

**Міністерство освіти і науки України
Тернопільський національний педагогічний університет
імені Володимира Гнатюка**

**Фізико-математичний факультет
Кафедра інформатики та методики її навчання**

**Кваліфікаційна робота
ВИКОРИСТАННЯ ЕЛЕКТРОННОЇ ОСВІТНЬОЇ СИСТЕМИ «МІЙ
КЛАС» ДЛЯ ВИВЧЕННЯ МОВИ ПРОГРАМУВАННЯ PYTHON В
ШКІЛЬНОМУ КУРСІ ІНФОРМАТИКИ**

Спеціальність 014 Середня освіта

**Освітня програма
«Середня освіта (Інформатика, математика, STEM-освіта)»**

**Здобувача другого (магістерського)
рівня вищої освіти
Лесика Тараса Ігоровича**

**НАУКОВИЙ КЕРІВНИК:
доктор педагогічних наук, професор
Романишина Оксана Ярославівна**

**РЕЦЕНЗЕНТ:
доктор педагогічних наук,
професор кафедри комп'ютерних
технологій Тернопільського
національного педагогічного
університету ім.В.Гнатюка
Потапчук Ольга Ігорівна**

Тернопіль – 2025

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1	7
ТЕОРЕТИЧНІ ЗАСАДИ ВИКОРИСТАННЯ ЕЛЕКТРОННИХ ОСВІТНІХ СИСТЕМ У НАВЧАННІ ІНФОРМАТИКИ.....	7
1.1. Аналіз сучасного стану викладання мови програмування Python у шкільному курсі інформатики.....	7
1.2. Теоретичні аспекти застосування електронних освітніх систем на прикладі «МійКлас» у навчальному процесі.....	12
1.3. Дидактичні можливості та особливості платформи «МійКлас» для вивчення програмування.....	19
Висновки до першого розділу	22
РОЗДІЛ 2	25
МЕТОДИЧНІ ОСОБЛИВОСТІ ВИКОРИСТАННЯ СИСТЕМИ «МІЙКЛАС» ПРИ ВИВЧЕННІ PYTHON.....	25
2.1. Методичні засади інтеграції платформи «МійКлас» у процес навчання програмування Python.....	25
2.2. Проєктування наповнення навчального курсу «Програмування мовою Python» в системі «МійКлас»	33
2.3. Організація та проведення педагогічного експерименту з використанням ЕОС «МійКлас».	40
2.4. Аналіз результатів педагогічного експерименту та визначення ефективності запропонованої методики.....	47
Висновки до другого розділу	54
ВИСНОВКИ	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
ДОДАТКИ	61

АНОТАЦІЯ

Лесик Т.І. Використання електронної освітньої системи «МійКлас» для вивчення мови програмування Python в шкільному курсі інформатики.

Наукова робота на здобуття кваліфікації магістра зі спеціальності 014 Середня освіта. Тернопільський національний педагогічний університет імені Володимира Гнатюка, 2025. 64 с.

Наукову роботу присвячено дослідженню методики застосування ЕОС «МійКлас» для навчання мови Python учнів 7-х класів. Проаналізовано стан викладання та дидактичний потенціал платформи, обґрунтовано її ефективність для подолання типових труднощів. Розроблено авторський 35-годинний курс з шести модулів, орієнтований на ООП, Tkinter та проєктну діяльність. Експериментальна перевірка з використанням U-критерію Манна-Уїтні підтвердила статистично значуще підвищення теоретичних знань, практичних навичок та мотивації в експериментальній групі. Надано методичні рекомендації щодо впровадження курсу.

Ключові слова: методика навчання, електронна освітня система, «МійКлас», Python, об'єктно-орієнтоване програмування, середня освіта, педагогічний експеримент, диференційоване навчання, Tkinter, мотивація учнів.

SUMMARY

Lesyk T.I. Using the electronic educational system «MyClass» for studying the Python programming language in a school computer science course.

Scientific work for obtaining a master's degree in the specialty 014 Secondary Education. Volodymyr Hnatyuk Ternopil National Pedagogical University, 2025. 64p.

The scientific work is devoted to the study of the methodology for using the EOS «MyClass» for teaching Python to 7th grade students. The state of teaching and the didactic potential of the platform are analyzed, and its effectiveness for overcoming typical difficulties is substantiated. An author's 35-hour course of six modules focused on OOP, Tkinter and project activities is developed. Experimental testing using the Mann-Whitney U-criterion confirmed a statistically significant increase in theoretical knowledge, practical skills and motivation in the experimental group. Methodological recommendations for the implementation of the course are provided.

Keywords: teaching methodology, electronic educational system, «MyClass», Python, object-oriented programming, secondary education, pedagogical experiment, differentiated instruction, Tkinter, student motivation.

ВСТУП

Сучасна система освіти перебуває на етапі глибокої трансформації, зумовленої потребою у формуванні в учнів ключових компетентностей XXI століття, серед яких особливе місце посідає цифрова грамотність та алгоритмічне мислення. У цьому контексті програмування визнається однією з фундаментальних навичок, а мова Python, завдяки своїй простоті, читабельності та широкому застосуванню в науці, аналізі даних та штучному інтелекті, стала одним з пріоритетних інструментів для впровадження в шкільну програму з інформатики.

Численні українські науковці досліджували використання електронних платформ для навчання програмування школярів, висвітлюючи як можливості, так і проблеми, пов'язані з їх впровадженням. Так, роботи Н. Морзе зосереджені на розробці методологічних основ для інтеграції інформаційно-комунікаційних технологій (ІКТ) у педагогічну освіту та впровадження штучного інтелекту для підтримки навчання в класі [23]. Т. Вакалюк у своїй роботі зосереджується на використанні хмарних сервісів, таких як Google Workspace, онлайн-компіляторів та автоматизованих систем оцінювання [11]. Праця академіка М. Жалдака базується на підготовці вчителів до ефективної інтеграції ІКТ у навчання, задля використання електронних платформ та ресурсів у навчанні програмування у закладах загальної середньої освіти [4].

Модернізація освітнього процесу в Україні, зокрема у контексті впровадження стандартів Нової української школи (НУШ), висуває на перший план необхідність формування ключових компетентностей у сфері інформаційно-комунікаційних технологій. Для учнів 7-х класів програма зосереджується на розвитку обчислювального мислення та практичному оволодінні мовами програмування, серед яких Python займає центральне місце. Традиційна методика навчання інформатики, яка часто передбачає лекційні форми подачі матеріалу та можливість миттєвого зворотного зв'язку, нерідко виявляється недостатньо адаптивною для забезпечення високого рівня самостійності та ефективного формування навичок налагодження коду. Це

створює методологічну прогалину, яку найдоцільніше заповнити за допомогою інтегрованого використання сучасних електронних освітніх систем (ЕОС). Однією із наймасовіших та доступних серед них є «МійКлас», що володіє значним потенціалом для подолання зазначених вище викликів. Проте питання цілеспрямованого та методично обґрунтованого застосування платформи саме для вивчення мови програмування Python залишається недостатньо дослідженим.

Актуальність даного дослідження полягає у необхідності розробки методики навчання програмуванню мовою Python, яка забезпечувала б достатній рівень розвитку знань та умінь учнів на рівні стандартів НУШ. Особливе значення має розробка таких навчальних підходів, що забезпечуватимуть індивідуалізацію навчання, підвищення мотивації учнів та ефективності контролю результатів.

Мета дослідження: теоретично обґрунтувати, розробити та експериментально перевірити методику використання електронної освітньої системи «МійКлас» для підвищення ефективності вивчення мови програмування Python у шкільному курсі інформатики.

Для досягнення мети були поставлені наступні **завдання**:

1. Проаналізувати сучасний стан викладання мови програмування Python у шкільному курсі інформатики.
2. Визначити дидактичний потенціал та особливості платформи «Мій Клас» для вивчення програмування.
3. Спроекувати та розробити методичні рекомендації щодо впровадження системи «Мій Клас» у практику викладання Python.
4. Експериментально перевірити ефективність розробленої методики та здійснити аналіз результатів дослідження.

Об'єкт дослідження: освітній процес підготовки старшокласників з основ програмування на мові Python в закладах загальної середньої освіти.

Предмет дослідження: методика використання функціональних можливостей електронної освітньої системи «МійКлас» як засобу підвищення

ефективності навчання мови програмування Python у шкільному курсі інформатики.

Наукова новизна кваліфікаційної роботи: отримала подальший розвиток методика навчання учнів старших класів програмуванню мовою Python, яка передбачає інтеграцію об'єктно-орієнтованого підходу та проєктної діяльності за допомогою електронної освітньої системи «МійКлас».

Практичне значення одержаних результатів полягає у розробленні та впровадженні в освітній процес закладів загальної середньої освіти методики використання комплексу навчально-методичних матеріалів для забезпечення ефективності процесу формування практичних навичок учнів з програмування мовою Python. Цей комплекс включає 35-годинну програму з 6 модулів, орієнтовану на формування об'єктно-орієнтованого підходу та структуровану під вимоги НУШ. Розроблена методика може бути використана вчителями для викладання поглибленого курсу програмування у якості факультативу або у якості допоміжного матеріалу при вивченні шкільного курсу інформатики.

Методи дослідження. Для розв'язання поставлених завдань використано комплекс загальнонаукових і спеціальних методів. На теоретичному етапі було застосовано методи аналізу, порівняння та узагальнення наукових джерел з педагогіки, інформатики та методики навчання програмування для визначення сучасного стану викладання мови Python та потенціалу електронних освітніх систем. Емпіричну частину дослідження проведено методом педагогічного експерименту (констатувального і формувального) з використанням анкетування (для оцінки мотиваційного компоненту), тестування (для перевірки теоретичних знань), практичних завдань з програмування (для оцінки навичок) та спостереження за навчальною діяльністю учнів. Для діагностики рівня засвоєння знань та сформованості практичних умінь було розроблено спеціальні інструменти (тестові завдання, практичну роботу «Калькулятор середнього балу»). Отримані кількісні дані оброблено методами математичної статистики, зокрема з використанням U-критерію Манна-Уїтні для порівняння результатів експериментальної та контрольної груп та визначення розміру ефекту.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ЗАСАДИ ВИКОРИСТАННЯ ЕЛЕКТРОННИХ ОСВІТНІХ СИСТЕМ У НАВЧАННІ ІНФОРМАТИКИ

1.1. Аналіз сучасного стану викладання мови програмування Python у шкільному курсі інформатики

У сучасному суспільстві, що стрімко змінюється під впливом цифрових технологій, формування навичок програмування стає однією з ключових складових інформаційної грамотності. Програмування в закладах загальної середньої освіти вже не розглядається як вузькоспеціалізований напрям, доступний лише окремим учням, — воно дедалі більше інтегрується в базову освітню програму як засіб розвитку логічного мислення, алгоритмічної культури та загальних навичок розв’язання проблем.

Динамічний розвиток ринку праці та інформаційних технологій в цілому забезпечує зростання вакансій для кваліфікованих фахівців, що володіють сучасними мовами програмування та можуть широко застосовувати їх у поставлених задачах. Відповідно до цього, актуальним стає і вивчення програмування у навчальних закладах, що надає достатню базу знань для формування світогляду та подальшого розвитку учнів у цій сфері.

У контексті таких змін постає потреба не лише у вдосконаленні змісту навчання, а й у переосмисленні підходів до організації освітнього процесу. Важливо забезпечити учням доступне, мотивуюче та практично орієнтоване середовище для вивчення програмування. Одним із засобів, що відповідає цим вимогам, є використання цифрових освітніх платформ, які підтримують як традиційні, так і інноваційні педагогічні стратегії. Багато українських та закордонних науковців досліджували це питання у своїх роботах та публікаціях, серед них найбільший внесок зробили Т. Вакалюк, М. Жалдак, У. Когут, Н. Морзе, І. Мінтій, О. Руденко, О. Семеніхіна, О. Струк, О. Ящик та інші. Розглянемо дослідження та різноманіття варіацій навчання програмування на прикладі.

Так, О. Ящик виділяє важливість компетентнісного підходу як основи сучасного навчання програмування, обґрунтовуючи це тим, що при подібних умовах відбувається перехід від засвоєння окремих знань і навичок до формування цілісної системи ключових, базових та спеціальних компетентностей. Знання подаються не як основна мета навчання, а як спосіб для реального застосування у практичній діяльності з урахуванням різноманіття лабораторних та робочих задач. Так, навчання об'єктно-орієнтованому програмуванню у межах цього підходу має орієнтуватися на розвиток когнітивних, особистісних та соціальних якостей, а не лише технічних умінь, тож потребує не лише змістовної підготовки, а й уміння формувати індивідуальні освітні траєкторії для учнів. Для цього вчитель має володіти відповідними професійними, методичними та інформаційними компетенціями [21].

Науковець О. Струк у своїх публікаціях приділяє увагу комплексному підходу, який забезпечує цілісність знань та формує системне мислення щодо об'єкта або проблеми, сприяє міцним міжпредметним зв'язкам та ширшому контексту навчання. Подібний напрям застосовується вчителем в умовах дистанційного навчання для підвищення мотивації здобувачів до самостійної роботи, а саме вивчення базових елементів мови, алгоритмічних структур, типів даних, об'єктно-орієнтованого програмування та інших важливих концепцій [18].

