



Єрмакова С.С, Іванова О.С.

ОСНОВИ ДИДАКТИКИ

для здобувачів вищої освіти першого (бакалаврського) рівня
освітньо-професійної програми «Інформатика та програмування»
зі спеціальності А4 Середня освіта
(Предметна спеціальність А4.09 – Середня освіта/Інформатика)
для денної та заочної форм навчання

КОНСПЕКТ ЛЕКЦІЙ

Єрмакова С.С. Іванова О. С. Конспект лекцій з дисципліни «Основи дидактики» для здобувачів вищої освіти першого (бакалаврського) рівня освітньо-професійної програми «Інформатика та програмування», спеціальності А4.09 Середня освіта (Інформатика) денної та заочної форми навчання / С. С. Єрмакова , Іванова О. С.; [Електронне видання]. – О.: МГУ, 2025. 38с.

Рецензенти:

Кічук Н. В. - доктор педагогічних наук, професор, професор кафедри дошкільної та початкової освіти, декан Педагогічного факультету, Ізмаїльський державний гуманітарний університет.

Стрелковська І. В. - доктор технічних наук, професор, декан Факультету кібербезпеки, програмної інженерії та комп'ютерних наук, Міжнародний гуманітарний університет.

Конспект лекцій з дисципліни «Основи дидактики» затверджений на
засіданні кафедри педагогіки та психології
(№ 1 від 29 серпня 2025 року)

Видання спрямоване закріпити та поглибити теоретичний навчальний матеріал з дисципліни «Основи дидактики»; містять методичні вказівки до структури курсу, перелік питань до підсумкового контролю; список рекомендованих джерел та додатки в якому подані критерії оцінювання за дисципліною. Також важливим результатом методичного видання є систематизація теоретичного навчального матеріалу з метою підвищення рівня засвоєння навчального матеріалу.

Зміст

Вступ.....	4
Очікуванні результати навчання	4
Тематичний зміст навчальної дисципліни	6
Структура навчальної дисципліни.....	9
Зміст лекцій з дисципліни	10
<i>Лекція 1. Дидактика як теорія освіти і навчання в цифрову епоху.....</i>	<i>10</i>
<i>Лекція 2. Процес навчання як цілісна система</i>	<i>12</i>
<i>Лекція 3. Закономірності та принципи навчання в ІТ-освіті</i>	<i>14</i>
<i>Лекція 4. Зміст освіти та державні стандарти навчання інформатики</i>	<i>15</i>
<i>Лекція 5. Методи навчання: від репродуктивних до дослідницьких.....</i>	<i>17</i>
<i>Лекція 6. Інноваційні технології навчання інформатики.....</i>	<i>19</i>
<i>Лекція 7. Форми організації навчання у комп'ютерному класі</i>	<i>21</i>
<i>Лекція 8. Дистанційне навчання та цифрові інструменти дидактики</i>	<i>23</i>
<i>Лекція 9. Діагностика та оцінювання навчальних досягнень учнів.....</i>	<i>25</i>
<i>Лекція 10. Проєктування та самоаналіз діяльності вчителя інформатики...27</i>	
Питання до підсумкового контролю	28
Глосарій до тематичної програми	30
Форми контролю успішності та критерії оцінювання	32
Перелік рекомендованої літератури	36

ВСТУП

Метою вивчення навчальної дисципліни «Основи дидактики» є підготовка фахівців спеціальності А4.09 Середня освіта (Інформатика) за освітнім ступенем бакалавра відповідно до державних стандартів, встановлених освітньо-професійною програмою «Інформатика та програмування»; формування у майбутніх учителів системи теоретичних знань про загальні закономірності, принципи, методи та форми навчання, а також розвиток практичних умінь і навичок проектування освітнього процесу в закладах середньої освіти.

Основними **завданнями** вивчення дисципліни є:

- **теоретичні:** засвоєння фундаментальних понять дидактики (викладання, учіння, зміст освіти, методи контролю) та розуміння логіки навчального процесу;

- **практичні:** оволодіння методиками розробки поурочних планів, структурування навчального матеріалу з інформатики та програмування, вибору оптимальних методів навчання відповідно до вікових особливостей учнів;

- **технологічні:** формування здатності інтегрувати сучасні цифрові інструменти та інформаційно-комунікаційні технології в традиційну дидактичну систему для підвищення ефективності засвоєння знань;

- **прогностичні:** розвиток умінь аналізувати результати навчання, здійснювати діагностику знань учнів та коригувати освітню траєкторію.

Передумови для вивчення дисципліни. Перед вивченням навчальної дисципліни «Основи дидактики» здобувачі мають вивчати такі навчальні дисципліни, як Загальна педагогіка, Академічна доброчесність та професійна етика, Основи загальної психології.

ОЧІКУВАННІ РЕЗУЛЬТАТИ НАВЧАННЯ

ОЧІКУВАНІ КОМПЕТЕНТНОСТІ, ЯКІ ПЛАНУЄТЬСЯ СФОРМУВАТИ ТА ДОСЯГНЕННЯ ПРОГРАМНИХ РЕЗУЛЬТАТІВ

У процесі реалізації програми дисципліни «Основи дидактики» формуються наступні компетентності із передбачених освітньою програмою:

Інтегральна компетентність

ІК1. Здатність розв'язувати складні спеціалізовані задачі у сфері середньої освіти.

Загальні компетентності (ЗК)

ЗК9. Здатність виявляти ініціативу та підприємливість.

Фахові компетентності

ФК1. Здатність виконувати вимоги Державного стандарту базової середньої освіти у професійній діяльності з урахуванням предметної спеціальності, моделювати зміст навчання відповідно до обов'язкових результатів навчання учнів, планувати, організовувати освітній процес з використанням різних видів і форм навчально-пізнавальної діяльності, прогнозувати результати освітнього процесу.

ФК3. Здатність здійснювати інтегроване навчання учнів, добирати і використовувати сучасні та ефективні методики й технології навчання, виховання і розвитку учнів, використовувати інновації у професійній діяльності.

ФК5. Здатність здійснювати оцінювання та моніторинг результатів навчання учнів на засадах компетентнісного підходу, оцінювати рівень навчальних досягнень учнів, аналізувати результати їхнього навчання, навчати учнів самооцінювання та взаємооцінювання.

ФК17. Здатність здійснювати моніторинг власної педагогічної діяльності, навчатися впродовж життя.

Навчальна дисципліна «Основи дидактики» забезпечує досягнення програмних результатів навчання (РН), передбачених освітньою програмою:

РН1. Знати та дотримуватися вимог Державного стандарту базової середньої освіти у професійній діяльності відповідно до предметної спеціальності; знати законодавчі вимоги щодо змісту загальної середньої освіти відповідного рівня та форм організації освітнього процесу (державні стандарти, типові освітні програми, модельні навчальні програми); планувати освітній процес на основі освітньої програми закладу освіти і навчальних програм з предметів (інтегрованих курсів) з урахуванням вимог Державного стандарту базової середньої освіти.

РН6. Добирати дидактичні матеріали для вивчення учнями окремих тем/розділів навчальної програми відповідно до обов'язкових результатів навчання; добирати доцільні сучасні методики й технології навчання, виховання і розвитку учнів засобами освітньої галузі/навчального предмета (інтегрованого курсу) відповідно до визначених теми, мети і завдань уроку; використовувати навчальний матеріал з метою розвитку в учнів ключових компетентностей і вмінь, спільних для всіх компетентностей; навчати учнів застосовувати їх на практиці; моделювати навчальні заняття на основі компетентнісного, діяльнісного, особистісно зорієнтованого підходів.

РН9. Володіти методиками та інструментами оцінювання й моніторингу результатів навчання учнів, здійснювати різні види оцінювання результатів навчання учнів (формувальне, поточне, підсумкове тощо) з використанням відповідних методик і критеріїв оцінювання, аналізувати результати навчання учнів з метою подальшого врахування їх в освітньому процесі; використовувати методи та прийоми розвитку в учнів здатності до самооцінювання та взаємооцінювання результатів навчання.

РН18. Організовувати навчальні заняття різних типів, застосовувати різні види й форми навчально-пізнавальної діяльності учнів.

РН19. Добирати та застосовувати інноваційні форми, методи, прийоми, засоби навчання у педагогічній діяльності, визначати власні освітні потреби та обирати відповідні види, форми, програми професійного розвитку.

Заплановані результати навчання за навчальною дисципліною у термінах НРК

Знання:

Зн1 Концептуальні наукові та практичні знання в області освіти та дидактики

Зн2 Критичне осмислення теорій, принципів, методів і понять у сфері дидактики та навчання

Уміння/Навички:

Ум1 Поглиблені когнітивні та практичні уміння/навички, майстерність та інноваційність на рівні, необхідному для розв'язання складних спеціалізованих задач і практичних проблем у сфері дидактики та навчання

Комунікація

K1 Донесення до фахівців і нефахівців інформації, ідей, проблем, рішень, власного досвіду та аргументації

K2 Збір, інтерпретація та застосування даних у сфері навчання

K3 Спілкування з професійних питань, у тому числі іноземною мовою, усно та письмово

Відповідальність і автономія

AB1 Управління складною професійною діяльністю чи проектами в області та навчання

AB2 Спроможність нести відповідальність за вироблення та ухвалення рішень у непередбачуваних навчальних контекстах

AB3 Формування суджень, що враховують соціальні, наукові та етичні аспекти

AB4 Організація та керівництво професійним розвитком осіб та груп

AB5 Здатність продовжувати навчання із значним ступенем автономії

3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Тема 1. Дидактика як теорія освіти і навчання в цифрову епоху

Предмет, завдання та категорії дидактики. Специфіка навчання покоління «цифрових аборигенів» (Digital Natives). Дидактичні аспекти викладання програмування як нової грамотності.

