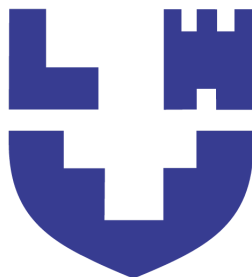


**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

Факультет цифрових, освітніх та соціальних технологій
Кафедра цифрових освітніх технологій



**ПРОЕКТУВАННЯ ТА РОЗРОБКА
ПЕДАГОГІЧНИХ ПРОГРАМНИХ ЗАСОБІВ**

Методичні вказівки до виконання практичних робіт
для здобувачів першого (бакалаврського) рівня вищої освіти
освітньої програми «Середня освіта. Інформатика» галузі знань А Освіта
спеціальності А4.09 Середня освіта (Інформатика)
денної форм навчання

УДК 004.9:37.091.3 (07)

П 79

До друку

Голова вченої ради факультету цифрових, освітніх та соціальних технологій ЛНТУ _____ Г. Герасимчук

Затверджено вченою радою факультету цифрових, освітніх та соціальних технологій ЛНТУ, протокол № _____ від «_____» _____ 2026 року.

Електронна копія друкованого видання передана для внесення в репозитарій ЛНТУ.

Директор бібліотеки _____ Н. Поліщук

Рекомендовано до видання на засіданні кафедри цифрових освітніх технологій ЛНТУ, протокол № _____ від «_____» _____ 2026 року.

Завідувач кафедри цифрових освітніх технологій _____ В. КАБАК

Укладач: _____ А. Лобацький, кандидат педагогічних наук, старший викладач кафедри цифрових освітніх технологій ЛНТУ.

Рецензент: _____ Ю. Мельничук, кандидат педагогічних наук, доцент кафедри цифрових освітніх технологій ЛНТУ.

Відповідальний за випуск: _____ В. Кабак, кандидат педагогічних наук, доцент, завідувач кафедри цифрових освітніх технологій ЛНТУ.

Проектування та розробка педагогічних програмних засобів:
методичні вказівки до виконання практичних робіт для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Середня освіта. Інформатика» галузі знань А Освіта спеціальності А4.09 Середня освіта (Інформатика) денної форм навчання / уклад. А. Лобацький. – Луцьк: ЛНТУ, 2026. – 96 с.

Методичне видання складене відповідно до чинної програми навчальної дисципліни «Проектування та розробка педагогічних програмних засобів» з метою допомогти у підготовці до виконання практичних робіт здобувачами першого (бакалаврського) рівня вищої освіти спеціальності А4.09 Середня освіта (Інформатика) і містить необхідний матеріал та рекомендації щодо виконання практичних завдань.

© А. Лобацький, 2026

ЗМІСТ

ВСТУП	4
Практична робота № 1	7
Практична робота № 2	10
Практична робота № 3	16
Практична робота № 4	19
Практична робота № 5	22
Практична робота № 6	24
Практична робота № 7	30
Практична робота № 8	36
Практична робота № 9	43
Практична робота № 10	49
Практична робота № 11	57
Практична робота № 12	65
Практична робота № 13	73
Практична робота № 14	82
ГЛОСАРІЙ	89
РЕКОМЕНДОВАНА ЛІТЕРАТУРА	95

ВСТУП

Дисципліна «Проектування та розробка педагогічних програмних засобів» посідає важливе місце у професійній підготовці майбутніх учителів інформатики за освітньою програмою «Середня освіта. Інформатика». Сучасний заклад загальної середньої освіти потребує педагога, який не лише користується готовими цифровими інструментами, а й здатний самостійно проектувати, розробляти, оцінювати та впроваджувати педагогічні програмні засоби (ППЗ) - від інтерактивних вправ і тренажерів до систем автоматизованого оцінювання з аналітикою навчальних результатів.

Метою вивчення дисципліни є формування загальних і професійних компетентностей у сфері проектування, розроблення, оцінювання та супроводу педагогічних програмних засобів, набуття спроможності діагностування освітніх потреб, уміння постановлення інженерно-педагогічних задач, формування компетентності добору дидактичних, педагогічних і технологічних архітектур, розвиток компетентності добору, створення, тестування та впровадження якісних, безпечних, доступних та інклюзивних цифрових рішень для формальної й неформальної освіти.

Практичні заняття з дисципліни утворюють цілісний проєктний цикл: від аудиту наявних ППЗ і моделювання життєвого циклу освітнього програмного забезпечення - через проектування цільових користувачів, дидактичних моделей, інструментів оцінювання й інформаційних моделей - до практичної реалізації навчальних активностей у no-code середовищах (Scratch, H5P, LearningApps, Google Forms, Quizizz), автоматизації оцінювання засобами Google Apps Script, побудови банку завдань і, нарешті, методичної аргументації та публічної демонстрації розробленого засобу.

Виконання практичних робіт забезпечує досягнення таких програмних результатів навчання: застосування сучасних інформаційно-комунікаційних та цифрових технологій у професійній діяльності (ПРН-9); володіння сучасними технологіями пошуку наукової інформації (ПРН-10); навички роботи в команді та адаптації до нових ситуацій (ПРН-11); використання ІКТ для подання,

редагування, збереження та перетворення інформації (ПРН-16); створення інформаційних моделей та їх реалізація засобами ІКТ (ПРН-21); реалізація алгоритмів розв'язання задач і розв'язування задач шкільного курсу інформатики різного рівня складності (ПРН-22).

Кожна практична робота у цих методичних вказівках має уніфіковану структуру: тема й мета роботи; теоретичні відомості, необхідні для виконання; завдання для виконання роботи; контрольні питання для захисту; рекомендована література. Наприкінці видання наведено зведений глосарій ключових термінів, що полегшує самостійне опрацювання матеріалу.

Загальні вимоги до виконання та оформлення звітів

Звіт про виконання практичної роботи оформлюється у Google Docs або у форматі .docx і має містити: титульну сторінку (назва дисципліни, номер і тема роботи, прізвище та ініціали здобувача, шифр групи, дата); мету роботи; результати виконання всіх завдань (заповнені таблиці, схеми, скріншоти з підписами, посилання на створені цифрові ресурси); відповіді на контрольні питання; висновок (4–7 речень) про набуті вміння та практичну цінність виконаної роботи.

Роботи здобувачів вищої освіти повинні бути оригінальними. Відсутність посилань на використані джерела, фабрикування та списування є прикладами академічної недоброчесності; виявлення її ознак у роботі здобувача є підставою для незарахування роботи викладачем. Використання інструментів штучного інтелекту дозволено для уточнення дослідницьких і пошукових питань, пошуку інформації для аналізу та синтезу, перевірки граматики і стилю оформлення.

Критерії оцінювання практичних робіт

Участь у практичному занятті та захист роботи оцінюються до 10 балів за заняття. Бали виставляються за такими критеріями:

Таблиця 1 – Критерії оцінювання роботи на практичному занятті

Критерії оцінювання знань здобувача	Бали
Розгорнутий, вичерпний виклад змісту питання; повний перелік необхідних категорій та понять; правильне розкриття їх змісту і механізму взаємозв'язку; здатність здійснювати порівняльний аналіз теорій, концепцій і підходів та самостійно робити логічні висновки; уміння застосовувати дидактичні принципи; аргументування власного ставлення до альтернативних поглядів; наведення прикладів застосування у навчальному процесі	10
Порівняно з відповіддю на найвищий бал не розкрито один із зазначених пунктів; або допущені неточності у формулюванні термінів і категорій, які здобувач швидко виправляє з допомогою викладача	8
Не розкрито два із зазначених пунктів; відповідь малообґрунтована, неповна; здобувач не наводить прикладів застосування у навчальному процесі; помилки виправляє лише з допомогою викладача	6
Не розкрито три чи більше пунктів; відсутні належні докази й аргументи; висновки хибні; здобувач неправильно зрозумів суть питання або не може виправити грубі помилки	2-4
Здобувач не готовий до заняття	0

Практична робота № 1

Тема: Аудит педагогічних програмних засобів (ППЗ).

Мета: ознайомитися з актуальними прикладами освітнього програмного забезпечення (функціональність, надійність, безпека даних і приватність, зручність, доступність, сумісність та інтеграції, підтримуваність, аналітика й оцінювання, локалізація); навчитися користуватися чек-лістом аудиту; сформувати навички порівняльної оцінки цифрових інструментів.

Теоретичні відомості

Сутність та класифікація ППЗ

Педагогічний програмний засіб (ППЗ) - це прикладне програмне забезпечення, призначене для вирішення дидактичних завдань, що безпосередньо реалізує технологію навчання, керує пізнавальною діяльністю учнів та сприяє взаємодії між викладачем і студентом у цифровому середовищі.

Класифікація ППЗ за дидактичним призначенням

Згідно з сучасними підходами (Гладких Г. В., Гулай О. та ін.), ППЗ поділяють на такі основні класи:

Таблиця 1.1 – Основні класи педагогічних програмних засобів

Тип ППЗ	Опис та призначення	Приклади
Тренажери	Призначені для відпрацювання практичних навичок, закріплення знань через багаторазове повторення.	Duolingo, MathGame
Симулятори	Моделюють реальні об'єкти або процеси (фізичні, хімічні, соціальні) для безпечних експериментів.	PhET Interactive Simulations
Тестові системи	Програми для контролю та діагностики рівня знань з автоматизованою перевіркою.	Quizizz, Google Forms
Інтерактивні середовища	Творчі лабораторії, де користувач може створювати власні об'єкти або проєкти.	Scratch, Canva, Padlet

Тип ППЗ	Опис та призначення	Приклади
LMS (Learning Management Systems)	Платформи для адміністрування навчальних курсів, комунікації та відстеження прогресу.	Moodle, Google Classroom, Canvas

Критерії аудиту ППЗ

Аудит освітнього ПЗ базується на стандарті **ISO/IEC 25010** та специфічних педагогічних вимогах:

- **Функціональність:** Чи відповідає інструмент заявленим освітнім цілям?
- **Зручність (Usability):** Інтуїтивність інтерфейсу та легкість навігації.

Доступність (Accessibility): Підтримка стандартів інклюзивності (WCAG 2.1) для людей з порушеннями зору/слуху.

- **Безпека та приватність:** Шифрування даних, відповідність GDPR.
- **Аналітика:** Можливість відстеження прогресу студента (Dashboard).

Чек-лист аудиту

Використовуйте цей шаблон для оцінювання обраного вами ППЗ. Оцінюйте кожен параметр за шкалою від 1 до 5.

Чек-лист оцінювання (Audit Scorecard)

- **Назва ППЗ:** _____
- **Тип (LMS, Симулятор тощо):** _____

Таблиця 1.2 – Картка аудиту педагогічного програмного засобу

№	Критерій	Питання для перевірки	Оцінка (1-5)
1	Функціональність	Чи достатньо інструментів для вивчення теми?	
2	Надійність	Чи стабільно працює засіб при слабкому інтернеті?	
3	Приватність	Чи потребує програма надлишкових персональних даних?	
4	UX/UI	Чи зрозуміло, куди натиснути без інструкції?	
5	Доступність	Чи є можливість змінити розмір шрифту або колір?	
6	Локалізація	Чи якісно перекладено інтерфейс українською?	

7	Інтеграція	Чи можна підключити сервіс до Google Drive або LMS?	
---	-------------------	---	--

Завдання для виконання роботи

- Оберіть один ППЗ за вашим напрямом підготовки (наприклад, *PhET* для фізиків, *Coursera* для економістів, *CodeCombat* для інформатиків).
- Пройдіть шлях користувача від реєстрації до виконання типового завдання в системі.
- Заповніть вищенаведений чек-лист. Обґрунтуйте кожну низьку оцінку (1-2 бали).
- Оберіть конкурентний аналог (наприклад, Google Classroom vs Microsoft Teams) та виділіть 3 ключові переваги одного над іншим у контексті *підтримуваності* та *аналітики*.

Контрольні питання для захисту роботи

- Чому симулятори вважаються ефективнішими за традиційні тренажери при вивченні природничих наук?
- Які ризики для приватності даних існують при використанні безкоштовних “хмарних” тестів?
- Як “імерсивні технології” (за Колмаковою В. О.) впливають на сучасну класифікацію ППЗ?
- Що таке SCORM-пакет і чому він важливий для сумісності ППЗ з різними LMS?

Вимоги до оформлення звіту

Звіт має містити: титульну сторінку, заповнений чек-лист, скріншоти роботи з обраним ПЗ та висновок щодо доцільності впровадження даного засобу в освітній процес.

Практична робота № 2

Тема: Побудова схеми життєвого циклу ППЗ та аналіз артефактів.

Мета: навчитися моделювати життєвий цикл шкільного ППЗ, пов'язувати етапи з артефактами, ролями та критеріями якості; сформувати базовий набір шаблонів артефактів і логіку їх трасування.

Теоретичні відомості

Життєвий цикл та моделі розроблення освітнього софту

Життєвий цикл педагогічного програмного засобу - це логічна та часова структура, що охоплює всі фази його існування. У контексті середньої освіти ППЗ розглядається як складна система, що включає контент, методику його подання, алгоритми взаємодії та засоби оцінювання результатів навчання. На відміну від загального програмного забезпечення, успіх ППЗ вимірюється не лише відсутністю технічних помилок, а й рівнем досягнення навчальних цілей користувачами.

Порівняльний аналіз моделей ЖЦ у контексті розробки освітнього софту

Вибір моделі життєвого циклу визначає стратегію управління проектом, швидкість виходу продукту на ринок (або у клас) та гнучкість у реагуванні на потреби вчителів та учнів. Для освітнього програмного забезпечення найчастіше розглядають каскадну модель (Waterfall), ітеративні підходи (Agile) та V-подібну модель (V-Model).

Таблиця 2.1 – Порівняння моделей життєвого циклу ППЗ

Характеристика	Каскадна модель (Waterfall)	Гнучка модель (Agile)	V-подібна модель (V-Model)
Логіка процесу	Лінійна та послідовна: кожен етап має завершитися перед початком наступного.	Циклічна та інкрементальна: продукт створюється невеликими частинами (спринтами).	Симетрична: кожен етап розроблення має відповідний етап тестування.

Характеристика	Каскадна модель (Waterfall)	Гнучка модель (Agile)	V-подібна модель (V-Model)
Вимоги до проєкту	Визначаються повністю на початку та фіксуються.	Можуть змінюватися та уточнюватися в процесі розроблення.	Чітко визначені та підлягають суворій верифікації.
Участь вчителя	Тільки на етапах збору вимог та фінального приймання.	Постійна участь як джерела зворотного зв'язку після кожної ітерації.	Активна роль у валідації вимог та критеріїв успіху.
Сфера застосування	Стабільні державні системи, електронні підручники за твердою програмою.	Експериментальні тренажери, ігрові навчальні середовища, стартапи.	Системи тестування (ЗНО), медичні та безпекові симулятори.
Ризики	Висока ймовірність виявлення методичних помилок наприкінці ЖЦ.	Ризик втрати загальної архітектури та безкінечного процесу змін.	Трудомісткість та великий обсяг документації.

Каскадна модель у педагогічному проєктуванні часто призводить до ситуації, коли розроблений за рік продукт виявляється методично застарілим або не враховує нові тренди у викладанні предмета. Agile-підхід дозволяє вчителю побачити “робочий шматочок” програми вже через два тижні та внести корективи у сценарій, що значно підвищує якість дидактичного наповнення. V-модель є найбільш дисциплінованою, оскільки вона змушує розробника думати про те, як він буде перевіряти знання учня ще на етапі написання першого речення технічного завдання.

Класифікація артефактів проєктування ППЗ

Артефакти в розробці ПЗ - це будь-які проміжні або фінальні продукти інтелектуальної праці, що документують хід проєкту. У проєктуванні ППЗ вони є ключовими для забезпечення трасування (traceability) - можливості простежити шлях від педагогічної мети до конкретного рядка коду.

Основні типи артефактів та їх призначення

Артефакти можна класифікувати за напрямками діяльності в межах ЖЦ.

Таблиця 2.2 – Класифікація артефактів життєвого циклу проєкту

Артефакт	Категорія	Опис та роль у проєкті
Технічне завдання (SRS)	Аналітичний	Фундаментальний документ, що описує всі вимоги до системи. Використовується як еталон для тестування та приймання робіт.
Дидактичний сценарій	Педагогічний	Опис логіки навчання: тези-визначення, вправи, умови переходу між рівнями, тексти підказок та повідомлень про помилки.
Інформаційна модель (ER-схема)	Проєктний	Графічне представлення структури даних: сутності (Учень, Тема, Оцінка), їхні атрибути та зв'язки між ними.
UX-макети (Wireframes)	Проєктний	Візуальні начерки інтерфейсу, що демонструють розташування кнопок, контенту та логіку навігації без детального графічного дизайну.
План тестування (Test Plan)	Якість	Опис методик перевірки системи, включаючи перевірку навчальних результатів (валідацію) та технічну надійність.

Важливо розуміти, що артефакти не є статичними. Наприклад, SRS у гнучких моделях (Agile) може трансформуватися у беклог продукту (Product Backlog), де кожна вимога представлена у вигляді історії користувача (User Story): “Я як учень хочу отримувати миттєву підказку після другої невдалої спроби розв’язання задачі”.⁴

Практичні завдання для студентів

Виконання практичних завдань має на меті переведення теоретичних знань у площину практичних умінь моделювання та документування.

Завдання 1: Вибір та обґрунтування моделі ЖЦ для конкретного кейсу

Студент має вибрати один із двох кейсів для розроблення та обґрунтувати вибір моделі ЖЦ (Waterfall, Agile або V-Model).

- **Кейс А: Мобільний застосунок для інтенсивної підготовки до ЗНО з математики.** Вимоги: сувора відповідність програмі УЦОЯО, великий банк завдань, висока надійність підрахунку балів, фіксована дата релізу (до початку сесії).
- **Кейс Б: Інтерактивне середовище (LMS) для сільської школи з елементами віртуальної лабораторії.** Вимоги: можливість роботи в умовах слабкого інтернету, потреба у поступовому додаванні нових практичних робіт вчителем фізики, необхідність адаптації під мінливі запити учнів.

Обґрунтування має містити:

- Оцінку стабільності вимог для обраного кейсу.
- Аналіз критичності помилок та потреб у тестуванні.
- Опис ролі вчителя у вибраній моделі.
- Висновок про доцільність обраної моделі (наприклад: “Для Кейсу Б доцільно обрати Agile, оскільки він дозволяє впроваджувати модулі віртуальної лабораторії ітераційно, збираючи відгуки учнів та удосконалюючи систему без зупинки навчального процесу” 8).

Завдання 2: Побудова візуальної схеми ЖЦ у Miro/Lucidchart

Студент має створити візуальну модель ЖЦ ППЗ для обраного у Завданні 1 кейсу. Схема повинна відображати етапи, артефакти та контрольні точки якості.