Крім вищеописаних, до використання та комбінування в освітньому процесі, будуть доречними і інші основні педагогічні підходи. Одним із поширених для навчання програмуванню є конструктивістський підхід. Він ґрунтується на освітній теорії, згідно з якою учні конструюють знання через власний досвід та активну взаємодію з концепціями, а не пасивно отримуючи інформацію від вчителя. У контексті освіти з програмування цей підхід наголошує на дослідженні, експериментуванні та вирішенні проблем. Школярів заохочують відкривати для себе принципи кодування, створюючи та тестуючи власні програми, а не запам'ятовуючи синтаксис чи дотримуючись покрокових інструкцій. Вчитель виступає в ролі посередника, керуючи учнями в процесі

навчання, ставлячи відкриті проблеми, заохочуючи співпрацю та сприяючи рефлексії. Завдання програмування часто представляють як реальні виклики або проєкти, що дозволяє учням контекстуалізувати своє навчання та розвивати глибше розуміння обчислювального мислення, алгоритмів та логічних структур. Цей підхід сприяє розвитку когнітивних навичок вищого порядку, таких як абстракція, стратегії налагодження та здатність аналізувати та вдосконалювати власні рішення. Він особливо ефективний у навчальних середовищах, що базуються на проєктах, і сумісний з використанням цифрових інструментів, що підтримують ітеративний дизайн та експериментування.

Проектний підхід – це тип навчання, який використовує практичний метод до розуміння нових концепцій [25]. Учні створюють проєкти, використовуючи мінімальну кількість навчальних посібників, спираючись на власні наявні знання, здатність вирішувати нові проблеми, засвоювати матеріал в процесі роботи. Цей метод підходить для практичної перевірки знань учнів, надаючи їм можливість самостійно обирати ідею чи проблему, що буде вирішуватися, вибір інструментів (наприклад, мова програмування, методи і середовище) для виконання проєкту на рівні власної креативності та досвіду. Розробка проєктів гарантує свободу творчості, поглиблює уявлення про фреймворки та технології, які використовуються у розробці, розширює вміння працювати з помилками та виправляти їх, а також розвиває навички роботи у команді (у випадку, якщо проєкт груповий) [8]. Проектне навчання спонукає до проведення досліджень, читання документації, запису ідей, пошуку нових помилок, а потім їх вирішення. І хоча підхід відповідає сучасним освітнім стандартам Нової Української Школи (НУШ) та Європейським рамкам компетентностей, здобувач освіти нерідко може зіткнутися зі складнощами реалізації проєкту без відповідної матеріальної та методичної бази і здатності використовувати програмування для вирішення практичних задач.

Крім того, існує також гейміфікований підхід, який передбачає застосування в неігрових навчальних середовищах елементів ігрового дизайну, таких як бали, рівні, значки, таблиці лідерів та винагороди, для підвищення

мотивації та залученості учнів. Використання цього підходу перетворює процес навчання на інтерактивний та корисний досвід, особливо для молодших або менш впевнених учнів. Важливо, що гейміфікація також сприяє поступовому навчанню, коли учні просуваються через дедалі складніші завдання у власному темпі. Вона також може сприяти відчуттю досягнення та конкуренції, що мотивує учнів покращувати свою успішність, а отже є найефективнішим у поєднанні з чіткими педагогічними цілями та узгоджується зі стандартами навчальної програми [5].

Різноманіття підходів не обмежується вищезазначеним переліком, змінюється та удосконалюється із плином часу та розвитком інформаційних технологій. Кожен із них має свої цілі, методи та дидактичні переваги, тому часто їх поєднують та комбінують залежно від віку учнів, змісту курсу, обраної мови програмування та освітнього середовища. У нашому дослідженні йде мова про програмування Python, що набуло популярності у навчальних курсах завдяки низці вагомих переваг. Синтаксис Python зрозумілий та зрозумілий, дуже схожий на англійську мову. Ця простота полегшує початківцям розуміння концепцій програмування, не заглиблюючись у складні синтаксичні правила. Наприклад, Python використовує відступи для визначення блоків коду, усуваючи потребу в дужках або ключових словах, таких як «end», які є поширеними в інших мовах програмування. Ця читабельність дозволяє учням зосередитися на вивченні логічних структур та алгоритмів, а не заплутуватися в тонкощах самої мови.

Python — це мова загального призначення, яка підтримує різноманітні парадигми програмування, включаючи процедурне, об'єктно-орієнтоване та функціональне програмування. Ця універсальність означає, що Python можна використовувати для широкого кола застосувань, від веб-розробки та аналізу даних до штучного інтелекту та автоматизації. Як результат, школярі можуть застосовувати свої навички Python до численних реальних проблем, покращуючи свій навчальний досвід та роблячи мову актуальною для різних галузей дослідження. Враховуючи різноманітні педагогічні підходи та вимоги сучасної освіти, вибір Python як основної мови програмування у школі є цілком

закономірним і ґрунтованим. По-перше, Python ідеально відповідає компетентнісному підходу, оскільки дозволяє учням не просто засвоювати синтаксичні правила, а відразу застосовувати знання для вирішення практичних завдань. Це сприяє формуванню цілісної системи компетентностей — від базових логічних навичок до спеціалізованих умінь у сфері обчислювального мислення. По-друге, Python підтримує конструктивістську модель навчання: завдяки інтуїтивно зрозумілому синтаксису та можливості швидкого прототипування учні можуть експериментувати, створювати власні проєкти та відкривати принципи програмування через досвід, а не пасивне сприйняття інформації.

Додатковими перевагами Python, які роблять його особливо привабливим для шкільної освіти, є:

- велика кількість навчальних ресурсів та спільнот. Python має безкоштовні документації, тисячі підручників, відеокурсів та онлайн-середовищ для практики (наприклад, Trinket, Replit), що робить його доступним для учнів і вчителів навіть з обмеженими технічними можливостями;

- потужні бібліотеки для різних галузей. Завдяки бібліотекам, таким як Matplotlib (візуалізація), Pandas (аналіз даних), Turtle (графіка для початківців) або Django (веб-розробка), Python дозволяє інтегрувати програмування з іншими предметами – математикою, природничими науками, мистецтвом, що відповідає вимогам міжпредметних зв'язків;

- висока популярність у реальному секторі. Python широко використовується в наукових дослідженнях, штучному інтелекті, автоматизації та стартапах, що надає учням мотивацію вивчати мову, яка має практичну цінність у майбутній кар'єрі;

- підтримка різних стилів навчання. Python добре поєднується з гейміфікацією (наприклад, через платформи типу CodeCombat), проєктним навчанням (створення ігор, сайтів, аналітичних звітів) та дистанційною освітою, завдяки інтерактивним середовищам виконання коду.

Отже, вивчення Python у школі є результатом сукупної дії педагогічних, технічних та соціальних чинників. Він не лише відповідає сучасним освітнім стандартам (зокрема, концепції Нової української школи), але й забезпечує гнучкість у навчанні, сприяючи розвитку алгоритмічного мислення, творчості та готовності учнів до життя в цифровому суспільстві. Тому Python можна вважати оптимальним інструментом для формування основ програмування.

1.2. Теоретичні аспекти застосування електронних освітніх систем на прикладі «МійКлас» у навчальному процесі

Завдяки розвитку інтернет-технологій, штучного інтелекту та появи різноманітних онлайн-інструментів, організація освітнього процесу стала більш комплексною та доступною як для учнів, так і для вчителів. Для забезпечення ефективного та дієвого освітнього процесу для запланованого курсу та обраних під нього підходів та методів навчання, необхідно залучити новітні інтернет-технології у вигляді онлайн-інструментів для організації дистанційного навчання [2]. Починаючи зі спалаху COVID-19 у 2019 році та після початку повномасштабної війни в Україні, дистанційні платформи для навчання здобули особливого поширення та вдосконалення: тепер учні можуть знайти різноманітні курси, тренінги та майстер-класи для отримання нових знань та навичок.

Поява онлайн-інструментів відкриває численні можливості для створення інтерактивних уроків навіть у денному форматі навчання, що надає додаткову наочність та методи зацікавлення учнів до навчання. Ці інструменти дозволяють вчителям створювати захопливі та змістовні уроки, використовуючи різні методи тестування, відеоматеріали та онлайн-дошки для взаємодії зі здобувачами освіти і надаючи новітні способи перевірити знання та навички школярів.

Найпростішою та найпопулярнішою веб-платформою для навчання є Google Classroom, розроблена Google, де вчителі можуть проводити онлайн-заняття, створювати навчальні програми, призначати тестування та ділитися завданнями з учнями без паперового використання. Платформа спрощує

співпрацю між вчителем та учнем, використовуючи різні сервіси, такі як Google Docs, Sheets, Slides, Forms та Meet [1]. Як перевірений засіб для навчання, Google Classroom має ряд переваг:

- Завдання. Вчителі можуть створювати завдання, використовуючи навчальні матеріали, такі як відео на Youtube, опитування Google Forms або інтерактивні PDF-файли з диска. Вони можуть призначати завдання всьому класу або окремим учням. Вчителі можуть або одразу опублікувати завдання, або запланувати його на певний день та час.

- Налаштовувана система оцінювання. Вчителі можуть вибрати систему оцінювання та створити категорії оцінок. Якщо вчителі хочуть, щоб загальні оцінки були доступні для учнів, вони можуть вибрати будь-яку з наступних систем:

- Оцінювання за загальною кількістю балів. Загальна кількість балів, які отримують учні, ділиться на максимально можливу кількість балів.

- Оцінювання за категоріями. Категоріям оцінок присвоюється вага. Потім обчислюються середні бали кожної категорії оцінок і множаться на вагу оцінки, щоб отримати загальну оцінку зі 100%.

- Віртуальні дискусії. Вчителі можуть запрошувати учнів відповідати на обговорення, що базуються на питаннях, та відповідати однокласникам. Коментарі в Google Docs дозволяють вести двостороннє обговорення, оскільки вчителі можуть надавати учням зворотний зв'язок; це чудовий спосіб підтримувати зацікавленість учнів. Вчителі можуть керувати обговореннями, за потреби блокуючи публікації чи коментарі для окремих учнів.

- Оголошення. Вчителі можуть надавати учням інформацію через оголошення. Оголошення – це публікації без завдань, повідомлення для учнів про терміни, тести чи будь-яку класну роботу. Вчителі можуть планувати оголошення та контролювати відповіді й коментарі, зроблені до повідомлення.

- Заняття в прямому ефірі. Це одна з найзатребуваніших функцій, яка дозволяє вчителям проводити заняття в реальному часі віртуально через Google Meet. Вчителі можуть додавати до 250 осіб до дзвінка, і близько 100 000 глядачів

можуть переглядати трансляцію в прямому ефірі. Ці зустрічі чи лекції також можна записувати, щоб учні, які пропустили сеанс у прямому ефірі, могли переглянути їх пізніше [24].

Як зазначено, Google Classroom пропонує надійну платформу для покращення процесу викладання та навчання. Використовуючи безперебійну інтеграцію інструментів Google Workspace, він дозволяє вчителям сприяти залученню та активному навчанню учнів. Для максимальної участі учнів та оптимізації управління класом на цій платформі можна реалізовувати віртуальні дискусії, інтерактивні тести, створювати аудиторні блоги, покращувати співпрацю групи над проєктами та експериментувати з гнучкістю подання завдань.

Другою за популярністю використання у закладах середньої освіти є Moodle – навчальна платформа, яку також називають системою управління навчанням (LMS) [19]. Завдяки цій платформі користувачі можуть керувати, проводити та контролювати онлайн-навчання та розвиток з однієї безпечної та інтегрованої системи. Вони також мають можливість створювати персоналізовані навчальні середовища. Окрім безпосереднього проведення онлайн-курсів, Moodle можна використовувати кількома способами для задоволення різних потреб та вимог у навчанні, створення персоналізованих навчальних середовищ шляхом вибору найкращого контенту для електронного навчання та його надання учням. Основними перевагами LMS Moodle можна назвати:

- Створення та управління курсами. Moodle має зручний інтерфейс для створення та організації курсів. Вчителі можуть легко структурувати зміст курсу, завантажувати ресурси та створювати інтерактивні завдання, такі як вікторини та форуми. LMS також підтримує інтеграцію мультимедіа, що дозволяє додавати відео, аудіофайли та інтерактивні елементи для збагачення навчальних матеріалів.

- Реєстрація та управління користувачами. Moodle дозволяє адміністраторам ефективно керувати реєстрацією користувачів. Учні можуть

самостійно реєструватися або додаватися вручну, а доступ до курсу може бути обмежений на основі критеріїв реєстрації. Адміністратори можуть призначати різні ролі користувачам – вчитель, школяр або запрошений лектор, з різними дозволами та рівнями доступу. Ця гнучкість дозволяє персоналізувати навчальний досвід та адаптувати доставку контенту.

- Комунікація та співпраця. Moodle пропонує різні інструменти комунікації та співпраці для сприяння залученню та взаємодії між учасниками. Форуми для обговорення дозволяють учням ставити запитання, брати участь у групових обговореннях та обмінюватися ресурсами. Функції обміну повідомленнями дозволяють здійснювати пряме спілкування між школярами, вчителями та адміністраторами. Крім того, Moodle підтримує співпрацю в режимі реального часу через плагіни веб-конференцій, заохочуючи віртуальні класні сесії та живу взаємодію.