(ІК, ЗК9, ФК17)

Тема 2. Процес навчання як цілісна система

Структура процесу навчання: цілі, зміст, діяльність вчителя та учня. Етапи засвоєння знань: від сприйняття інтерфейсу до логічного абстрагування при написанні коду. Рушійні сили навчального процесу в умовах інформатизації освіти.

(РН1, РН6, ФК1, ФК3)

Тема 3. Закономірності та принципи навчання в ІТ-освіті

Система загальнодидактичних принципів (науковість, наочність, доступність). Принцип системності та послідовності у вивченні мов програмування: від візуальних (Scratch) до текстових (Python, C++). Реалізація принципу зв'язку теорії з практикою через виконання комп'ютерних проєктів.

(РН6, РН18, ФК3)

Тема 4. Зміст освіти та державні стандарти навчання інформатики

Аналіз Державного стандарту базової середньої освіти (інформатична галузь). Навчальні плани та модельні програми за напрямом «Інформатика та програмування». Дидактичні вимоги до шкільних підручників та електронних освітніх ресурсів.

(РН1, ФК1)

Тема 5. Методи навчання: від репродуктивних до дослідницьких

Класифікація методів навчання за джерелом знань та характером пізнавальної діяльності. Специфіка пояснювально-ілюстративного методу при викладанні синтаксису мов програмування. Метод «чорної скриньки» та евристичні бесіди під час налагодження алгоритмів (debugging).

(ФК3, ФК14, ІК)

Тема 6. Інноваційні технології навчання інформатики

Метод проєктів (PBL) як ключова технологія в ІТ-освіті. STEM-технології та міжпредметна інтеграція (інформатика + математика/фізика). Гейміфікація навчального процесу: використання ігрових платформ (Code.org, Swift Playgrounds).

(РН6, РН19, ЗК9, ФК3).

Тема 7. Форми організації навчання у комп'ютерному класі

Урок як основна форма навчання: структура, типи, санітарно-гігієнічні вимоги. Специфіка проведення уроків-практикумів, лабораторних робіт та хакатонів. Змішане навчання (Blended Learning): моделі «Перевернутий клас» та «Ротація станцій».

Інклюзивна освіта та індивідуалізація навчання. Принципи інклюзії; універсальний дизайн навчання (UDL); адаптації та модифікації матеріалів; ІПР; педагогічний супровід учнів з ООП.

(РН16, РН18, РН19, ФК3, ФК14)

Тема 8. Дистанційне навчання та цифрові інструменти дидактики

Особливості дистанційної взаємодії «вчитель – комп'ютер – учень». Використання LMS (Moodle, Google Classroom) та хмарних сервісів для організації самостійної роботи. Дидактичний потенціал Штучного Інтелекту в допомозі вчителю (генерація завдань, перевірка коду).

(РН18, РН19, ФК3, ФК14, ІК)

Тема 9. Діагностика та оцінювання навчальних досягнень учнів

Функції та види контролю: від попереднього до підсумкового. Критерії оцінювання результатів навчання за освітньою програмою «Інформатика та програмування». Формувальне оцінювання: використання чек-листів, рубрик та автоматизованих систем тестування.

(РН9, ФК5)

Тема 10. Проєктування та самоаналіз діяльності вчителя інформатики

Технологія підготовки вчителя до уроку: складання технологічної карти та плану-конспекту. Дидактична рефлексія та педагогічний аналіз уроку. Шляхи підвищення професійної майстерності вчителя в умовах стрімкого розвитку технологій.

(РН19, ФК17, ЗК9, ІК).

4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістов их модулів і тем	Кількість годин							
	Денна форма				Заочна форма			
	Усього	У тому числі			Усього	У тому числі		
		Лекц	Прак т	Сам		Лекц	Прак т	Сам
Тема 1. Дидактика як теорія освіти і навчання в цифрову епоху.	18	4	4	10	18	2	-	16
Тема 2. Процес навчання як цілісна система.	18	4	4	10	18	2	-	16
Тема 3. Закономірності та принципи навчання в ІТ- освіті.	18	4	4	10	18	2	2	14

[illegible]

ЗМІСТ ЛЕКЦІЙ З ДИСЦИПЛІНИ

Лекція 1. Дидактика як теорія освіти і навчання в цифрову епоху

1. Предмет, завдання та категорії дидактики

Дидактика (від грец. *didaktikos* — той, хто навчає) — це галузь педагогіки, яка розробляє теорію освіти і навчання. Якщо педагогіка відповідає на питання «як виховувати?», то дидактика — «чому вчити?» та «як навчити ефективно?».

- Об'єкт дидактики: реальні процеси навчання (взаємодія вчителя та учня).
- Предмет: закономірності та принципи, за якими відбувається передача знань та формування компетенцій.

Основні категорії дидактики:

1. Навчання: двосторонній процес взаємодії викладання (діяльність учителя) та учіння (діяльність учня). В цифрову епоху цей процес часто стає *опосередкованим* (через платформу LMS).
2. Освіта: результат навчання — система знань, умінь і цінностей, яка стає надбанням особистості.
3. Зміст освіти: відібрана система знань, яка закріплена в навчальних програмах.
4. Метод навчання: спосіб досягнення мети (лекція, практична робота, проєкт).
5. Форма навчання: організаційні рамки (урок, вебінар, хакатон).

2. Специфіка навчання покоління «цифрових аборигенів»

Термін «*Digital Natives*», запроваджений Марком Пренські, описує дітей, які народилися в епоху повсюдного інтернету та гаджетів. Для них цифрові технології є «рідною мовою».

Дидактичні особливості «аборигенів»:

- Багатозадачність (Multitasking): здатність швидко перемикає увагу між різними потоками інформації, але часто за рахунок глибини занурення.
- Пріоритет візуального над текстовим: інформація краще сприймається через відео, інфографіку та інтерактивні моделі (інстаграмний/тік-ток формат).
- Потреба в миттєвому фідбеку: вони звикли до швидкої реакції (лайки, сповіщення), тому в навчанні очікують негайної перевірки результатів (автоматичні тести, коментарі в чаті).
- «Кліпове» vs «Лінійне» мислення: складність у читанні довгих лонгвідів; краще працюють із короткими модулями (мікронавчання).
- Горизонтальна комунікація: вони сприймають вчителя не як недосяжний авторитет, а як ментора, коуча або партнера.

3. Програмування як «нова грамотність» (New Literacy)

У ХХІ столітті програмування перестало бути вузькотехнічною навичкою і перейшло в розряд базової грамотності, поруч із читанням та письмом.

Дидактичні аспекти викладання коду:

- Код як засіб вираження: навчання програмуванню — це не вивчення синтаксису мови, а навчання *мисленню*. Учень використовує код для створення власного світу (ігри, мистецтво, симуляції).
- Перехід від споживання до створення: дидактика зміщується від використання готових програм до створення власних інструментів розв'язання проблем.
- Помилка як дидактичний ресурс: у програмуванні помилка (*bug*) не є приводом для покарання; це природний етап дослідження. Це докорінно змінює традиційний підхід до оцінювання.
- Викладання через «Action-Learning»: дидактика коду вимагає негайного застосування теорії на практиці. Спершу «Hello World», потім — теорія того, як це працює.

Висновок. Дидактика цифрової епохи трансформується з «теорії передачі знань» у *«теорію організації пізнавальної діяльності»*. Робота з «цифровими аборигенами» вимагає від педагога гнучкості, використання мікромодулів та візуалізації. Викладання програмування стає фундаментом для розвитку критичного та логічного мислення, формуючи нову мову комунікації людини з технологічним світом.

Питання до самоперевірки

Блок 1: Категоріальний апарат

1. Чим дидактика відрізняється від методики викладання конкретного предмета (наприклад, інформатики)?
2. Як змінилася категорія «зміст освіти» під впливом доступності будь-якої інформації в Google?
3. Дайте визначення процесу «учіння» в умовах дистанційної освіти.

Блок 2: Цифрові аборигени

4. Які переваги та ризики несе «багатозадачність» учнів на уроці інформатики?
5. Як вчителю змінити структуру уроку, щоб адаптувати її під «кліпове мислення» учнів?
6. Наведіть приклади «миттєвого фідбеку» в цифровій дидактиці.

Блок 3: Нова грамотність

7. Чому програмування називають «новою грамотністю» і як це впливає на обов'язкові результати навчання в школі?
8. Поясніть концепцію «Помилка як позитивний досвід» на прикладі уроку з написання коду.
9. У чому полягає дидактична складність викладання синтаксису мови програмування для учнів із кліповим сприйняттям?

Лекція 2. Процес навчання як цілісна система

1. Структура процесу навчання: цілі, зміст, діяльність

Процес навчання — це цілісна система, де зміна одного елемента неминуче веде до трансформації інших. В епоху Digital структура виглядає так:

- Цільовий компонент. Визначає «навіщо вчимо?». Сьогодні мета зміщується від накопичення знань до формування цифрової суб'єктності — здатності учня самостійно діяти та створювати продукти в ІТ-середовищі.
- Змістовий компонент. «Чому вчимо?». Включає не лише теорію (що таке цикл?), а й контекст (де цей цикл застосовується в реальному додатку?). Зміст стає динамічним: підручник доповнюється документацією та форумами.
- Операційно-діяльнісний компонент:
 - Діяльність вчителя (викладання). Переходить від трансляції знань до педагогічного дизайну — проектування середовища, де учень може знайти відповіді самостійно.
 - Діяльність учня (учіння). Активна пізнавальна діяльність. Учень перестає бути «приймачем» інформації, стаючи її дослідником та розробником.
- Контрольно-регулювальний компонент. Постійний моніторинг через цифрові інструменти (LMS-аналітика), що дозволяє коригувати процес миттєво.

2. Етапи засвоєння знань в ІТ-освіті

Процес перетворення зовнішньої інформації у внутрішній досвід учня проходить через кілька специфічних етапів:

1. Сприйняття (етап інтерфейсу):

Перше знайомство з інструментом. Учень бачить вікна, кнопки, палітру блоків у Scratch або редактор коду. Важливо, щоб на цьому етапі «когнітивний шум» (складний інтерфейс) не заважав сприйняттю суті.

