348, 219, 98, 77, 27, 12, 10]
 Частоти другої вибірки: [385, 413, 431, 347, 211, 105, 61, 29, 15, 3]

Обчислене значення $\chi^2 = 7.5178$
 Критичне значення $\chi^2 = 16.919$
 Висновок: немає підстав відхиляти гіпотезу про однорідність розподілів.

Інтервал	m1	m2	1/(m1+m2)	diff^2	Добуток
1	357	385	0.001348	0.00019600	2.64150943e-07
2	418	413	0.001203	0.00000625	7.52105897e-09
3	434	431	0.001156	0.00000225	2.60115607e-09
4	348	347	0.001439	0.00000025	3.59712230e-10
5	219	211	0.002326	0.00001600	3.72093023e-08
6	98	105	0.004926	0.00001225	6.03448276e-08
7	77	61	0.007246	0.00006400	4.63768116e-07
8	27	29	0.017857	0.00000100	1.78571429e-08
9	12	15	0.037037	0.00000225	8.33333333e-08
10	10	3	0.076923	0.00001225	9.42307692e-07

Рисунок 1 — Реалізація перевірки на мові Python з генерацією вибірок завдань та перевірки однорідності

Обчислення за формулою (6) дають результат $\chi^2 = 0,8698$. Кількість ступенів свободи дорівнює 9, а для рівня значущості $\alpha = 0,005$ критичне значення становить $\chi_{кр}^2 = 16,919$. Отже, генератор формує вибірки завдань статистично узгоджено, а модель (5) може бути використана для побудови сценаріїв тестування в чат-боті.

4. Генерація номера тестового завдання за заданою аналітичною функцією розподілу

У системі оцінювання персоналу з використанням чат-ботів вибір конкретного тестового завдання з банку питань може здійснюватися не лише за дискретними вагами або за належністю до інтервалів складності, а й на основі неперервної аналітичної функції розподілу. Такий підхід доцільний у тих випадках, коли потрібно забезпечити плавне керування ймовірністю вибору завдань залежно від їх складності, діагностичної цінності або близькості до цільового профілю працівника за (3).

У межах моделі (5) вважається, що всі завдання попередньо впорядковані за нормованим індексом складності $x \in [0; 2]$, де центральна область відповідає завданням середньої складності, а значення, близькі до меж відрізка, відображають простіші або складніші питання. Для генерації номера завдання приймається густина розподілу у вигляді $f(x) = A \left(1 - |x-1|^{\frac{2}{3}}\right)$, $x \in [0; 2]$. Така функція є симетричною відносно точки $x = 1$, набуває найбільшого значення в центральній частині відрізка і зменшується до країв. Для задачі тестування персоналу це означає, що система частіше добирає завдання середньої складності, які найкраще диференціюють рівень підготовки, дотримуючись (3), а крайні рівні використовуються рідше — переважно для уточнення результату.

Один із способів генерації випадкової величини з таким законом розподілу полягає у використанні методу зворотної функції. Якщо R — випадкове число, рівномірно розподілене на інтервалі $(0,1)$, тоді необхідно знайти таке значення X , для якого виконується рівність $F(X) = R$, де $F(X)$ — інтегральна функція розподілу, що відповідає густині $f(x)$.

Для вирішення задачі пропонується такий алгоритм генерації методом зворотної функції:

- Крок 1. Генерується випадкове число $R \in (0,1)$.
- Крок 2. На відрізку $[0; 2]$ розв'язується рівняння $F(x) - R = 0$.
- Крок 3. Корінь знаходиться методом дихотомії до заданої точності.
- Крок 4. Отримане значення x інтерпретується як нормований номер завдання.
- Крок 5. Отримане значення x переноситься на індекс реального питання з переліку тестових завдань.

Альтернативним способом є метод прийняття-відкидання. Він не потребує обернення інтегральної функції, а базується на порівнянні випадкової точки з графіком густини. Для цього генеруються два незалежні рівномірно розподілені числа R_1 і R_2 , після чого обчислюються залежності

$$y_1 = a + (b - a)R_1,$$

$$y_2 = f_{\max}R_2$$

за умови

$$y_2 \leq f(y_1).$$

Для задачі формування множини тестових завдань цей метод є корисним тим, що його легко модифікувати для складніших функцій густини, а також для випадків, коли потрібно оперативно змінювати форму розподілу в залежності від сценарію діалогу чат-бота.

Для порівняння двох методів було згенеровано по 2000 значень. Перша вибірка була сформована методом зворотної функції, друга — методом прийняття-відкидання. Інтервал $[0; 2]$ розбито на 10 рівних частин довжиною 0,2. Далі для кожного інтервалу підраховано абсолютні частоти попадання та значення складника статистики Пірсона. Результати порівняння наведені у табл. 2.