Алгоритм побудови у Miro:

1. **Створення полотна:** Відкрийте Miro та виберіть інструмент “*Shapes*”.
2. **Розмітка зон (Swimlanes):** Додайте об’єкт “Swimlane” (доріжки). Назвіть доріжки за ролями: “Методист/Вчитель”, “Аналітик/Архітектор”, “Розробник”, “Тестувальник/Учень”.³³

3. **Візуалізація етапів:** Використовуйте прямокутники для позначення фаз ЖЦ (Аналіз, Проектування тощо). Розміщуйте їх на доріжці тієї ролі, яка є провідною на цьому етапі.
4. **Додавання артефактів:** Біля кожного етапу додайте іконку документа або Callout (виноску) з назвою артефакту (наприклад, SRS, Сценарій, Код).
5. **Встановлення зв'язків:** З'єднайте фігури стрілками. У точках переходу додайте ромби (Decision point) - це моменти контролю якості, де приймається рішення про перехід до наступної фази.
6. **Оформлення:** Застосуйте кольорове кодування: наприклад, блакитний для педагогічних етапів, зелений для технічних, червоний для тестування.

Завдання 3: Робота з артефактами. Заповнення шаблону SRS

Студент має заповнити фрагмент специфікації вимог (SRS) для одного модуля ППЗ (наприклад, “Модуль реєстрації та ведення профілю учня” або “Модуль генерації випадкових тестів”).

Фрагмент шаблону SRS:

1. Назва модуля та призначення: <наприклад: Модуль автоматичного зворотного зв'язку>.

2. Функціональні вимоги (FR):

FR.01: Система має виводити текст підказки після першої помилки.

FR.02: Система має блокувати перехід до наступного рівня, якщо відсоток правильних відповідей менш як 75%.

3. Нефункціональні вимоги (NFR):

NFR.01 (Доступність): Кольорова гама інтерфейсу має відповідати стандартам WCAG для дітей з вадами зору.

NFR.02 (Безпека): Паролі учнів мають зберігатися у зашифрованому вигляді.

4. Користувацькі характеристики: Опис типового учня (вік, рівень володіння ІКТ).

Контрольні питання для самоперевірки

1. Дайте визначення життєвого циклу ППЗ. Чим він відрізняється від ЖЦ програмного забезпечення для бізнесу?.1
2. Які переваги та ризики застосування Agile-моделі під час створення ППЗ для шкільної інформатики?.7
3. Які розділи SRS є критичними для дотримання прав дитини в цифровому середовищі?.22
4. Поясніть сутність комплексного показника якості ППЗ за методикою Ю. І. Грицюка. Як визначити вагові коефіцієнти для різних типів ППЗ?.6
5. Опишіть структуру дидактичного сценарію. Чому він є основним артефактом для методиста?.5
6. У чому полягає складність моделювання предметної області в довідково-інформаційних системах вчителя (за Нестеренко Є. В.)?.1
7. Які специфічні модулі входять до складу інтелектуальних навчальних систем і як це відображається на схемі ЖЦ?.28
8. Що таке матриця трасування артефактів (RTM) і як вона допомагає забезпечити повне тестове покриття вимог ППЗ?.25

Практична робота № 3

Тема: Цільові користувачі, сценарії, цілі: основи проєктування ППЗ.

Мета: навчитися описувати цільових користувачів, формулювати сценарії взаємодії з ППЗ та ставити вимірювані навчальні цілі під конкретний шкільний урок.

Практична частина (Кейс-стаді): Проєктування модуля «Алгоритми» для 6 класу

Опис ситуації:

Ви розробляєте інтерактивний навчальний модуль «Алгоритми та їх виконавці» для 6 класу в межах впровадження НУШ. Ваш аудит цільової аудиторії показав, що клас є вкрай гетерогенним:

- **Група «Апатичні» (30%):** Мають низьку мотивацію до вивчення інформатики. Вважають програмування занадто складним і відірваним від життя. Люблять візуальний контент, але швидко втрачають інтерес при читанні довгих текстів.
- **Група «Нормативні» (50%):** Сумлінно виконують інструкції вчителя. Добре працюють за алгоритмом «дивись і повторюй», але мають труднощі з творчими завданнями, де немає чіткого зразка.
- **Група «Лідери» (20%):** Випереджають шкільну програму. Мають досвід роботи в Scratch або Minecraft Education Edition. Їм нудно виконувати лінійні вправи, вони прагнуть складності та автономії.

Завдання для виконання роботи

Завдання 1: Аналіз навчальної програми та ресурсних обмежень

Як проєктувальник освітнього середовища, проаналізуйте модельну програму з інформатики для 6 класу. Виділіть ядро знань для теми «Алгоритми».

- Сформулюйте три ключові цілі (ОРН), що стосуються циклічних та розгалужених процесів.

- Визначте технічні обмеження: ППЗ має працювати на планшетах (мобільність) та десктопах. Чи можливе використання ШІ-модулів для аналізу прогресу в офлайн-режимі?.5
- Опишіть «вхідний поріг» - які навички роботи з файлами та введенням тексту вже мають бути сформовані в учнів згідно з програмою 5 класу.

Завдання 2: Створення 2-х полярних персон

На основі даних кейсу створіть два детальні профілі учнів. Використовуйте підхід «Емпатія понад статистику».

Персона А: «Геймер Макс» (Група «Апатичні»)

- *Цифрові звички:* Грає в ігри 4+ години на день, звик до швидкого дофаміну.
- *Когнітивний бар'єр:* Низька витривалість при роботі з текстовою інформацією.
- *Технічна грамотність:* Висока в іграх, але низька в офісних застосунках.
- *Мотиваційний тригер:* Змагання, нагороди, «ігровий наратив».

Персона Б: «Старанна Анна» (Група «Нормативні»)

- *Цифрові звички:* Використовує планшет для перегляду YouTube та підготовки дошкільних завдань.
- *Когнітивний бар'єр:* Страх зробити помилку, що призведе до «збою програми».
- *Технічна грамотність:* Добре слідує візуальним підказкам.
- *Мотиваційний тригер:* Схвалення, чітке розуміння «правильності» кроку.

Завдання 3: Побудова сценарію взаємодії з ППЗ (User Flow)

Створіть схему User Flow для конкретної навчальної задачі: «Створення циклічного алгоритму для малювання візерунку».

Опишіть словами (або намалюйте схематично), як система реагує на дії Макса (якщо він поспішає і припускається помилок) та Анни (якщо вона вагається і не робить крок протягом 30 секунд).

- Вкажіть точки, де ШІ надає адаптивну підтримку на основі профілю учня.

- Опишіть імерсивні елементи (за Колмаковою В. О.), які занурюють учня в ситуацію «вирішення реальної проблеми».

Завдання 4: Заповнення матриці узгодження

Створіть таблицю, що пов'язує методичні вимоги з технічною реалізацією в ППЗ.

Таблиця 3.1 – Зв'язок навчальних цілей за Блумом із механіками ППЗ

Рівень за Блумом	Навчальна ціль	Механіка завдання в ППЗ	Критерій оцінювання
Remember	Називає компоненти циклу	Сортування блоків «Тіло циклу», «Умова», «Лічильник»	Правильність класифікації елементів
Understand	Пояснює, скільки разів виконається цикл	Вибір результату з 4-х варіантів після перегляду анімації коду	Правильна ідентифікація кількості ітерацій
Apply	Складає алгоритм за зразком	Конструктор коду з обмеженим набором блоків	Скрипт працює без помилок
Analyze	Знаходить логічну помилку в циклі	Режим «Debug»: клік по хибному блоку в готовому скрипті	Точність знаходження помилки
Create	Створює оригінальний циклічний малюнок	Free-mode конструктор з можливістю ділитися результатом	Наявність складних структур (вкладені цикли)

Практична робота № 4

Тема: Модель навчання для ППЗ: аналіз і рішення.

Мета: навчитися обирати та обґрунтовувати дидактичну модель (blended / flipped / PBL) під конкретну тему уроку, цілі, цільового користувача й умови класу; зафіксувати роль ППЗ на етапах уроку.

Теоретичні відомості

Педагогічний програмний засіб (ППЗ) - це прикладне програмне забезпечення, яке реалізує технологію навчання та керує пізнавальною діяльністю учнів. У сучасній освіті ППЗ не існують автономно, а інтегруються в конкретні дидактичні моделі.

Основні моделі цифрового навчання:

- **Blended Learning (Змішане навчання):** поєднання традиційного очного навчання з дистанційними формами, де цифрові ресурси забезпечують ефективність освітнього процесу.
- **Flipped Classroom (Перевернутий клас):** модель, де теоретичний матеріал опрацьовується учнями самостійно за допомогою ІКТ поза класом, а аудиторний час використовується для практичної діяльності.
- **PBL (Project-Based Learning):** навчання через виконання довготривалих проєктів, де ППЗ виступають інструментами дослідження та створення продукту.

Роль ППЗ на етапах уроку

Ефективність моделі залежить від того, яку функцію виконує ППЗ на кожному етапі:

Таблиця 4.1 – Функції ППЗ на різних етапах уроку

Етап уроку	Функція ППЗ	Приклади інструментів
Підготовка	Актуалізація знань, мотивація, самостійне вивчення теорії.	Google Forms, YouTube, H5P
Опрацювання	Формування практичних навичок, процесів, симуляція	Scratch, PhET, CodeCombat +1

Етап уроку	Функція ППЗ	Приклади інструментів
Оцінювання	Контроль знань, рефлексія, автоматизований фідбек.	Quizizz, Google Forms

Завдання для виконання роботи

1. Кейс-аналіз: Вибір моделі

Оберіть тему з курсу інформатики (наприклад: “*Основи алгоритмізації*”, “*Веб-дизайн*” або “*Кібербезпека*”) та заповніть матрицю обґрунтування моделі:

- **Тема:** _____
- **Цільова аудиторія (клас):** _____
- **Ресурси:** (наявність гаджетів, інтернет)

Таблиця 4.2 – Обґрунтування вибору моделі навчання

Компонент	Обрана модель (Blended/Flipped/PBL)	Обґрунтування вибору
Теорія		Чому саме такий спосіб подачі?
Практика		Як ППЗ допоможе у діяльності?
Взаємодія		Як організована співпраця?

2. Проєктування навчального треку (Learning Path)

Створіть ланцюжок використання цифрових інструментів для обраної теми. Заповніть таблицю, вказавши конкретні ППЗ:

- **Етап “****До уроку****”** (якщо Flipped):** Який ресурс учень опрацює самостійно?
- **Етап “****У класі****”**: Який засіб (наприклад, Scratch) використовується для практики?
- **Етап “****Контроль****”**: Яка система (наприклад, Quizizz) використана для формуального оцінювання?

3. Обґрунтування інклюзивності

Згідно з критеріями аудиту (Accessibility), вкажіть, чи дозволяють обрані вами ППЗ адаптувати контент для учнів з особливими освітніми потребами (зміна шрифту, озвучення тексту тощо).

Контрольні питання для самоперевірки

1. Як інтеграція змішаного навчання впливає на розвиток фахової компетентності студента/учня?
2. У чому полягає роль викладача як “тьютора” в моделі PBL порівняно з традиційним уроком?
3. Які технічні вимоги (стабільність інтернету, локалізація) є критичними для реалізації моделі Flipped Classroom?
4. Чому для проектної діяльності (PBL) інтерактивні середовища (наприклад, Canva або Scratch) є більш доцільними за звичайні тренажери?

Практична робота № 5

Тема: Формувальне та підсумкове оцінювання: рубрики й тести.

Мета: спроектувати інструменти оцінювання під цілі уроку: рубрику для формувального оцінювання та міні-тест для підсумкового; узгодити їх з рівнями таксономії Блума, принципом SMART і роллю ППЗ.

Завдання для виконання роботи

Завдання 1. Проєктування SMART-цілей та підбір ППЗ (20 хв.)

Оберіть одну тему з курсу шкільної інформатики (наприклад: «Алгоритми з розгалуженням», «Моделювання в 3D», «Створення запитів у БД»).

- Сформулюйте **дві цілі** за SMART: одна на рівні «Застосування», інша на рівні «Аналіз» або «Створення».
- Визначте **ППЗ**, у якому учні виконуватимуть завдання (наприклад, Scratch, Blender, Access).

Завдання 2. Конструювання рубрики для формувального оцінювання (40 хв.)

Розробіть рубрику оцінювання для практичного завдання в обраному ППЗ.

- **Кількість критеріїв:** 3–4 (наприклад: правильність алгоритму, оптимізація коду, оформлення інтерфейсу, дотримання інструкцій).
- **Шкала:** 4 рівні (Початковий, Середній, Достатній, Високий).
- **Вимога:** Опишіть рівні так, щоб учень міг сам зрозуміти, чого йому не вистачає для вищого балу.

Завдання 3. Розробка підсумкового міні-тесту (30 хв.)

Створіть у цифровому середовищі (Google Forms/Quizizz) міні-тест із 5 запитань, що перевіряють досягнення поставлених SMART-цілей.

- *Запитання 1-2:* Вибір однієї правильної відповіді (рівень: Знання/Розуміння).
- *Запитання 3-4:* На встановлення відповідності або послідовності (рівень: Розуміння/Застосування).
- *Запитання 5:* Аналіз фрагмента (наприклад, «Що виведе даний код?» або «Де помилка на скріншоті?») (рівень: Аналіз).

Контрольні питання для захисту роботи

- У чому полягає різниця між формувальним і підсумковим оцінюванням?
- Як узгодити критерії формувального оцінювання з навчальними цілями SMART і рівнями Блума?
- Як забезпечити об'єктивність і надійність тестових завдань у цифровому середовищі?
- Які типові помилки роблять при створенні тестових завдань і як їх уникнути?

Практична робота № 6

Тема: Академічна доброчесність у цифровому оцінюванні.

Мета: опанувати практики запобігання списуванню та забезпечення чесності в онлайн-оцінюванні; налаштувати інструменти (Google Forms / H5P Quiz / Quizizz / Kahoot) з випадковими варіантами, часовими лімітами, банком питань і чіткими інструкціями для учнів.

Теоретичні відомості

1. Академічна доброчесність: поняття і законодавча база

1.1. Визначення і принципи

Академічна доброчесність - це сукупність етичних принципів і визначених законом правил, якими мають керуватись учасники освітнього процесу під час навчання, викладання та проведення оцінювання. Закон України «Про освіту» (ст. 42) визначає академічну доброчесність через перелік заборон та обов'язків для учнів і педагогів.

Основні принципи академічної доброчесності, що безпосередньо стосуються оцінювання:

- **Самостійність:** результат оцінювання відображає власні знання учня, а не знання Інтернету, ШІ або однокласника.
- **Достовірність:** надані відповіді є правдивими і відповідають реальному рівню знань учня.
- **Справедливість:** всі учні працюють у рівних умовах: однаковий обсяг часу, однакова складність, однакові правила.
- **Відповідальність:** учень усвідомлює наслідки порушень - як для власної репутації, так і для педагогічного процесу.

1.2. Законодавча база в Україні

Академічна доброчесність регулюється кількома нормативними актами: 2

Таблиця 6.1 – Нормативні документи з академічної доброчесності

Документ	Що регулює стосовно оцінювання
ЗУ «Про освіту», ст. 42	Визначає академічну доброчесність, перелічує порушення (плагіат, фабрикація, обман, списування), встановлює відповідальність

Документ	Що регулює стосовно оцінювання
ЗУ «Про повну загальну середню освіту», ст. 12	Обов'язок учня дотримуватись академічної доброчесності; право вчителя застосовувати різні форми перевірки
Положення про організацію освітнього процесу (школа)	Внутрішній документ: конкретні правила для онлайн-оцінювання, наслідки порушень
ЗУ «Про захист персональних даних»	Обмеження на збір і зберігання персональних даних учнів у тестових платформах

Ключова норма: ЗУ «Про освіту», ст. 42 - списування визначається як отримання неправомірної переваги під час оцінювання, у тому числі з використанням технічних засобів. Вчитель зобов'язаний вживати заходів для забезпечення доброчесності, а не просто фіксувати порушення.

2. Загрози академічній доброчесності в цифровому оцінюванні

2.1. Класифікація загроз

Загрози доброчесності в онлайн-оцінюванні умовно поділяються на три групи залежно від засобів:3

Таблиця 6.2 – Загрози академічній доброчесності в онлайн-оцінюванні

Група загроз	Конкретні прийоми	Наскільки поширено	Технічний захист
Соціальні (між учнями)	Передача відповідей у чаті; показ екрану; повідомлення «правильних» варіантів після проходження	Дуже поширено	Рандомізація порядку; обмеження часу; заборона повторного перегляду
Технологічні (ШІ, пошук)	ChatGPT / Copilot для генерації відповідей; Google Search під час тесту; переклад питань у перекладач	Зростає стрімко	Завдання на Apply/Analyze рівні Блума (ШІ важче генерує); відкриті питання; контекстні задачі

Група загроз	Конкретні прийоми	Наскільки поширено	Технічний захист
Технічні (обхід платформ)	Використання кількох пристроїв; підміна акаунту; відкриття форми в інкогніто після перегляду	Помірно поширено	Прив'язка до акаунту домену; обмеження 1 спроба; збір email
Завчасна підготовка питань	Запам'ятовування питань з попереднього проходження; обмін питаннями між паралелями	Системно поширено при фіксованих тестах	Банк питань з рандомним відбором; різні варіанти для різних паралелей

3. Технічні механізми забезпечення доброчесності: детальний огляд

3.1. Рандомізація питань і варіантів відповідей

Рандомізація - найефективніший і найменш стресовий технічний захід - реалізується на двох рівнях: перемішування порядку питань (кожен учень бачить питання у різній послідовності, тому «злите» в чат «питання 3» не допоможе іншому) та перемішування варіантів відповідей (правильна відповідь «В» в одного учня може бути «А» в іншого). Для підсумкового тесту обидва рівні є обов'язковими, для формувального бліцу - рекомендованими; у навчальній вправі з фіксованою послідовністю пояснень рандомізація може бути педагогічно недоречною, адже порушує продуману логіку подання матеріалу.

3.2. Часові ліміти: як встановлювати правильно

Таймер - це не лише засіб боротьби зі списуванням. Це і **педагогічний інструмент**: він відображає реальні умови оцінювання і формує навичку тайм-менеджменту. Водночас невірний встановлений таймер є Extraneous Load - він відволікає від змісту і підвищує тривогу.

Таблиця 6.3 – Рекомендований час на типи тестових питань

Тип питання	Рекомендований час	Обґрунтування
Remember (визначення, терміни)	20–30 сек	Факт або знаєш, або ні - довга пауза не допоможе

Тип питання	Рекомендований час	Обґрунтування
Understand (класифікація, пояснення)	45–60 сек	Потрібно обдумати, але не дуже довго
Apply (задача, вибір рішення)	90–120 сек	Потрібна оцінка ситуації і вибір підходу
Analyze (знайти помилку, порівняти)	2–3 хв	Потрібен детальний розгляд матеріалу
Відкрите питання (есе, коротка відповідь)	5–10 хв	Потрібно сформулювати і написати власну думку

Загальний час тесту = сума часів на питання × 1.3 (коефіцієнт запасу для повільніших учнів). Для учнів з ООП (СДУГ, тривожні розлади) - × 1.5–2.0.