2. Розуміння (етап логічних зв'язків).

Усвідомлення того, як елементи пов'язані між собою. Наприклад: «Якщо я натисну цю кнопку, спрацює цей блок коду». Це перехід від візуального образу до першої логічної гіпотези.

3. Узагальнення та абстрагування (Етап коду).

Найвищий рівень. Учень відривається від конкретної картинки та починає мислити абстрактними категоріями: змінна, функція, масив. Написання коду — це процес перекладу логічної моделі з голови в синтаксис мови.

4. Закріплення та застосування.

Виконання практичних завдань. У програмуванні це етап «вирішення проблем» (Problem Solving), де знання перевіряються працездатністю програми.

3. Рушійні сили навчального процесу в умовах інформатизації

Рушійною силою навчання є **суперечність** між виниклою у учня потребою в знаннях та реальними можливостями їх задовольнити.

- Внутрішні суперечності:
 - Між бажанням створити круту гру та відсутністю знань про координатні осі або цикли.
 - Між впевненістю «я все знаю» та помилкою в коді, яку видає компілятор.
- Зовнішні стимули (Інформатизація як прискорювач):
 - Гейміфікація: Досягнення, рівні та прогрес-бари створюють штучну, але потужну потребу рухатися далі.
 - Доступність інструментів: Наявність безкоштовних IDE та онлайн-курсів знижує бар'єр входу, дозволяючи рушійним силам працювати швидше.
 - Соціальне визнання: Можливість опублікувати свій проект на GitHub або в Play Market мотивує учня долати труднощі навчання.

Висновок. Процес навчання в ІТ-освіті — це не просто передача інструкцій, а *технологічно насичена система*. Успіх навчання залежить від того, наскільки вчитель допомагає учню пройти шлях від пасивного сприйняття красивого інтерфейсу до глибокого логічного абстрагування, що втілюється в працюючому коді.

Питання до самоперевірки

Блок 1: Структура системи

1. Як цифровізація змінила «Змістовий компонент» навчання інформатики (порівняйте підручник 2000-х та сучасні ресурси)?
2. У чому полягає нова роль вчителя як «педагогічного дизайнера» в цілісній системі навчання?
3. Чому контроль-регулювальний компонент у цифровому навчанні є більш об'єктивним, ніж у традиційному?

Блок 2: Етапи засвоєння

4. Опишіть перехід учня від стадії сприйняття до стадії абстрагування на прикладі вивчення теми «Змінні».
5. Чому занадто складний інтерфейс середовища розробки може зупинити процес навчання на етапі сприйняття?
6. На якому етапі засвоєння знань найчастіше виникає «страх перед білим аркушем» (чистим файлом коду)?

Блок 3: Рушійні сили

7. Наведіть приклад внутрішньої суперечності, яка виникає в учня під час розробки власного ІТ-проекту.
8. Як інформатизація (інтернет, ШІ-помічники) може як допомагати, так і заважати рушійним силам навчання?
9. Чому «миттєва помилка» в коді є кращою рушійною силою, ніж зауваження вчителя в зошиті?

Лекція 3. Закономірності та принципи навчання в ІТ-освіті

1. Система загальнодидактичних принципів в ІТ

Принципи навчання — це фундаментальні положення, що визначають вимоги до змісту та методів викладання.

- Принцип науковості. В ІТ-освіті цей принцип вимагає використання актуальних стандартів. Не можна вчити «застарілому» коду. Науковість означає розуміння учнями того, що за візуальним блоком Scratch стоїть реальна математична логіка та архітектура фон Неймана.
- Принцип наочності. В інформатиці наочність еволюціонувала від малюнка на дошці до динамічної візуалізації. Це демонстрація роботи пам'яті («трасування коду»), візуалізація сортування масивів та використання інтерактивних середовищ.
- Принцип доступності. Навчання має бути на межі «зони найближчого розвитку». Для 2-го класу доступним є керування роботом-мишею, для 10-го — робота з бібліотеками штучного інтелекту. Доступність не означає «примітивність», а означає відповідність когнітивним можливостям учня.

2. Принцип системності та послідовності: від Scratch до C++

Цей принцип забороняє хаотичне вивчення тем. В ІТ-освіті він реалізується через *трансфер навичок*:

1. Візуальний рівень (Scratch). Формування базових концепцій (цикл, умова, подія) без стресу через синтаксичні помилки. Учень бачить логіку «цеглинками».
2. Перехідний рівень (Blockly / Python). Поступове введення текстових команд. Учень розуміє, що текстове слово `while` — це той самий блок «повторювати поки», який він знав раніше.
3. Текстовий рівень (Python, C++). Перехід до повної абстракції. Системність полягає в тому, що ми не вчимо мову як набір команд, а вчимо алгоритмічні структури, які є універсальними для всіх мов.

Послідовність також передбачає шлях:

«алгоритм у голові

блок-схема

візуальний код

текстовий код».

3. Зв'язок теорії з практикою: проєктний підхід

В ІТ-освіті теорія без практики є мертвою. Принцип реалізується через девіз: *«Знаю, щоб створювати»*.

- Комп'ютерний проєкт як дидактична одиниця. Теорія (наприклад, типи даних) засвоюється найкраще, коли учню потрібно розв'язати практичну задачу: «Який тип даних обрати для збереження кількості життів у грі?».

- Цикл «Теорія — Спроба — Фідбек». На відміну від математики, де результат перевіряє вчитель, в ІТ практику перевіряє компілятор. Це створює негайний зв'язок: «Я зрозумів теорію невірно — програма не працює».
- Метапредметність. Практика часто виходить за межі ІТ. Створення калькулятора площ для геометрії або симуляції фізичного процесу — це ідеальна реалізація зв'язку навчання з реальним життям.

Висновок. Закономірності ІТ-освіти вимагають від учителя дотримання балансу: не перевантажувати учня складним синтаксисом (доступність), забезпечувати логічний перехід між мовами (системність) та завершувати кожну теоретичну тему створенням реального цифрового продукту (зв'язок із практикою).

Питання до самоперевірки

Блок 1: Загальні принципи

1. Як принцип науковості змушує вчителя інформатики постійно змінювати свою навчальну програму?
2. Наведіть приклад «сучасної наочності» при поясненні теми «Як працює Інтернет».
3. Чому ігнорування принципу доступності при вивченні мов програмування призводить до швидкої втрати мотивації учнів?

Блок 2: Системність та послідовність

4. У чому полягає дидактична небезпека занадто тривалого перебування учнів у середовищі Scratch?
5. Як реалізувати принцип послідовності, якщо учень хоче створити мобільний додаток, але ще не знає, що таке змінні?
6. Опишіть траєкторію вивчення теми «Цикли» від початкової до старшої школи, дотримуючись принципу системності.

Блок 3: Теорія та практика

7. Чому в ІТ-освіті неможливо реалізувати принцип зв'язку з практикою через просте переписування коду з підручника?
8. Які критерії має мати «комп'ютерний проєкт», щоб він дійсно сприяв глибокому засвоєнню теорії?
9. Як вчителю реалізувати зворотний зв'язок «Практика_Теорія», коли учень знайшов рішення в Google, але не розуміє, як воно працює?

Лекція 4. Зміст освіти та державні стандарти навчання інформатики

1. Аналіз Державного стандарту базової середньої освіти

Державний стандарт — це документ, який визначає вимоги до обов'язкових результатів навчання учнів. У контексті Нової української школи (НУШ) стандарт інформатичної галузі базується не на переліку знань, а на *групах результатів*.

Основні групи результатів (компетентності):

1. Пошук та опрацювання даних. Учень має вміти знаходити, фільтрувати та критично оцінювати інформацію.
2. Створення інформаційних продуктів. Навичка проектування та розробки (тексти, медіа, код, бази даних).
3. Алгоритмічне мислення та програмування. Здатність будувати логічні моделі та реалізовувати їх у програмному середовищі.
4. Безпека та етика. Цифрове громадянство, захист даних, розуміння впливу технологій на суспільство.

Важливо: Стандарт не фіксує конкретну мову програмування (напр. Python), він фіксує *здатність* використовувати алгоритмічні конструкції.

2. Навчальні плани та модельні програми

Зміст освіти деталізується через ієрархію документів, що забезпечують системність навчання.

- Типова освітня програма (ТОП). Задає загальні рамки та кількість годин. Вона гарантує, що учень в будь-якому куточку країни отримає базовий обсяг ІТ-підготовки.
- Модельні навчальні програми (МНП). Це "сценарії" навчання. Вчитель обирає одну з кількох рекомендованих МОН програм.
 - *Програми для 5-6 класів* зазвичай акцентують на візуальному програмуванні та цифровій творчості.
 - *Програми для 7-9 класів* вводять текстове програмування та складні обчислювальні моделі.
- Навчальний план закладу Документ, де школа визначає конкретну кількість годин на інформатику (наприклад, збільшення годин за рахунок варіативної складової для математичних класів).

3. Дидактичні вимоги до підручників та ЕОР

Оскільки ІТ-сфера змінюється швидше, ніж цикли друку книг, дидактичні вимоги до ресурсів стали специфічними.

Вимоги до шкільного підручника:

- Інваріантність. Підручник має висвітлювати фундаментальні поняття (алгоритм, об'єкт, цикл), які не залежать від версії мови програмування.
- Інтерактивність (через QR-коди). Сучасний підручник — це "портал" до відеоуроків, онлайн-компіляторів та файлів-заготовок.
- Проблемність. Замість інструкції "натисніть сюди", підручник має ставити задачу: "як зробити так, щоб герой стрибав?".