Таблиця 2 — Порівняння двох способів генерації номера завдання

Ліва межа	Права межа	Метод зворотної функції	Метод прийняття-відкидання	$\frac{1}{m_{1i} + m_{2i}} \left(\frac{m_{1i}}{n_1} - \frac{m_{2i}}{n_2} \right)^2$
0,0	0,2	36	42	4,92308E-07
0,2	0,4	104	96	3,20000E-07
0,4	0,6	188	203	1,47315E-07
0,6	0,8	271	284	1,14054E-07
0,8	1,0	402	389	1,06827E-07
1,0	1,2	417	431	1,15660E-07
1,2	1,4	276	263	1,51020E-07
1,4	1,6	174	181	6,90141E-08
1,6	1,8	90	79	3,60947E-07
1,8	2,0	42	32	6,75676E-07
Сума				2,55222E-06

Для $n_1 = n_2 = 2000$ статистика Пірсона дорівнює $\chi^2 = 10,20888$. Кількість ступенів свободи 9, для рівня значущості $\alpha = 0,005$ критичне значення становить $\chi_{кр}^2 = 16,919$. Оскільки $10,20888 < 16,919$, підстав для відхилення гіпотези про однорідність вибірок немає. Отже, обидва способи генерації формують статистично узгоджені результати. Обидва підходи можуть використовуватися як складові механізму формування множини тестових завдань у системі оцінювання персоналу з використанням чат-ботів.

5. Поєднання моделі вибору завдань із чат-ботом

Для практичного використання запропонованого підходу в системі оцінювання персоналу генерація номера тестового завдання повинна бути інтегрована в логіку функціонування чат-бота. У цьому випадку чат-бот виступає не лише інтерфейсом діалогової взаємодії з працівником, а й керуючим модулем, який ініціює добір завдань, контролює послідовність їх подання, фіксує результати відповідей та коригує подальшу траєкторію тестування. Чат-бот, отримавши інформацію про користувача, посаду, сценарій оцінювання або результати попередніх відповідей, формує допустиму множину завдань. Після цього з цієї множини за допомогою генератора (рис. 1) вибирається наступне завдання. Отже, математична модель генерації випадкового номера завдання виконує в чат-боті функцію механізму стохастичного керування тестовою сесією. Її призначення полягає не в хаотичному виборі питань, а в контрольованому формуванні індивідуальної послідовності завдань. Це дозволяє забезпечити одночасно дві важливі вимоги: з одного боку — варіативність тестування, а з іншого — збереження його структури та порівнюваності результатів.

Роботу чат-бота доцільно організувати як послідовність етапів.

На першому етапі чат-бот виконує ідентифікацію користувача та визначає сценарій тестування: первинний добір кандидата, атестація працівника, періодичний контроль знань, перевірка після навчання тощо. На основі цього формується профіль оцінювання за моделлю (3).

На другому етапі з банку завдань формується поточна допустима множина. До неї включаються лише ті питання, які:

- відповідають потрібним компетентностям;
- належать до дозволеного діапазону складності;
- не були використані раніше в поточній сесії;
- відповідають обмеженням за часом;
- узгоджуються з логікою діалогу чат-бота.

На третьому етапі чат-бот запускає модуль генерації наступного завдання. Саме тут використовується або метод зворотної функції, або метод прийняття-відкидання. Згенероване значення $x \in [0; 2]$ ще не є безпосередньо номером питання, а виконує роль індексу вибору в межах нормованої шкали складності. Далі це значення зіставляється з реальною підмножиною доступних завдань. Тобто генератор формує цільову точку на шкалі складності, а чат-бот вибирає найближче доступне завдання, яке ще не було використано.

На четвертому етапі чат-бот подає завдання користувачу в діалоговій формі, отримує відповідь, фіксує правильність, час виконання, факт використання підказки та інші параметри взаємодії.

На п'ятому етапі відбувається адаптація подальшої траєкторії тестування. Якщо відповідь правильна і дана швидко, система може зміщувати допустимий діапазон у бік складніших завдань. Якщо відповідь помилкова або потребує підказки, чат-бот може або знизити складність, або залишитися в тій самій зоні, змінивши тематику для уточнення рівня знань.

Алгоритм формування послідовності тестових завдань у чат-боті можна подати так:

Крок 1. Чат-бот ініціює тестову сесію та визначає профіль оцінювання працівника.

Крок 2. Із загального банку формується початкова множина допустимих завдань.