- Оцінювання та зворотний зв'язок. Оцінювання відіграє життєво важливу роль у вимірюванні прогресу здобувачів освіти, і Moodle пропонує низку варіантів оцінювання. Викладачі можуть створювати вікторини, завдання та інтеракції для оцінки розуміння учнями. LMS підтримує автоматичне оцінювання, надаючи учням миттєвий зворотний зв'язок. Moodle також дозволяє взаємне оцінювання, що дозволяє школярам оцінювати та надавати зворотний зв'язок щодо роботи один одного, сприяючи спільному навчанню.

- Відстеження прогресу та звітність. Moodle пропонує комплексні функції відстеження та звітності для моніторингу прогресу учнів. Вчителі можуть переглядати рівень участі, статуси завершення та оцінки для окремих школярів або цілих класів. Ці дані допомагають вчителям виявляти осіб, які мають труднощі, відстежувати загальну успішність курсу та надавати цілеспрямовану підтримку, де це необхідно. Адміністратори можуть створювати звіти за різними показниками для оцінки ефективності курсів та приймати рішення на основі даних.

- Налаштування та розширюваність. Moodle має високі можливості налаштування, що дозволяє установам адаптувати LMS до своїх конкретних

потреб. Він підтримує теми та шаблони для брендингу та естетичного налаштування. Крім того, архітектура плагінів Moodle дозволяє інтеграцію сторонніх плагінів для розширення функціональності, таких як додавання інтерактивного контенту, віртуальних класів або розширеної аналітики [22].

Система управління навчанням Moodle надає надійну платформу для створення, управління та проведення онлайн-курсів. Її інтуїтивно зрозумілий інтерфейс, інструменти комунікації, можливості оцінювання, відстеження прогресу та опції налаштування сприяють захопливому та персоналізованому навчанню, хоч платформа і потребує більшого часу для наповнення та впровадження в освітній процес, особливу увагу треба приділити навчанню її використання учнями. Використовуючи функції Moodle, навчальні заклади можуть покращити свої ініціативи онлайн-освіти, сприяти співпраці та надавати учням можливості досягати своїх академічних або професійних цілей [16].

Великого попиту серед закладів загальної середньої освіти здобула електронна інформаційно-освітня система «МійКлас», розроблена українцем Олександром Єр'оміним за підтримки компанії «Uzdevumi», на основі адаптації латвійської освітньої платформи «Uzdevumi.lv» [20]. Платформа пропонує низку функцій, розроблених для покращення освітнього процесу для учнів та оптимізації навчального процесу для вчителів. Ключові переваги включають комплексне узгодження навчальних програм та різні моделі підписки для задоволення різноманітних потреб користувачів. «МійКлас» прагне оцифрувати освіту, покращити доступність навчальних ресурсів та підвищити ефективність викладання, орієнтуючись на учнів широкого діапазону рівнів освіти та предметів. Подвійна спрямованість платформи передбачає універсальне та потенційно надійне освітнє рішення, яке можна адаптувати до різних національних освітніх систем.

Основна місія «МійКлас» зосереджена на трьох фундаментальних принципах: цифровізації, доступності та ефективності. Платформа прагне допомогти вчителям інтегрувати цифрові інструменти у свої методики викладання, тим самим модернізуючи процес навчання. Ключовою метою є

забезпечення доступності якісних цифрових навчальних ресурсів для кожного, хто має підключення до Інтернету, що усуває географічні та соціально-економічні бар'єри для освіти. Крім того, «МійКлас» прагне надати учням можливість самостійно вдосконалювати свої знання в будь-який час і з будь-якого місця, дозволяючи їм вчитися на своїх помилках без прямого втручання вчителя. З точки зору вчителя, «МійКлас» прагне полегшити тягар рутинних і повторюваних завдань, дозволяючи вчителям приділяти більше часу створенню індивідуальних навчальних завдань та аналізу моделей навчання учнів. Зрештою, головна мета «МійКлас» полягає в тому, щоб зробити дітей розумнішими за допомогою цих цифрових освітніх рішень. Ця місія підкреслює прагнення платформи не лише покращити педагогічну практику, але й забезпечити практичну користь як для вчителів, так і для учнів [9].

«МійКлас» орієнтований на учнів від 1 до 12 класу. Такий широкий діапазон свідчить про намір підтримувати учнів протягом всього періоду навчання, задовольняючи різноманітні освітні потреби та академічні цілі. Окрім учнів, платформа також задовольняє потреби вчителів, надаючи інструменти для керування завданнями, відстеження прогресу та доступу до величезної бази даних навчальних матеріалів. Батьки є ще одним важливим сегментом цільової аудиторії, а «МійКлас» пропонує функції, що дозволяють їм контролювати прогрес та результати навчання своєї дитини.

Пропонуючи школярам численні варіації та пояснення, «МійКлас» сприяє самостійному процесу навчання, дозволяючи учням навчатися у власному темпі та зосереджуватися на тому матеріалі, де їм потрібна більша підтримка. Крім того, «МійКлас» запровадив навчальні завдання, спеціально розроблені для державних тестів, що надає цінну підтримку учням, які готуються до цих важливих іспитів. Школярі, які користуються преміум-підпискою Мій+, також отримують доступ до детальних кроків вирішення завдань та відповідей на них. Платформа включає інтерактивне навчальне середовище завдяки використанню вікторин, опитувань та ігрового навчального досвіду, спрямованого на підвищення залученості школярів. Акцент платформи на інтерактивних

елементах свідчить про прагнення забезпечити динамічний та персоналізований навчальний досвід, який враховує різні стилі навчання та сприяє глибшому концептуальному розумінню.

«МійКлас» пропонує надійний набір функцій, розроблених для оптимізації різних аспектів освітнього процесу. Вчителі можуть легко створювати та розповсюджувати домашні завдання та тести своїм учням через онлайн-портал. Платформа також надає вчителям інструменти для моніторингу прогресу та результатів навчання своїх учнів, пропонуючи інформацію про їхню успішність. Основна місія «МійКлас» полягає в тому, щоб допомогти вчителям оцифрувати свої навчальні процеси, надаючи їм ресурси та інструменти, необхідні для ефективної інтеграції технологій у свої класи. Автоматизуючи кілька завдань, «МійКлас» прагне зменшити навантаження буденної та повторюваної роботи для вчителів, звільняючи їхній час для більш ефективної діяльності (рис. 1).



Рис.1.1 Витрати часу вчителя до та після використання «МійКлас»

Платформа додатково розширює можливості вчителів, дозволяючи їм створювати індивідуальні навчальні завдання, адаптовані до конкретних потреб їхніх учнів, та аналізувати моделі навчання для формування своїх стратегій викладання. Передплатники преміум-сервісу Мій+ отримують доступ до ще потужніших функцій, включаючи автоматичне оцінювання домашніх завдань та тестів. Система також автоматично генерує статистику ефективності та звіти, які можна завантажити у зручних форматах, таких як docx або PDF. Мій+ також

дозволяє вчителям створювати власну індивідуальну методичну базу за допомогою розділу «Тестові роботи», де вони можуть створювати індивідуальні тести та домашні завдання, спираючись на велику базу даних «МійКлас» або створюючи власні унікальні питання. Щоб збагатити свої завдання, вчителі також можуть додавати додаткові навчальні матеріали, такі як відео, аудіофайли або відповідні веб-посилання в середовищі Мій+. Співпраця між вчителями також полегшується, оскільки створені користувачами «Предмети» можна використовувати разом з іншими вчителями в одній школі. Ці функції разом демонструють прагнення «МійКлас» надавати вчителям комплексну платформу, яка не лише спрощує адміністративні завдання, але й розширює їхню здатність персоналізувати навчання та покращувати результати учнів.

1.3. Дидактичні можливості та особливості платформи «МійКлас» для вивчення програмування

Для забезпечення якісного процесу навчання, освітні платформи мають в першу чергу мати достатньо простий та інтуїтивно зрозумілий інтерфейс, який водночас не відволікатиме учнів від навчання. «МійКлас» може слугувати чудовим стартом для опанування інформаційно-комунікативних технологій учнями, залучити в процес навчання батьків та розширити для вчителів способи подачі навчального матеріалу. Платформа забезпечує цілісний процес і підготовку до навчання. Так, однією з головних переваг платформи є унікальний підхід «навчання не форсуючи» та технічна складова, що базується на генерації умов завдань та налаштованій автоматичній перевірці відповідей учнів [9]. З кожним перезапуском завдання оновлюються та змінюють умови задля того, щоб сформувати у школярів навички розв'язування однотипних задач з будь-якого предмету у безкінечному режимі. При підключенні преміального режиму, учні мають можливість засвоїти матеріал на власних помилках завдяки функції «Кроки розв'язання», що не тільки сприяє покращенню розв'язку типових завдань, а й вчить мислити, аналізувати, розуміти і відтворювати [15].

Але це не єдина перевага цієї освітньої платформи. «МійКлас» забезпечує користувачів електронним розкладом та щоденником із журналом, інтерактивним контентом, а для вчителів особливо цінною функцією ресурсу є конструктор уроку, що надає можливість створювати власний унікальний контент. Широка база питань та автоматична генерація умов завдань унеможлиблює можливість списування. Крім того, сервіс забезпечує автоматичну перевірку відповідей і неупереджене оцінювання знань учнів, тож вчитель має більше часу для створення нового навчального контенту. Загалом, крім вже описаних, сервіс забезпечує користувачам такі переваги:

- Варіативність. Кожне зі створених завдань має близько 50 альтернативних варіантів, що забезпечує самостійну роботу школярів;
- Статистика. Платформа «МійКлас» автоматично генерує звітність про успішність учнів, додатково сортуючи їх за класом;
- Наявність сертифікатів. Учителі, що використовують цей сервіс, автоматично отримують сертифікати «Вчитель «МійКлас» та «Апробатор електронних освітніх технологій»;
- Шкільний тренажер. Допомогає учням самостійно засвоїти тему та містить аналогічні до річних контрольних та державних іспитів питання;
- Помічник вчителя. Інструмент для швидких опитувань, який допомагає з роботою над помилками на офлайн уроках;
- Іспити. Полегшене створення та повністю підготовлений курс підготовки, який не потребує проведення в один час. Кожний студент має індивідуальний варіант та знає результат іспиту одразу після завершення.

Крім того, для розширення функціоналу, сервіс надає додаткову платну опцію «Мій+». Вперше вона пропонується у якості пробного періоду після підтвердження статусу вчителя. Порівняння різниці доступу та можливостей базового та преміум тарифів наведено у таблиці 1.1.

Порівняння безкоштовного користування та з опцією «Мій+»

Безкоштовна	З «Мій+»
Спільне	
Навчальні матеріали для 1-11 класів: теорія, завдання та домашні та перевірочні роботи	
Отримання та виконання електронних домашніх завдань, перевірочних робіт	
	Правильні відповіді, пояснення та кроки розв'язання до всіх завдань на сайті
	При участі в конкурсах - доступ до підготовчих завдань
	Підсумкові контрольні роботи
	Доступ до швидких тестів
	Відсутня реклама

Загалом, підключення розширеного тарифу надає учителям можливість заощадити більшість часу на підготовку до уроку, доступ до завдань курсів з усіх шкільних предметів та автоматизує рутинні справи учителів без додаткових навантажень або складнощів.

Важливим аспектом платформи «МійКлас» є її адаптивність до вимог сучасного навчання програмування, зокрема мови Python. Платформа дозволяє інтегрувати інтерактивні вправи з кодування, де учні можуть відпрацьовувати синтаксис, логічні конструкції та алгоритмічні завдання в безпечному середовищі з миттєвим зворотним зв'язком. Автоматизована перевірка завдань з програмування, підтримана генерацією варіантів, дає змогу учням багаторазово тренуватися без страху помилитися, що відповідає принципам конструктивістського та компетентнісного підходів [3]. Крім того, функція «Кроки розв'язання» в преміум-режимі може бути особливо корисною для розбору алгоритмічних помилок, що сприяє глибшому розумінню логіки програмування.

Для вчителів платформа пропонує інструменти, що спрощують індивідуалізацію навчання програмування. Наприклад, конструктор уроків дозволяє створювати завдання різного рівня складності, враховуючи різний темп та можливості учнів, що є ключовим для формування особистісних освітніх

траєкторій [17]. Статистика успішності та автоматичні звіти допомагають вчителю швидко ідентифікувати проблемні моменти в групі та корегувати навчальний процес. Це особливо важливо при вивченні Python, де на початкових етапах учні часто стикаються з типовими помилками синтаксису чи логіки.

Дослідження також показують, що використання цифрових платформ, таких як «МійКлас», сприяє розвитку саморегуляції та відповідальності в учнів, оскільки вони можуть самостійно планувати своє навчання, відстежувати прогрес і повторювати матеріал у зручному темпі [7]. Це узгоджується з вимогами Нової української школи щодо формування в учнів навичок самоосвіти та вміння вчитися протягом життя.

Окрім того, платформа може бути використана для реалізації проєктного підходу: вчителі можуть створювати довгострокові проєктні завдання з програмування, де учні поступово розробляють власні програми, отримуючи проміжні перевірки та консультації через систему. Це імітує реальний процес розробки ПЗ і формує досвід роботи над складними завданнями [13].

Таким чином, «МійКлас» виступає не лише як засіб автоматизації перевірки знань, але й як комплексне середовище для розвитку алгоритмічного мислення, креативності та індивідуальних навичок учнів у сфері програмування, що робить її цінним інструментом в арсеналі сучасного вчителя інформатики.