Вимоги до електронних освітніх ресурсів (ЕОР):

- Мультимедійність. Поєднання тексту, анімації виконання коду та аудіо-супроводу.
- Адаптивність. Ресурс має коректно відображатися і на ПК в класі, і на смартфоні учня.
- Інтерактивний зворотний зв'язок. Можливість миттєвої перевірки коду або тесту (автоперевірка).

- Гіпертекстуальність: наявність посилань на офіційну документацію мов програмування для виховання навички самоосвіти.

Висновок. Зміст ІТ-освіти сьогодні — це баланс між **Державним стандартом** (що гарантує фундаментальність) та **модельними програмами** (що дають вчителю свободу вибору інструментів). Якість навчання залежить від того, наскільки підручники та ЕОР здатні перетворити суху теорію на живий, інтерактивний процес створення цифрового продукту.

Питання до самоперевірки

Блок 1: Державний стандарт

1. Чому Державний стандарт НУШ не містить назв конкретних програм (наприклад, Microsoft Word чи Scratch)?
2. Які чотири групи результатів навчання є обов'язковими для інформатичної галузі?
3. Як стандарт забезпечує наступність між початковою школою та ліцеєм?

Блок 2: Програми та плани

4. У чому полягає різниця між "Типовою" та "Модельною" навчальною програмою?
5. На основі чого вчитель створює власну "календарно-тематичну розробку"?
6. Як академічна свобода вчителя проявляється при виборі модельної програми?

Блок 3: Ресурси та підручники

7. Чому класичний паперовий підручник з інформатики вважається "допоміжним", а не "основним" засобом навчання?
8. Сформулюйте 3 дидактичні переваги електронного підручника над паперовим при вивченні теми "Програмування".
9. Які елементи ЕОР допомагають учню самостійно контролювати свій прогрес?

Лекція 5. Методи навчання: від репродуктивних до дослідницьких

1. Класифікація методів навчання в ІТ

Метод навчання — це спосіб взаємодії вчителя та учня для досягнення цілі. В ІТ-дидактиці використовують дві основні класифікації:

А. За джерелом знань:

- Словесні: розповідь, лекція, інструктаж (пояснення логіки алгоритму).
- Наочні: демонстрація екрана, скринкасти, робота з блок-схемами.
- Практичні: написання коду, тестування програм, лабораторні роботи.

Б. За характером пізнавальної діяльності (за Лернером та Скаткіним):

1. Пояснювально-ілюстративний (репродуктивний): учень сприймає готову інформацію.

2. Проблемний виклад: вчитель ставить проблему і сам показує шлях її розв'язання.
3. Частково-пошуковий (евристичний): учень розв'язує частину задачі самостійно.
4. Дослідницький: учень самостійно проходить шлях від постановки задачі до створення продукту.

2. Специфіка пояснювально-ілюстративного методу

Хоча сучасна дидактика прагне до активності, *пояснювально-ілюстративний метод* залишається критично важливим на етапі знайомства з новим *синтаксисом*.

- Чому він потрібен? Учень не може «винайти» назву функції чи правила розстановки дужок у мові Python. Це конвенційні знання, які треба просто передати.
- *Дидактичні вимоги в ІТ:*
 - Лаконічність: пояснення синтаксису не має перевищувати 10–15 хвилин (мікронавчання).
 - Паралельна демонстрація: вчитель пише код на екрані (Live Coding), а учні бачать результат виконання миттєво.
 - Еталонний зразок: надання «чистого» прикладу коду, який учні використовують як шаблон для подальших модифікацій.

3. Метод «чорної скриньки» та евристичні бесіди

Коли базовий синтаксис засвоєно, дидактика переходить до вищих рівнів пізнання.

Метод «чорної скриньки» (Black Box Testing):

Це метод дослідження системи, де ми знаємо, що подаємо на **вхід** і що отримуємо на *виході*, але не знаємо внутрішнього механізму.

- Застосування. Вчитель дає учням програму, код якої приховано. Учні мають шляхом тестування (введення різних чисел) зрозуміти алгоритм її роботи. Це розвиває аналітичне мислення та вміння будувати логічні гіпотези.

Евристична бесіда та налагодження (Debugging):

Налагодження коду — це інтелектуальний квест. Вчитель не має вказувати на помилку («У тебе тут двокрапка пропущена»). Замість цього використовується *евристична бесіда*:

- Вчитель: «Подивись на результат у консолі. Що він тобі каже?»
- Учень: «Помилка в 5 рядку».
- Вчитель: «А що саме комп'ютер очікував побачити в цьому рядку?»
Таким чином, учень через серію навідних запитань сам «відкриває» рішення.

Висновок. Ефективна ІТ-дидактика — це перехід від *репродукції* (копіювання коду з дошки) до *дослідження* (пошуку власних алгоритмів). Вчитель має

майстерно поєднувати чітке пояснення синтаксису з методами, що змушують учня думати самостійно, аналізувати помилки та тестувати гіпотези.

Питання до самоперевірки

Блок 1: Класифікація

1. Чому суто словесні методи (лекція без демонстрації ПК) малоефективні при вивченні програмування?
2. Наведіть приклад переходу від частково-пошукового до дослідницького методу під час вивчення теми «Цикли».
3. Який метод за джерелом знань є провідним на етапі перевірки працездатності коду?

Блок 2: Пояснення та ілюстрація

4. Чому «Live Coding» (написання коду вчителем у реальному часі) є кращим за показ скріншота з кодом у презентації?
5. Як реалізувати принцип науковості під час пояснення синтаксису нової мови програмування?
6. Сформулюйте недоліки надмірного використання репродуктивного методу в ІТ-освіті.

Блок 3: Активне навчання

7. Опишіть алгоритм проведення вправи за методом «чорної скриньки» для теми «Математичні оператори».
8. Чому «евристична бесіда» є психологічно складнішою для вчителя, ніж проста вказівка на помилку в коді?
9. Як культура «Debugging» (налагодження) сприяє формуванню наполегливості та критичного мислення учня?

Лекція 6. Інноваційні технології навчання інформатики

1. Метод проєктів (PBL) у навчанні інформатики

Project-Based Learning (PBL) — це технологія, за якої учні здобувають знання та навички шляхом тривалого дослідження та вирішення реальної, складної задачі.

- Дидактична цінність. Учень перестає питати: «Навіщо мені вчити масиви?». Він вчить їх, бо без них неможливо реалізувати таблицю лідерів у його власній грі.
- Етапи PBL.
 - Запуск. Постановка проблемного питання («Як автоматизувати полив шкільної клумби?»).
 - Планування. Створення технічного завдання (ТЗ), розподіл ролей у команді.
 - Виконання. Написання коду, розробка дизайну, тестування.
 - Презентація. Захист продукту перед аудиторією (пітчінг).
- Результат. Не просто оцінка, а *матеріалізований продукт* (сайт, додаток, бот).

2. STEM-технології та міжпредметна інтеграція

STEM (Science, Technology, Engineering, Math) — це підхід, де інформатика виступає не як окремий предмет, а як інструмент для наукового дослідження.

- Інтеграція з математикою. Використання програмування для побудови графіків функцій або обчислення площ складних фігур (алгоритми замість формул).
- Інтеграція з фізикою. Робототехніка (Arduino, LEGO Spike). Учні пишуть код, щоб керувати фізичними процесами (датчики світла, відстані, мотори).
- Дидактичний ефект. Учні бачать цілісну картину світу. Інформатика дає «мозок» (код), фізика — «тіло» (конструкцію), а математика — «логіку» (розрахунки).

3. Гейміфікація та ігрові навчальні платформи

Гейміфікація — це впровадження ігрових механік (рівні, досвід, квести) у навчальний процес. В інформатиці це реалізується через спеціалізовані платформи:

- Code.org. Світовий стандарт пропедевтики. Навчання через ігри з відомими героями (Minecraft, Frozen). Учні складають алгоритми, допомагаючи героям проходити квести.
- Swift Playgrounds. Інтерактивна гра від Apple, яка вчить професійній мові Swift через 3D-головоломки.

Дидактичні переваги:

- Зниження страху помилки: В грі програш — це просто привід спробувати ще раз.
- Візуалізація виконання: Учень миттєво бачить, куди пішов персонаж через його помилку в коді.
- Потік (Flow): Ігри створюють стан глибокої залученості, де складність задачі зростає паралельно з навичками учня.

Висновок. Інновації в ІТ-освіті зміщують фокус із *вивчення інструменту* на *вирішення проблеми*. Метод проєктів вчить працювати в команді, STEM-технології об'єднують знання про світ у єдину систему, а гейміфікація підтримує високу мотивацію, роблячи складне навчання природним та захопливим.

Питання до самоперевірки

Блок 1: Метод проєктів (PBL)

1. Чим відрізняється «проєктний метод» від звичайного виконання «практичної роботи» за інструкцією вчителя?
2. Які навички (Soft Skills) розвиває учень під час командного ІТ-проєкту?
3. Як вчителю оцінити індивідуальний внесок кожного учня в спільний програмний продукт?

Блок 2: STEM

4. Наведіть приклад навчальної теми, де інформатика нерозривно пов'язана з біологією чи географією.
5. Чому робототехніка вважається ідеальною реалізацією STEM-підходу?
6. Як використання Python на уроках математики допомагає краще зрозуміти поняття «функція»?

Блок 3: Гейміфікація

7. У чому полягає різниця між «використанням ігор на уроці» та «гейміфікацією освітнього процесу»?
8. Які механіки платформи Code.org допомагають реалізувати принцип наочності?
9. Чи може гейміфікація зашкодити навчанню старшокласників? Обґрунтуйте відповідь.

Лекція 7. Форми організації навчання у комп'ютерному класі

1. Урок як основна форма навчання

У комп'ютерному класі урок набуває специфічних рис через наявність персонального обладнання для кожного учня.