Педагогічне правило: Таймер має створювати помірний тиск, а не паніку. Якщо 30% учнів не встигають - таймер занадто короткий, а не учні «занадто повільні». Перегляньте час, а не очікування.

4. Налаштування доброчесності у конкретних платформах

Порівняння можливостей чотирьох платформ для забезпечення академічної доброчесності наведено в таблиці.

Таблиця 6.4 – Порівняння платформ за засобами забезпечення доброчесності

Платформа	Ключові налаштування доброчесності	Обмеження
Google Forms	Перемішування порядку запитань і варіантів відповідей; збір e-mail та обмеження до однієї відповіді; режим заблокованого екрана (Chromebook у Google Workspace for Education)	Немає вбудованого таймера (потрібні доповнення); журнал дій недоступний
Quizizz	Банк питань і випадковий відбір; таймер на кожне питання; приховування правильних відповідей; контрольоване повторне проходження	Ігрова механіка може відволікати; частина налаштувань - у платному тарифі
Kahoot	Таймер на питання; нік-генератор проти образливих імен; випадковий порядок питань	Орієнтований на синхронну гру в класі; слабкий контроль самотійного проходження

H5P Quiz	Повний контроль на власному сервері (LMS): пул питань, рандомізація, обмеження спроб, журнал подій xAPI	Потребує власного розміщення (Moodle, WordPress); складніше початкове налаштування
----------	---	--

5. Формування культури доброчесності в учнів

5.1. Чому технічних заходів недостатньо

Технічні заходи усувають **можливість** списувати, але не усувають **бажання**. Справжня академічна доброчесність - це культурна характеристика, що формується роками. Вчитель відіграє ключову роль у її формуванні: 11

- **Поясніть цінність самостійного результату:** учень розуміє, що підказки «знищують» його власні знання. Аргумент: «Якщо ти списав сьогодні - завтра ти не зможеш самостійно на ДПА».
- **Зробіть помилку прийнятною:** elaborative фідбек після тесту показує, де учень помилився і чому - це знижує страх перед «провалом» і зменшує мотивацію до списування.
- **Оцінюйте процес, а не лише результат:** якщо завдання передбачає рефлексію («що ти дізнався, що найважче?»), скопіювати її набагато важче, ніж MC-відповіді.
- **Чіткі правила до початку:** учні мають знати умови тесту ДО того, як він розпочнеться. Сюрпризний таймер або заборона повторного перегляду без попередження - це несправедливо, навіть якщо технічно легально.

Завдання для виконання роботи

1. Проведіть аудит власного тесту (розробленого на практичній роботі № 5) на вразливості до академічної недоброчесності: складіть перелік ризиків (списування, обмін відповідями, підглядання у сторонні вкладки, використання генеративного ШІ) і оцініть кожен ризик за шкалою «низький / середній / високий».
2. Налаштуйте обраний інструмент оцінювання (Google Forms, Quizizz, Kahoot або H5P Quiz): увімкніть перемішування питань і варіантів відповідей, часовий ліміт, обмеження кількості спроб та збір ідентифікаційних даних. Зафіксуйте налаштування скріншотами.

3. Складіть чіткі інструкції для учнів перед тестуванням (до 150 слів): що дозволено, що заборонено, які наслідки порушень, як технічно організовано контроль. Інструкції мають формувати культуру відповідальності, а не атмосферу недовіри.
4. Розробіть фрагмент банку питань: 6 питань у двох еквівалентних варіантах кожне (однакова складність і рівень за Блумом, різні формулювання та числові дані) для випадкового відбору.

Контрольні питання для захисту роботи

1. Що означає поняття академічної доброчесності в сучасній освіті?
2. Які основні принципи академічної доброчесності визначені законодавством України (ЗУ «Про освіту», ст. 42)?
3. Як технологічні засоби (чат-боти, скріншоти, підглядання) впливають на об'єктивність оцінювання?
4. Які технічні налаштування цифрових платформ можна використати для контролю чесності (обмеження часу, перемішування запитань, випадкові варіанти)?
5. Як формувати у здобувачів освіти культуру відповідальності за результати власного навчання?

Практична робота № 7

Тема: Моделювання предметної області: сутності, атрибути, зв'язки (ER-схема).

Мета: навчитися виділяти сутності й атрибути та будувати зв'язки для шкільної задачі; оформити ER-схему та таблицю словника даних під подальшу реалізацію ППЗ (тести, трекінг прогресу тощо).

Перш ніж реалізовувати будь-який ППЗ - тест, тренажер, систему трекінгу прогресу - необхідно відповісти на запитання: **які дані зберігатиме ця система і як вони пов'язані між собою?** Відповідь на це запитання і є предметом інформаційного моделювання.

ER-схема (Entity-Relationship Diagram) - це графічна модель предметної області, що описує **сутності** (що зберігається), **атрибути** (які властивості має кожна сутність) і **зв'язки** (як сутності пов'язані між собою). Навіть для по-code ППЗ, де немає жодного рядка SQL-коду, ER-схема є корисним проектним артефактом: вона змушує розробника до початку роботи чітко визначити, що саме система «знатиме» про учня, завдання і результат.

1. Основні поняття: сутності, атрибути, зв'язки

1.1. Сутність (Entity)

Сутність - це клас реальних або концептуальних об'єктів, про які система зберігає інформацію; у ER-схемі зображається прямокутником з назвою всередині. Правила виокремлення: сутність - це те, про що можна поставити кілька різних питань («Учень» - сутність, бо має ім'я, клас, результати; «Правильно» - лише значення); назва - іменник в однині (Учень, Питання, Тест); сутність існує незалежно від інших. Типові сутності шкільного ППЗ: Учень, Питання, Тест, Відповідь, Результат, Тема.

1.2. Атрибут (Attribute)

Атрибут - це властивість сутності, що зберігається в системі; у класичній нотації - еліпс, на практиці атрибути записують списком усередині прямокутника сутності. Ключові типи атрибутів: первинний ключ (PK) - унікальний ідентифікатор запису (question_id); простий - неподільне значення (question_text); похідний - обчислюється з інших (is_correct); зовнішній ключ

(FK) - посилання на ключ іншої сутності, що реалізує зв'язок (test_id у сутності Питання).

1.3. Зв'язок (Relationship)

Зв'язок описує, як пов'язані дві сутності; його найважливіша характеристика - кардинальність: один-до-одного (1:1 - один Учень має один Профіль налаштувань), один-до-багатьох (1:N - один Тест містить багато Питань), багато-до-багатьох (M:N - багато Учнів проходять багато Тестів). Практичне правило: зв'язок M:N завжди розв'язується через третю, асоціативну сутність (наприклад, Спроба між Учнем і Тестом), яка містить зовнішні ключі обох сторін.

2. ER-схема для шкільного ППЗ: покроковий алгоритм

2.1. П'ять кроків від задачі до схеми

Моделювання ER-схеми для шкільного ППЗ виконується за стандартним алгоритмом. Кожен крок відповідає на конкретне питання:4

- **Виокремити сутності** - записати всі іменники з технічного завдання (постановки задачі). Кожен іменник-кандидат = можлива сутність. Відсіяти ті, що є лише значеннями або властивостями.
- **Визначити атрибути кожної сутності** - для кожної сутності запитати: «Що система повинна пам'ятати про цей об'єкт?» Виділити ключовий атрибут (унікальний ідентифікатор, РК).
- **Визначити зв'язки між сутностями** - для кожної пари сутностей запитати: «Чи пов'язані вони? Як саме?» Назвати зв'язок дієсловом: «Учень складає Тест», «Тест містить Питання».
- **Визначити кардинальність** - для кожного зв'язку визначити: скільки один може мати другого і навпаки (1:1, 1:N, M:N). Якщо M:N - додати асоціативну сутність.
- **Намалювати схему** - прямокутники (сутності) + лінії зі значками кардинальності + список атрибутів. Перевірити: чи можна відновити весь потрібний ФФнкціонал, маючи тільки схему?

2.2. Приклад: ER-схема для простого тесту

Завдання: «Учні здають тест. Тест складається з питань. Відповіді учнів записуються для подальшого аналізу.»

Крок 1 - іменники: учень, тест, питання, відповідь, результат.

Крок 2 - атрибути:

Таблиця 7.1 – Сутності та атрибути предметної області

Сутність	Атрибути (PK підкреслено)
Учень	student_id, прізвище, ім'я, клас, email
Тест	test_id, назва, тема, дата_проведення, ліміт_часу
Питання	question_id, текст_питання, варіант_a, варіант_b, варіант_v, правильна_відповідь, рівень_блума
Відповідь	answer_id, обрана_відповідь, час_відповіді, is_correct
Результат	result_id, загальний_бал, кількість_правильних, дата_здачі

Крок 3 - зв'язки:

- Учень *складає* Тест (один учень може скласти кілька тестів → 1:N)
- Тест *містить* Питання (один тест має кілька питань → 1:N)
- Учень *дає* Відповідь (учень відповідає на кожне питання → Відповідь є асоціативною сутністю між Учнем і Питанням)
- Проходження (асоціативна) *має* Результат (1:1)

Крок 4 - M:N між Учнем і Питанням: одне питання відповідають багато учнів, один учень відповідає на багато питань. Вводимо асоціативну сутність **Відповідь** з атрибутами: answer_id (PK), student_id (FK), question_id (FK), обрана_відповідь, is_correct.

2.3. Типові ER-патерни для шкільного ППЗ

Три найпоширеніших патерни, які зустрічаються у проєктах студентів:4

Таблиця 7.2 – Типові патерни проектування бази даних

Патерн	Структура (стисло)	Коли застосовувати
Тест з відповідями	Учень - Тест - Питання - Відповідь; Відповідь є асоціативною між Учнем і Питанням	Google Forms тест, Quizizz квіз, будь-яке оцінювання
Прогрес учня	Учень - Тема - Результат_по_темі; Тема - Активність - Завершення	Трекінг навчального прогресу в LMS або Sheets Dashboard
Банк питань	Тема - Питання (1:N); Питання - Тест (M:N через Включення); Включення має порядковий_номер	Автоматична генерація варіантів тесту з банку (лаб. №13)

3. Словник даних

3.1. Що таке словник даних

Словник даних (Data Dictionary) - це **таблиця-специфікація всіх атрибутів кожної сутності** з точним описом типу, обмежень і призначення кожного поля. Якщо ER-схема показує *що існує*, то словник даних показує *як точно* це зберігається.

Для no-code ППЗ (Google Forms → Sheets) словник даних відповідає **заголовкам стовпців таблиці відповідей**: кожен стовпець = атрибут = рядок у словнику.

Словник даних оформлюють як таблицю, в якій для кожного атрибута фіксують сутність, назву атрибута, тип даних, обмеження (обов'язковість, унікальність, діапазон) і короткий коментар. Приклад фрагмента словника даних для сутності УЧЕНЬ:

Таблиця 7.3 – Словник даних предметної області

Сутність	Атрибут	Тип	Обмеження	Коментар
УЧЕНЬ	id_учня	ціле	PK, унікальний	ідентифікатор запису
УЧЕНЬ	ПІБ	текст (100)	обов'язковий	повне ім'я учня

УЧЕНЬ	клас	текст (5)	обов'язковий	наприклад, «9-А»
-------	------	-----------	--------------	---------------------

4. ER-схема у draw.io: практичний мінімум

4.1. Чому draw.io для навчального проєкту

diagrams.net (draw.io) - це безкоштовний веб-редактор схем, що: не потребує реєстрації, зберігається у Google Drive, підтримує стандартні форми для ER-діаграм (Entity Relation shapes у боковому меню) і може бути вбудований як посилання у звіт. Саме тому він є рекомендованим інструментом для практичної роботи №7.6

4.2. Покрокова інструкція: проста ER-схема у draw.io

- Відкрийте app.diagrams.net → «Create New Diagram» → «Entity Relationship» (або просто Blank).
- Для кожної сутності: перетягніть прямокутник → введіть назву сутності великими літерами (УЧЕНЬ).
- Атрибути: всередині прямокутника або нижче - перерахуйте поля. РК позначте як *_student_id (РК)* або підкресліть.
- Зв'язки: лінія між сутностями → клацніть правою кнопкою на лінію → Edit Connection → задайте кардинальність (наприклад: «1» і «N» з обох боків).
- Збережіть: File → Save to Google Drive (або Export as PNG для звіту).

Альтернативні інструменти: Miro (студентський акаунт безкоштовно), Lucidchart (обмежений безкоштовний), Google Slides (простий варіант - прямокутники і стрілки вручну).

Завдання для виконання роботи

1. Оберіть шкільну задачу для моделювання (тематичний тест із автоперевіркою, трекінг прогресу учня або банк питань). Випишіть із постановки задачі всі іменники-кандидати у сутності та відсійте ті, що є лише значеннями або властивостями.
2. Для кожної відібраної сутності складіть таблицю атрибутів із зазначенням ключового атрибута (РК) і зовнішніх ключів (FK).

3. Побудуйте ER-схему в середовищі draw.io (diagrams.net): мінімум 3 сутності, зв'язки з кардинальністю; зв'язок M:N реалізуйте через асоціативну сутність. Експоруйте схему у PNG для звіту.
4. Заповніть словник даних (мінімум 8 полів): назва поля, сутність, тип даних, обов'язковість, обмеження, призначення.

Контрольні питання для захисту роботи

1. Що таке предметна область і чому її важливо моделювати перед створенням ППЗ?
2. Які переваги має побудова ER-схеми для навчального програмного продукту?
3. Як правильно встановлювати зв'язки між сутностями в контексті навчального ППЗ?
4. Які графічні елементи використовуються для побудови ER-схеми (прямокутники, овали, ромби, лінії)?

Практична робота № 8

Тема: Матриця вибору інструментів (no-code стек).

Мета: навчитися добирати інструменти без коду (Scratch / H5P / LearningApps / Google Forms / Quizizz / Kahoot / хмарні диски) під цілі уроку, умови класу та вимоги до оцінювання.

Після виконання практичної роботи студент зможе:

- характеризувати основні no-code платформи за призначенням, перевагами та обмеженнями;
- будувати зважену матрицю вибору інструментів для конкретного педагогічного контексту;
- обґрунтовувати вибір технологічного стеку (2–3 інструменти) на рівні 3 (теорія + дані);
- аргументовано відповідати на 4 питання обговорення з програми.

Компетентності, що формуються:

Таблиця 8.1 – Компетентності, що формуються у роботі

Компетентність	Реалізація у цій роботі
ЗК-1: Аналіз та синтез	Порівняльний аналіз інструментів за критеріями; прийняття обґрунтованого рішення
ЗК-5: Навички ІКТ	Практична оцінка і налаштування реальних no-code платформ
ПРН-9: Цифрові технології	Проектування технологічного стеку ППЗ під конкретні педагогічні умови

Стислі теоретичні відомості

Зверніться до теоретичних відомостей до теми 8. Нижче - довідкова таблиця для виконання завдань.

Орієнтиром для зіставлення платформ слугують їхні профілі та порівняльна матриця «платформа × сценарій», докладно розглянуті у практичній роботі № 9.

Наскрізний кейс практичної роботи

Ситуація: Вчителька інформатики 9 класу Ірина Сергіївна починає розробляти цифрові матеріали до розділу «Алгоритмізація та програмування»

(28 учнів). Школа має Google Workspace, стабільний Wi-Fi у класі, всі учні мають ПК або ноутбуки. Ірина Сергіївна планує три типи активностей: (1) підсумковий тест наприкінці теми (оцінка у журнал), (2) щорочний 5-хвилинний бліц-опитування для поточного контролю, (3) практична вправа на класифікацію типів алгоритмів (drag & drop). Усі 4 завдання виконуються від її імені.

Завдання 1. Аналіз вимог і характеристик платформ

(Репродуктивне - 2 бали)

Ваше завдання: визначити вимоги до кожного з трьох типів активностей і порівняти кандидатів-платформ за заданими критеріями.

1.1. Вимоги до кожної активності

Заповніть таблицю вимог до кожного з трьох типів активностей Ірини Сергіївни.

Таблиця 8.2 – Вимоги до типів навчальних активностей

Параметр	(1) Підсумковий тест	(2) Бліц-опитування	(3) Drag & drop вправа**
Потрібна email-ідентифікація?	Так - оцінка у журнал		
Потрібен таймер?			
Потрібна аналітика результатів?			
Важливо: рандомізація питань?			
Потрібна ДРА-угода платформи?	Так - офіційне оцінювання		
Рівень Блума (1–3)			
Учні можуть виконати вдома?			

1.2. Порівняльна оцінка платформ-кандидатів

Оцініть кожну платформу за шкалою 1–5 (1 = не відповідає, 5 = ідеально). Перший рядок - зразок.

Таблиця 8.3 – Оцінювання платформ для створення активностей

Критерій	Google Forms	Quizizz	Kahoot	H5P	LearningApps
Підтримує email-ідентифікацію	5	3	1	3	1
Підтримує таймер					
Підтримує drag & drop / складні типи					
Якість аналітики					
Безпека / наявність DPA					
Зручність для учня (UX 1–5)					
Безкоштовний тариф достатній?					

Завдання для звіту:

- Заповнені таблиці 1.1 і 1.2 - всі клітинки.
- Відповідь (2–3 речення): «Що означає підхід no-code у створенні освітніх продуктів? Чому Ірина Сергіївна може обійтись без програміста?»

Завдання 2. Побудова зваженої матриці вибору

(Аналітичне - 3 бали)

Ваше завдання: побудувати три окремих зважених матриці вибору - по одній для кожного типу активності Ірини Сергіївни. Результат кожної матриці - аргументований вибір.

Як рахувати зважену суму: Зважений бал = Оцінка (1–5) × Вага критерію. Потім сума зважених балів по всіх критеріях = Підсумок інструменту. Найвищий підсумок - рекомендований вибір (якщо немає dealbreaker).

2.1. Матриця 1: Підсумковий тест (оцінка у журнал)

Ваги для підсумкового оцінювання: Безпека ×3; Педагогічна відповідність ×3; Функціональність ×2; UX ×2; Вартість ×1. Оцінюйте Google Forms, Quizizz, Kahoot.

Таблиця 8.4 – Зважене оцінювання платформ оцінювання

Критерій (вага)	Google Forms Оцін × Вага	Quizizz Оцін × Вага	Kahoot Оцін × Вага
Безпека / DPA (×3)			
Педагогічна відповідність (×3)			
Функціональність (×2)			
UX для учня (×2)			
Вартість (×1)			

Таблиця 8.5 – Підсумковий вибір інструмента

Рекомендований інструмент	Обґрунтування (3–4 речення)

Матрицю 2 (бліц-опитування для поточного контролю) і матрицю 3 (drag & drop вправа на класифікацію) побудуйте за тим самим шаблоном, що й матрицю 1: визначте 4–5 критеріїв, призначте ваги (сума дорівнює 1,0), оцініть платформи-кандидати за шкалою 1–5, обчисліть зважені суми та зафіксуйте платформу-переможця для кожної активності з коротким обґрунтуванням.