Висновки до першого розділу

Проведений аналіз підкреслює зростаючу важливість освіти з програмування в закладах середньої освіти як життєво важливого елементу інформаційної грамотності та інструменту для розвитку важливих когнітивних навичок, таких як логічне мислення та вирішення проблем. Це вимагає розвитку педагогічних підходів та інтеграції технологій у освітній процес. У дослідженні виділено педагогічні підходи: компетентнісний, комплексний, функціональний, імперативний, блоковий, об'єктно-орієнтований, проєктний та алгоритмічний підходи. Визначено їх переваги, а саме: компетентнісний підхід забезпечує перехід до формування цілісної системи вмінь для вирішення реальних

практичних задач, комплексний сприяє створенню міцних міжпредметних зв'язків і системного мислення, конструктивістський розвиває когнітивні навички вищого порядку через дослідження та експериментування. Проектний підхід забезпечує глибоке практичне засвоєння матеріалу та розвиває креативність, а гейміфікований – підвищує мотивацію та залученість учнів за рахунок елементів ігрового дизайну. Застосування та комбінування цих підходів дозволяє адаптувати навчання до різного віку учнів, змісту курсу та обраного освітнього середовища, формуючи динамічну та ефективну освітню практику.

Цифрові освітні платформи стали незамінними для організації ефективного та захопливого онлайн- та змішаного навчального середовища. Такі платформи, як Google Classroom, Moodle та «МійКлас», пропонують широкий спектр функцій, що підтримують сучасні стратегії навчання. До них належать інструменти для створення та управління курсами, сприяння комунікації та співпраці, надання різноманітних варіантів оцінювання, відстеження прогресу учнів та налаштування для задоволення конкретних освітніх потреб. Використовуючи ці платформи та різноманітність підходів до навчання, вчителі можуть створити більш динамічний, персоналізований та ефективний навчальний досвід для учнів у галузі програмування. Синергія між відповідними педагогічними методами та передовими цифровими інструментами є ключем до підготовки учнів до вимог сучасного цифрового світу.

Інтеграція платформи «МійКлас» в освіту з програмування ілюструє зростаючу актуальність цифрових інструментів у узгодженні навчальних практик із сучасними педагогічними вимогами. Її комплексна інфраструктура, починаючи від автоматизації завдань та миттєвого зворотного зв'язку до створення налаштованого контенту, підтримує не лише технічний розвиток учнів, але й розвиває важливі когнітивні процеси, такі як вирішення проблем, міркування та абстракція.

Включаючи покрокові інструкції з вирішення проблем та функції адаптивного тестування, «МійКлас» створює середовище, яке особливо корисне для вивчення програмування, предмета, який часто створює труднощі для

початківців. Крім того, відповідність платформи національним навчальним програмам та включення тем ООП, таких як класи, об'єкти та конструктори, гарантує, що учні отримають як теоретичну основу, так і практичний досвід кодування.

Не менш важливим є внесок платформи в оптимізацію робочого процесу вчителів. Завдяки автоматизованому оцінюванню, структурованій аналітиці та централізованій системі управління контентом, «МійКлас» зменшує рутинне навантаження, пов'язане з навчанням програмування, дозволяючи викладачам зосередитися на змістовних педагогічних втручаннях та диференційованій підтримці.

Загалом, платформа являє собою ефективну модель того, як технологічно вдосконалені навчальні середовища можуть відповідати конкретним потребам освіти з програмування в школах. Його потенціал для покращення залученості учнів, розвитку обчислювального мислення та підтримки автономії вчителів підтверджує його цінність як педагогічно обґрунтованого та технологічно надійного інструменту для навчання інформатиці.

РОЗДІЛ 2

МЕТОДИЧНІ ОСОБЛИВОСТІ ВИКОРИСТАННЯ СИСТЕМИ «МІЙКЛАС» ПРИ ВИВЧЕННІ PYTHON

2.1. Методичні засади інтеграції платформи «МійКлас» у процес навчання програмування Python

Платформа «МійКлас» з початку створення запланована для використання у першу чергу закладами загальної середньої освіти, тому безкоштовний тарифний план розрахований на використання для шкільних курсів. Наразі у вільному доступі для вчителя можна використовувати 26 курсів, але актуальним для дослідження є лише курс «Інформатика», в той час як «Інформатика НУШ» створена для учнів 1–4 класів, які ще не вивчають програмування. У предметі «Інформатика» теми розподілені на класи, починаючи від «5 клас НУШ» до «7 класу НУШ», надалі переходячи на традиційні програми навчання 8–10 класів. Наявний курс «Інформатика» охоплює широкий спектр тем, тоді як для глибокого засвоєння Python потрібна більш спеціалізована, поетапна програма, орієнтована на практику та проєктну діяльність, яку і дає змогу реалізувати «МійКлас».

Так, школярі починають вивчати основи програмування у п'ятому класі на основі алгоритмів Python, надалі практикуючи навички їх побудови та розгалуження. Для цього використовується динамічна візуальна мова програмування Scratch. Саме названі дві мови формують уявлення, знання та уміння учнів в контексті програмування і вивчаються поступово протягом наступних шкільних років. Розглянемо наповненість запропонованих тем до вивчення на прикладі розділу VIII «Алгоритми та програми. Python» 7 класу Нової Української школи, а саме теми «Об'єктно-орієнтоване програмування». Сторінка теми містить в собі основні матеріали для вивчення, вигляд яких представлений у таблиці 2.1.

Таблиця 2.1.

Зміст теми «Об'єктно-орієнтоване програмування»

Пункт	Опис	Бал
Зміст		
1. Методична рекомендація	Опис вмісту курсу та завдань, тестів та перевірочних тестів з теми.	
Теорія		
1. Опис класів	Теоретичний матеріали до теми.	
2. Принципи ООП	Поняття класу та об'єкту як основи об'єктно-орієнтованого програмування (ООП).	
Завдання		
1. Обери правильну відповідь Складність: легке	Завдання на знання понять класу та об'єкту.	1
2. Вибери опис поняття Складність: легке	Завдання на знання понять класу та об'єкту.	1
3. Поясни принципи ООП Складність: легке	Завдання на знання основних принципів ООП.	1
4. Так чи ні Складність: середнє	Завдання на знання понять класу та об'єкту.	3
5. Вибери заголовок конструктора Складність: середнє	Завдання на знання понять класу та об'єкту.	3
6. Як створено об'єкт? Складність: середнє	Завдання на знання понять класу та об'єкту.	3
7. Знайди результат Складність: середнє	Завдання на уміння створювати екземпляри класів.	3
Додаткові завдання (доступні у режимі «Мій+»)		
1. Що надрукує програма? Складність: важке	Завдання на знання принципів ООП.	4
2. Яку людину представляє об'єкт? Складність: важке		4
3. Як створено список об'єктів? Складність: важке		4
Додаткові завдання, приховані від учнів («Мій+»)		
1. Знайди опис поняття Складність: легке	Завдання на знання принципів ООП.	1
2. Визнач значення атрибутів Складність: важке		4
3. Обчисли площу		4

Складність: важке		
Тести		
1. Тренування №1 з теми «Об’єктно-орієнтоване програмування» Складність: середнє	Удосконалення знань з теми «Об’єктно-орієнтоване програмування».	9
2. Тренування №2 з теми «Об’єктно-орієнтоване програмування» Складність: важке		18
Перевірочні тести (приховані від учнів)		
1. Домашня робота з теми «Об’єктно-орієнтоване програмування» Складність: важке	Закріплення знань з теми «Об’єктно-орієнтоване програмування».	27
2. Перевірочна робота з теми «Об’єктно-орієнтоване програмування» Складність: важке	Перевірка засвоєння знань з теми «Об’єктно-орієнтоване програмування».	26

Усі теоретичні матеріали, представлені на платформі, узгоджені з рекомендаціями Міністерства освіти і науки України та посилаються на підручники, що використовуються у школі фактично. Так, у цьому розділі використовується підручник «Інформатика» для 7 класу від авторів О. Бондаренко, В. Ластовецького та інших 2020 року видання [12]. Формат адаптовано під формат веб-сайту для полегшення сприйняття та навчання учнів, демонструючи їм не лише теоретичний зміст уроку, а й частини коду, які вони можуть скопіювати та протестувати (рис. 2.1).

Опис класів та створення об'єктів

Програмування мовою Python є об'єктно-орієнтованим, тобто всі величини є об'єктами певного класу.



Клас — це тип даних, створений користувачем. Клас описує властивості об'єктів, створених на його основі, та дії, які об'єкти класу можуть виконувати.

Класи можна використовувати для організації фрагментів кода в Python-програмі.

Приклад:

Модуль `tkinter` містить описи класів графічних компонентів, і ми створюємо віджети як екземпляри цих класів. Кожний компонент графічного інтерфейсу — це об'єкт певного класу графічних компонентів, що має властивості та методи: об'єкт `Label` є екземпляром класу `Label`, об'єкт `Button` є екземпляром класу `Button`.

Під час створення об'єктів можна задати значення їхніх властивостей для налаштування вигляду, та описати методи — прикріплені до цих віджетів обробники подій.

```
Синтаксис опису класу:
class <Назва_класу>():
def __init__(self, <перелік атрибутів (властивостей) класу >):
    self.<змінна = <значення>
```

Рис. 2.1. Представлення вигляду уроку на платформі «МійКлас»

І відповідно до матеріалу, вказаного на курсі з цієї теми, після вивчення та засвоєння матеріалу, учні проходять тестування, де мають можливість перевірити свій рівень знань. Це можуть бути поодинокі питання після теми, які готують здобувачів до проходження повноцінного тестування, у якому змінюється складність та тип завдань (рис. 2.2).

Отже, «МійКлас» надає вчителям інформатики інструменти, які підвищують ефективність навчання, покращують залученість учнів та сприяють автономному та керованому навчанню, що робить його актуальним та практичним вибором для навчання програмуванню у закладах загальної середньої освіти. Для вчителя інформатики «МійКлас» полегшує управління індивідуальними та груповими навчальними шляхами, дозволяючи учням працювати у власному темпі та забезпечуючи своєчасний зворотний зв'язок. Платформа підтримує планування завдань, оцінювання та моніторинг прогресу, зменшуючи адміністративне навантаження вчителя та дозволяючи більше зосередитися на педагогічній підтримці.

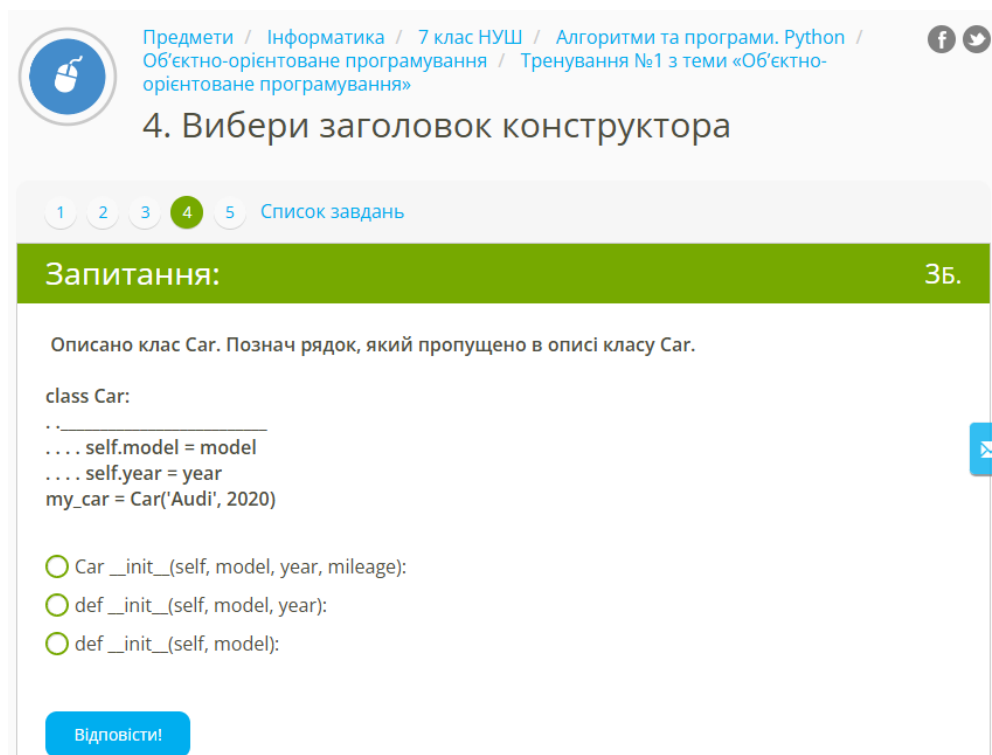


Рис. 2.2. Приклад проходження тестування за темою

Розробка змістовної основи навчального курсу «Програмування на Python» для учнів 7-го класу вимагає системного підходу до інтеграції з функціональністю сучасної освітньої платформи. Методологія використання цифрового середовища базується на принципах диференційованого навчання, формувального оцінювання та розвитку компетенцій, визначених концепцією Нової української школи.

Проаналізуємо стан вивчення інформатики у Бродівській спеціалізованій школі I-III ступенів №2. Згідно з освітньою програмою закладу на 2024-2025 навчальний рік, загальний обсяг навчального навантаження для 7-их класів складає 35 годин на тиждень [10]. Програмою для загальноосвітніх навчальних закладів, затвердженою для загальноосвітніх навчальних закладів, для курсу інформатики з цього обсягу присвячується лише одна година на тиждень і цього часу може бути недостатньо для вивчення та засвоєння особливо складних концепцій об'єктно-орієнтованого програмування [14]. Тож для покращення вивчення програмування запропоновано впровадження курсу програмування на Python.