- *Типологія уроків:*
 - Введення нового матеріалу (мінімум теорії, максимум демонстрації екрана).
 - Урок формування навичок (практична робота за зразком).
 - Комбінований урок (найпопулярніший: 15 хв теорія + 25 хв практика).
- *Санітарно-гігієнічні вимоги (згідно з регламентом):*
 - Час за монітором. 5-7 класи — до 20 хв; 8-9 класи — до 25 хв; 10-11 класи — до 30 хв за урок.
 - Фізкультхвилинки. Обов'язкова гімнастика для очей та вправи для постави кожні 15-20 хвилин.
 - Освітлення та провітрювання. Запобігання відблискам на екранах та накопиченню озону від техніки.

2. Специфіка практичних форм навчання

В ІТ-освіті акцент зміщується з пасивного слухання на активну дію.

- *Урок-практикум.* Короткі завдання на відпрацювання конкретної навички (наприклад, написання циклу `for`).
- *Лабораторна робота.* Дослідження. Учень отримує готову програму і має змінити параметри, щоб зрозуміти, як вони впливають на результат (метод "дослідження коду").

- *Шкільний хакатон*. Інтенсивна командна робота (2-4 години), де учні створюють міні-проект (гру чи бота) з нуля. Це виховує стресостійкість та навички тайм-менеджменту.

3. Змішане навчання (Blended Learning)

Це поєднання очного навчання в класі та онлайн-навчання з елементами самоконтролю.

- *Перевернутий клас (Flipped Classroom)*:
 - Вдома. Учні дивляться відеопояснення вчителя про нову тему (напр., "Масиви").
 - У класі. Весь час витрачається на розв'язання складних задач та допомогу тим, хто не впорався.
- *Ротація станцій*. Клас ділиться на групи, які змінюють локації кожні 15 хв:
 - Станція вчителя (пояснення складного).
 - Станція онлайн-навчання (тренажери, тести).
 - Станція проектної роботи (командне створення коду).

4. Інклюзивна освіта та індивідуалізація

ІТ-освіта має унікальний потенціал для інклюзії, оскільки комп'ютер може бути асистивним інструментом.

- *Принципи інклюзії*. Рівний доступ, відсутність дискримінації, командна робота (вчитель + асистент).
- *Універсальний дизайн навчання (UDL)*. Створення контенту, який зручний для всіх одразу (відео з субтитрами, текстові інструкції з аудіо-супроводом).
- *Адаптація та модифікація*:
 - *Адаптація*. Те саме завдання, але інша форма (збільшений шрифт, спеціальна миша).
 - *Модифікація*. Спрощення змісту (менша кількість команд в алгоритмі).
- *ІПР (Індивідуальна програма розвитку)*. Документ, що фіксує потреби дитини з ООП та необхідні для неї кроки підтримки на уроці інформатики.

Висновок. Сучасна форма навчання в комп'ютерному класі — це гнучкий мікс *санітарної безпеки, активної практики та цифрової доступності*. Вчитель має майстерно перемикатися між ролями лектора, фасилітатора в "перевернутому класі" та архітектора інклюзивного простору, де кожна дитина, незалежно від потреб, може стати творцем коду.

Питання до самоперевірки

Блок 1: Урок та норми

1. Чому традиційний урок-лекція тривалістю 45 хвилин є неможливим у комп'ютерному класі згідно з нормами?

2. Як вчителю контролювати час роботи учнів за монітором, не перериваючи їхній творчий потік?

3. Які особливості Classroom Management (керування класом) з'являються, коли в кожного учня є доступ до інтернету?

Блок 2: Форми та змішане навчання

4. У чому полягає дидактична перевага "Перевернутого класу" при вивченні складних мов програмування?

5. Опишіть ризики моделі "Ротація станцій" у класі, де технічне обладнання є застарілим.

6. Чим хакатон відрізняється від звичайної контрольної роботи за результатами та атмосферою?

Блок 3: Інклюзія та UDL

7. Як реалізувати принцип Універсального дизайну (UDL), створюючи інструкцію до практичної роботи?

8. Наведіть приклад адаптації середовища Scratch для учня з порушенням зору.

9. Яка роль асистента вчителя під час проведення уроку-практикуму з програмування?

Лекція 8. Дистанційне навчання та цифрові інструменти дидактики

1. Особливості дистанційної взаємодії: нова дидактична тріада

У дистанційному навчанні комп'ютер перестає бути просто інструментом і стає *активним середовищем взаємодії*.

- *Опосередкованість спілкування*. Вчитель не бачить екрана учня «через плече». Це вимагає використання інструментів віддаленого моніторингу та спільного редагування коду (напр., *Replit Teams*, *VS Code Live Share*).
- *Зміна акцентів у тріаді*:
 - Вчитель – Комп'ютер: Вчитель стає архітектором цифрового контенту (педагогічний дизайн).
 - Учень – Комп'ютер: Комп'ютер виконує роль «першого вчителя», який видає помилки компіляції та підказки.
 - Вчитель – Учень: Взаємодія стає партнерською, спрямованою на консультування та фасилітацію.
- Проблема мотивації. Дистанційне навчання вимагає від учня високого рівня *саморегуляції*. Дидактика тут спирається на цікавий контент та гейміфікацію.

2. LMS та хмарні сервіси: Організація самостійної роботи

LMS (Learning Management Systems) — це «цифровий кабінет» вчителя, де навчальний процес триває 24/7.

- *Google Classroom*: Оптимальний для швидкого розповсюдження завдань та збирання робіт. Дидактична перевага — інтеграція з Google Диском для спільної роботи над документами та презентаціями.

- *Moodle / Canvas*: Дозволяють будувати **складні освітні траєкторії**. Учень не отримає доступу до Теми №2, поки не пройде тест за Темою №1. Це забезпечує принцип системності та послідовності в дистанційному форматі.
- *Хмарні IDE (Replit, GitHub Codespaces)*: Дозволяють учням писати код у браузері без встановлення ПЗ. Вчитель може зайти в код учня в реальному часі та залишити коментар до конкретного рядка.

3. Дидактичний потенціал Штучного Інтелекту (ШІ)

У 2026 році ШІ став повноцінним партнером вчителя інформатики, автоматизуючи рутину та персоналізуючи навчання.

- Генерація навчальних матеріалів:
 - Створення 10 варіантів аналогічних завдань за секунди (проти дія списування).
 - Генерація текстових описів задач на основі коду.
 - Створення «зипованого» коду, де учні мають знайти помилки (Debugging-вправи).
- Перевірка та аналіз коду:
 - ШІ може миттєво пояснити учню помилку зрозумілою мовою, не даючи готового рішення.
 - Аналіз стилю коду (Clean Code) та пропозиції щодо його оптимізації.
- ШІ як асистент у плануванні:
 - Створення планів-конспектів уроків та технологічних карт.
 - Адаптація складного тексту підручника для учнів з особливими потребами (спрощення мови).

Висновок. Дистанційна дидактика перетворює вчителя на *куратора цифрового досвіду*. Використання LMS структурує процес, а Штучний Інтелект звільняє час вчителя від перевірки синтаксису, дозволяючи зосередитися на розвитку критичного мислення та креативності учнів. Технології не замінюють вчителя, але вони стають потужним підсилювачем його педагогічної майстерності.

Питання до самоперевірки

Блок 1: Дистанційна взаємодія

1. Чому при дистанційному навчанні інформатики зростає роль «зворотного зв'язку» (фідбеку)?
2. Назвіть 3 дидактичні ризики дистанційного навчання для учнів початкової школи.
3. Як вчителю забезпечити «ефект присутності» під час асинхронного навчання?

Блок 2: LMS та хмари

4. Порівняйте Google Classroom та Moodle: у якому випадку доцільно обрати кожен з систем?

5. Як хмарні сервіси (напр., Replit) реалізують принцип наочності при поясненні помилок у коді?

6. Запропонуйте структуру дистанційного модуля на тему «Масиви» в LMS.

Блок 3: Штучний Інтелект

7. Як використання ШІ для генерації завдань допомагає реалізувати принцип індивідуалізації?

8. Чи є етичним використання ШІ вчителем для оцінювання творчих проєктів учнів? Обґрунтуйте.

9. Наведіть приклад завдання, яке вчитель може згенерувати за допомогою ChatGPT за 1 хвилину для перевірки знань про цикли.

Лекція 9. Діагностика та оцінювання навчальних досягнень учнів

1. Функції та види контролю: від діагностики до результату

В ІТ-дидактиці контроль виконує не лише перевірну, а й *коригувальну* та *мотиваційну* функції. Комп'ютерне середовище дозволяє зробити контроль безперервним.

- *Види контролю:*

- Попередній (діагностичний). Проводиться перед початком нової теми. Мета — виявити «прогалини» (наприклад, чи знають учні математичні операції перед вивченням арифметики в Python).

- Поточний. Контроль на кожному уроці під час виконання практичних вправ. Допомагає вчителю зрозуміти, хто «застряг» на етапі синтаксису.

- Тематичний (періодичний). Оцінювання завершеного розділу (наприклад, модуля «Графіка»).

- Підсумковий: Результат за семестр/рік. В ІТ найчастіше реалізується через захист фінального проєкту.

2. Критерії оцінювання: що саме ми оцінюємо?

В програмуванні оцінка за принципом «працює/не працює» є недостатньою. Дидактика виділяє *багаторівневі критерії*:

1. Функціональність. Чи виконує програма поставлене завдання? Чи обробляє вона некоректні вхідні дані?
2. Алгоритмічна складність. Наскільки ефективно розв'язана задача? (Чи не використовує учень 10 циклів там, де достатньо одного?).
3. Якість коду (Style Guide). Чи зрозумілі назви змінних? Чи є коментарі? Чи дотримано відступів?
4. Самостійність та креативність. Наскільки учень доповнив базове завдання власними ідеями.
5. Презентаційні навички. Вміння пояснити роботу свого коду (захист логіки).