Крок 3. Відповідно до заданої аналітичної функції розподілу генерується нормоване значення складності наступного завдання.

Крок 4. Серед допустимих невикористаних завдань обирається питання, найближче до згенерованого значення.

Крок 5. Завдання подається користувачу у діалоговому форматі.

Крок 6. Чат-бот аналізує відповідь, час виконання та факт використання підказки.

Крок 7. На основі результату оновлюється стан тестової сесії та коригується допустима множина наступних завдань.

Кроки 3–7 повторюються до досягнення умови завершення тестування.

Для перевірки працездатності запропонованого підходу було проведено імітаційне випробування модуля формування послідовності тестових завдань у ChatGPT. У межах експерименту моделювалася серія тестових сесій, у яких чат-бот здійснював вибір наступного завдання на основі аналітичної функції розподілу складності з урахуванням поточного діапазону допустимих значень, відсутності повторів та результатів попередніх відповідей користувача.

Під час випробування оцінювалися такі показники:

- загальна кількість поданих завдань;
- кількість правильних відповідей;
- частка використаних підказок;
- середнє значення складності вибраних завдань;
- частка неповторюваних питань;
- середній час тестової сесії;
- інтегральний показник успішності.

Для експерименту було змодельовано 12 тестових сесій чат-бота для працівників із різними умовними профілями підготовки. У кожній тестовій групі було по 12 осіб. Кожна

сесія містила вісім завдань. Банк завдань був попередньо структурований за компетентністю, тематикою та рівнем складності в межах нормованої шкали [0; 2]. Тестування відбувалося на базі Академії праці, соціальних відносин та туризму (м. Київ). Результати тестування наведені у табл. 3.

Таблиця 3 — Результати випробування моделі формування послідовності завдань у чат-боті

№ сесії	Кількість завдань	Правильних відповідей	Частка правильних відповідей	Використано підказок	Середня складність	Неповторюваність завдань	Середній час сесії, с	Інтегральний бал
1	8	6	0,75	1	0,94	1,00	412	78,6
2	8	5	0,63	2	0,88	1,00	437	71,4
3	8	7	0,88	0	1,12	1,00	395	86,9
4	8	4	0,50	2	0,81	1,00	446	65,2
5	8	6	0,75	1	1,03	1,00	408	79,8
6	8	7	0,88	1	1,17	1,00	401	87,5
7	8	5	0,63	1	0,91	1,00	429	72,8
8	8	6	0,75	0	1,06	1,00	404	81,3
9	8	4	0,50	3	0,79	1,00	453	63,7
10	8	7	0,88	0	1,21	1,00	392	88,4
11	8	5	0,63	2	0,90	1,00	434	70,9
12	8	6	0,75	1	1,00	1,00	415	79,1
Середнє	8,0	5,67	0,71	1,17	0,99	1,00	418,8	77,1

Як видно з даних табл. 3, у всіх тестових сесіях було забезпечено повну неповторюваність завдань у межах однієї сесії, тобто жодне питання не подавалося повторно. Це підтверджує коректність реалізації механізму контролю множини вже використаних тестових елементів. Середня складність завдань за всіма сесіями становила 0,99, що є близьким до центру нормованого діапазону. Такий результат узгоджується з обраною аналітичною функцією розподілу, яка орієнтує генератор на частіший вибір завдань середньої складності. Водночас індивідуальні відмінності між сесіями свідчать, що чат-бот динамічно змінював траєкторію тестування залежно від правильності відповідей та використання підказок. Середня частка правильних відповідей склала 0,71, що свідчить про достатню узгодженість між складністю поданих завдань і рівнем підготовки змодельованих користувачів. При цьому у сильніших профілях працівників середня складність поданих завдань зростала до 1,17–1,21, тоді як у слабших сценаріях вона знижувалася до 0,79–0,81. Це підтверджує працездатність механізму адаптивного переходу між рівнями складності.

Середній час тестової сесії становив 418,8 с, що для восьми завдань є прийнятним показником з погляду діалогового формату чат-бота. При зростанні кількості підказок та зменшенні частки правильних відповідей тривалість сесії збільшувалася, що також є логічно обґрунтованим. Інтегральний бал коливався в межах від 63,7 до 88,4. Це свідчить про здатність моделі не лише формувати варіативні послідовності запитань, а й забезпечувати диференціацію результатів оцінювання персоналу.

6. Висновки

1. Запропоновано логіко-ймовірнісний механізм вибору завдань із банку питань, який поєднує випадковість, неповторюваність і контроль структури тесту. Встановлено, що використання ймовірнісного розподілу для добору інтервалів складності дозволяє забезпечити варіативність тестових сесій без порушення загальної логіки оцінювання. Проведене моделювання підтвердило статистичну однорідність сформованих вибірок.