Розробка тематичної структури курсу вимагає переходу від загальних методологічних міркувань до більш систематизованого представлення його навчальної архітектури. Базові засади програмування учні НУШ вивчають, починаючи із 5 класу, отже розроблена методика має починатися з повторення та закріплення раніше вивчених тем та умінь. Після уточнення педагогічних підходів, організаційних обмежень та дидактичних цілей стає важливим окреслити внутрішню структуру курсу таким чином, щоб вона відображала узгодженість між результатами навчання, змістовними блоками та формами діяльності учнів. Таким чином, у наступній таблиці підсумовано змістовну лінію дисципліни, основний теоретичний матеріал та відповідні навчальні види діяльності, які разом визначають модульну структуру курсу (таблиця 2.2).

Таблиця 2.2

Змістова лінія навчальної дисципліни

Очікувані результати навчання	Зміст навчального предмета	Види навчальної діяльності
Модуль 1. Систематизація знань та робота з даними (5 годин)		
Учень/учениця: Використовує методи роботи зі списками та словниками для розв'язання практичних задач. Пояснює різницю між локальними та глобальними змінними. Читає та записує дані у текстові файли.	Поглиблена робота зі списками (методи, зрізи); Словники та їх методи; Поглиблена робота з функціями (аргументи, область видимості); Робота з текстовими файлами (читання, запис).	• <i>Пояснення теоретичного матеріалу та демонстрація.</i> • <i>Практикум:</i> «Сортування та аналіз даних» (аналіз списку оцінок). • <i>Проект:</i> «Англо-український словник» з використанням словників Python. • <i>Лабораторні роботи:</i> «Читання конфігураційних файлів» та «Ведення логу дій програми».
Модуль 2: Вступ до ООП. Класи та об'єкти (6 годин)		
Учень/учениця: Вміє створювати власні класи з атрибутами та методами. Пояснює поняття «екземпляр класу». Використовує конструктор <code>__init__</code> для ініціалізації об'єктів.	Концепція ООП. Клас як тип даних; Атрибути та методи; Конструктор <code>__init__</code> та <code>self</code> ; Створення екземплярів класу.	• <i>Пояснення теоретичного матеріалу та демонстрація.</i> • <i>Обговорення:</i> Аналогії з реальним світом (автомобіль, тварина) для пояснення класів та об'єктів. • <i>Моделювання:</i> Створення класів <code>Student</code>, <code>Book</code>, <code>GameCharacter</code>.

		• <i>Практична робота:</i> «Створення власного типу даних».
Модуль 3: ООП. Спадкування та поліморфізм (5 годин)		
Учень/учениця: Використовує спадкування для створення ієрархій класів. Вміє перевизначати методи у дочірніх класах. Розуміє та демонструє принцип поліморфізму.	Концепція спадкування. Базові та дочірні класи; Метод super(); Перевизначення методів; Поліморфізм: однакова дія для різних об'єктів.	• <i>Пояснення теоретичного матеріалу та демонстрація.</i> • <i>Проект «Графічні фігури»:</i> Базовий клас Shape, дочірні Circle, Rectangle з методами area() та draw(). • <i>Проект «Транспорт»:</i> Ієрархія Vehicle → Car, Bicycle, Motorcycle. • <i>Візуалізація:</i> Малювання ієрархії класів у вигляді діаграми.
Модуль 4: Основи створення GUI з Tkinter (7 годин)		
Учень/учениця: Створює графічні інтерфейси з використанням бібліотеки Tkinter. Вміє розміщувати віджети (мітки, поля, кнопки) та обробляти події.	Створення головного вікна; Основні віджети: Label, Entry, Button; Менеджери геометрії: pack() та grid(); Обробка подій (подія натискання кнопки).	• <i>Пояснення теоретичного матеріалу та демонстрація.</i> • <i>Практикум «Мій перший GUI»:</i> Створення програми, яка вітає користувача по імені. • <i>Проект «Простий калькулятор»:</i> Логіка обчислень та графічний інтерфейс. • <i>Взаємооцінювання інтерфейсів однокласників.</i>
Модуль 5: Інтеграція знань: ООП та Tkinter (7 годин)		
Учень/учениця: Створює інтерактивні інтерфейси з використанням різноманітних віджетів (Text, Listbox). Використовує діалогові вікна для роботи з файлами. Пояснює принцип розділення коду на логіку (модель) та інтерфейс (подання). Впроваджує принципи ООП для побудови архітектури GUI-додатків.	Поглиблена робота з віджетами Tkinter: • Багаторядкове текстове поле Text. • Список елементів Listbox. • Діалогові вікна для відкриття та збереження файлів (filedialog). Архітектура GUI-додатків: • Концепція "Модель-Подання" на простому рівні. • Створення окремих класів для бізнес-логіки програми. • Створення класу інтерфейсу, який взаємодіє з класом-логікою.	• <i>Пояснення теоретичного матеріалу та демонстрація.</i> • <i>Практикум «Опанування інструментів»:</i> Створення форми з текстовим полем (Text) та списком (Listbox). • <i>Індивідуальна/парна робота над основним проектом модуля:</i> • Варіант «Простий текстовий редактор»: Логіка (клас TextEditor) та Інтерфейс (клас TextEditorGUI). • Варіант «Список завдань»: Логіка (клас TodoList) та Інтерфейс (клас TodoListGUI).

	Інтеграція ООП та Tkinter на практиці.	
Модуль 6: Фінальний індивідуальний або груповий проєкт (5 годин)		
Учень/учениця: Демонструє комплексне застосування набутих знань та навичок шляхом реалізації власного проєкту. Захищає свій проєкт, пояснюючи прийняті рішення.	Вибір теми та архітектури проєкту; Планування та розподіл завдань (у групах); Кодування, тестування; Підготовка презентації та демонстрації.	• <i>Пояснення теоретичного матеріалу та демонстрація.</i> • <i>Проектна діяльність:</i> Розробка повноцінної програми (на вибір: гра, корисна утиліта, тренажер). • <i>Консультації.</i> • <i>Захист проєктів:</i> Публічна презентація своєї роботи перед класом.

Встановивши необхідність розпочати курс з повторення раніше набутих знань, загальну структуру програми необхідно представити у формі, яка чітко відображає пропорційний розподіл годин, присвячених кожному тематичному компоненту. Таке представлення дозволяє прозоро зрозуміти, як навчальна програма базується на фундаментальних концепціях, поступово вводить складніші теми та виділяє достатньо часу як для теоретичного навчання, так і для практичного навчання на основі проєктів. У наступній таблиці 2.3 наведено розподіл годин між модулями курсу, демонструючи, як кожен компонент сприяє узгодженому впровадженню програми.

Таблиця 2.3

Розподіл годин для курсу

№	Назва	Години
1	<i>Систематизація знань та робота з даними</i>	5
	Поглиблена робота з функціями (аргументи, область видимості)	1
	Словники та вкладені структури даних. Практика: «Англо-український словник».	2
	Робота з текстовими файлами (читання, запис). Практика: «Зчитування та аналіз даних з файлу».	2
2	<i>Вступ до ООП. Класи та об'єкти</i>	6
	Теорія: що таке клас, об'єкт, конструктор <code>__init__</code> .	2
	Практика: створення атрибутів та методів.	2
	Проект: «Створення власного типу даних».	2
3	<i>ООП. Спадкування та поліморфізм</i>	5
	Теорія спадкування. Метод <code>super()</code> .	2

	Практика поліморфізму та перевизначення методів.	2
	Проект: «Графічні фігури».	1
4	Основи створення GUI з Tkinter	7
	Знайомство з Tkinter. Віджети Label, Entry. Менеджер grid.	2
	Кнопка (Button) та обробка подій за допомогою command.	2
	Практичний проєкт: «Простий калькулятор»	3
5	Інтеграція знань: ООП та Tkinter	7
	Огляд віджетів Text та Listbox, їх властивостей і методів.	2
	Практикум «Опанування інструментів»: Створення форми з текстовим полем (Text) та списком (Listbox).	2
	Індивідуальна/парна робота над основним проєктом модуля: «Простий текстовий редактор» або «Список завдань»	3
6	Фінальний індивідуальний або груповий проєкт	5
	Вибір теми, планування, технічне завдання.	1
	Активна розробка проєкту та індивідуальні консультації.	3
	Презентація та захист проєктів.	1

Структурований план, розроблений на основі окреслених тематичних модулів та розподілу годин, встановлює узгоджену основу для організації навчального процесу в рамках курсу програмування на Python. Після чіткого визначення змісту, результатів навчання та навчальних заходів наступним кроком є реалізація цього плану з використанням функціональних можливостей електронної освітньої системи «МійКлас», яка надає необхідні цифрові інструменти для реалізації, моніторингу та підтримки запропонованої навчальної програми.

2.2. Проєктування наповнення навчального курсу «Програмування мовою Python» в системі «МійКлас»

Наповнення запланованого курсу передбачає у собі декілька етапів. Як зазначалося раніше, можливість створювати та розробляти свій власний курс мають лише ті педагоги, що оформили підписку на «МійКлас+». Так, їхні ресурси можуть як бути у відкритому доступі для користувачів порталу, так і бути доступними лише за посиланням або за призначенням вчителя. Після оформлення преміум доступу до сервісу, у розділі «Предмети» бачимо зверху

поле «Мої предмети», де будуть висвітлені ті курси, що будуть створені та опубліковані. А поки обираємо кнопку «Створити предмет» та переходимо до нової сторінки. Тут необхідно вказати обов'язкові параметри нашої нової дисципліни: назву предмета необхідно вказати самостійно, але з розрахунку на те, щоб школярі не переплутали авторський курс з тим, що надає банк системи «МійКлас». В графі «Назва навчальної програми» доцільно вказувати клас, для якого власне розрахована розроблена методика. Решта з полів містять виключно списки з обмеженими варіантами іконки для предмета, тож користувач не зможе виділити авторську розробку. Останній блок пропонує обрати вчителю «Тип предмета», тобто підібрати той предмет, який є найближчим до створеного серед міжпредметних зв'язків. Отже, заповнивши усі поля, можемо переходити до наступного кроку (рис. 2.3)

Рис. 2.3. Створення нового предмету

Після цього на платформі «МійКлас» створено порожню чернетку нашого майбутньої дисципліни. Саме в цьому полі, представленому на рисунку 2.4 буде виконуватися наповнення розроблений курс у відповідності з уже обраними темами. Інтерфейс цієї сторінки простий та інтуїтивно зрозумілий: кнопка публікації недоступна, так як предмет не має наповнення, поруч є значок довідки з необхідними підказками та вказівками для тих фахівців, що вперше розробляють власний курс.

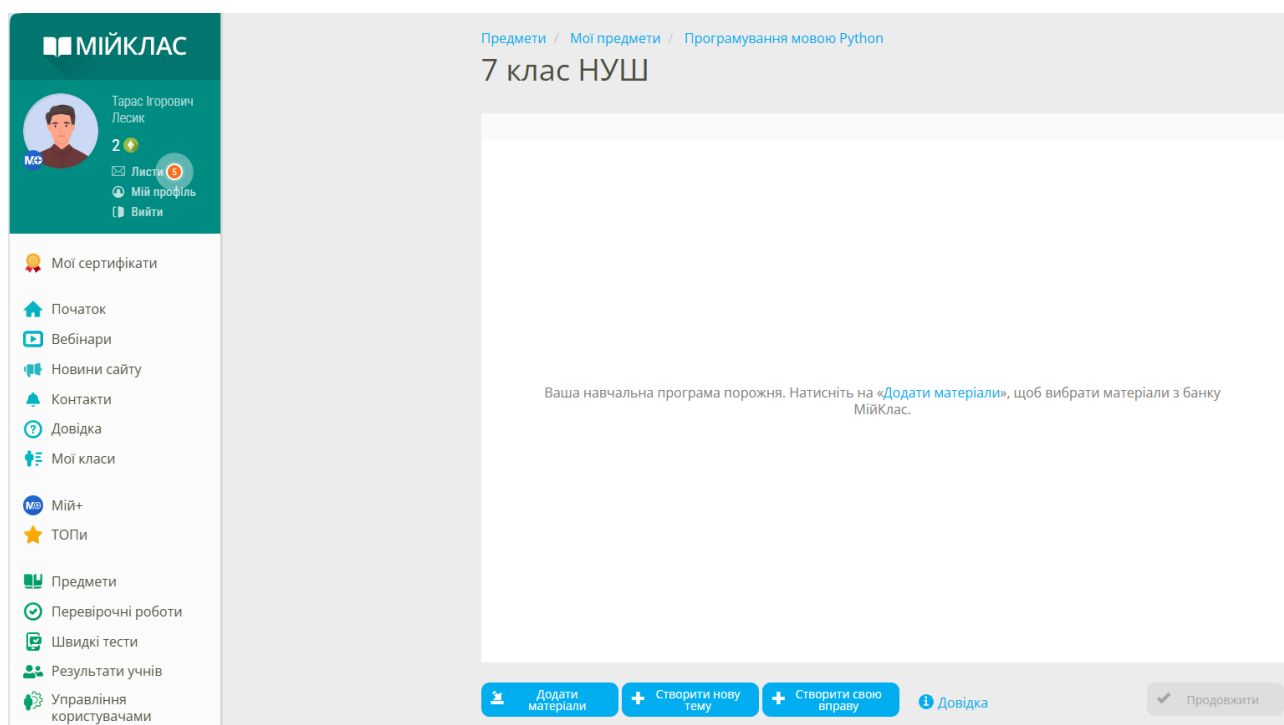


Рис. 2.4. Стартовий вигляд нового предмету

Основна робота вчителя відбувається через три кнопки: «Додати матеріали», «Створити нову тему» та «Створити свою вправу». Розглянемо їх, починаючи з найпростішої – створення нової теми. Це каркас, який надає розробнику можливість виокремити теми та розділити матеріали візуально, що значно допоможе школярам ліпше орієнтуватися у процесі навчання. Усі створені в цьому розділі теми можна вільно переміщати між собою для необхідного групування. Корисною функцією на цьому етапі є автозбереження, яке фіксує усі внесені зміни, тож розробник може бути впевнений, що у разі перебоїв електроенергії через обстріли чи планові відключення, його напрацювання буде збережено. Після внесення назв модулів та тем, предмет має наступний вигляд, представлений на рисунку 2.5.