Завдання 3. Технологічний стек і роль хмарних сервісів

(Проектувальне - 2 бали)

Ваше завдання: зібрати фінальний технологічний стек Ірини Сергіївни з трьох обраних інструментів і пояснити роль Google Drive у цьому стеку.

3.1. Технологічний стек

Таблиця 8.6 – Технологічний стек проекту ППЗ

Компонент ППЗ	Обраний інструмент	Чому саме він (з матриці)	Де зберігаються результати
Підсумкове оцінювання			
Поточний контроль / бліц			
Практична вправа (drag & drop)			
Зберігання матеріалів і результатів	Google Drive	Інфраструктурний компонент стеку; G Suite DPA	Google Sheets - автоматично від Forms

3.2. Роль хмарних сервісів у стеку

Відповідайте від імені Ірини Сергіївни на питання обговорення:

Таблиця 8.7 – Відповіді на питання для обговорення

Питання	Відповідь Ірини Сергіївни (3–4 речення)
Які можливості пропонують хмарні сервіси (Google Drive, OneDrive) для організації спільної роботи та зберігання ресурсів ППЗ?	
У яких випадках варто комбінувати кілька no-code рішень? Наведіть приклад зі свого стеку.	

Завдання для звіту:

- Заповнена таблиця технологічного стеку (3.1).
- Дві відповіді на питання обговорення (3.2).

Завдання 4. Відповіді на питання обговорення і методична аргументація

(Рефлексивне - 3 бали)

Ваше завдання: дати розгорнуті відповіді на всі 4 питання обговорення з програми дисципліни і підготувати обґрунтування вибору для педради.

4.1. Обґрунтування вибору для педради

Ірина Сергіївна представляє стек на методичній раді. Заповніть аргументацію рівня 3 (теорія + дані + норма):

Таблиця 8.8 – Аргументація вибору стеку на методичній раді

Запитання від колеги	Аргументована відповідь Ірини Сергіївни
«Навіщо три інструменти? Чи не достатньо одного?»	
«Чому ви не обрали Kahoot для підсумкового тесту - він же зручний?»	
«Як ви гарантуєте безпеку даних учнів у цих платформах?»	

Завдання для звіту:

- Відповіді на 4 питання обговорення (4.1).
- Відповіді на 3 заперечення колег (4.2).

Питання для обговорення та захисту роботи

Обирається по одному питанню з кожної групи.

Група А: Знання

- Що означає підхід no-code? Назвіть 3 переваги і 2 обмеження порівняно з традиційною розробкою.
- Що таке матриця вибору інструментів? Які 5 обов'язкових елементів вона має містити?
- Яку роль виконують Google Drive і OneDrive у структурі ППЗ? Чому їх не вважають «активними» інструментами?

Група Б: Аналіз

- Порівняйте Google Forms і Quizizz для підсумкового оцінювання: безпека, функціональність, аналітика.
- Чому Kahoot не підходить для підсумкового оцінювання? Що саме відсутнє?

Група В: Застосування

- Покажіть вашу найкращу матрицю наживо. Поясніть кожну оцінку і підсумок.

- Ірина Сергіївна дізнається, що школа переходить на Microsoft 365. Як це вплине на стек? Що потрібно змінити?

Практична робота № 9

Тема: Створення інтерактивної навчальної активності.

Мета: спроектувати й реалізувати мінімально життєздатну активність (MVP) у середовищі без коду (Scratch / H5P / LearningApps / Google Forms / Quizizz), що має інструкцію для учня, основну взаємодію та базове оцінювання і зворотний зв'язок.

Тема 8 дала відповідь на запитання «який інструмент обрати і чому». Тема 9 відповідає на інше: **«як саме ці інструменти влаштовані, які типи активностей вони реалізують і в яких сценаріях кожен є найсильнішим?»** Якщо тема 8 - це стратегічне проектування, то тема 9 - це тактичне занурення: студент переходить від порівняльних таблиць до практичного розуміння архітектури кожної платформи.

Особливість цієї теми полягає у тому, що п'ять розглянутих платформ - **Scratch, H5P, LearningApps, Genially та Google Forms** - є якісно різними за своєю природою. Scratch є середовищем програмування. H5P - фреймворком для упаковки інтерактивного HTML5-контенту. LearningApps - конструктором ігрових вправ. Genially - візуальним редактором інтерактивних презентацій. Google Forms - системою збирання даних з можливостями тестування. Кожна з цих платформ виникла з різних потреб і вирішує різні педагогічні задачі.

Знання архітектурних особливостей кожної платформи дозволяє проектувальнику ППЗ не просто “використовувати інструменти”, але й **проектувати навчальний досвід** - планувати, на якому етапі уроку і з якою метою активується кожен компонент, як інструменти взаємодіють між собою і як вимірюється результат навчання учня.

1. Концептуальні засади: від редактора до навчального середовища

1.1. MVP-підхід у проектуванні no-code ППЗ

Поняття MVP (Minimum Viable Product - мінімально життєздатний продукт) прийшло у освітнє проектування зі стартап-культури. У контексті no-code ППЗ MVP означає: **найпростіша версія навчальної активності, що реалізує основну педагогічну функцію** і може бути протестована з реальними учнями без доопрацювань.

MVP у no-code ППЗ складається з трьох обов'язкових компонентів:

Таблиця 9.1 – Обов'язкові компоненти MVP no-code ППЗ

Компонент MVP	Що включає	Чому обов'язковий
Інструкція для учня	Чіткий текст або візуальна підказка на старті: що робити, скільки часу, що буде після завершення	Без інструкції учень витрачає час на «вгадування правил» замість навчання; когнітивне навантаження зростає
Основна взаємодія	Тип завдання, що реалізує навчальну мету: відповідь на питання, перетягування, заповнення, програмування	Це і є серцевина ППЗ - без взаємодії немає навчальної активності
Базове оцінювання / фідбек	Мінімальна реакція системи на дії учня: правильно/неправильно, бал, коментар, або хоча б підтвердження отримання відповіді	Без фідбеку учень не знає, чи досяг мети; вчитель не може виміряти результат

Принцип MVP: Краще мати просту, але завершену активність з інструкцією, взаємодією і фідбеком - ніж складну, але «розпочату». MVP можна запустити, отримати зворотний зв'язок від учнів і вдосконалити. Незавершений проєкт неможливо ні запустити, ні поліпшити.

2. Профілі no-code платформ

П'ять платформ, що утворюють базовий no-code стек учителя інформатики, відрізняються педагогічною нішею, моделлю розміщення та умовами безкоштовного використання. Стислі профілі платформ наведено в таблиці, а розгорнуте порівняння за сценаріями уроку - в наступному підрозділі.

Таблиця 9.2 – Базовий no-code стек учителя інформатики

Платформа	Педагогічна ніша	Сильні сторони	Обмеження
Scratch	Конструктивістське навчання через створення проєктів:	Повна свобода логіки (події, змінні, цикли); велика спільнота і	Немає вбудованого журналу результатів;

	тренажери, ігрові вправи, симуляції (8–16 років)	галерея прикладів; безкоштовний і відкритий	потребує продуманої архітектури проекту
H5P	Стандартизовані інтерактивні модулі (Interactive Video, Course Presentation, Branching Scenario, Quiz) для вбудовування в LMS	Понад 40 типів контенту; аналітика xAPI; адаптивні сценарії розгалуження; відкритий код	Потребує розміщення (Moodle, WordPress, Lumi); обмежена кастомізація дизайну
LearningApps.org	Швидкі ігрові вправи на класифікацію, відповідність, послідовності	Найнижчий поріг входу; десятки готових шаблонів; колекції вправ для класу; безкоштовний	Застарілий інтерфейс; примітивний зворотний зв'язок; немає детальної аналітики
Genially	Інтерактивні презентації, інфографіки, квести й мікрокурси для перевернутого класу	Висока візуальна якість; інтерактивні шари і гарячі точки; гейміфіковані шаблони	Аналітика та експорт - у платних тарифах; немає повноцінного тестування
Google Forms	Тести з автоперевіркою, опитування, збір даних з аналітикою в Sheets	Глибока інтеграція з Google Workspace і Classroom; безкоштовний без обмежень; розширюваність через Apps Script	Обмежені типи інтерактивності; дизайн майже не налаштовується

3. Порівняльна характеристика платформ: сценарії та компроміси

3.1. Матриця «платформа × сценарій»

Жодна з п'яти платформ не є «найкращою» абсолютно - кожна є найкращою у своєму сценарії. Наступна матриця фіксує рекомендоване використання: 13

Таблиця 9.3 – Відповідність no-code платформ навчальним сценаріям

Навчальний сценарій	Scratch	H5P	LearningApps	Genially	Google Forms
Пояснення нового матеріалу (вчитель показує)	✗ (не для цього)	★★★★ Interactive Video, Presentation	✗	★★★★★ Презентація, інфографіка	✗
Домашнє вивчення матеріалу (учень сам)	★★ (якщо є вправа-тренажер)	★★★★ Course Presentation, Branching	★★ (прості вправи)	★★★★★ Мікрокурс, Flipped Classroom	★★ (з розгалуженням)
Закріплення та повторення (після пояснення)	★★★★ Квіз або тренажер	★★★★ Question Set, Drag & Drop	★★★★ (ідеальний сценарій)	★★★★ Ігрова дошка, Escape Room	★★★★ Тест з фідбеком
Формування оцінювання (бліц, exit ticket)	✗ (немає аналітики)	★★★★ (хAPI при наявності LMS)	✗ (немає аналітики)	★★ (базова аналітика)	★★★★★ Ідеальний сценарій
Підсумкове оцінювання	✗	★★★★ (при Moodle)	✗	✗ (публічні в безкоштовному)	★★★★★ з DPA та рандомізацією
Розвиток алгоритмічного мислення	★★★★★ Основне призначення	★★ (Branching Scenario)	✗	✗	✗
Учнівський творчий проєкт	★★★★★ Основне призначення	★★★★ Course Presentation	✗	★★★★★ Власна презентація /інфографіка	★★ (форма-портфоліо)
Робота при слабкому Wi-Fi	★★★★ (Scratch Desktop офлайн)	★★ (частково офлайн)	★★ (кеш браузера)	★★ (потребує завантаження)	★★★★ (Forms кешується)

Навчальний сценарій	Scratch	H5P	LearningApps	Genially	Google Forms
Адаптивне навчання (різні шляхи)	★★ (з логікою)	★★★★★ Branching Scenario	×	★★★ Нелінійна навігація	★★★ Розгалуження розділів

3.2. Типовий no-code стек для різних дидактичних моделей

Таблиця 9.4 – Типовий no-code стек для дидактичних моделей

Дидактична модель	Роль інструментів у моделі	Рекомендований стек
Пряме навчання (Direct Instruction)	Вчитель пояснює → учні практикують → вчитель перевіряє	Genially (пояснення) + LearningApps (практика) + Google Forms (перевірка)
Перевернутий клас (Flipped Classroom)	Вдома: учень вивчає матеріал самостійно → у класі: вирішує задачі	Genially або H5P (домашній матеріал) + Scratch або H5P (класна вправа) + Google Forms (exit ticket)
Проектне навчання (PBL)	Учні створюють власний продукт як відповідь на дослідницьке питання	Scratch (навчальна програма або гра) + Genially (презентація результатів) + Google Forms (peer review)
Змішане навчання (Blended Learning)	Частина навчання онлайн, частина очно; LMS об'єднує	Google Classroom + H5P (вправи) + Google Forms (оцінювання) + Scratch (проектні задачі)

Завдання для виконання роботи

1. Оберіть тему шкільного курсу інформатики та платформу для активності за «правилом 5 секунд вибору» (див. шпаргалку). Обґрунтуйте вибір у 2–3 реченнях з посиланням на дидактичну архітектуру платформи.
2. Спроектуйте MVP-активність: складіть таблицю трьох обов'язкових компонентів (інструкція для учня, основна взаємодія, базове оцінювання та фідбек) із конкретним змістом кожного.

3. Реалізуйте активність в обраному середовищі (мінімум 5 завдань або взаємодій). Додайте у звіт пряме посилання та 3 скріншоти: стартовий екран, екран взаємодії, екран результату.
4. Протестуйте активність двома проходженнями - «як сильний учень» (усе правильно) і «як учень, що помиляється». Зафіксуйте виявлені дефекти та внесені виправлення.

Контрольні питання для захисту роботи

1. Що таке інтерактивна навчальна активність у контексті ППЗ?
2. Які платформи краще підходять для різних типів активностей?
3. Що означає поняття MVP у педагогічному проєктуванні ППЗ?
4. Які обов'язкові елементи повинна містити мінімальна версія активності (інструкція, взаємодія, оцінювання)?
5. Як описати логіку проходження активності у вигляді сценарію або flowchart?

Практична робота № 10

Тема: Навігація та зворотний зв'язок: проєктуємо шлях учня.

Мета: навчитися проєктувати зрозумілий маршрут дій учня (екрани, кроки) і налаштовувати чіткий зворотний зв'язок (підказки, повідомлення, бали) у вибраному no-code інструменті.

Після виконання практичної роботи студент зможе:

- ідентифікувати тип навігаційної архітектури і виявляти навігаційні дефекти (тупики, відсутній фідбек);
- будувати User Flow для активності рівня 2 (лінійна навігація з elaborative фідбеком);
- писати elaborative фідбек (принцип 4E) та 2-рівневі підказки;
- реалізовувати спроектовану навігацію у no-code платформі і тестувати всі маршрути.

Компетентності, що формуються:

Таблиця 10.1 – Компетентності, що формуються у роботі

Компетентність	Реалізація у цій роботі
ЗК-1: Аналіз та синтез	Аудит навігаційних дефектів; декомпозиція активності на стани та переходи
ЗК-5: Навички ІКТ	Практична реалізація навігаційної схеми у no-code інструменті
СК-23: Проєктування «людина–техніка–середовище»	Проєктування навчального UX з урахуванням когнітивного навантаження учня
ПРН-9: Цифрові технології у профдіяльності	Реалізація ППЗ з якісною навігацією і педагогічно обґрунтованим фідбеком

Стилі теоретичні відомості

Для виконання завдань зверніться до теоретичних відомостей до теми 10.1

Типи навігаційних архітектур та елементи User Flow

Таблиця 10.2 – Типи навігаційних архітектур та User Flow

Тип навігації	Логіка переходів	Приклад у no-code
Лінійна	Строго від кроку 1 до N, без розгалужень	Google Forms без розгалуження; LearningApps
Розгалужена	Різні маршрути залежно від відповіді або результату	H5P Branching Scenario; Google Forms з розгалуженням розділів
Вільна	Учень обирає маршрут самостійно	Genially (нелінійна карта)

Принцип elaborative feedback (4E)

Таблиця 10.3 – Принцип elaborative feedback (4E)

Критерій 4E	Що означає	Приклад для теми «Цикли»
Empathetic (співчутливий)	Тон підтримує, не карає	«Майже! Подивись на умову виходу.»
Explicit (явний)	Вказує ЧИМ саме неправильно	«Ти обрав while, але тут кількість повторів відома.»
Explanatory (пояснювальний)	Пояснює ЧОМУ правильна відповідь правильна	«for використовується, коли ітерацій фіксована кількість.»
Enabling (орієнтуючий)	Дає наступний крок для виправлення	«Спробуй ще раз: запитай себе - скільки разів виконається тіло?»

В User Flow використовують стандартні позначення: прямокутник - екран або стан, ромб - точка розгалуження (умова), стрілка - перехід між станами, подвійний контур - початковий і кінцевий стани маршруту.

Наскрізний кейс практичної роботи

Ситуація: Після практичної роботи №9, де Ігор Степанович створив базовий MVP, учні написали у зворотному зв'язку: «незрозуміло, що робити далі», «не знаємо, чи правильно відповіли», «застрягли після третього питання». Це три навігаційні та фідбек-проблеми. Ігор Степанович вирішує

вдосконалити MVP - спроектувати повноцінну навігацію з elaborative фідбеком і підказками для вправи «Тренажер: for vs while» (9 клас, 5 питань, Google Forms або H5P). Всі 4 завдання - послідовний проєктний цикл від аудиту до готової реалізації.

Завдання 1. Аудит навігаційних дефектів у MVP

(Репродуктивне - 2 бали)

Ваше завдання: проаналізувати існуючий ППЗ-ресурс і виявити навігаційні дефекти та проблеми із зворотним зв'язком за структурованим чек-листом.

1.1. Об'єкт аудиту

Відкрийте будь-який тест або вправу на тему «Алгоритми» / «Програмування» (Google Forms, LearningApps, H5P - знайдіть самостійно або отримайте від викладача). Пройдіть активність повністю - як учень, що знає матеріал, і як учень, що навмисно помиляється.

1.2. Чек-лист навігаційних дефектів

Перевірте MVP за вісьмома критеріями, фіксуючи для кожного оцінку (виконано / порушено / частково) та обов'язкове пояснення для порушень: 1) стартовий екран має назву, мету, інструкцію і кнопку «Почати»; 2) із кожного стану є вихід - тупиків немає; 3) учень бачить прогрес («питання X з Y»); 4) кнопка «Далі» завжди на тому самому місці; 5) учень отримує реакцію на кожен відповідь; 6) фідбек пояснює, чому відповідь правильна чи ні, а не лише констатує; 7) при помилці є підказка або пояснення; 8) фінальний екран показує підсумок і наступний крок.

1.3. Висновок аудиту

Таблиця 10.4 – Висновок аудиту: контрольні питання

Питання	Ваша відповідь
Тип навігаційної архітектури та обґрунтування (2 речення)	
Найкритичніший дефект - який і чому він шкодить навчанню?	
Що зроблено добре - найсильніший навігаційний елемент	

Завдання для звіту:

- Заповнений чек-лист (1.2) з поясненнями.
- Заповнений висновок (1.3).
- Скріншот тупика або відсутнього фідбеку (якщо виявлено) з підписом.

Завдання 2. Побудова User Flow для тренажера

(Аналітичне - 2 бали)

Ігор Степанович хоче мати схему шляху учня ще до того як відкрив редактор платформи. **Ваше завдання:** побудувати User Flow для тренажера «for vs while» з 3 питаннями, elaborative фідбеком і петлею повтору.

2.1. Специфікація станів

Опишіть кожен унікальний стан активності (мінімум 8):

Таблиця 10.5 – Опис станів навчальної активності

Назва стану	Тип стану	Що бачить учень	Доступні дії / кнопки
Стартовий екран	інструкція	Назва, мета, час, кнопка «Почати»	«Почати»
Питання 1			
Фідбек+ до П1			
Фідбек- до П1 (рівень 1)			
Фідбек- до П1 (рівень 2)			
Питання 2			
Питання 3			
Фінальний екран			

2.2. User Flow (текст або схема)

Побудуйте схему, використовуючи позначення з теоретичних відомостей. Текстовий варіант або вставте зображення (Miro / draw.io / від руки + фото).