2. Представлено підхід до використання неперервної аналітичної функції розподілу для керованого вибору номера тестового завдання. Обґрунтовано вибір густини, яка забезпечує вищу ймовірність потрапляння завдань середнього рівня складності та меншу для крайніх значень, що відповідає практиці оцінювання персоналу. Для реалізації генерації було розглянуто два методи: метод зворотної функції з чисельним розв'язанням рівняння та метод прийняття-відкидання. За результатами експериментального порівняння встановлено, що обидва методи формують статистично однорідні вибірки, а тому можуть бути використані як альтернативні інструменти вибору завдання з банку.

3. Доведено, що модуль генерації завдань доцільно розглядати як складову частину логіки функціонування чат-бота, а не як окремий технічний елемент. Чат-бот у цьому випадку виконує функції керування тестовою сесією: визначає допустиму множину завдань, виключає повтори, подає питання в діалоговому форматі, фіксує відповіді та коригує подальшу траєкторію тестування. Проведене імітаційне випробування моделі формування послідовності тестових завдань у чат-боті. Результати показали, що в усіх змодельованих сесіях забезпечується неповторюваність завдань у межах одного проходження, а середня складність поданих питань залишається близькою до цільового діапазону, визначеного аналітичною функцією розподілу. Це підтверджує практичну придатність запропонованого підходу для побудови систем оцінювання персоналу в діалоговому форматі.

СПИСОК ДЖЕРЕЛ

1. Strohmeier S. Digital human resource management: A conceptual clarification. *German Journal of Human Resource Management: Zeitschrift Für Personalforschung*. 2020. Vol. 34 (3). P. 345–365. DOI: <https://doi.org/10.1177/2397002220921131>.
2. Marler J.H., Boudreau J.W. An evidence-based review of HR Analytics. *The International Journal of Human Resource Management*. 2017. Vol. 28 (1). P. 3–26. DOI: <https://doi.org/10.1080/09585192.2016.1244699>.
3. Elements of adaptive testing. Springer / W.J. van der Linden, C.A.W. Glas (eds.). 2010. DOI: <https://doi.org/10.1007/978-0-387-85461-8>.
4. Wainer H. Computerized adaptive testing: A primer. *Psychological Methods*. 2000. Vol. 5 (1). P. 121–122. DOI: <https://doi.org/10.1037/1082-989X.5.1.121>.
5. Kryazhych O., Ivanov I., Iushchenko K., Kuprin O., Vasenko O., Riznyk V., Ryzhkov O. Devising an approach to preventing information chaos in chat bots using generative artificial intelligence. *Eastern-European Journal of Enterprise Technologies*. 2025. Vol. 2 (134). P. 84–95. DOI: <https://doi.org/10.15587/1729-4061.2025.324957>.
6. Bennett R.E. Inexorable and Inevitable: The Continuing Story of Technology and Assessment. *The Journal of Technology, Learning and Assessment*. 2002. Vol. 1 (1). URL: <https://ejournals.bc.edu/index.php/jtla/article/view/1667>.
7. Vössing J., Stamoov-Roßnagel C., Heinitz K. Text difficulty affects metacomprehension accuracy and knowledge test performance in text learning. *Journal of Computer Assisted Learning*. 2017. Vol. 33. P. 282–291. DOI: <https://doi.org/10.1111/jcal.12179>.
8. Espinoza-Acero H., Galarza-Minaya T., Vidal E. Exploring the Role of Chatbots in the Recruitment Process in Latin America. *Revista De Gestão Social E Ambiental — RGSA*. 2024. Vol. 18 (1). e07047. DOI: <https://doi.org/10.24857/rgsa.v18n1-166>.

9. Rukadikar A., Khandelwal K. Navigating change: a qualitative exploration of chatbot adoption in recruitment. *Cogent Business & Management*. 2024. Vol. 11 (1). DOI: <https://doi.org/10.1080/23311975.2024.2345759>.
10. Linear models for optimal test design / W.J. van der Linden (ed.). Springer, 2005. URL: <https://link.springer.com/book/10.1007/0-387-29054-0>.
11. Marler J.H., Boudreau J.W. An evidence-based review of HR Analytics. *The International Journal of Human Resource Management*. 2017. Vol. 28 (1). P. 3–26. DOI: <https://doi.org/10.1080/09585192.2016.1244699>.

Стаття надійшла до редакції 09.02.2026 / прийнята до друку 28.04.2026