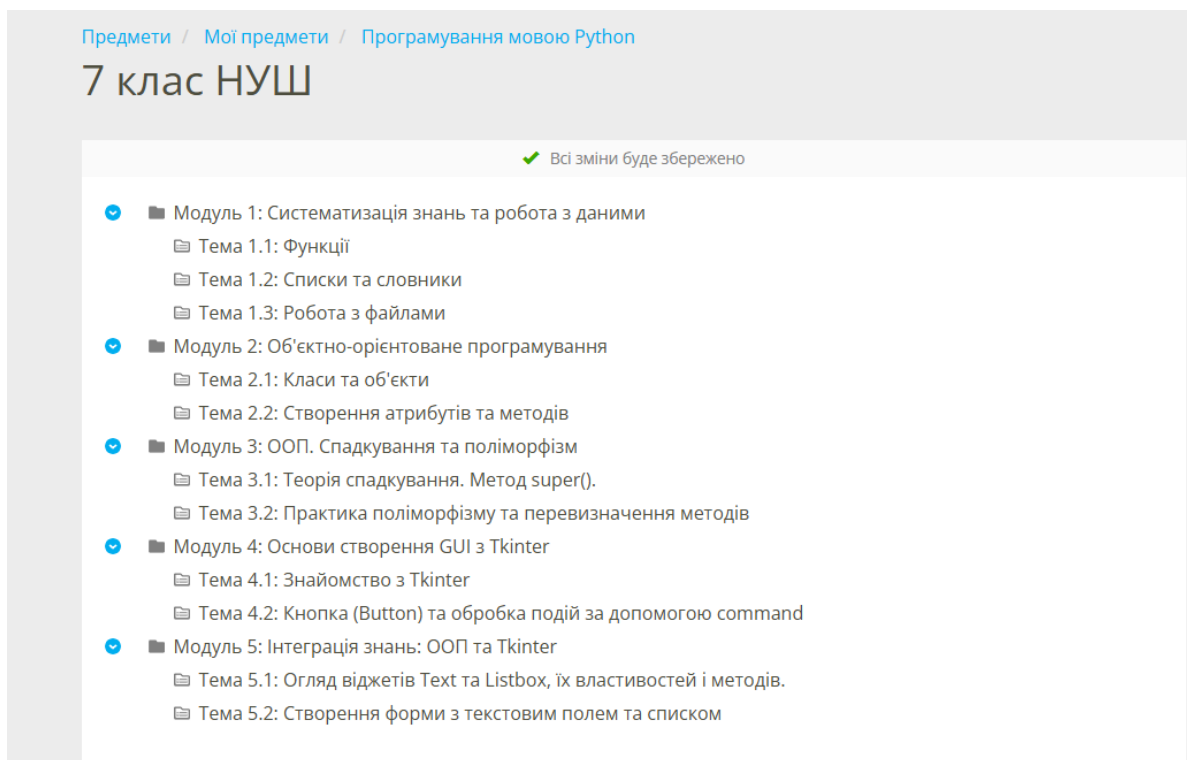


Рис.2.5. Створення тем та підтем

Особливістю електронної освітньої програми «МійКлас» також можна виділити функцію «Додати матеріали», при виборі якої, вчитель може додати у свій предмет теоретичний або практичний блок із вже представлених та доступних на сайті предметів, в залежності від предмету, класу та теми (рис 2.6).

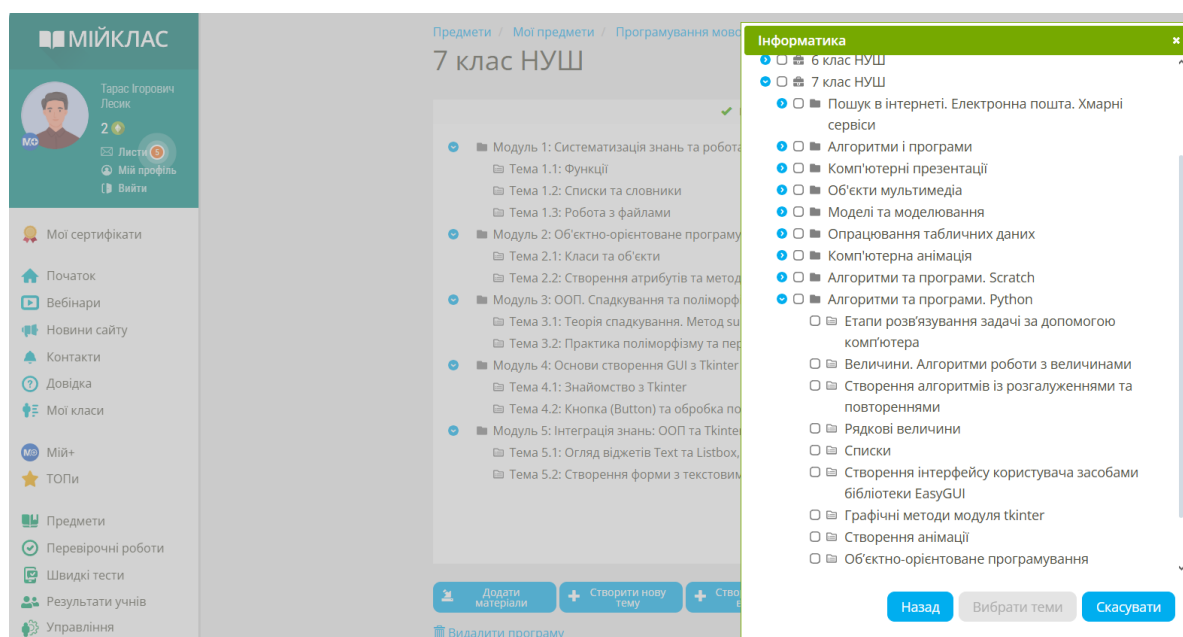


Рис. 2.6. Доступні матеріали ЕОС «МійКлас»

Але, окрім вже наданих платформою матеріалів, кожен викладач, що має авторську розробку, має можливість завантажити власні матеріали на курс. Для цього необхідно обрати кнопку «Створити свою вправу», після чого з'явиться нове вікно браузера із доступними типами завдань (рис 2.7).

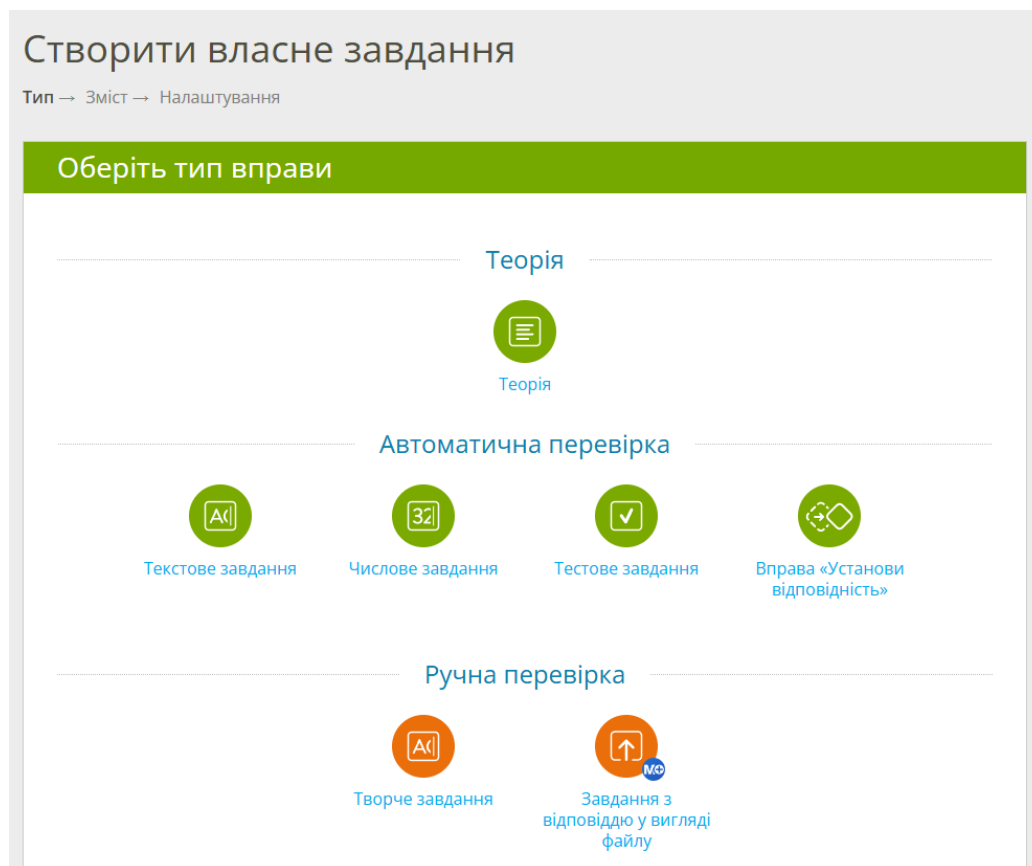


Рис.2.7. Типи вправ, доступних для створення

В першу чергу при створенні навчального курсу необхідно наповнити його теоретичними відомостями на обрані теми для того, щоб учні не лише мали для себе опорний конспект з матеріалу, вивченого на уроці, а й могли дослідити нову тему самостійно, у разі пропуску заняття. Крім описаного, це стане в нагоді і у разі впровадження дистанційного навчання або під час відключень електроенергії.

Коли необхідний тип завдання, у нашому випадку – теорія, обрано, з'являється наступне поле (рис. 2.8). Тут є стандартне поле для створення лекційного матеріалу: портал надає можливість робити візуальні акценти на ключових поняттях теми, а й доповнювати матеріал наглядними фото- та відео-

ресурсами, таблицями і формулами. Як і у попередніх розділах, тут також є підказки для користувачів, що займаються наповненням вперше.

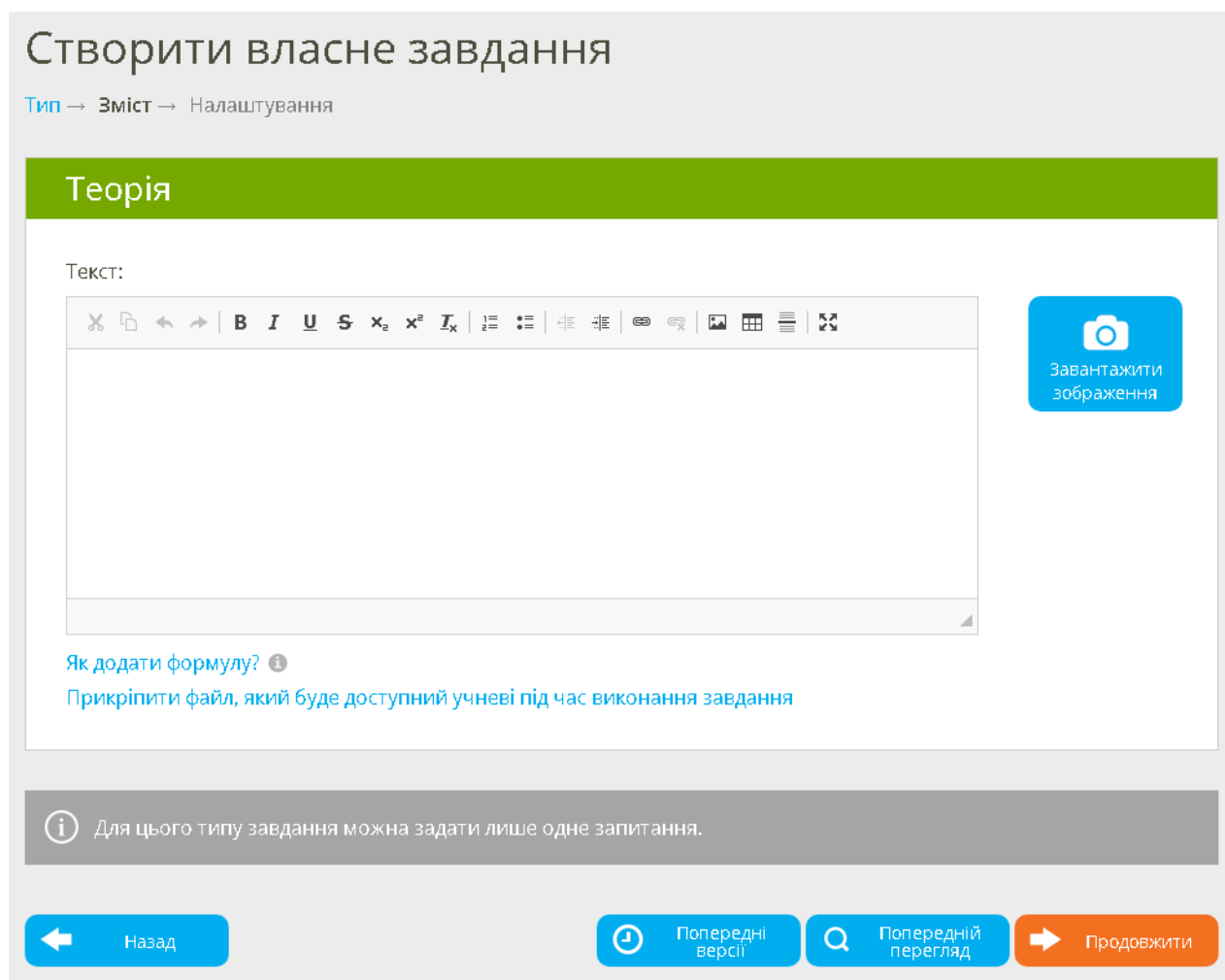


Рис. 2.8. Вікно створення теоретичного матеріалу.