[User Flow - текстова схема або зображення]

2.3. «Сухий прогін» - перевірка маршрутів

Перелічіть у таблиці всі можливі маршрути від СТАРТ до КІНЕЦЬ (номер маршруту, стисла послідовність станів, умова, чи досягнуто кінець) і переконайтеся, що тупиків немає; мінімум - чотири маршрути, серед них «усі відповіді правильні», «усі неправильні» та маршрути з петлею повтору.

До звіту увійдуть: специфікація станів (п. 2.1, від 8 рядків), User Flow (п. 2.2, текст або зображення), таблиця «сухого прогону» (від 4 маршрутів без тупиків) і відповідь на питання «яка педагогічна мета петлі повтору у вашому User Flow?» (2 речення).

Завдання 3. Написання elaborative фідбеку та підказок

(Проектувальне - 3 бали)

Схема готова. Тепер - зміст. **Ваше завдання:** написати elaborative фідбек і 2-рівневі підказки для трьох питань тренажера. Спочатку сформулюйте питання.

3.1. Зміст питань

Таблиця 10.6 – Зміст питань навчальної активності

Питання	Повний текст питання	Правильна відповідь	Типова помилкова відповідь
П1	Виберіть тип циклу: «Вивести числа від 1 до 10». А) for Б) while В) Обидва підходять	А) for - кількість ітерацій відома	В) Обидва (технічно можливо, але треба вчити обирати оптимальний)
П2			
П3			

3.2. Elaborative фідбек (принцип 4Е)

Для кожного питання напишіть два варіанти фідбеку: при правильній та при неправильній відповіді. Максимум 2–3 речення кожен.

Таблиця 10.7 – Зворотний зв'язок до питань активності

Питання	Фідбек при правильній відповіді(Explicit + Explanatory)	Фідбек при неправильній відповіді(Empathetic + Explicit + Enabling)
П1 (for vs while)	«Правильно! for використовується, коли кількість ітерацій відома до початку - саме так: «від 1 до 10» = 10 повторень, фіксовано.»	«Майже! Ти обрав [варіант]. Але подумай: скільки разів виконається тіло? Якщо кількість відома - це for. Спробуй ще раз.»
П2		
П3		

3.3. Прогресивні підказки (2 рівні)

Рівень 1 - активується після 1-ої помилки (контекстна). Рівень 2 - після 2-ої (стратегічна).

Таблиця 10.8 – Дворівневі підказки до питань

Питання	Підказка рівня 1 (контекстна)Мах 1–2 речення	Підказка рівня 2 (стратегічна)Мах 2 речення
П1	«Запитай себе: скільки разів виконається тіло циклу? Чи відома ця кількість до початку?»	«Правило: якщо кількість повторень відома заздалегідь - for. Якщо залежить від умови - while.»
П2		
П3		

Завдання 4. Реалізація та методична аргументація

(Рефлексивне / презентаційне - 3 бали)

Маємо схему і зміст фідбеку. **Ваше завдання:** реалізувати навігаційну схему у вибраній платформі, протестувати всі маршрути та підготувати методичне обґрунтування.

4.1. Реалізація у платформі

Оберіть одну платформу - Google Forms або H5P (h5p.org):

- **Google Forms:** 3 питання → для кожного налаштуйте «Відповідь» (поле фідбеку при правильній та неправильній відповіді). Підказку реалізуйте як перший рядок фідбеку при неправильній відповіді (рівень 1), окремий розділ «Підказка рівня 2» - через розгалуження.
- **H5P Question Set:** налаштуйте поля «Тір» (підказка рівня 1), «Correct answer feedback» та «Incorrect answer feedback» для кожного питання. У «Overall Feedback» - фінальне повідомлення за діапазонами балів.

4.2. Документування реалізації

У звіті зафіксуйте: обрану платформу та URL активності; де саме в налаштуваннях реалізовано підказки рівнів 1 і 2; три скріншоти - elaborative фідбек при правильній відповіді, фідбек із підказкою при неправильній відповіді та фінальний екран з підсумком і наступним кроком.

4.3. Методична аргументація (для Ігоря Степановича)

Підготуйте коротке обґрунтування активності (5 розділів):

Таблиця 10.9 – Обґрунтування навчальної активності

Розділ	Зміст (заповнити)
1. Навчальна мета Яку прогалину вирішує тренажер?	
2. Тип навігації та її педагогічна логіка Чому лінійна з петлею повтору, а не просто лінійна?	
3. Якість зворотного зв'язку Чим elaborative фідбек кращий за «правильно/неправильно»? (1–2 речення з посиланням на Hattie & Timperley)	
4. Місце в уроці Коли використовувати: після пояснення / перед контрольною / вдома?	
5. Відомі обмеження Що активність НЕ перевіряє? Що буде у версії 2.0?	

Завдання для звіту:

- Пряме посилання на реалізовану активність.
- Таблиця документування (4.2) - всі рядки + 3 скріншоти з підписами.

- Таблиця тестування (4.3) - без **X** або з поясненням.
- Методичне обґрунтування (4.4) - всі 5 розділів.

Питання для обговорення та захисту роботи

Обирається по одному питанню з кожної групи. Активність демонструється наживо.

Група А: Знання понять

- Що таке «тупик» у навігації ППЗ? Наведіть приклад і поясніть, як виявити тупик ще до публікації.
- Поясніть принцип 4E elaborative фідбеку. Наведіть приклад «поганого» фідбеку і виправте його за кожним критерієм.
- Чим підказка рівня 1 відрізняється від підказки рівня 2? Чому не можна давати одразу підказку рівня 2?

Група Б: Аналіз та порівняння

- Порівняйте реалізацію elaborative фідбеку у Google Forms та H5P Question Set. Яка платформа дає більше педагогічних можливостей для фідбеку?
- У вашому тренажері є петля повтору після неправильної відповіді. Скільки разів учень може повторити питання? Яка педагогічна логіка обмеження або відсутності обмеження?

Група В: Демонстрація та застосування

- Продемонструйте активність наживо: пройдіть маршрут «дві помилки на П1 → підказки рівня 1 і 2 → правильна відповідь → наступне питання». Поясніть кожен перехід.
- Учень отримав 1/3 правильних. Що він бачить на фінальному екрані? Чи достатньо цих даних Ігорю Степановичу для педагогічного рішення?

Практична робота № 11

Тема: Прототип навчальної вправи у Scratch.

Мета: засвоїти вміння створення прототипу вправи у Scratch з чіткою метою, простим геймплеєм, підрахунком балів і зрозумілим зворотним зв'язком для учня.

1. Поняття навчальної вправи у Scratch

Scratch - це візуальне середовище програмування, розроблене Lifelong Kindergarten Group у MIT Media Lab (М. Resnick та ін., запущено у 2007 р., десктопна версія - 2014, веб-версія Scratch 3.0 - 2019). Призначене насамперед для учнів 8–16 років, але активно використовується у методичній підготовці майбутніх учителів інформатики як інструмент швидкого прототипування навчальних вправ. У Scratch код будується з готових блоків-команд, що з'єднуються як деталі конструктора, - це усуває синтаксичні бар'єри і дозволяє зосередитися на алгоритмічній логіці.

Навчальна вправа у Scratch - це інтерактивний цифровий ресурс, побудований у середовищі Scratch, який має чітку навчальну ціль (узгоджену з ОРН шкільної програми), передбачає взаємодію учня (вибір, введення, перетягування, рух), забезпечує оцінювання результату (підрахунок балів, повідомлення) і надає зворотний зв'язок (фідбек), що допомагає учневі побачити правильність дій і зрозуміти причину помилки.

На відміну від суто ігрової активності, навчальна вправа має вимірювану дидактичну мету, перевіряє конкретний навчальний результат і дає учневі зворотний зв'язок щодо правильності дій; ігрові елементи (сюжет, бали, персонажі) виконують у ній допоміжну мотиваційну роль.

2. Класифікація типів навчальних вправ у Scratch

Робоча програма дисципліни виокремлює п'ять типів завдань, які найкраще реалізуються у Scratch: класифікація, обчислення, реакція, послідовності, логіка. Цей перелік покриває переважну більшість навчальних потреб шкільного курсу інформатики 5–11 класів.

Таблиця 11.1 – Типи навчальних вправ у Scratch

Тип вправи	Що перевіряє	Приклад теми	Ключові блоки Scratch
Квіз з варіантами	Розпізнавання, відтворення	Типи алгоритмів, типи даних	when green flag clicked, ask, if...then, change score
Класифікація / drag-and-drop	Аналіз, упорядкування за критерієм	Розрізнення «змінна / стала / константа»	when this sprite clicked, go to mouse-pointer, if touching...
Обчислення	Застосування формули, точність	Перетворення систем числення, формули	ask, if answer = ..., change score
Реакція / тренажер	Швидкість прийняття рішення	Тренажер логічних операторів	wait, timer, if key ... pressed
Послідовності	Алгоритмічне мислення	Порядок етапів алгоритму	list, add to list, order check
Логічна симуляція	Розуміння причинно-наслідкових зв'язків	Робота умовних операторів, циклів	if...then...else, repeat, variables

Орієнтири вибору типу: для перевірки знань (рівень Remember/Understand за Блумом) - квіз; для застосування правил (Apply) - обчислення або тренажер; для аналізу і синтезу (Analyze/Evaluate) - класифікація або послідовності; для творчості (Create) - логічна симуляція з вільним експериментом.

3. Архітектура Scratch-проєкту

Перш ніж відкривати редактор, методист має спроектувати архітектуру майбутньої вправи на папері. Це дозволяє уникнути типової помилки - «нашаровування» блоків без логіки, коли проєкт швидко стає некерованим.

3.1. Базові компоненти

Сцена (Stage) - фон, на якому відбувається дія. У навчальній вправі використовується для зміни «контексту» (стартовий екран → екран запитання

→ екран результату). Зміна фону через блок `switch backdrop to ...` підтримує загальну навігацію.

Спрайти (Sprites) - об'єкти, з якими взаємодіє учень. У навчальній вправі типово використовуються:

- **спрайти-кнопки** - для вибору варіанта відповіді;
- **спрайт-помічник** - персонаж, який дає фідбек (костюми «думає», «радіє», «пояснює»);
- **спрайт-фон-питання** - окремий спрайт для тексту запитання, що дозволяє легко його змінювати без зміни сцени;
- **декоративні спрайти** - мінімум, лише ті, що несуть функцію.

Події (Events) - точки запуску скриптів. Ключові у навчальній вправі: `when green flag clicked` (старт), `when this sprite clicked` (вибір кнопки), `when I receive...` (broadcast-сигнал від іншого спрайта).

Змінні (Variables) - пам'ять проєкту. У навчальній вправі обов'язкові:

- **score** (бали учня);
- **currentQuestion** (індекс поточного запитання);
- **answerLocked** (прапорець «відповідь захищена», запобігає подвійному нарахуванню балів);
- **difficulty** (рівень складності, якщо реалізовано диференціацію).

Списки (Lists) - впорядковані набори даних. У навчальній вправі використовуються для зберігання банку запитань (текст запитання, варіанти відповідей, правильний індекс), банку фідбеку, історії дій учня.

Broadcast-повідомлення - внутрішній «сигнал», який один спрайт надсилає, а інші отримують. Це основний механізм координації між спрайтами без прямого зв'язку. Типові повідомлення у навчальній вправі: `startGame`, `showQuestion`, `checkAnswer`, `showResult`, `endGame`.

3.2. Принцип розділення відповідальності

Кожен спрайт має робити одну річ. Якщо спрайт-кнопка реагує на клік, нараховує бали, змінює фон і ще й виводить фідбек - це поганий дизайн. Краще: кнопка лише надсилає broadcast «обрано варіант №2», а далі окремий «логічний» спрайт обробляє вибір.

4. SMART-ціль і узгодження з ОРН

Сильна навчальна вправа починається не з блоків, а з мети. Стандартний інструмент - формулювання за принципом SMART (Specific, Measurable, Achievable, Relevant, Time-bound).

4.1. Формулювання SMART-цілі

Слабке формулювання: *«Учень навчиться розрізняти типи алгоритмів».*

Сильне формулювання за SMART: *«Після виконання вправи учень 7-го класу за 10 хвилин самостійно правильно класифікує не менше 4 з 5 запропонованих алгоритмів за типом (лінійний / розгалужений / циклічний)».*

Що сильніше:

- **S (specific):** не «розрізняти», а «класифікувати за типом»;
- **M (measurable):** «не менше 4 з 5» - є чисельний поріг успіху;
- **A (achievable):** обсяг 5 завдань адекватний 10 хвилинам;
- **R (relevant):** тип алгоритму - базова тема ОРН з інформатики;
- **T (time-bound):** 10 хв - ясне обмеження.

4.2. Узгодження з ОРН шкільної програми інформатики

Будь-яка Scratch-вправа має обслуговувати конкретний результат навчання шкільної програми. Без цього вправа залишається «технологією заради технології». Узгодження проводиться через таблицю відповідності, де кожна механіка вправи прямо співвідноситься з пунктом ОРН.

Приклад таблиці узгодження вправи «Три типи алгоритмів» (7 клас, тема «Алгоритми та виконавці»):

Таблиця 11.2 – Узгодження вправи з очікуваними результатами навчання

ОРН (за програмою МОН)	Механіка у Scratch-вправі	Чим перевіряється
Учень розрізняє лінійний, розгалужений і циклічний алгоритми	Квіз: 5 запитань на класифікацію псевдокоду	Кількість правильних відповідей $\geq 4/5$
Учень наводить приклади алгоритмів кожного типу	Спрайт-помічник вимагає назвати приклад до кожного типу	Учень вводить текст; вчитель перевіряє якісно

ОРН (за програмою МОН)	Механіка у Scratch-вправі	Чим перевіряється
Учень аналізує помилки у класифікації	Elaborative фідбек: пояснення «чому це не лінійний»	Учень читає пояснення після помилки

Методичний наслідок: якщо студент-розробник не може заповнити такої таблиці - ймовірно, його вправа не має чіткого освітнього призначення, і її потрібно або переформулювати, або розробити іншу.

5. Педагогічні механіки

5.1. Підрахунок балів зі захистом від подвійного нарахування

Типова помилка: учень клацає правильну кнопку двічі, бали додаються двічі. Рішення - використання прапорця `answerLocked`. Перед кожним новим запитанням він скидається на 0, після першого кліку встановлюється на 1, і подальші кліки ігноруються.

Псевдокод обробника кнопки:

```
when this sprite clicked «Кнопка А»* // подія*
```

```
if answerLocked = 0:* // ще не відповів*
```

```
set answerLocked = 1
```

```
if правильна_відповідь = «А»:
```

```
change score by 1
```

```
broadcast «showCorrectFeedback»
```

```
else:
```

```
broadcast «showWrongFeedback»
```

5.2. Прогрес-індикатор

Навіщо: знизити невизначеність учня («скільки ще лишилось?»).
Реалізація - окремий спрайт-смужка або текстовий індикатор «Питання 2 з 5», що оновлюється через broadcast при переході між запитаннями.

6. Диференціація у Scratch-вправі

Робоча програма дисципліни прямо вимагає, щоб прототип вправи передбачав «налаштування рівня складності та диференціації». Диференціація - це адаптація вправи до різних можливостей учнів так, щоб кожен працював у своїй зоні найближчого розвитку (Виготський).

6.1. Три рівні диференціації у Scratch

Таблиця 11.3 – Три рівні диференціації у Scratch

Рівень	Що варіюється	Реалізація у Scratch
Рівень 1: за змістом (складність питань)	Кількість і складність запитань	Стартовий екран із вибором: «Базовий рівень - 5 запитань» / «Поглиблений - 8 запитань». Змінна difficulty керує, який список запитань зчитувати
Рівень 2: за підтримкою (адаптивні підказки)	Наявність і обсяг підказки	Кнопка «Підказка» з'являється тільки на «Базовому» рівні. На «Поглибленому» - підказки немає або вона прихована за рахунок балів
Рівень 3: за темпом	Часові обмеження	На «Базовому» рівні - без таймера; на «Поглибленому» - таймер 30 секунд на запитання

6.2. Адаптивна підказка (за Виготським)

Більш просунутий механізм: підказка з'являється не за натиском кнопки, а автоматично, коли учень помилився більше N разів поспіль. Це моделює дію вчителя, який втручається лише тоді, коли бачить системну проблему.

```
when I receive «wrongAnswer»:
  change wrongStreak by 1
  if wrongStreak >= 2 and hintsEnabled = 1:
    show sprite «Підказка»
```

```
set wrongStreak = 0
```

7. Тестування і запуск

Перед демонстрацією вправи учням обов'язковою є фаза тестування - систематична перевірка коректності всіх можливих сценаріїв проходження. Це не «один раз пройду, як учень би пройшов», а структурований чек-лист тест-кейсів.

Тест-кейси охоплюють чотири категорії: позитивні (правильні відповіді - бали нараховуються, фідбек коректний); негативні (неправильні відповіді - бали не нараховуються, фідбек пояснює помилку); граничні (учень нічого не натискає, натискає двічі, закриває проєкт посередині); диференціаційні (обидва рівні складності проходяться до кінця, обмеження працюють). Типові помилки цього етапу - подвійне нарахування балів через нескинутий `answerLocked` і показ попереднього запитання через неочищену змінну `currentQuestion`.

8. Експорт і шеринг проєкту

Готовий прототип потрібно зробити доступним для учнів. Scratch пропонує три способи:

8.1. Публікація на scratch.mit.edu (основний шлях)

- Зайти у власний акаунт на scratch.mit.edu (потрібна електронна пошта вчителя; для учнів - обліковий запис класу через Scratch Teacher Account).
- Натиснути кнопку Share у верхньому правому куті редактора.
- Заповнити поля Instructions (інструкція учневі) і Notes and Credits (методичні нотатки, джерела).
- Скопіювати посилання вигляду <https://scratch.mit.edu/projects/123456789>.
- Розповсюдити посилання через Google Classroom, чат класу або QR-код.

Завдання для виконання роботи

1. Сформулюйте SMART-ціль вправи з числовим порогом успіху та складіть таблицю узгодження «механіка вправи → очікуваний результат навчання» (мінімум 3 рядки).
2. Спроектуйте архітектуру проєкту до відкриття редактора: перелік спрайтів із функціями, змінні (score, currentQuestion, answerLocked, difficulty), broadcast-повідомлення та схема переходів між фонами сцени.
3. Реалізуйте прототип у Scratch: мінімум 5 запитань або ігрових завдань, підрахунок балів із захистом від подвійного нарахування, elaborative-фідбек до кожного запитання, два рівні складності (базовий із підказками / поглиблений з таймером).
4. Протестуйте вправу за чотирма категоріями тест-кейсів (позитивні, негативні, граничні, диференціації), опублікуйте проєкт (Share) із заповненими полями Instructions та Notes і додайте посилання у звіт.