3. Формувальне оцінювання: шлях до майстерності

Формувальне оцінювання (*Assessment for Learning*) — це моніторинг, який відбувається *під час* навчання і дає учню можливість покращити результат до моменту виставлення підсумкового балу.

- *Чек-листу (Checklists)*: Списки вимог, які вчитель видає *перед* початком роботи. Учень відмічає виконані пункти (напр., «1. Змінні оголошені. 2. Цикл працює. 3. Вивід даних оформлено»). Це вчить самоконтролю.
- *Рубрики (Assessment Rubrics)*: Таблиці, де розписано, що саме відповідає початковому, середньому, достатньому та високому рівням. Вони роблять оцінювання прозорим та позбавляють суб'єктивізму («чому в нього 10, а в мене 8?»).
- *Автоматизовані системи тестування*:
 - *Auto-graders (Stepik, CodeHS, LeetCode)*: Системи, що автоматично перевіряють код учня на тестах.
 - *Дидактична перевага*. Учень отримує фідбек миттєво, коли він ще «в темі», а не через тиждень, коли вчитель перевірів зошити.

Висновок. Сучасна діагностика в ІТ-освіті — це перехід від «полювання на помилки» до *аналізу зростання*. Оцінка має стимулювати учня виправляти код, оптимізувати алгоритми та прагнути до стандартів професійної розробки. Використання рубрик та автоматизації робить цей процес справедливим і технологічним.

Питання до самоперевірки

Блок 1: Види контролю

1. У чому полягає дидактична небезпека ігнорування *діагностичного контролю* на початку вивчення програмування?
2. Чому підсумковий контроль у формі «тесту на знання термінів» є менш ефективним, ніж «захист проєкту»?
3. Як поточний контроль допомагає вчителю реалізувати принцип *диференціації* на уроці?

Блок 2: Критерії

4. Чому за працюючу програму можна виставити низький бал (спираючись на критерій «якості коду»)?
5. Як оцінювати проєкт, якщо він виконаний групою з трьох учнів? Які критерії допоможуть виділити внесок кожного?
6. Сформулюйте критерій «креативності» для завдання «Створення анімації у Scratch».

Блок 3: Формувальне оцінювання

7. Як чек-лист допомагає розвивати в учня навичку *Debugging* (налагодження)?
8. У чому перевага «Рубрики» над звичайним оголошенням теми контрольної роботи?
9. Які обмеження мають *автоматизовані системи* перевірки коду? Чого вони не можуть перевірити на відміну від вчителя?

Лекція 10. Проектування та самоаналіз діяльності вчителя інформатики

1. Технологія підготовки вчителя до уроку

Підготовка до уроку інформатики — це процес педагогічного дизайну. Оскільки технології змінюються швидко, вчитель має проектувати не «застиглий» конспект, а динамічний сценарій.

- *Технологічна карта уроку.* Це сучасна форма планування, що візуалізує взаємодію вчителя та учнів.
 - Цілепокладання. Цілі мають бути сформульовані через очікувані результати учня (напр., «учень створює цикл», а не «ознайомити з циклом»).
 - Проектування засобів. Перевірка працездатності ПЗ, наявність інтернет-з'єднання, підготовка файлів-заготовок.
- *Складання плану-конспекту:*
 - Вступ. Створення «проблемної ситуації» (запуск інтересу).
 - Основна частина. Баланс між поясненням (теорія) та «кодуванням» (практика).
 - Завершення. Обов'язкова демонстрація результатів (презентація робіт).

2. Дидактична рефлексія та самоаналіз

Рефлексія — це здатність вчителя подивитися на свій урок очима дослідника. Без самоаналізу педагогічний досвід не перетворюється на майстерність.

- *Алгоритм педагогічного аналізу уроку:*
 - Чи досягнуто мети? (Скільки учнів реально виконали код без критичних помилок?).
 - Часовий менеджмент. Чи не зайняло пояснення занадто багато часу, залишивши учнів без практики?
 - Дидактична адаптація. Чи були завдання зрозумілими для найслабших учнів і чи було чим зайнятися найсильнішим (диференціація)?
 - Реакція на непередбачуване. Як я діяв, коли вимкнули світло, завис сервер або учень поставив питання, на яке я не знаю відповіді?

3. Шляхи підвищення професійної майстерності

Вчитель інформатики — це професія, що вимагає *Lifelong Learning* (навчання впродовж життя) у квадраті.

- *Формальна та неформальна освіта:*
 - Проходження сертифікації МОН.
 - Курси на платформах *Coursera*, *Udemy*, *EdEra* (опанування нових мов програмування та стеку технологій).

- *Участь у професійних спільнотах:*
 - Мережева взаємодія (групи в Telegram/Facebook, участь у конференціях).
 - Ведення власного професійного блогу або YouTube-каналу.
- *Педагогічне експериментування:*
 - Апробація нових методик (напр., спробувати «парне програмування» або «перевернутий клас»).
 - Вивчення можливостей *III* для автоматизації підготовки до уроків.

Висновок. Проектування уроку інформатики — це мистецтво поєднання *педагогічного розрахунку та технологічної гнучкості*. Майстерність вчителя полягає не в знанні всіх мов програмування світу, а у вмінні аналізувати свій досвід, визнавати помилки та постійно вчитися разом зі своїми учнями.

Питання до самоперевірки

Блок 1: Проектування

1. Чим *технологічна карта* відрізняється від традиційного плану уроку і які її переваги для вчителя інформатики?
2. Які специфічні ризики (технічні, дидактичні) має врахувати вчитель під час проектування практичного заняття?
3. Як правильно сформулювати «навчальну мету» уроку за *методикою SMART*?

Блок 2: Рефлексія

4. Чому «успішний» урок з точки зору тиші в класі може бути «провальним» з точки зору дидактики інформатики?
5. Назвіть 3 запитання, які вчитель має поставити собі відразу після виходу з кабінету.
6. Як залучити учнів до процесу аналізу уроку (рефлексія учня)?

Блок 3: Розвиток майстерності

7. Як ви розумієте вислів: «Вчитель інформатики має вчитися швидше за свої пристрої»?
8. Яку роль відіграє «*цифрове портфоліо*» вчителя у його професійному зростанні?
9. Як вчителю подолати «синдром самозванця», коли учні знають певну вузьку ІТ-технологію краще за нього?

ПИТАННЯ ДО ПІДСУМКОВОГО КОНТРОЛЮ

1. Дидактика як наука про навчання: предмет, завдання та її значення для вчителя інформатики.
2. Основні категорії дидактики: освіта, навчання, викладання, учіння, знання, вміння, навички.

3. Процес навчання як цілісна система: компоненти (цільовий, стимулюючо-мотиваційний, змістовий, операційно-діяльнісний, контрольно-регулювальний, оцінно-результативний).
4. Взаємозв'язок викладання та учіння в умовах комп'ютеризованого навчання.
5. Реалізація освітньої, виховної та розвивальної функцій навчання при вивченні мов програмування.
6. Психолого-педагогічні особливості пізнавальної діяльності учнів під час роботи з цифровим контентом.
7. Закономірності навчання: зовнішні (соціальні) та внутрішні (педагогічні).
8. Принцип науковості та його реалізація через актуалізацію стеку технологій у шкільній програмі.
9. Принцип систематичності та послідовності: від візуального програмування (Scratch) до текстового (Python/C++).
10. Принцип наочності: використання мультимедіа, динамічних моделей алгоритмів та скринкастів.
11. Принцип доступності: дидактична обробка складних понять (ООП, рекурсія, бази даних) для учнів середньої школи.
12. Принцип свідомості й активності: способи залучення учнів до активного кодування та розв'язання логічних задач.
13. Принцип міцності знань та особливості тренувальних вправ у програмуванні.
14. Поняття змісту освіти та чинники його формування в галузі «Інформатика».
15. Аналіз Державного стандарту базової та повної середньої освіти щодо інформатичної галузі.
16. Класифікація методів навчання за джерелом знань та їх застосування в комп'ютерному класі.
17. Пояснювально-ілюстративний та репродуктивний методи: межі використання при вивченні софту.
18. Проблемне навчання: створення проблемних ситуацій через «баги» у коді та пошук їх розв'язання.
19. Евристичний та дослідницький методи у проєктній діяльності учнів.
20. Метод проєктів як провідна технологія реалізації програми «Інформатика та програмування».
21. Урок як основна форма організації навчання: вимоги, структура, типологія.
22. Специфіка уроку-практикуму та комбінованого уроку в кабінеті інформатики.
23. Дидактичні вимоги до використання засобів ІКТ та хмарних сервісів (Google Classroom, GitHub).
24. Технологія змішаного навчання (Blended Learning): моделі «Перевернутий клас» та «Ротація станцій».

25. Позакласна робота з інформатики: хакатони, олімпіади з програмування, гуртки робототехніки.
26. Дистанційне навчання: дидактичні виклики та інструменти взаємодії.
27. Контроль у навчанні: види (попередній, поточний, тематичний, підсумковий) та їх специфіка в ІТ.
28. Методи та цифрові інструменти перевірки результатів навчання (автоматизоване тестування, Code Review).
29. Критерії оцінювання навчальних досягнень учнів з програмування (рівні: початковий, середній, достатній, високий).
30. Формувальне оцінювання: використання рефлексії та чек-листів для самоконтролю учнів.

ГЛОСАРІЙ ДО ТЕМАТИЧНОЇ ПРОГРАМИ

Тема 1.

- **Дидактика** — галузь педагогіки, що розробляє теорію освіти та навчання (принципи, методи, форми, зміст).
- **Цифрові аборигени (Digital Natives)** — покоління, що народилося в епоху високих технологій і сприймає цифрове середовище як природне.
- **Нова грамотність (New Literacy)** — сукупність навичок (зокрема програмування та медіаграмотності), необхідних для повноцінної життєдіяльності в інформаційному суспільстві.
- **Об'єкт дидактики** — навчання як особливий вид діяльності, спрямований на передачу та засвоєння соціокультурного досвіду.