Тепер, коли школярі мають теоретичний матеріал, на який можуть спиратися в ході навчання програмуванню, варто також додати тестові питання, мета яких закріпити та перевірити отримані знання. Платформа «МійКлас» надає можливість створити до 10 унікальних варіантів завдання для випадкового розподілу між учнями, тим самим надаючи викладачам урізноманітнити тести та зменшити вірогідність списування. Кожен варіант нумерується автоматично, але варто зауважити, що вони не пов'язані між собою, тобто учневі може бути призначено варіант 1 для першого запитання і варіант 2 для другого. Заповнене поле питання має вигляд, представлений на рисунку 2.9.

Тестове завдання

Варіанти: 1 2 +

Питання, яке буде поставлено школяреві:

✂ 📄 ↶ ↷ | **B** *I* U ~~S~~ $\times_2$ $\times^2$ $\int_x$ | $\frac{1}{2}$ $\frac{1}{3}$ $\frac{1}{4}$ $\frac{1}{5}$ | ∞ $\frac{1}{\infty}$ |

Завантажити зображення

Що виведе наступний код?

```
def func(x, y=5):
    return x + y

print(func(3))
```

body p

Як додати формулу? i

Прикріпити файл, який буде доступний учневі під час виконання завдання

Варіанти відповідей:

☒

8

🗑

☐

35

🗑

☐

Помилка

🗑

☐

3

🗑

☒ Тільки одна відповідь правильна

Додати відповідь

Відмітьте правильні відповіді

Сума балів:

1

Рис.2.9. Приклад оформлення тестового завдання.

В той час як для учня під час проходження тесту матиме наступний вигляд (рис.2.10).



Завдання

Який результат виконання програми?

```
def outer():
    n = 1
    def inner():
        nonlocal n
        n += 1
        return n
    return inner()

print(outer())
```

- ☐ 1
- ☐ 2
- ☐ None

Виберіть два правильні твердження про аргументи функцій:

- ☐ Аргументи з значенням за замовчуванням можуть бути лише в кінці списку параметрів
- ☐ Позиційні аргументи повинні йти перед іменованими
- ☐ args використовується для отримання словника іменованих аргументів
- ☐ kwargs дозволяє передавати довільну кількість іменованих аргументів

Рис.2.10. Вигляд готового тесту.

Таким чином відбувається наповнення усього курсу факультативу, що складається із 6 модулів, поділених на 3 теми, що вивчаються протягом шкільного року один раз на тиждень.

2.3. Організація та проведення педагогічного експерименту з використанням ЕОС «МійКлас».

Експериментальне дослідження було проведено у двох паралельних класах 7-х класах Бродівської спеціалізованої загальноосвітньої школи №2. Воно включало в себе етапи вхідного та підсумкового контролю у формі тестування, що надало можливість провести порівняльний аналіз динаміки змін у експериментальній (7-А, що складається з 13 учнів) та контрольній (7-Б, 12 учнів) групах. Перша група засвоювала навчальний матеріал з Python за

допомогою системи «МійКлас», тоді як друга використовувала традиційну модель навчання. Головною метою аналізу було не лише констатувати наявність статистично значущих відмінностей у освітніх показниках, але й кількісно визначити практичну значущість цих відмінностей, використовуючи сучасні методи статистичного моделювання.

Педагогічний експеримент проводився у природних умовах освітнього процесу, що є характерною ознакою квазі-експериментального підходу. Дослідження мотиваційної складової здійснювалося через адаптоване анкетування за п'ятибальною шкалою Лайкерта, що включало три ключові аспекти: інтерес до програмування як навчальної дисципліни, готовність до подолання труднощів під час виконання складних завдань та схильність до самостійного опанування нового матеріалу. Кожен аспект оцінювався окремо з подальшим розрахунком інтегрального показника мотивації (таблиця 2.4).

Таблиця 2.4

Мотиваційний опитувальник

№	Твердження	Зовсім не згоден/на	Швидше не згоден	Нейтрально	Швидше згоден	Повністю згоден
1	Вивчення Python є для мене захоплюючим заняттям, і я радий/рада, що його викладають у школі.					
2	Я готовий/готова витратити додатковий час, щоб зрозуміти складну тему або виправити помилки у коді.					
3	Мені подобається самостійно шукати інформацію та опановувати нові функції чи модулі Python.					

4	Я вважаю, що навички програмування будуть важливими для мого майбутнього успіху.					
5	Під час уроків я часто намагаюся вийти за межі завдання, шукаючи альтернативні чи більш ефективні рішення.					

Окрім того, на початку дослідження було проведено попереднє тестування (пре-тест) для встановлення початкової еквівалентності груп, що здійснювалося за трьома основними напрямками, необхідними для якісного засвоєння знань:

- *Когнітивний (Знання)*: оцінка засвоєння теоретичного матеріалу, синтаксису Python та базових алгоритмічних конструкцій (у відсотках);
- *Компетентісний (Практичні навички)*: оцінка рівня самостійності та здатності до налагодження коду (у відсотках);
- *Афективний (Мотиваційний)*: оцінка внутрішньої зацікавленості, використовуючи опитувальник на основі п'ятибальної шкали Лікерта.

Пре-тестування, як стартова діагностична процедура, було спрямоване на встановлення початкового рівня сформованості ключових компетентностей з програмування та вимірювання мотиваційного потенціалу учнів [3]. Теоретичний блок складався з 15 тестових завдань закритого типу, розроблених з урахуванням принципу поступового ускладнення. Перша група завдань перевіряла засвоєння базових синтаксичних конструкцій мови Python, включаючи оголошення змінних, роботу з основними типами даних та використання операторів. Друга група завдань була спрямована на оцінювання розуміння принципів роботи з функціями, зокрема механізмів передачі параметрів, областей видимості змінних та організації повернення результатів. Критерієм успішності виступав відсоток правильних відповідей з максимально можливим балом 100%.

Практична складова діагностики передбачала розробку функціональної програми «Калькулятор середнього балу», що вимагала застосування комплексних навичок програмування. Перед учнями було поставлено наступні умови для виконання: програма надає користувачеві ввести три змінні, які обчислюватимуться для пошуку середнього арифметичного з використанням визначеної функції, та у результаті виводить на екран форматований вигляд результату (наприклад, із двома знаками після коми). Завдання оцінювалося за п'ятибальною шкалою, де критеріями виступали коректність роботи програми, логічна цілісність алгоритму, якість коду з точки зору читабельності та дотримання стандартів оформлення, а також ефективність використання функцій для структурування програми (таблиця 2.4).

Таблиця 2.4

Критерії оцінювання практичного завдання (5-бальна шкала)

Бал	Критерій оцінки	Опис
5	Коректність роботи	Програма функціонує без помилок, правильно обчислює середній бал, обробляє введення користувача.
4	Логічна цілісність	Алгоритм є логічним та ефективним. Використано функцію, але є невеликі недоліки у структурі.
3	Якість коду	Код працює, але має помітні проблеми з читабельністю, відсутнє форматування результату
2	Часткова реалізація	Здійснено спробу реалізувати програму, але функція не використовується або містить значні логічні/синтаксичні помилки.
1	Нереалізоване завдання	Завдання не виконано або код не має функціональної цінності.

Обґрунтування вибору практичного завдання «Калькулятор середнього балу» для вхідної діагностики ґрунтується на його комплексному та репрезентативному характері, що дозволяє одночасно оцінити низку фундаментальних компетентностей учнів у програмуванні. Дане завдання інтегрує кілька ключових аспектів роботи з мовою Python: організацію введення даних за допомогою стандартної функції **input()**, перетворення типів даних, виконання арифметичних операцій, структурування коду шляхом виокремлення

логіки обчислення в окрему функцію та форматування виведення результату. Така багатоаспектність робить завдання ефективним інструментом для вимірювання як синтаксичних знань, так і алгоритмічного мислення на початковому етапі навчання.

Діагностична цінність завдання полягає у його чутливості до типових труднощів, з якими стикаються учні-початківці. Аналіз результатів дозволяє чітко ідентифікувати проблемні зони: нерозуміння різниці між типами даних, помилки в оголошенні та виклику функцій, логічні помилки в алгоритмах або невиконання умов форматування виведення [6]. Таким чином, виконання або невиконання умов завдання дає об'єктивну картину початкового рівня, класифікуючи учнів від тих, хто не може реалізувати базовий алгоритм, до тих, хто демонструє вміння створювати структурований, технічно вірний і повністю відповідний умові код. Критерії оцінювання, представлені у таблиці 2.4 дослідження, забезпечують стандартизований та об'єктивний підхід до аналізу якості рішень.

Практична релевантність та зрозумілість контексту завдання — обчислення середнього балу — усуває когнітивне навантаження, пов'язане з розумінням складного предметного поля, і дозволяє учням сконцентруватися суто на програмній реалізації. Крім того, завдання має властивість масштабованості: воно містить мінімальний бар'єр входу для усіх учнів, водночас залишаючи простір для демонстрації більш якісного рішення, наприклад, через використання циклів, списків або обробку винятків, що дозволяє диференціювати рівні підготовки.

Отже, «Калькулятор середнього балу» є не випадковим, а методично вагомим інструментом вхідної діагностики. Його комплексність, об'єктивність оцінювання та прогностична здатність щодо подальшого успіху в опануванні Python роблять його оптимальним вибором для констатації початкового рівня сформованості ключових компетентностей у рамках педагогічного експерименту.

На підставі результатів теоретичного аналізу та експериментальної перевірки для ефективного впровадження розробленого курсу у практику шкільного навчання пропонується дотримуватися комплексу методичних принципів. Організаційно-педагогічною основою має стати інтеграція курсу як факультативного компоненту для учнів 7-х класів, що дозволить компенсувати часові обмеження базової програми та забезпечити поглиблене опанування об'єктно-орієнтованого програмування. Ключовими передумовами є стабільне технічне забезпечення (інтернет-з'єднання, доступ до середовищ виконання Python) та попередня підготовка педагога, який має пройти верифікацію в системі та активувати преміум-доступ «Мій+» для отримання повноцінних авторських та аналітичних інструментів.

Методика реалізації навчального процесу ґрунтується на дотриманні чіткої модульної послідовності, що забезпечує перехід від систематизації базових знань до інтеграції складних концепцій ООП з розробкою графічних інтерфейсів засобами Tkinter. Оптимальною організаційною формою визнається змішана модель, що поєднує очні заняття для пояснення ключових концепцій та організації проєктної взаємодії з самостійною роботою учнів в середовищі ЕОС «МійКлас» [11]. При цьому необхідно цілеспрямовано використовувати функціональні можливості платформи: «Конструктор уроку» для публікації авторських матеріалів, завдання з автоматичною генерацією варіантів для тренування навичок, модульні тести для поетапного контролю, а також аналітичні звіти для виявлення індивідуальних труднощів та диференціації навчальних траєкторій.

Використання розробленого курсу в середовищі «МійКлас» є доцільним не як універсальна заміна традиційного уроку, а як його системний компонент, інтегрований у конкретні етапи для досягнення певних дидактичних цілей. На *етапі актуалізації опорних знань та мотивації* платформу ефективно застосовувати для швидкого вхідного тестування (інструмент «Швидкий тест») з метою діагностики рівня засвоєння попереднього матеріалу та концентрації

уваги учнів на майбутній темі. Це дозволяє вчителю отримати миттєву об'єктивну картину та оперативно скоригувати план пояснення нового матеріалу.

Під час *основного етапу вивчення нового матеріалу* ЕОС виступає потужним інструментом для організації самостійної роботи учнів із авторськими теоретичними модулями курсу. Особливість полягає в тому, що пояснення ключових концепцій (наприклад, синтаксису конструктора `__init__` або принципів роботи менеджера геометрії `grid()` у Tkinter) відбувається в очному форматі, після чого учні переходять до платформи для детального ознайомлення з текстово-графічними матеріалами, перегляду вбудованих відеофрагментів та аналізу готових прикладів коду, які можна безпосередньо копіювати для подальшого експериментування.