Контрольні питання для захисту роботи

1. Що таке навчальна вправа у контексті середовища Scratch?
2. У чому різниця між ігровою активністю та навчальним завданням?
3. Як сформулювати чітку навчальну мету вправи (за принципом SMART)?
4. Які типи завдань найкраще реалізуються у Scratch (класифікація, обчислення, реакція, послідовності, логіка)?
5. Як поєднати сюжет, логіку та оцінювання в одній вправі?

Практична робота № 12

Тема: Створення інтерактивного оцінювання з елементами кодування.

Мета: навчитися застосовувати елементи кодування (налаштування скриптів, параметрів, інтеграцій, умовних логік) для вдосконалення тестових і аналітичних інструментів.

Теми 1–11 охоплювали проєктування і реалізацію окремих елементів ППЗ: вправ, тестів, навігації, фідбеку. Тема 12 виходить на наступний рівень - **системну інтеграцію**. Поодинокий тест або Scratch-вправа, що стоять окремо, - це інструменти. Але Google Forms, що автоматично надсилає результати у структуровану таблицю, яка живить живий дашборд для вчителя, що миттєво реагує на помилки класу - це вже **система оцінювання**. Різниця між ними така ж, як між цеглою і будинком.

Елементи кодування у контексті no-code ППЗ - це не написання повноцінних програм. Це **точково застосовані скрипти**, що автоматизують те, що неможливо налаштувати через інтерфейс: автоматичне відправлення email з результатами, динамічне наповнення тесту з банку питань у Sheets, умовне форматування дашборду, що сигналізує про відставання учня. Google Apps Script - основний інструмент цього рівня - є JavaScript-середовищем, вбудованим у Google Workspace, і не потребує жодних додаткових інсталяцій.

Ця тема важлива не лише технічно. Вона формує у майбутнього вчителя інформатики системне мислення: здатність бачити ППЗ не як набір ізольованих компонентів, а як **інтегровану систему збору, обробки і представлення навчальних даних**. Саме цього бракує більшості шкільних впроваджень EdTech сьогодні.

1. Концептуальні засади: інтеграція як принцип проєктування

1.1. Архітектура системи «Forms → Sheets → Dashboard»

Система трьох компонентів є найпоширенішим і найдоступнішим шаблоном інтеграції для шкільного оцінювання без жодних сторонніх платформ - лише Google Workspace.

- **Google Forms (збір):** учень відповідає на питання → Forms автоматично записує кожну відповідь у прив'язану Google Sheets таблицю. Кожен

рядок = одна відправка форми. Столпці: Timestamp, Email, відповідь на П1, відповідь на П2... Це рівень 1 інтеграції.

- **Google Sheets (обробка):** другий аркуш «Аналітика» містить формули, що обробляють сирі дані: AVERAGEIF для середнього балу по учню, COUNTIF для частоти помилок по питанню, IFS для класифікації рівня учня. Оновлюється автоматично при кожній новій відповіді. Рівень 2–3.
- **Dashboard (представлення):** третій аркуш «Дашборд» або окремий Google Slides, що містить графіки та таблиці, прив'язані до аналітики. Вчитель бачить візуальну картину класу без перегляду сирих рядків. Рівень 3.
- **Apps Script (автоматизація):** додатковий шар - скрипт, що запускається при подачі нової форми і виконує дії: надсилає email учню, оновлює зведену таблицю, генерує підсумковий звіт. Рівень 4.

2. Google Sheets як аналітичний двигун оцінювання

2.1. Структура таблиці відповідей Forms

Розуміння структури автоматично генерованої таблиці є передумовою будь-якого аналізу. Кожна відправка форми додає новий рядок. Перший стовпець завжди - Timestamp (час відправки). Якщо увімкнено збір email - другий стовпець - Email Address. Далі - по одному стовпцю на кожне питання форми у порядку їх розміщення у формі.

Таблиця 12.1 – Структура автоматично генерованої таблиці результатів

Стовпець	Назва (автоматична)	Тип даних	Призначення в аналізі
A	Timestamp	DateTime (рядок)	Час відправки; для аналізу часового патерну, виявлення «нічних» відправок
B	Email Address	String	Ідентифікатор учня; JOIN з журналом класу
C	Текст питання 1	String (відповідь учня)	Порівняння з еталонною

Стовпець	Назва (автоматична)	Тип даних	Призначення в аналізі
D	Текст питання 2	String	відповіддю; COUNTIF для частоти варіантів
...
Авто	Score (якщо тест)	Number	Автоматичний бал Forms; база для AVERAGEIF, PERCENTILE

2.2. Ключові аналітичні формули для оцінювання

Наступні формули є «швидким стартом» для будь-якого аналітичного аркуша. Всі вони посилаються на таблицю відповідей Forms.

2.2.1. Середній бал по класу і по учню

// Середній бал всього класу (стовпець Score = J):
=AVERAGE(Responses!J2:J)

// Середній бал конкретного учня (email у D1):
=AVERAGEIF(Responses!B:B, D1, Responses!J:J)

// Медіана класу (більш стійка до «відмінників»):
=MEDIAN(Responses!J2:J)

2.2.2. Частота помилок по питанню

Цей аналіз відповідає на запитання: «Яке питання виявилось найважчим?» - і є прямою підставою для педагогічного рішення «що повторити завтра».

// Кількість правильних відповідей на питання 1 (правильна = «Варіант А»): =COUNTIF(Responses!C:C, "Варіант А")

// Відсоток правильних: =COUNTIF(Responses!C:C, "Варіант А") / COUNTA(Responses!C2:C)

// Відсоток форматований: =ТЕХТ(COUNTIF(Responses!C:C, "Варіант А") / COUNTA(Responses!C2:C), "0%")

// Найпоширеніша відповідь (для виявлення misconception):
=INDEX(Responses!C2:C, MATCH(MAX(COUNTIF(Responses!C2:C, Responses!C2:C)), COUNTIF(Responses!C2:C, Responses!C2:C), 0))

2.2.3. Класифікація рівнів досягнення

// IFS для визначення рівня досягнення за балом у комірці J2: =IFS(J2>=90, “Високий”, J2>=75, “Достатній”, J2>=60, “Середній”, J2<60, “Початковий”)

// VLOOKUP для отримання імені учня за email: =VLOOKUP(B2, ClassList!A:C, 2, FALSE) // де ClassList!A = emails, ClassList!B = прізвища

2.2.4. Умовне форматування для візуального виявлення проблем

Умовне форматування (Format → Conditional Formatting) є no-code способом створення «теплові карти» результатів - без жодних формул:5

- Правило 1: якщо значення < 60 → фон червоний (учень потребує уваги).
- Правило 2: якщо значення між 60 і 75 → фон жовтий (є труднощі).
- Правило 3: якщо значення ≥ 75 → фон зелений (навчальна ціль досягнута).
- Правило 4: кольорова шкала (Color Scale) для стовпця балів - автоматичний градієнт від червоного (мін.) до зеленого (макс.).

3. Google Apps Script: автоматизація оцінювання

Google Apps Script - вбудована в Google Workspace мова автоматизації на основі JavaScript: скрипти створюються безпосередньо з Google Sheets (меню «Розширення → Apps Script»), не потребують окремого сервера і мають прямий доступ до Forms, Sheets, Gmail та Календаря. Для вчителя це означає можливість автоматизувати перевірку, розсилання результатів і формування звітів без розгортання власної інфраструктури.

3.1. Базова структура скрипту для оцінювання

Розглянемо найбільш корисний сценарій: скрипт, що запускається при відправці Google Forms і автоматично надсилає учню email з результатами.

// Скрипт запускається при кожній відправці форми function onFormSubmit(e) { // 1. Отримати відповіді учня з об'єкту події var responses = e.values; // масив відповідей у порядку питань var email = responses[1]; //

```

стовець B = email (індекс 1) var score = responses[responses.length - 1]; //
останній = Score var timestamp = responses[0]; // стовець A = Timestamp
    // 2. Визначити рівень досягнення var level = getLevel(parseFloat(score));
    // 3. Сформувати тіло email var subject = "Результати тесту з інформатики";
var body = "Привіт!" + "Твій результат:" + score + " балів" + "Рівень
досягнення:" + level + "" + "Дата:" + timestamp + "" + "Успіхів у навчанні!";
    // 4. Надіслати email GmailApp.sendEmail(email, subject, body);
    // 5. Записати результат у журнал (необов'язково) logResult(email, score,
level, timestamp); }
    // Допоміжна функція: визначення рівня function getLevel(score) { if (score
>= 90) return "Високий"; if (score >= 75) return "Достатній"; if (score >= 60)
return "Середній"; return "Початковий"; }

```

3.2. Налаштування тригерів у Apps Script

Тригер визначає, коли запускається скрипт. Для шкільного ППЗ найважливіші: onFormSubmit - спрацьовує при кожній надісланій відповіді форми (автоматична розсилка результатів, запис у журнал); time-driven - за розкладом (щотижневий зведений звіт, нагадування про незавершені завдання); onOpen - при відкритті таблиці (додавання власного меню «Аналітика»). Тригери налаштовуються в редакторі Apps Script через меню Triggers → Add Trigger, де обирають функцію, джерело події та її тип.

4. Побудова аналітичного дашборду

4.1. Ключові метрики шкільного оцінювання

Набір показників дашборду має відображати педагогічні питання, які вчитель ставить щодня: 11

Таблиця 12.2 – Метрики навчального дашборду

Метрика	Педагогічне питання	Формула у Sheets	Де відобразити
Середній класу (%) бал	Чи засвоїв клас матеріал прийнятному рівні (≥75%)?	=AVERAGE(Responses!J2:J)	Велика цифра у верхній частині дашборду; ліворуч

Метрика	Педагогічне питання	Формула у Sheets	Де відобразити
Кількість учнів нижче порогу (<60%)	Скільки учнів потребують негайної допомоги?	=COUNTIF(Responses!J2:J,"<60")	Велика цифра з червоним фоном (якщо > 0)
Найважча тема / питання	Що потрібно пояснити ще раз?	=MINIFS(QuestionStats!B:B,QuestionStats!B:B,">0")	Гістограма «Відсоток правильних по питаннях»
Динаміка класу (остання - передостання)	Клас покращується чи погіршується?	=(AvgLastTest - AvgPrevTest) / AvgPrevTest	Стрілка вгору/вниз з відсотком зміни
Список учнів, що не здали	Хто не виконав завдання взагалі?	=FILTER(ClassList!A:A, COUNTIF(Emails!B:B, ClassList!A:A) = 0)	Таблиця «Не здано» у нижній частині

4.2. Покрокова побудова дашборду у Google Sheets

Крок 1. Підготовка структури аркушів:

- *Responses* - автоматично від Forms (не чіпати).
- *QuestionStats* - аркуш із формулами COUNTIF для кожного питання.
- *StudentStats* - аркуш з AVERAGEIF по кожному учню.
- *Dashboard* - аркуш лише з посиланнями на метрики і вбудованими діаграмами.

Крок 2. Побудова аркушу QuestionStats:

// Аркуш QuestionStats: // Стовець А: Питання (вручну або посилання на заголовки Forms) // Стовець В: Кількість правильних відповідей // Стовець С: Загальна кількість відповідей // Стовець D: Відсоток правильних

// Приклад для питання 1 (правильна відповідь = "Варіант А"): // B2: =COUNTIF(Responses!C:C,"Варіант А") // C2: =COUNTA(Responses!C2:C) // D2: =IFERROR(B2/C2, 0)

Крок 3. Побудова діаграм на аркуші Dashboard:

- Гістограма «Відсоток правильних по питаннях»: Insert Chart → Chart type: Bar Chart → Data range: QuestionStats!A:D → Series: стовпець D.
- Кругова діаграма «Розподіл рівнів»: підрахувати COUNTIF для кожного рівня → Pie Chart.
- Таблиця «Топ-3 проблемних питання»: SMALL() для виявлення мінімальних відсотків → посилання в Dashboard.

Крок 4. Автооновлення (якщо потрібно): За умовчанням Sheets перераховує формули при кожній зміні. Але діаграми оновлюються з затримкою. Для примусового оновлення - Apps Script з функцією *SpreadsheetApp.flush()* у тригері onFormSubmit.

5. Безпека та приватність даних при автоматизації оцінювання

5.1. GDPR і Apps Script: що треба знати

Автоматизація оцінювання через Apps Script підпадає під вимоги GDPR так само, як і будь-яка обробка персональних даних. Ключові питання при розробці скриптів:

- **Email учнів - персональні дані:** скрипт, що зберігає або передає email учнів, обробляє персональні дані. Школа як оператор даних несе відповідальність за законність обробки.
- **Мінімізація даних у скриптах:** скрипт має обробляти лише те, що потрібно для педагогічної мети. Зберігати email у зовнішніх сервісах або надсилати у сторонні API - заборонено без явної згоди.
- **Дані залишаються у Google Workspace:** якщо скрипт працює лише у межах Google Workspace For Education (Sheets, Forms, Gmail) - дані не виходять за межі DPA-захищеного середовища Google. Це безпечно.
- **Доступ до скрипту:** скрипт у спільній Sheets отримує доступ до всіх даних таблиці. Не надавати доступ до «аналітичних» Sheets учням - лише до їх власних результатів.

Завдання для виконання роботи

1. Зв'яжіть Google Forms із Google Sheets та побудуйте аркуш аналітики з мінімум п'ятьма формулами: середній бал класу (AVERAGE), кількість учнів нижче порогу (COUNTIF), середній бал конкретного учня

(AVERAGEIF), відсоток правильних відповідей на найважче питання, класифікація рівня досягнення (IFS).

2. Налаштуйте умовне форматування «теплової карти» результатів: червоний фон для балів < 60 , жовтий для $60-74$, зелений для ≥ 75 .
3. Напишіть Google Apps Script із функцією onFormSubmit, що автоматично надсилає учневі email із балом і рівнем досягнення; налаштуйте тригер «From form - On form submit» і перевірте спрацювання тестовою відправкою.
4. Побудуйте міні-дашборд на окремому аркуші: дві діаграми (гістограма «відсоток правильних по питаннях», кругова «розподіл рівнів») і три ключові метрики з педагогічною інтерпретацією.

Контрольні питання для захисту роботи

1. У чому різниця між користуванням готовою платформою і створенням власного рішення на її основі?
2. Як налаштувати інтеграцію між платформами (наприклад, Forms → Sheets → Dashboard)?
3. Як використовувати скрипти або API для спрощення аналізу відповідей студентів?
4. Які прості скрипти чи налаштування можна використовувати у Forms, H5P, Quizizz чи LMS?
5. Які готові інструменти ви могли б вдосконалити через власні скрипти або інтеграції?

Практична робота № 13

Тема: Створення банку завдань, критеріїв та автоматизованого наповнення.

Мета: сформувати структурований банк завдань, а також налаштувати автоматизоване наповнення тестів.

Після виконання практичної роботи студент зможе:

- проектувати структуру банку завдань з повним набором метаданих (item_id, topic, bloom_level, difficulty, item_type, correct, feedback);
- наповнювати банк мінімум 15 якісними завданнями різних рівнів Блума та типів;
- реалізувати автоматичний відбір питань з банку за темою, рівнем складності та типом через FILTER + RAND();
- розробити аналітичну рубрику (5 критеріїв × 4 рівні) для оцінювання Scratch-вправи або іншого складного завдання;
- аргументувати методичну доцільність банку і рубрики для шкільного ППЗ.

Компетентності, що формуються:

Таблиця 13.1 – Компетентності, що формуються у роботі

Компетентність	Реалізація у цій роботі
ЗК-1: Аналіз, синтез та застосування знань	Класифікація завдань за таксономією Блума; перевірка балансу рівнів у банку
ЗК-5: Навички ІКТ	Реалізація структури банку у Sheets; формули FILTER, RAND, INDEX для автовідбору
СК-13: Використання програмних засобів	Google Sheets як сховище банку; формульна логіка для генерації варіантів
ПРН-9: Цифрові технології у профдіяльності	Системна організація оцінювання через структурований банк і рубрику

Стислі теоретичні відомості

Мінімальний набір метаданих тестового завдання

Кожне завдання банку описується рядком метаданих: `item_id` - унікальний ідентифікатор (наприклад, INF9-ALG-001); `topic` - тема для фільтрації; `bloom_level` - рівень за Блумом (1–4) для збалансованого розподілу; `difficulty` - складність 0,0–1,0 (після апробації уточнюється за `p-value`); `item_type` - тип завдання (MC / TF / FIB / Match); `question_text`, `option_a–d`, `correct_answer` - зміст завдання і ключ для автоматичної перевірки; `feedback` - `elaborative` пояснення при помилці; `usage_count` - лічильник використань для ротації (питання з `usage_count > 5` тимчасово виключаються). Саме ці поля роблять банк придатним для автоматичної генерації варіантів.

Формула автоматичного відбору через FILTER + RAND()

// Крок 1: Допоміжний стовпець з випадковим числом (наприклад, стовпець L): // у L2: =IF(A2<>"", RAND(),"") → розтягнути на всі рядки банку

// Крок 2: Відбір 5 питань з теми «Алгоритми», рівень Блума 1–2: =FILTER(Items!A2:J50, (Items!B2:B50 = "Алгоритми") (Items!C2:C50 <= 2) (Items!L2:L50 <> "") // Потім відсортувати за стовпцем L (RAND) і взяти перші 5 рядків - INDEX/SMALL

Обмеження формульного підходу: Формула RAND() перераховується при кожній зміні таблиці, що змінює вибірку. Після генерації варіанту - скопіюйте результат через Paste Special → Paste values only, щоб зафіксувати варіант.

Наскрізний кейс практичної роботи

Ситуація: Методичне об'єднання вчителів інформатики Луцького ліцею (4 вчителі, 8 паралельних класів) вирішує перейти від «папок з питаннями» до спільного банку завдань. Завдання виконавця - Марії Тимофіївни: (1) створити структуру банку у Google Sheets для теми «Алгоритмізація та програмування», 9 клас; (2) наповнити мінімум 15 якісними завданнями трьох рівнів Блума; (3) реалізувати автоматичну генерацію варіанту з 10 питань через формули; (4) розробити рубрику для оцінювання Scratch-вправи (практичної роботи № 11), щоб різні вчителі виставляли однакові оцінки.

Всі 4 завдання виконуються від імені Марії Тимофіївни як послідовний проєктний цикл.

Завдання 1. Проєктування та наповнення структури банку завдань

(Репродуктивне - 2 бали)

Ваше завдання: створити структуровану таблицю банку у Google Sheets і наповнити її 15 якісними завданнями для теми «Алгоритмізація».