Тема 2.

- **Викладання** — діяльність учителя з організації, стимулювання та управління пізнавальною діяльністю учнів.
- **Учіння** — активна пізнавальна діяльність учня, спрямована на засвоєння знань, умінь та навичок.
- **Дидактичний цикл** — сукупність етапів навчання від постановки мети до аналізу результатів.
- **Когнітивне навантаження** — обсяг зусиль, що витрачаються робочою пам'яттю для опрацювання нової інформації.

Тема 3.

- **Науковість** — принцип, що вимагає відповідності змісту освіти сучасним досягненням науки та технологій.
- **Трасування коду (Code Tracing)** — метод наочного покрокового відстеження виконання алгоритму для розуміння його логіки.
- **Зона найближчого розвитку (ЗНР)** — рівень завдань, які дитина ще не може виконати самостійно, але долає з допомогою вчителя.
- **Трансфер навичок** — здатність застосовувати логіку, вивчену в одному середовищі (напр., Scratch), у новому контексті (напр., Python).

Тема 4.

- **Державний стандарт** — нормативний документ, що визначає обов'язкові результати навчання в інформатичній галузі.
- **Інваріантна складова** — частина навчального плану, обов'язкова для всіх закладів освіти країни.
- **Електронний освітній ресурс (ЕОР)** — навчальний засіб на цифрових носіях (інтерактивні підручники, симулятори, відео).
- **Модельна програма** — концептуальний план курсу, на основі якого вчитель будує власну навчальну траєкторію.

Тема 5. Методи навчання

- **Евристична бесіда** — метод навчання, за якого вчитель через серію запитань підводить учня до самостійного відкриття істини.
- **Debugging (Налагодження)** — дидактичний метод пошуку та виправлення помилок, що формує критичне мислення.
- **Метод «Чорної скриньки»** — вивчення системи через аналіз вхідних та вихідних даних без доступу до внутрішнього коду.
- **Репродуктивний метод** — навчання через відтворення зразка (наприклад, копіювання коду з дошки для вивчення синтаксису).

Тема 6.

PBL (Project-Based Learning) — метод проєктів, де знання здобуваються під час створення реального продукту.

- **STEM-освіта** — інтегрований підхід, що поєднує природничі науки, технології, інженерію та математику.
- **Гейміфікація** — використання ігрових механік (XP, бейджі, лідерборди) для підвищення мотивації у навчанні.
- **Пітчинг** — коротка презентація проєкту з метою переконати аудиторію в його цінності.

Тема 7.

- **Перевернутий клас (Flipped Classroom)** — модель, де теорія вивчається вдома самостійно, а практика виконується в класі з учителем.
- **Ротація станцій** — модель змішаного навчання, де групи учнів змінюють види діяльності (онлайн, офлайн, проєкт) за графіком.
- **Універсальний дизайн навчання (UDL)** — підхід, що забезпечує доступність матеріалів для всіх учнів, включаючи дітей з ООП.
- **Хакатон** — форма інтенсивного командного змагання з розробки програмного рішення за обмежений час.

Тема 8.

- **LMS (Learning Management System)** — платформа для управління навчанням (напр., Google Classroom, Moodle).
- **Синхронне навчання** — взаємодія вчителя та учня в реальному часі (вебінар, відеочат).
- **Промпт-інжиніринг** — навичка формулювання точних запитів до Штучного Інтелекту для отримання якісного результату.
- **Адаптивна система** — ПЗ, що підлаштовує складність завдань під індивідуальний темп учня.

Тема 9. Діагностика та оцінювання

- **Формувальне оцінювання** — поточне оцінювання прогресу учня для корекції навчання, а не для підсумкового балу.
- **Рубрикатор (Матриця)** — таблиця з чіткими критеріями для оцінювання складних творчих або технічних робіт.
- **Чек-лист** — список дій або ознак, які учень має перевірити в роботі перед її здачею.
- **Автогрейдер** — система автоматичної перевірки програмного коду на відповідність тестам.

Тема 10. Проектування та самоаналіз

- **Технологічна карта** — графічний або табличний сценарій уроку, що деталізує дії вчителя та учнів на кожному етапі.
- **Дидактична рефлексія** — процес усвідомлення вчителем результативності своєї діяльності та пошук шляхів її покращення.
- **Цикл Гіббса** — структура рефлексії:
опис
почуття
оцінка
аналіз
висновок
план.
- **Lifelong Learning** — концепція безперервного навчання впродовж усього життя для підтримки професійної актуальності.

ФОРМИ ПІДСУМКОВОГО КОНТРОЛЮ УСПІШНОСТІ ТА КРИТЕРІЇ ОЦІНЮВАННЯ ЗДОБУВАЧІВ

Види контролю	Складові оцінювання	
	Для екзамену	Для заліку
поточний контроль , який здійснюється у ході: проведення практичних занять, виконання самостійної роботи, індивідуальних завдань, дослідна робота здобувача тощо.	50%	100%
підсумковий контроль	50%	

Методи діагностики знань (контролю)	фронтальне опитування; реферати, усне повідомлення, індивідуальне опитування; робота у групах; розв'язання практичних завдань, екзамен
--	--

КРИТЕРІЇ ОЦІНЮВАННЯ

**ОЦІНЮВАННЯ ЗДОБУВАЧІВ ВИЩОЇ ТА ФАХОВОЇ ПЕРДВИЩОЇ
ОСВІТИ ЗА НАВЧАЛЬНУ ДИСЦИПЛІНУ У МІЖНАРОДНОМУ
ГУМАНІТАРНОМУ УНІВЕРСИТЕТІ**

(поточний контроль, підсумковий контроль, загальний результат
оцінювання)

ЕКЗАМЕН

<i>Поточний контроль</i>			
Види роботи	Планові терміни виконання	Форми контролю та звітності	Максимальний відсоток оцінювання
Систематичність і активність роботи на практичних (лабораторних) заняттях			
1.1. Підготовка до практичних (лабораторних) занять	Відповідно до робочої програми та розкладу занять	Перевірка обсягу та якості засвоєного матеріалу під час практичних (лабораторних) занять	30
Виконання завдань для самостійного опрацювання			
1.2. Індивідуальне тестування, завдання, підготовка реферату (есе) за заданою тематикою	Відповідно до робочої програми тарозкладу занять	Розгляд відповідного матеріалу під час аудиторних занять або ІКР ¹ , перевірка конспектів навчальних текстів тощо	10
Виконання індивідуальних завдань, дослідна робота здобувача			
1.3. Інші види індивідуальних завдань, в т.ч. підготовка наукових публікацій, участь у роботі круглих столів, конференцій тощо.	Відповідно до робочої програми тарозкладу занять	Обговорення результатів проведеної роботи під час аудиторних занять, наукових конференцій та круглих столів.	10
Разом балів за поточний контроль			50
Підсумковий контроль екзамен			50
<i>Загальний результат оцінювання</i>			100

Поточний контроль знань здобувачів освіти при формі контролю **екзамен** (максимум 50 балів) оцінюється за шкалою («2», «3», «4», «5»), з обов'язковим переведенням балів за кожний вид роботи таким чином:

¹ Індивідуально-консультативна робота викладача зі здобувачами

- за підготовку до аудиторних занять (максимум 30 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості практичних (лабораторних) занять, для переводу балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 6;
- за виконання завдань для самостійного опрацювання (тестування, завдання, підготовка реферату (есе) за заданою тематикою) (максимум 10 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості програмного матеріалу, що виноситься на самостійне вивчення, для переводу зазначених балів середньоарифметична оцінка множиться на коефіцієнт 2;
- за інші види індивідуальних завдань, в т.ч. підготовку наукових публікацій, участь у роботі круглих столів, конференцій тощо (максимум 10- балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості інших видів індивідуальних завдань, для переводу балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 2.

КРИТЕРІЇ ОЦІНЮВАННЯ ЗА ПОТОЧНИЙ КОНТРОЛЬ

- «2»- виставляється за неправильну відповідь (письмову роботу), яка не відповідає змісту теми та свідчить про нерозуміння її основних положень;
- «3» -виставляється за відповідь (письмову роботу), яка відповідає змісту теми, але є неповною і неточною у визначенні понять, свідчить про неповне розуміння основних положень навчального матеріалу, свідчить про те, що знання студента мають фрагментарний, поверховий характер;
- «4» - оцінюється правильна і обґрунтована відповідь (письмова робота), з якої видно, що студент знає й розуміє теоретичний матеріал в обсязі навчальної теми, володіє необхідними навичками, не допускає суттєвих помилок.
- «5» - оцінюється повна, правильна, ґрунтовна відповідь (письмова робота) на запитання теми, що передбачає логічність і послідовність викладу матеріалу, володіння термінологією, показує вміння повно й глибоко використовувати теоретичні знання в практичній площині.

КРИТЕРІЇ ОЦІНЮВАННЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ ЗДОБУВАЧІВ

ЗА ЕКЗАМЕН (максимум 50 балів)

Рівень знань оцінюється:

- від 40 до 50 балів - студент виявляє особливі творчі здібності, вміє самостійно знаходити та опрацьовувати необхідну інформацію, демонструє знання матеріалу, проводить узагальнення і висновки;
- 30 до 40 балів - студент володіє знаннями матеріалу, але допускає незначні помилки у формуванні термінів, категорій, проте за допомогою викладача швидко орієнтується і знаходить правильні відповіді;
- 20 - 30 балів - студент відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, з допомогою викладача може аналізувати навчальний матеріал, але дає недостатньо обґрунтовані, невичерпні відповіді, допускає помилки.

- 10 - 20 балів - володіє навчальним матеріалом на середньому рівні, допускає помилки, серед яких є значна кількість суттєвих;
- 0 10 балів - володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні, на всі запитання дає необґрунтовані, невичерпні відповіді, допускає помилки.