1.1. Створення структури таблиці

- Відкрийте Google Sheets → нова таблиця → назва: «Банк_Алгоритмізація_9клас_[ПрізвищеАвтора]».
- Перший аркуш назвіть «Items» (банк завдань).
- Створіть заголовки у рядку 1 у такому порядку (по стовпцях А–J):

Таблиця 13.2 – Структура заголовків банку завдань

Стовп.	Заголовок	Стовп.	Заголовок
A	item_id	F	question_text
B	topic	G	option_a
C	bloom_level	H	option_b
D	difficulty	I	option_c
E	item_type	J	correct_answer
K	feedback	L	usage_count

- Зафіксуйте перший рядок: View → Freeze → 1 row.
- Додайте аркуш «Topics» - довідник тем: стовпці topic_id | topic_name | min_items_required. Заповніть 3–4 підтеми теми «Алгоритмізація» (наприклад: Цикли, Умовні оператори, Масиви, Функції).

1.2. Наповнення банку завданнями

Заповніть банк 15 завданнями з теми «Алгоритми і програми» (9 клас): 6 завдань рівнів «початковий/середній», 6 - «достатній», 3 - «високий»; типи завдань - одиничний вибір, множинний вибір, відповідність, коротка відповідь. Кожне завдання опишіть повним рядком метаданих за структурою з п. 1.1. Зразок трьох рядків банку:

Таблиця 13.3 – Банк завдань з теми «Алгоритми і програми»

ID	Тип	Рівень	Формулювання (скорочено)	Відповідь
A-01	одиничний вибір	середній	Що таке лінійний алгоритм?	послідовне виконання команд
A-07	відповідність	достатній	Установіть відповідність між алгоритмічними структурами та блок-схемами	1–Б, 2–А, 3–В
A-13	коротка відповідь	високий	Запишіть результат виконання поданого фрагмента алгоритму	24

1.3. Перевірка якості завдань

Перевірте кожне завдання за базовими правилами item writing: формулювання однозначне і не містить подвійного заперечення; дистрактори правдоподібні та однорідні; правильна відповідь не вирізняється довжиною чи деталізацією; завдання перевіряє саме той результат, що зазначений у метаданих; рівень складності відповідає маркуванню. Завдання, що не пройшли перевірку, відредагуйте і зафіксуйте зміни у звіті.

1.4. Аналіз балансу банку

Побудуйте зведену таблицю розподілу завдань за рівнями складності й типами (меню «Вставити → Зведена таблиця»). Переконайтеся, що фактичний розподіл відповідає запланованому (6/6/3) і що кожен тип завдань представлений принаймні двома позиціями; за потреби доповніть банк.

Завдання 2. Автоматизоване наповнення варіанту тесту

(Аналітичне - 3 бали)

Банк наповнено. Тепер потрібно реалізувати автоматичний відбір 10 питань для варіанту тесту. **Ваше завдання:** побудувати аркуш «TestGenerator» з формулами, що генерують збалансований варіант, і перевірити повторну генерацію.

2.1. Специфікація тесту - визначте параметри варіанту

Заповніть специфікацію до побудови генератора:

Таблиця 13.4 – Специфікація генератора завдань

Параметр специфікації	Ваше рішення	Обґрунтування (1–2 речення)
Загальна кількість питань у варіанті	10	Оптимально для 20-хвилинного тесту
Розподіл за рівнями Блума (кількість питань)		
Цільовий діапазон складності (difficulty)		
Розподіл за типами (MC / TF / FIB)		
Чи обмежувати за темою? Якщо так - якою?		
Що робити з питаннями з usage_count > 5?		

2.2. Реалізація генератора на аркуші TestGenerator

- Додайте аркуш «TestGenerator».
- **Крок 1 - допоміжний стовець RAND у банку:** на аркуші Items у стовпці L (після feedback) введіть у L2: =IF(A2<>“”,RAND(),””). Розтягніть на всі рядки банку.
- **Крок 2 - фільтрація пулу питань:** на аркуші TestGenerator у A1 введіть заголовок «Пул питань (відфільтровано)». У A2 введіть формулу FILTER відповідно до вашої специфікації 2.1. Приклад для рівнів 1–2, теми «Цикли»:
 =FILTER(Items!A2:K50, (Items!B2:B50 = “Цикли”) (Items!C2:C50 <= 2) (Items!D2:D50 >= 0.35) * (Items!L2:L50 <> “”))
- **Крок 3 - сортування за RAND і відбір 10 питань:** використайте SORT для впорядкування пулу за стовпцем L (RAND), потім INDEX/ARRAY для відбору перших 10 рядків. Або - простіший варіант: оновлюйте RAND кнопкою F9 і копіюйте перші 10 рядків вручну.
- **Крок 4 - таблиця варіанту:** на аркуші «Variant_1» збережіть результат відбору через Paste Special → Paste values only. Це зафіксує варіант.

2.3. Перевірка генератора

Згенеруйте три варіанти тесту поспіль (перерахунок RAND - зміна будь-якої клітинки або повторне відкриття файлу). Переконайтеся, що варіанти не повторюються повністю, відповідають специфікації з п. 2.1 і не містять двох однакових завдань в одному варіанті.

2.4. Журнал варіантів - запобігання повторам

Додайте аркуш «TestLog» - журнал згенерованих варіантів. Після кожної генерації записуйте:1

// Аркуш TestLog - стовпці: // A: test_id B: date C: class D: item_ids (список через кому)

// Приклад рядка: // T-2025-01 | 2025-10-15 | 9-A | INF9-ALG-001,INF9-ALG-007,INF9-ALG-012,...

Це дозволяє при наступній генерації виключати вже використані item_id. Для практичної роботи - достатньо заповнити 2–3 записи журналу вручну (реальний банк з 15 питань ще не вимагає автоматичного виключення).

Завдання для звіту:

- Аркуш TestGenerator з формулою FILTER (скріншот №3 або посилання).
- Аркуш Variant_1 з зафіксованими 10 питаннями (скріншот №4).
- Заповнена таблиця перевірки (2.3) - без X.
- Аркуш TestLog з 2 записами (скріншот №5).
- Заповнена специфікація тесту (2.1) - всі рядки з обґрунтуваннями.
- Відповідь (2–3 речення): *«Чому RAND() у допоміжному стовпці є кращим підходом, ніж =RAND() безпосередньо у формулі FILTER?»*

Завдання 3. Аналітична рубрика для оцінювання Scratch-вправи

(Проектувальне - 3 бали)

Банк - для автоматизованого тестування. Рубрика - для складних завдань без єдиної правильної відповіді. **Ваше завдання:** розробити аналітичну рубрику для оцінювання Scratch-вправи (продукт з практичної роботи № 11) у Google Sheets.

3.1. Розробка повної рубрики (матриця 5×4)

Заповніть кожен клітинку матриці конкретним, спостережуваним описом. Уникайте загальних слів («добре», «задовільно»). **Конкретний опис:** «elaborative фідбек є для всіх 5 питань і пояснює причину помилки». **Неконкретний:** «фідбек є і непоганий».

Критерій 1: Відповідність навчальній меті

Таблиця 13.5 – Детскриптори критерію «Відповідність навчальній меті»

Рівень	Бал	Опис (конкретний, спостережуваний)
Початковий	1	
Середній	2	
Достатній	3	SMART-ціль сформульована; більшість завдань вправи відповідають їй; є посилання на тему уроку
Високий	4	

Критерії 2–5 розробіть за зразком критерію 1: для кожного сформулюйте дескриптори чотирьох рівнів (відмінно - 4 бали, добре - 3, задовільно - 2, незадовільно - 1). Орієнтири для дескрипторів наведено в таблиці.

Таблиця 13.6 – Критерії оцінювання та їх ваги

Критерій	Вага	Що описують дескриптори
2. Технічна коректність	0,25	відсутність логічних помилок, коректний підрахунок балів, захист від подвійного нарахування
3. Якість elaborative фідбеку	0,20	пояснення причини помилки, навідні підказки, доброзичливий тон (принцип 4E)
4. Навігація та UX	0,15	зрозумілий маршрут «старт → вправа → результат», помітні кнопки, інструкція для учня
5. Оформлення та дизайн	0,10	візуальна ієрархія, читабельність, відсутність перевантаження екрана

3.2. Формула розрахунку зваженого балу в Sheets

Реалізуйте рубрику у Google Sheets на аркуші «Rubric». Стовпці: Критерій | Вага | Оцінка (1–4) | Зважений бал:

// Аркуш Rubric - структура: // А: Критерій В: Вага С: Оцінка (вводиться вчителем) D: Зважений бал

// D2 (Зважений бал для критерію 1): =B2 * C2

// D7 (Загальна зважена сума): =SUM(D2:D6)

// D8 (Максимально можливий бал): =SUMPRODUCT(B2:B6, {4;4;4;4;4})

// або: =SUM(B2:B6) * 4

// D9 (Відсоток досягнення): =D7/D8

// D10 (Оцінка за 12-бальною шкалою): =ROUND(D9 * 12, 0)

3.3. Апробація рубрики

Оцініть за розробленою рубрикою Scratch-вправу, створену у практичній роботі № 11 (власну або однокласника): визначте рівень за кожним критерієм, обчисліть зважений бал за формулою з п. 3.2 і сформулюйте 2–3 рекомендації щодо вдосконалення вправи.

Завдання 4. Методична аргументація та тест якості банку

(Рефлексивне - 2 бали)

Ваше завдання: підготувати методичне обґрунтування банку завдань і рубрики для методичного об'єднання, та провести аналіз якості завдань банку.

4.1. Аналіз якості завдань після тестування

Після проведення тесту обчисліть для кожного завдання індекс складності p - частку учасників, які відповіли правильно. Завдання з $p > 0,9$ - занадто легкі, з $p < 0,2$ - занадто складні або некоректні: позначте їх у банку для перегляду і сформулюйте висновок про якість банку загалом.

4.2. Методичне обґрунтування для методичного об'єднання

Підготуйте для методичного об'єднання обґрунтування з п'яти пунктів (по 2–3 конкретних речення, без загальних фраз): 1) яку проблему об'єднання вирішує банк завдань; 2) чому обрано саме ці поля метаданих і які з них обов'язкові; 3) навіщо класифікація за Блумом, а не лише рівень Remember; 4) у чому педагогічна цінність автоматичної рандомізованої генерації варіантів (з

посиланням на ст. 42 Закону України «Про освіту»); 5) як рубрика забезпечує консистентність оцінювання між учителями.

Питання для обговорення та захисту роботи

Обирається по одному питанню з кожної групи. Банк і рубрика демонструються в Sheets.

Група А: Знання понять

- Що таке банк завдань? Назвіть 5 ключових відмінностей від «папки з питаннями».
- Що таке p-value тестового завдання і що він означає для педагогічного рішення? Наведіть приклад для значень 0.1, 0.5 і 0.9.
- Поясніть різницю між голістичною і аналітичною рубрикою. Коли доцільна кожна?

Група Б: Аналіз та порівняння

- У вашому банку 15 завдань, 10 з яких - рівень Remember. Чому це проблема? Як перебудувати розподіл?
- Порівняйте формульний підхід (FILTER + RAND) і Apps Script для генерації варіантів. Яка різниця у стабільності, складності та сценаріях застосування?
- Чому рубрика з вагами ($\times 3$, $\times 2$, $\times 1$) є точнішим інструментом оцінювання, ніж рубрика, де всі критерії рівнозначні?

Група В: Демонстрація та застосування

- Покажіть у Sheets: (а) банк з 15 завданнями, (б) формулу FILTER з вашою специфікацією, (в) зафіксований варіант Variant_1.
- Оцініть вашу власну Scratch-вправу (практичної роботи № 11) за рубрикою наживо. Поясніть кожну оцінку.
- Марія Тимофіївна хоче поділитися банком з трьома колегами. Як правильно налаштувати доступ, щоб вони могли додавати нові питання, але не видаляти чужі?

Практична робота № 14

Тема: Методична аргументація та демонстрація ППЗ.

Мета: навчитися коротко й переконливо презентувати свій ППЗ: показати, що він робить, навіщо у шкільному уроці, як оцінюється результат і чому обрані підходи методично обґрунтовані.

Курс проєктування та розробки ППЗ завершується темою, яка повертає нас до питання, з якого все починалось: «**Навіщо?**» Навіщо проєктувати User Flow замість того, щоб одразу відкрити редактор? Навіщо писати elaborative фідбек, а не просто «правильно / неправильно»? Навіщо будувати банк завдань, якщо можна скопіювати з інтернету? Тема 14 дає на ці питання** методичну відповідь** - систематичне, аргументоване обґрунтування педагогічних рішень, покладених в основу розробленого ППЗ.

«Показати що» (демонстрація) і «пояснити чому» (аргументація) - це два різних рівні педагогічної зрілості. Більшість вчителів, що впроваджують EdTech, зупиняються на першому: «подивіться, як це працює». Педагог-проєктувальник йде далі: «ось чому ця навігаційна архітектура обрана саме такою, ось яка педагогічна теорія стоїть за цим фідбеком, і ось яких змін у навчальних результатах ми очікуємо». Ця здатність є принципово важливою для трьох ситуацій: захист КППЗ перед комісією, впровадження ППЗ у школі (переконати адміністрацію і батьків), та рецензування чужих ППЗ (критичний аналіз, а не просто «подобається / не подобається»).

Методична аргументація - це мова, якою розробник ППЗ говорить з педагогічною спільнотою.

1. Методична аргументація: що це і чому вона необхідна

Методична аргументація - це обґрунтування педагогічної доцільності ППЗ через ланцюжок «освітня потреба → навчальна мета → обраний засіб → спосіб оцінювання → очікуваний результат». Вона відповідає не на питання «що вміє програма?», а на питання «чому саме цей засіб найкраще розв'язує конкретну навчальну задачу?».

Аргументація адресується різним аудиторіям - адміністрації закладу, методичному об'єднанню, батькам, атестаційній комісії - і для кожної з них

акценти зміщуються: для адміністрації важать ресурси та безпека даних, для методистів - відповідність навчальній програмі й результатам навчання, для батьків - користь і навантаження для дитини.

2. Структура методичного обґрунтування ППЗ

2.1. Стандартна структура методичної записки до ППЗ

Методична записка (Methodical Note) - це формальний документ, що супроводжує ППЗ і містить повне методичне обґрунтування. Для КПЗ або портфолію вона є обов'язковою. Структура відповідає логіці «від контексту до рефлексії».

Таблиця 14.1 – Структура методичної записки

Розділ методичної записки	Що включає	Орієнтовний обсяг
1. Назва та загальна інформація	Повна назва ППЗ; предмет; клас; тема навчальної програми; автор; дата	3–5 рядків
2. Навчальна мета та прогалина	Яку педагогічну проблему або прогалину вирішує ППЗ; SMART-ціль; рівень за таксономією Блума	5–8 речень
3. Цільова аудиторія	Вік, клас, рівень підготовки; особливості (ООП); технічні умови класу (пристрої, інтернет, LMS)	3–5 речень
4. Тип ППЗ та педагогічна модель	Тип активності (тренажер, тест, симуляція); дидактична модель (blended, flipped, PBL); місце в уроці	5–7 речень
5. Опис навчального контенту	Теми, питання, типи завдань; рівні Блума у розподілі; кількість питань; якісні характеристики	Таблиця + 3–5 речень
6. Навігаційна архітектура	Тип навігації (лінійна / розгалужена); User Flow (схема або опис); умови переходів	Схема + 3–5 речень

Розділ методичної записки	Що включає	Орієнтовний обсяг
7. Зворотний зв'язок і підказки	Тип фідбеку (elaborative, adaptive); система підказок; педагогічне обґрунтування	5–8 речень з прикладами
8. Технологічний стек	Обрана платформа і обґрунтування; матриця вибору або ключові критерії	Таблиця + 3–5 речень
9. Безпека та приватність	Що збирається; DPA; відповідність GDPR; рольова модель доступу	5–8 речень
10. Результати тестування	Протокол тестування (технічне, педагогічне); виявлені проблеми; виправлення	Таблиця тестування + коментар
11. Дорожня карта розвитку	Версія 2.0; відомі обмеження; умови ефективного застосування	3–5 речень

2.2. Від чек листу до аргументації: три рівні обґрунтування

Розрізняють три рівні обґрунтування проєктних рішень. Рівень 1 - описовий: «для тесту використано Google Forms» - констатація без аргументів. Рівень 2 - порівняльний: «Forms обрано замість Quizizz, бо школа має Google Workspace» - є порівняння, але без теоретичної опори. Рівень 3 - теоретично обґрунтований: вибір спирається на матрицю критеріїв, педагогічний принцип, нормативну вимогу (GDPR, ст. 42 Закону України «Про освіту») і результат тестування. Більшість початківців зупиняються на першому рівні; мета курсу - впевнений третій.

3. Демонстрація ППЗ: структура та техніки

3.1. Структура демонстраційної презентації

Незалежно від формату (PowerPoint, Google Slides, жива демонстрація), структура ефективної презентації ППЗ є фіксованою.

Таблиця 14.2 – Структура презентації-демонстрації ППЗ

Частина презентації	Зміст та функція	Орієнтовний час	Типова помилка
1. Гачок (Hook)	Відкрити проблемою або запитанням: «Скільки часу ви витрачаєте на перевірку тестів щотижня?». Не з назви ППЗ.	1–2 хвилини	Починати зі слів «Я розробив...» - аудиторія ще не залучена
2. Контекст і проблема	Навчальна прогалина або педагогічна проблема, яку вирішує ППЗ. SMART-ціль.	2–3 хвилини	Перейти одразу до демонстрації без контексту - незрозуміло навіщо
3. Демонстрація (живе або відео)	Показати ППЗ у дії: пройти шлях учня від старту до фінального екрану. Коментувати ключові рішення.	5–8 хвилин	Показувати технічні деталі замість педагогічного досвіду учня
4. Методичне обґрунтування (highlight)	3–4 ключових рішення з обґрунтуванням. Не все - лише найбільш нетривіальне.	3–4 хвилини	Зачитувати всю методичну записку - аудиторія засинає
5. Результати та докази	Що показало тестування? Яка метрика успіху? Що учні сказали після перегляду?	2–3 хвилини	Говорити «я думаю, що це добре» замість даних
6. Дорожня карта і відкриті питання	Версія 2.0; чого не вистачає; запрошення до обговорення.	1–2 хвилини	Завершувати на «недоліках» без позитивного висновку
7. Запитання та відповіді	Відповідати впевнено; якщо не знаєш - «це важливе питання, я дослідю і повернусь». Не захищатись.	5–10 хвилин	Захищатись від критики замість приймати і відповідати на неї

4. Дорожня карта розвитку ППЗ

4.1. Від MVP до версії 1.0

Дорожня карта розвитку ППЗ - це план перетворення MVP на повноцінний засіб: версія 0.x (MVP) - одна активність, базовий фідбек, ручне оцінювання; версія 1.0 - повний цикл «інструкція → вправа → результат», банк завдань, автоматичне оцінювання; подальші версії - диференціація рівнів, аналітика результатів, інтеграція з LMS. Для кожного кроку фіксують очікуваний педагогічний ефект, потрібні ресурси та орієнтовний термін.

5. Підготовка до захисту КПЗ: практичні рекомендації

5.1. Структура захисту КПЗ з ППЗ

Комплексне практичне індивідуальне завдання у 6 семестрі є одним з видів контролю, де студент демонструє сформовані компетентності педагога-проектувальника. Формат захисту - публічна презентація з демонстрацією ППЗ.

Таблиця 14.3 – Структура комплексного практичного завдання (КПЗ)

Компонент КПЗ	Що містить	Критерій повноти
Пояснювальна записка	Методична записка (11 розділів), технічна документація, джерела	Мінімум 25 сторінок; усі 11 розділів заповнені
Проектна частина	User Flow, ER-схема (якщо є база даних), специфікація банку завдань, рубрика	Усі артефакти присутні і відповідають реалізації
Розробна частина	Готовий ППЗ з функціонуючим посиланням; Apps Script (якщо є); банк питань у Sheets	ППЗ доступний за посиланням; всі компоненти працюють
Аналітична частина	Протокол тестування; аналітичний дашборд; результати бета-тестування	Мінімум 2 типи тестування; є показники якості

Компонент КПЗ	Що містить	Критерій повноти
Презентація	10–15 слайдів; демонстрація живого ППЗ; методичне обґрунтування 3–4 рішень	20–25 хвилин; демонстрація наживо; впевнені відповіді на питання

5.2. Чек-лист готовності до захисту

Перед захистом переконайтеся, що: 1) ППЗ відкривається за посиланням з будь-якого пристрою; 2) демонстраційний сценарій відрепетирувано і він триває до 7 хвилин; 3) методична записка містить мету, цільову аудиторію, обрану дидактичну модель і спосіб оцінювання; 4) підготовлено відповіді на типові запитання про вибір платформи, безпеку даних і доброчесність; 5) є резервний варіант демонстрації (відеоскрінкаст) на випадок технічного збою; 6) сформульовано дорожню карту розвитку засобу.

Завдання для виконання роботи

1. Складіть методичну записку до власного ППЗ (розробленого впродовж практичних робіт № 9–13) за стандартною структурою з 11 розділів - стисло, по 2–4 речення на розділ.
2. Побудуйте матрицю узгодження «очікуваний результат навчання → рівень за Блумом → тип завдання у ППЗ → метрика вимірювання» (мінімум 3 рядки).
3. Підготуйте демонстраційну презентацію захисту (10–12 слайдів) за семичастинною структурою (гачок, контекст, демонстрація, методичне обґрунтування, результати, дорожня карта, запитання) із застосуванням принципу «покажи - поясни - зв'яжи».
4. Підготуйте письмові відповіді на три типові запитання аудиторії (вибір платформи, захист від списування, докази ефективності) за моделлю PREP (Position - Reason - Example - Position restate).

Контрольні питання для захисту роботи

1. Що означає «методична аргументація» у контексті презентації педагогічного програмного засобу?
2. Чому важливо не лише показати функціональність ППЗ, а й обґрунтувати його педагогічну доцільність?
3. Як ППЗ підтримує реалізацію навчальних цілей і розвиток компетентностей учнів?
4. Як продемонструвати зв'язок між ППЗ та програмними результатами навчання?

ГЛОСАРІЙ

Нижче наведено зведений глосарій ключових термінів, що вживаються у практичних роботах.

Таблиця 14.4 – Глосарій ключових термінів

Термін	Визначення
Академічна доброчесність	Сукупність етичних принципів і правил, що регулюють поведінку учасників освітнього процесу.
Академічна культура	Розуміння і прийняття учнями цінності власних результатів.
Асоціативна сутність	Сутність, що виникає при зв'язку M:N і зберігає додаткові атрибути зв'язку.
Атрибут (Attribute)	Властивість сутності, що зберігається. Зображається еліпсом або списком.
Банк питань	Колекція з N питань, з якої система автоматично обирає K для конкретного учня.
Бета-тестування (<i>Beta Testing</i>)	Тестування ППЗ з реальними або умовними учнями для перевірки педагогічної ефективності.
Гачок (<i>Hook</i>)	Відкриваючий елемент презентації - запитання, проблема або статистика, що одразу залучає аудиторію.
Демонстраційна презентація (<i>Demo Presentation</i>)	Структурований виступ, що поєднує живу демонстрацію ППЗ з методичним поясненням ключових рішень.
Диференціація (<i>Differentiation</i>)	Адаптація вправи до рівня учня (зміст, темп, підтримка)
Дорожня карта (<i>Roadmap</i>)	План розвитку ППЗ через версії (MVP → v1.0 → v2.0), де кожна версія додає нову педагогічну або технічну цінність.
Зв'язок (<i>Relationship</i>)	Асоціація між двома або більше сутностями.
Змінна (<i>Variable</i>)	Іменована комірка пам'яті проєкту
Зовнішній ключ (<i>FK</i>)	Атрибут сутності, що посилається на первинний ключ іншої сутності.
Кардинальність	Кількісна характеристика зв'язку: 1:1, 1:N, M:N.
Конструктивна критика ППЗ	Аналіз сильних і слабких сторін ППЗ з конкретними пропозиціями щодо покращення, що базується на педагогічних принципах, а не на особистих уподобаннях.
Конструктивне узгодження (<i>Constructive Alignment</i>)	Принцип Biggs (1996): ефективне навчання виникає, коли навчальна мета, зміст, метод навчання і метод оцінювання узгоджені між собою.
Конструкціонізм (<i>Constructionism</i>)	Педагогічна теорія Сеймура Пейперта: найефективніше навчання відбувається, коли учень створює публічно значимий артефакт.
Костюм (<i>Costume</i>)	Альтернативний візуальний стан спрайта

Матриця узгодження (<i>Alignment Matrix</i>)	Таблиця, що візуально демонструє відповідність між ОРН навчальної програми і компонентами ППЗ (тип завдання, рівень Блума, метрика вимірювання).
Методична аргументація (<i>Pedagogical Argumentation</i>)	Систематичний виклад обґрунтування педагогічних рішень, прийнятих при проектуванні ППЗ, з посиланням на навчальні цілі, педагогічні теорії, нормативні документи та результати тестування.
Методична записка (<i>Methodical Note</i>)	Формальний документ, що супроводжує ППЗ і містить повне педагогічне обґрунтування у стандартизованій структурі (11 розділів).
ОРН (<i>Learning Outcome</i>)	Освітній результат навчання шкільної програми
Оффлайн-режим (<i>Offline Mode</i>)	Можливість використовувати інструмент або навчальний контент без підключення до інтернету.
Педагогічне висловлювання (<i>Pedagogical Statement</i>)	Метафора для розуміння ППЗ: кожне проєктне рішення є аргументом щодо того, як має відбуватись навчання; ППЗ є матеріалізацією педагогічних переконань розробника.
Педагог-користувач vs педагог-проектувальник	Два різних рівні компетентності у роботі з ППЗ.
Педагог-проектувальник (<i>Learning Designer</i>)	Вчитель, що здатний проектувати нові ППЗ з нуля: від аналізу прогалин і SMART-цілі до повного технологічного стеку з методичним обґрунтуванням.
Подія (Event)	Точка запуску скрипта (натискання, отримання сигналу)
Предметна область	Частина реального світу, що моделюється системою.
Принцип PREP	Структура відповіді на складне запитання: Position (позиція) → Reason (причина) → Example (приклад) → Position restate (повтор позиції).
Прототип / MVP (<i>Prototype / Minimum Viable Product</i>)	Мінімально життєздатна версія вправи з усіма ключовими механіками
Рандомізація варіантів	Налаштування, що переставляє варіанти відповідей А/Б/В у випадковому порядку.
Рандомізація питань	Технічне налаштування тесту, що виводить питання у випадковому порядку для кожного учня.
Словник даних	Таблиця-специфікація всіх атрибутів кожної сутності: назва поля, тип, обмеження, призначення.
Список (<i>List</i>)	Впорядкований набір значень (банк запитань, відповідей)
Спрайт (<i>Sprite</i>)	У Scratch: інтерактивний персонаж або об'єкт з власними костюмами, звуками і скриптами (набором блоків коду).
Сутність (<i>Entity</i>)	Клас реальних або концептуальних об'єктів, про які система зберігає інформацію.
Сцена (<i>Stage</i>)	Фон проєкту; зміна сцени відбиває зміну контексту
Таймер-лічильник	Показує витрачений час без блокування - учень бачить скільки витратив, але тест не завершується примусово.

Тригер (<i>Trigger</i>)	Механізм у Apps Script, що визначає умову запуску скрипту: при відправці форми (onFormSubmit), за часом (timeBased), при відкритті документу (onOpen).
Умови ефективного використання	Педагогічний і технічний контекст, в якому ППЗ дає максимальний навчальний результат.
Умовне форматування (<i>Conditional Formatting</i>)	No-code інструмент Google Sheets: автоматична зміна кольору комірки залежно від значення.
AVERAGEIF / COUNTIF	Функції Google Sheets для умовного обчислення середнього значення або підрахунку комірок, що відповідають заданому критерію.
Branching Scenario (<i>Розгалужений сценарій</i>)	Контент-тип H5P для створення нелінійних навчальних сценаріїв, де кожна відповідь або вибір учня веде до іншого шляху розвитку сценарію.
Broadcast-повідомлення (<i>Broadcast Message</i>)	Внутрішній сигнал для координації між спрайтами
Content Type (<i>Контент-тип</i>)	У H5P: стандартизований модуль для конкретного типу інтерактивного контенту (Question Set, Drag & Drop, Fill in Blanks тощо).
Dashboard (<i>Дашборд</i>)	Єдиний екран з ключовими показниками навчального процесу.
Data-Informed Teaching	Підхід до викладання, що базується на систематичному аналізі даних про навчальний прогрес учнів для прийняття обґрунтованих педагогічних рішень.
Discriminator-блокування (<i>Answer Locking</i>)	Захист від повторного нарахування балів
DPA (<i>Data Processing Agreement</i>)	Угода між школою і постачальником платформи щодо умов обробки персональних даних учнів.
draw.io / diagrams.net	Безкоштовний онлайн-редактор схем для побудови ER-діаграм.
Elaborative фідбек	Зворотний зв'язок, що пояснює причину правильності або неправильності відповіді (не лише «правильно/неправильно»).
Elaborative-фідбек (<i>Elaborative Feedback</i>)	Зворотний зв'язок з поясненням причини помилки
Embed-код (<i>Embed Code</i>)	iframe-фрагмент для вставлення проєкту на сайт
Facility Index (<i>p-value</i>)	Частка учнів, що правильно відповіли на питання. $p > 0.7$ - легке; $p < 0.3$ - важке.
Flipped Classroom (<i>Перевернутий клас</i>)	Дидактична модель, де опанування нового матеріалу відбувається вдома (через відео або мікрокурс), а урок використовується для практики, задач та обговорення.
Gamification (<i>Гейміфікація</i>)	Застосування ігрових механік (очки, рейтинги, значки, рівні, змагання) у неігровому навчальному контексті для підвищення мотивації та залученості.
Genially	Онлайн-редактор для створення інтерактивних та анімованих візуальних матеріалів: презентацій, інфографік, ігрових дощок, escape room, мікрокурсів.

Google Apps Script (GAS)	JavaScript-середовище, вбудоване у Google Workspace, що дозволяє автоматизувати дії з Docs, Sheets, Forms, Gmail і Calendar.
Hotspot (Клікабельна зона)	Визначена область на зображенні, карті або схемі, що реагує на клік і відкриває додаткову інформацію.
H5P (HTML5 Package)	Відкритий (open-source) фреймворк і специфікація для створення переносного інтерактивного HTML5-контенту з понад 50 стандартизованими типами вправ.
Interactive Video (Інтерактивне відео)	Контент-тип H5P: відео з вбудованими в конкретні хронологічні точки питаннями, кнопками, поясненнями.
LearningApps.org	Безкоштовна веб-платформа з відкритою бібліотекою ігрових вправ.
Low-code підхід	Підхід до розробки, де використовуються мінімальні елементи кодування (скрипти, параметри API) поверх готових no-code платформ для розширення їх можливостей.
MVP (Minimum Viable Product)	Мінімально життєздатний продукт у контексті ППЗ: найпростіша версія навчальної активності, що реалізує основну педагогічну функцію та містить три обов'язкові компоненти: інструкцію, взаємодію та фідбек.
Pivot Table (Зведена таблиця)	Інтерактивний інструмент Google Sheets для агрегації та групування великих масивів даних без формул.
Privacy by Design	Принцип захисту приватності через проектування: збирати лише необхідні дані, зберігати лише необхідний час.
.sb3-файл (Scratch Project File)	Локальний файл-резервна копія Scratch-проєкту
Scratch	Візуальне середовище блочного програмування MIT Media Lab для дітей та підлітків 8–16 років.
SMART-ціль (SMART Goal)	Specific, Measurable, Achievable, Relevant, Time-bound
SpreadsheetApp (GAS)	Клас Apps Script для роботи з Google Sheets: відкриття таблиці, читання/запис діапазонів, управління аркушами.

ЛІТЕРАТУРА

Основна

1. Гладких Г. В., Шарова Т. М. Організація самостійної діяльності здобувачів вищої освіти засобами ІКТ. *Педагогіка формування творчої особистості у вищій і загальноосвітній школах* : зб. наук. пр. 2020. Т. 2, № 69. С. 70–74.
2. Колмакова В. О. Імерсивні технології як сучасна освітня стратегія підготовки майбутніх фахівців. *Українські студії в європейському контексті* : зб. наук. пр. 2022. № 5. С. 177–182.
3. Нестеренко Є. В., Сириця О. О. Розробка довідково-інформаційної системи вчителя школи. *Українські студії в європейському контексті* : зб. наук. пр. 2021. № 4. С. 160–168.
4. Шаров С. В. Сучасний стан розвитку штучного інтелекту та напрямки його використання. *Українські студії в європейському контексті* : зб. наук. пр. 2023. № 6. С. 136–144.
5. Грицюк Ю. І. Система комплексного оцінювання якості програмного забезпечення. *Scientific Bulletin of UNFU*. 2022. Т. 32, № 2. С. 81–95.
6. Гулай О., Кабак В., Герасимчук Г. Засоби та технології цифрового навчання: теоретичний та практичний аспекти. Луцьк : ЛНТУ, 2025. 179 с.
7. Марчук Н. А. Цифрові інструменти в професійній освіті України: історія виникнення, особливості впровадження та перспективи. *Професійно-прикладні дидактики*. 2024. № 2. С. 48–53.
8. Anishchenko O., Kotun K., Kupalnyi V. Digital educational resources as a means of ensuring the effectiveness of blended learning. *UNESCO Chair Journal Lifelong Professional Education in the XXI Century*. 2024. № 2(10). С. 57–72.
9. Цапко А., Юденкова О., Сусретов А. Навчальні тести як засіб формування професійних компетентностей у здобувачів освіти. *Актуальні проблеми гуманітарних і природничих наук (APHN-journal)*. 2022. № 57/3.
10. Кухаренко В. М. Формування комунікаційних компетентностей тьютора. *Матеріали Десятої міжнародної конференції з адаптивних технологій управління навчанням*. 2024. С. 6–8.

11. Алека Г. І. Особливості використання середовища Scratch при підготовці майбутніх вчителів інформатики початкової школи. *Наукові записки. Серія: Педагогічні науки*. 2024. № 216. С. 82–87.
12. Білик В., Пан Ахмед Аль-Нашар. Сервіс Kahoot для проведення інтерактивних опитувань здобувачів освіти. *International Science Journal of Education & Linguistics*. 2023. Т. 2, № 1. С. 77–85. DOI: 10.46299/j.isjel.20230201.08.
13. Копус О. А. Інструменти інтерактивного тестування в контексті Assessment for Learning. *Українські студії в європейському контексті*. 2023. № 6.
14. Москаленко О. М., Федяй І. О., Бакуменко Т. К., Косенюк Г. В. Використання Google інструментів для освітнього процесу: Google Classroom як інноваційне рішення для дистанційного навчання. *Академічні візії*. 2023. № 19.
15. Dzen V., Borzov Y., Dzen D. Інтеграція Smart-систем в освітнє середовище закладів вищої освіти. *Вісник Львівського державного університету безпеки життєдіяльності*. 2024. № 30. С. 56–66. DOI: 10.32447/20784643.30.2024.06.
16. Лубко Д. В., Шаров С. В. Напрямки використання інтелектуальних систем в освітньому процесі. *Українські студії в європейському контексті*. 2021. № 3. С. 305–310.
17. Morze N., Vuinytska O. Модернізація освіти в цифровому вимірі. Київ, 2021. 300 с.

Додаткова

1. Гулай О., Шемет В., Фурс Т. Чат GPT у організації наукових досліджень студентів. *Цифрова трансформація: виклики та стратегії* : матеріали Міжнар. наук.-практ. конф., 25 лютого 2025 р. Луцьк : ЛНТУ, 2025. С. 216–218.
2. Лобацький А. Підготовка фахівців цифрових технологій в умовах дуальної освіти. *Академічні студії. Серія «Педагогіка»*. 2023. № 4. С. 81–85.

3. Лобацький А. Інтеграція змішаного навчання в курси комп'ютерних наук у контексті впливу на розвиток фахової компетентності студентів. *Health & Education*. 2024. № 1. С. 217–222.
4. Філіпенко Л. В., Думанський О. В., Козак О. В. Академічна доброчесність в науковому та освітньому середовищі закладів освіти України: погляд крізь призму наявності штучного інтелекту. *Академічні візії*. 2023. Вип. 19.
5. Лобацький А. Педагогічні умови формування фахової компетентності бакалаврів сфери комп'ютерних технологій в умовах змішаного навчання з елементами дуальної освіти. *Інноваційна педагогіка*. 2024. № 74. С. 164–169.

Електронні ресурси

1. Як навчати в цифровому світі. URL: <https://prometheus.org.ua/prometheus-free/how-to-teach-in-a-digital-world/> (дата звернення: 15.05.2026).
2. H5P Official Tutorials. URL: <https://h5p.org/documentation/for-authors/tutorials> (дата звернення: 15.05.2026).
3. Scratch Educator Resources. URL: <https://scratch.mit.edu/educators/> (дата звернення: 15.05.2026).
4. Google Apps Script: документація розробника. URL: <https://developers.google.com/apps-script> (дата звернення: 15.05.2026).

Проектування та розробка педагогічних програмних засобів: методичні вказівки до виконання практичних робіт для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Середня освіта. Інформатика» галузі знань А Освіта спеціальності А4.09 Середня освіта (Інформатика) денної форм навчання / уклад. А. Лобацький. – Луцьк: ЛНТУ, 2026. – 96 с.

Комп'ютерний набір
Редактор

А. ЛОБАЦЬКИЙ
А. ЛОБАЦЬКИЙ

Підп. до друку «__»_____ 2026 р. Формат 60x84/16. Папір офс.
Гарн. Таймс. Ум. друк. арк. 6.
Тираж 50 прим.

Луцький національний технічний університет
43018, м. Луцьк, вул. Львівська, 75