КРИТЕРІЇ ОЦІНЮВАННЯ ЗА ПОТОЧНИЙ КОНТРОЛЬ

- «2»- виставляється за неправильну відповідь (письмову роботу), яка не відповідає змісту теми та свідчить про нерозуміння її основних положень;
- «3» -виставляється за відповідь (письмову роботу), яка відповідає змісту теми, але є неповною і неточною у визначенні понять, свідчить про неповне розуміння основних положень навчального матеріалу, свідчить про те, що знання студента мають фрагментарний, поверховий характер;
- «4» - оцінюється правильна і обґрунтована відповідь (письмова робота), з якої видно, що студент знає й розуміє теоретичний матеріал в обсязі навчальної теми, володіє необхідними навичками, не допускає суттєвих помилок.
- «5» - оцінюється повна, правильна, ґрунтовна відповідь (письмова робота) на запитання теми, що передбачає логічність і послідовність викладу матеріалу, володіння термінологією, показує вміння повно й глибоко використовувати теоретичні знання в практичній площині.

КРИТЕРІЇ ЗАГАЛЬНИХ РЕЗУЛЬТАТІВ ОЦІНЮВАННЯ ЗНАНЬ ЗДОБУВАЧІВ

Рівень знань оцінюється:

- «відмінно» / «зараховано» А - від 90 до 100 балів. Студент виявляє особливі творчі здібності, вміє самостійно знаходити та опрацьовувати необхідну інформацію, демонструє знання матеріалу, проводить узагальнення і висновки. Був присутній на лекціях та семінарських заняттях, під час яких давав вичерпні, обґрунтовані, теоретично і практично правильні відповіді, має конспект з виконаними завданнями до самостійної роботи, презентував реферат (есе) за заданою тематикою, проявляє активність і творчість у науково-дослідній роботі;
- «добре» / «зараховано» В - від 82 до 89 балів. Студент володіє знаннями матеріалу, але допускає незначні помилки у формуванні термінів, категорій, проте за допомогою викладача швидко орієнтується і знаходить правильні відповіді. Був присутній на лекціях та семінарських заняттях, має конспект з виконаними завданнями до самостійної роботи, презентував реферат (есе) за заданою тематикою, проявляє активність і творчість у науково-дослідній роботі;
- «добре» / «зараховано» С - від 74 до 81 балів. Студент відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, з допомогою викладача може аналізувати навчальний матеріал, але

дає недостатньо обґрунтовані, невичерпні відповіді, допускає помилки. При цьому враховується наявність конспекту з виконаними завданнями до самостійної роботи, реферату та активність у науково-дослідній роботі;

- «задовільно» / «зараховано» D - від 64 до 73 балів. Студент був присутній не на всіх лекціях та семінарських заняттях, володіє навчальним матеріалом на середньому рівні, допускає помилки, серед яких є значна кількість суттєвих. При цьому враховується наявність конспекту з виконаними завданнями до самостійної роботи, рефератів (есе);

- «задовільно» / «зараховано» E - від 60 до 63 балів. Студент був присутній не на всіх лекціях та семінарських заняттях, володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні, на всі запитання дає необґрунтовані, невичерпні відповіді, допускає помилки, має неповний конспект з завданнями до самостійної роботи.

- «незадовільно з можливістю повторного складання» / «не зараховано» FX – від 35 до 59 балів. Студент володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу.

- «незадовільно з обов'язковим повторним вивченням дисципліни» / «не зараховано» F – від 0 до 34 балів. Студент не володіє навчальним матеріалом.

ТАБЛИЦЯ ВІДПОВІДНОСТІ ЗАГАЛЬНИХ РЕЗУЛЬТАТІВ ОЦІНЮВАННЯ ЗНАНЬ ЗА РІЗНИМИ ШКАЛАМИ

100-бальною шкалою	Шкала за ECTS	За національною шкалою	
		екзамен	залік
90-100	A	Відмінно	Зараховано
82-89	B	Добре	
74-81	C		
64-73	D	Задовільно	
60-63	E		
35-59	Fx	Незадовільно	Не зараховано
1-34	F		

ПЕРЕЛІК РЕКОМЕНОВАНОЇ ЛІТЕРАТУРИ

Основна

1. Волкова Н. П. Педагогіка : навч. посіб. Київ : Академія, 2012. 616 с.
2. Дичківська І. М. Інноваційні педагогічні технології : навч. посіб. Київ : Академвидав, 2015. 304 с.
3. Морзе Н. В. Методика навчання інформатики : посібник : у 3 ч. Київ : Навчальна книга, 2003–2004.
4. Савченко О. Я. Дидактика початкової освіти : підручник. Київ : Грамота, 2018. 448 с.

5. Спірін О. М. Теоретичні та методичні засади навчання програмування майбутніх учителів інформатики. — Житомир: Вид-во ЖДУ ім. І. Франка, 2007.

6. Триус Ю. В. Комп'ютерно-орієнтовані методичні системи навчання математичних та інформатичних дисциплін. — Черкаси: Брама-Україна, 2005.

7. Фіцула М. М. Педагогіка : навч. посіб. Київ : Академвидав, 2022. 456 с.

8. Єрмакова С.С. Іванова О. С. Конспект лекцій з дисципліни «Основи дидактики» для здобувачів вищої освіти першого (бакалаврського) рівня освітньо-професійної програми «Інформатика та програмування», спеціальності А4.09 Середня освіта (Інформатика) денної та заочної форми навчання / С. С. Єрмакова , Іванова О. С.; [Електронне видання]. – О.: МГУ, 2025. – 94 с.

9. Єрмакова С.С. Іванова О. С. Навчально-методичні рекомендації до практичних занять та самостійної роботи з навчальної дисципліни «Основи дидактики» для здобувачів вищої освіти першого (бакалаврського) рівня освітньо-професійної програми «Інформатика та програмування», спеціальності А4.09 Середня освіта (Інформатика) денної та заочної форми навчання / С. С. Єрмакова , Іванова О. С.; [Електронне видання]. Одеса: МГУ, 2025. 45 с.

10. Єрмакова С.С. Іванова О. С. Конспект лекцій з дисципліни «Загальна педагогіка» для здобувачів вищої освіти першого (бакалаврського) рівня освітньо-професійної програми «Інформатика та програмування», спеціальності А4.09 Середня освіта (Інформатика) денної та заочної форми навчання / С. С. Єрмакова , Іванова О. С.; [Електронне видання]. – О.: МГУ, 2025. – 104 с.

11. Єрмакова С.С. Іванова О. С. Навчально-методичні рекомендації до практичних занять та самостійної роботи з навчальної дисципліни «Загальна педагогіка» для здобувачів вищої освіти першого (бакалаврського) рівня освітньо-професійної програми «Інформатика та програмування», спеціальності А4.09 Середня освіта (Інформатика) денної та заочної форми навчання / С. С. Єрмакова , Іванова О. С.; [Електронне видання]. Одеса: МГУ, 2025. 41 с.

Допоміжна

1. Биков В. Ю. Моделі організаційних систем відкритої освіти: Монографія. Київ: Атіка, 2008.

2. Гризун Л. Е. Дидактичні основи проектування сучасних комп'ютерних навчальних програм. Харків: ХНПУ, 2014.

3. Дичківська І. М. Інноваційні педагогічні технології. — К.: Академвидав, 2015.

4. Морзе Н. В., Барна О. В., Вембер В. П. Формувальне оцінювання: від теорії до практики. Київ: Морзе, 2021.

5. Семеріков С. О. Фундаменталізація навчання інформатичних дисциплін у вищій школі: Монографія. Кривий Ріг: Мінерал, 2009.

Нормативно-правові акти:

1. Закон України Про повну загальну середню освіту. (Зі змінами, внесеними згідно з Законом № 4265-IX від 15.06.2025). Офіційний портал Верховної Ради України. Режим доступу: <https://zakon.rada.gov.ua/laws/show/463-20#Text>.
2. Державний стандарт базової середньої освіти (Постанова КМУ № 898 від 30.09.2020 року). Офіційний сайт Міністерства освіти і науки України. Режим доступу: <https://mon.gov.ua/osvita-2/zagalna-serednya-osvita/nova-ukrainska-shkola-2/derzhavniy-standart-bazovoi-serednoi-osviti>.
3. Типові освітні програми. Офіційний сайт Міністерства освіти і науки України. Режим доступу: <https://mon.gov.ua/osvita-2/zagalna-serednya-osvita/osvitni-programi/tipovi-osvitni-programi-2>.
4. Навчальні програми на основі модельних. Офіційний сайт Міністерства освіти і науки України. Режим доступу: <https://mon.gov.ua/osvita-2/zagalna-serednya-osvita/osvitni-programi/navchalni-prohramy-na-osnovi-modelnykh>.
5. Інформатика в школі : наук.-метод. журн. Харків : Видавнича група «Основа». URL: journal.osnova.com.ua
6. Інформаційні технології і засоби навчання : електрон. наук. фахове вид. / Ін-т цифровізації освіти НАПН України. URL: <https://journal.iitta.gov.ua>
7. Нова українська школа : сайт. URL: <https://nus.org.ua>
8. Coursera : освітня онлайн-платформа. URL: <https://www.coursera.org>
9. edX : освітня онлайн-платформа. URL: <https://www.edx.org>

Інформаційні ресурси

1. Бібліотека Міжнародного гуманітарного університету: офіційний сайт. Режим доступу: https://mgu.edu.ua/science_library
2. Бібліотека Університету Ушинського : офіційний сайт. Режим доступу: <https://library.pdpu.edu.ua>
3. Міністерство освіти і науки України: офіційний сайт. Режим доступу: <http://www.mon.gov.ua>
4. Національна бібліотека України імені В.І.Вернадського: офіційний сайт. URL: <http://www.nbuv.gov.ua/>
5. Одеська національна наукова бібліотека: офіційний сайт. Режим доступу: <http://odnb.odessa.ua/>