На *етапі первинного закріплення та відпрацювання навичок* особлива цінність «МійКлас» полягає у можливості виконання вправ з автоматичною перевіркою. Учні можуть одразу отримати зворотний зв'язок щодо правильності відповідей на теоретичні запитання або результатів виконання нескладних програмних завдань (наприклад, на визначення результату роботи невеликого фрагмента коду). Ця миттєвість сприяє усвідомленню помилок та мотивує до їх виправлення. Для практичних навичок програмування, що вимагають написання власного коду, платформа використовується як проміжний ланцюжок: учні ознайомлюються з умовою завдання в системі, а безпосереднє кодування та налагодження відбувається в зовнішньому середовищі (IDE), після чого готовий код або скріншот результату його роботи може бути завантажений як відповідь для перевірки вчителем.

Етап домашнього завдання та поточного контролю знаходить в ЕОС свою найбільш органічну реалізацію. Вчитель може призначити як обов'язкові завдання з базового курсу, так і додаткові вправи різного рівня складності з розділу «Мій+», забезпечуючи тим самим диференціацію. Особливістю є автоматизоване оцінювання більшості завдань та накопичення всієї статистики у цифровому щоденнику учня та журналі вчителя. Це дозволяє перетворити домашню роботу з рутинної перевірки на інструмент формувального

оцінювання, де кожен учень отримує можливість працювати у власному темпі, а вчитель — аналізувати типові помилки класу за зручними автоматично згенерованими звітами.

Таким чином, методично обґрунтоване впровадження курсу в «МійКлас» перетворює платформу з простого репозиторію завдань на середовище організації повного навчального циклу: від мотивації та пояснення через самостійну роботу до автоматизованого тренування, диференційованого домашнього завдання та об'єктивної аналітики результатів, що значно підвищує ефективність управління навчальним процесом.

Система оцінювання досягнень учнів має мати комплексний характер, інтегруючи кількісні показники тематичного тестування в ЕОС, якісну оцінку практичних програмних робіт за критеріями коректності, логіки та чистоти коду, а також результати захисту фінального індивідуального або групового проєкту, що демонструє рівень інтеграції набутих знань. Практична реалізація методики вимагає від вчителя проведення вступного інструктажу щодо роботи з платформою, поєднання групових обговорень складних тем на очних заняттях з подальшим індивідуальним закріпленням онлайн, а також використання даних аналітики для формування збалансованих проєктних груп. Таким чином, запропоновані рекомендації спрямовані на трансформацію ролі педагога з інформатора в фасилітатора, що забезпечує умови для розвитку алгоритмічного мислення, самостійності та творчого потенціалу учнів у процесі навчання програмуванню.

2.4. Аналіз результатів педагогічного експерименту та визначення ефективності запропонованої методики

Для отримання узагальнених балів експериментальної та контрольної груп у відсотковому вираженні використовувалась процедура нормалізації оцінок до єдиної стобальної шкали. Це дозволило порівнювати результати за різними за своєю природою компонентами діагностики: когнітивним, афективним та компетентісним.

Когнітивний аспект оцінювався за допомогою тесту з 15 завдань закритого типу. Кількість набраних учнем балів ділилась на максимально можливу суму та множилась на 100%, що давало індивідуальний відсотковий показник теоретичних знань. Афективний аспект вимірювався за допомогою опитувальника з п'яти тверджень за шкалою Лікерта. Сума балів учня (від 5 до 25) переводилась у відсотки від максимуму (25 балів). Компетентісний аспект оцінювався за п'ятибальною шкалою за практичне завдання «Калькулятор середнього балу». Отриманий бал ділився на 5 та множився на 100%.

Середній груповий показник по кожному аспекту розраховувався як середнє арифметичне індивідуальних відсоткових оцінок усіх учнів відповідної групи (Додаток А). Така процедура забезпечила можливість прямого порівняння рівня сформованості ключових компетентностей між групами на початковому та підсумковому етапах експерименту, а також подальшу статистичну обробку даних непараметричним методом Манна-Уїтні на основі індивідуальних відсоткових значень.

Аналіз отриманих даних підтвердив, що на початковому етапі рівень сформованості ключових компетентностей з програмування був загалом низьким в обох групах, що є типовим для початку вивчення нової дисципліни. Щодо мотиваційного компонента, середній інтегральний показник становив 2.8 бала в ЕГ та 2.9 бала в КГ за п'ятибальною шкалою Лікерта, тобто 56% у ЕГ та 58% КГ відповідно. Ці значення відповідають помірному або нейтрально-позитивному вихідному рівню інтересу до предмета, але без високої готовності до самостійного подолання труднощів.

За результатами оцінювання теоретичного блоку, представлених у Додатку А, учні експериментальної групи продемонстрували посередні бали: переводячи успішність групи у відсотки, рівень знань ЕГ на етапі пре-тесту складає 48%, тоді як у контрольній групі – 51%. Тут і далі у кваліфікаційній роботі вказані саме середні показники груп у відсотковому співвідношенні. Це незначне відхилення у 3 процентні пункти не є статистично значущим. За практичною складовою, яка вимагала розробки функціональної програми «Калькулятор середнього балу»,

результати також були низькими та подібними: 42% в експериментальній групі проти 45% у контрольній. Ці низькі показники свідчать про фрагментарне володіння базовим синтаксисом та нездатність до застосування комплексних алгоритмічних структур, зокрема, коректного використання функцій, що підтверджує необхідність систематичного навчання – результати пре-тесту представлено у таблиці 2.5 та рисунку 2.11.

Таблиця 2.5

Узагальнені результати констатувального етапу

Показник	ЕГ (13 людей)	КГ (12 людей)
Навчальна Мотивація (Афективний аспект)	56%	58%
Теоретичні Знання (Когнітивний аспект)	42%	45%
Практичні Навички (Компетентнісний аспект)	48%	51%

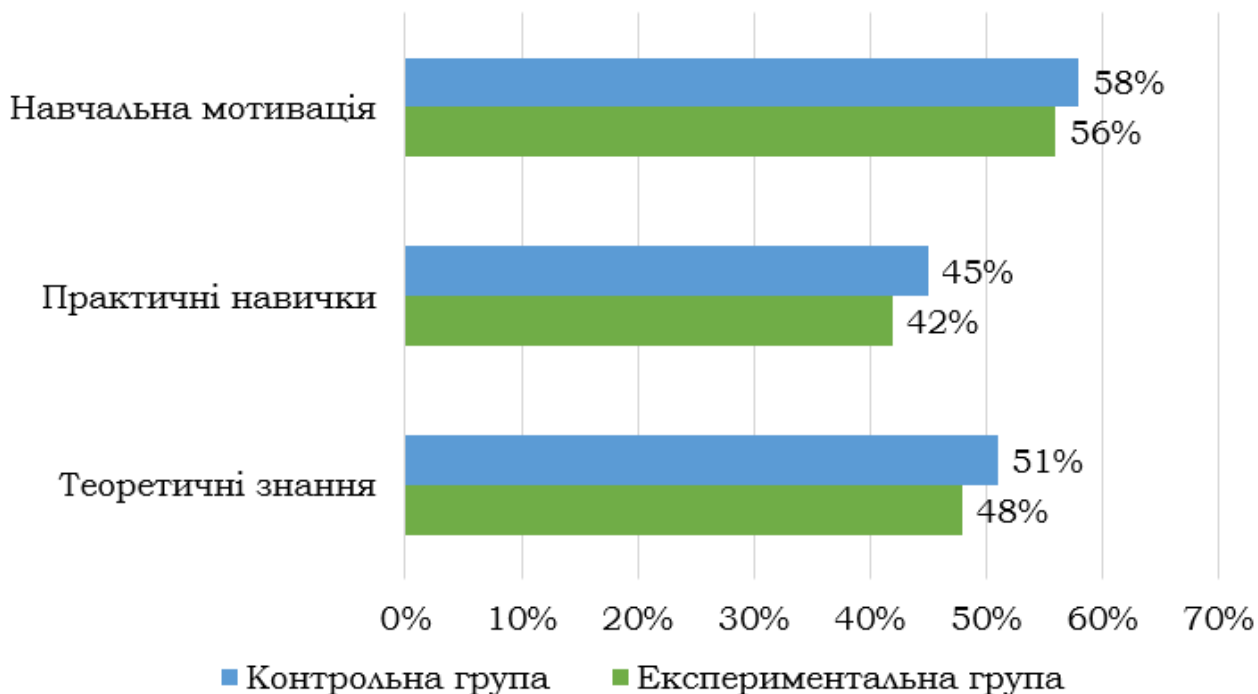


Рис 2.11. Діаграма результатів першого етапу експерименту у відсотках.

На основі наведених даних для таблиці індивідуальних результатів пре-тесту зробимо класифікацію учнів за рівнем знань, виходячи із їх середнього

значення у відсотках. Так, низький рівень відповідає результатам менше 50%, середній у діапазоні від 50% до 80%, а високий – вище 80%. На початковому етапі в обох групах переважає середній та низький рівень засвоєння матеріалу. На основі результативних даних у Додатку А, у контрольній групі спостерігається трохи більша частка учнів зі середнім рівнем (8 з 12) порівняно з експериментальною (6 з 13). Високого рівня не має жоден учень, що свідчить про загально низький стартовий рівень компетентностей з програмування, що є характерним для початку навчання новій дисципліні.

Для статистичної оцінки різниць між експериментальною (ЕГ, $n_1=13$) та контрольною (КГ, $n_2=12$) групами на етапах вхідної та підсумкової діагностики було застосовано непараметричний U-критерій Манна-Уїтні. Даний критерій дозволяє перевірити нульову гіпотезу про відсутність систематичних зрушень у розподілі рангів між групами. Розрахунок виконувався на основі індивідуальних середніх процентних оцінок учнів, отриманих за трьома аспектами (мотивація, теоретичні знання, практичні навички), які були попередньо проранжовані в межах об'єднаної вибірки ($N = 25$) (Додаток А та В). Для ЕГ сума рангів пре-тестування мають значення $\sum R_1 = 121$, а для КГ $\sum R_2 = 174$ У статистика для кожної групи обчислюється за формулами:

$$U_1 = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - \sum R_1$$

$$U_2 = n_1 n_2 + \frac{n_2(n_2 + 1)}{2} - \sum R_2$$

де $n_1 n_2 = 13 \times 12 = 156$.

Отже, для пре-тесту результати обчислень виглядатимуть наступним чином:

$$U_1 = 156 + \frac{13 \times 14}{2} - 121 = 156 + 91 - 121 = 126$$

$$U_2 = 156 + \frac{12 \times 14}{2} - 174 = 156 + 84 - 174 = 66$$

Для визначення значущості отриманих значень U було використано табличні критичні значення для двостороннього тесту на рівні значущості $\alpha =$

0,05. При $n_1 = 13$ та $n_2 = 12$ критичне значення становить $U_{\text{крит}} = 37$. $U_{\text{пре}} = 60 > U_{\text{крит}} = 37$, отже, нульова гіпотеза про відсутність різниць між групами на вхідному етапі не відхиляється ($p > 0,05$). Коли обидві вибірки є великими (близько $n = 20$) і нульова гіпотеза вірна (різниця між групами мінімальна), розподіл U-статистики Манна-Уїтні має тенденцію наближатися до нормальної форми. В цьому випадку гіпотези Манна-Уїтні можуть бути оцінені з використанням статистики z-показника і одиничного нормального розподілу. Процедура для цього нормального наближення наступна:

1. Знайти значення U для вибірки А і вибірки В, як і раніше Манн-Уїтні U - менше з цих двох значень.

2. Коли обидві вибірки відносно великі (близько $n = 20$ або більше), розподіл U-статистики Манна-Уїтні має тенденцію формувати нормальний розподіл з

$$\mu = \frac{n_1 n_2}{2} \text{ та } \sigma = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}$$

Манна-Уїтні U, отриманий з даних вибірки, може бути розташований в цьому розподілі, використовуючи z-показник:

$$z = \frac{U - \mu}{\sigma} = \frac{60 - \frac{156}{2}}{\sqrt{\frac{156(13 + 12 + 1)}{12}}} = -0,979$$

Критичне значення z для $\alpha = 0,05$ двостороннього тесту становить $\pm 1,96$. І після виконання цих дій, стає можливо визначити стартову величину ефекту за рекомендаціями Коена, а саме за формулою:

$$r = \frac{|z|}{\sqrt{N}} = \frac{0,979}{\sqrt{25}} \approx 0,196$$

Відповідно до інтерпретаційних стандартів Коена, розмір ефекту на етапі пре-тесту є малим ($r < 0,3$), що підтверджує практичну однорідність груп на початку експерименту. Таким чином, було встановлено статистичну еквівалентність вибірок, що дозволяє віднести будь-які статистично значущі

позитивні зрушення на контрольному етапі виключно за рахунок ефективності запропонованої експериментальної методики.

Після завершення навчального модуля було проведене підсумкове тестування з дотриманням визначених напрямів за умови підвищення складності та актуалізації питань до вивченого матеріалу, результати якого виявили суттєві відмінності між групами. Експериментальна група продемонструвала середній бал знань на рівні 87%, що на 39% вище початкових показників, тоді як контрольна група досягла лише 65% з приростом на 14%. У сфері практичних навичок спостерігалась подібна динаміка: експериментальна група покращила результати на 40% до 82%, а контрольна група лише на 13% до 58% (таблиця 2.6 та рис. 2.12).

Таблиця 2.6

Узагальнені результати підсумкового етапу

Показник	ЕГ (13 людей)	КГ (12 людей)
Навчальна Мотивація (Афективний аспект)	86%	64%
Теоретичні Знання (Когнітивний аспект)	87%	65%
Практичні Навички (Компетентнісний аспект)	82%	58%

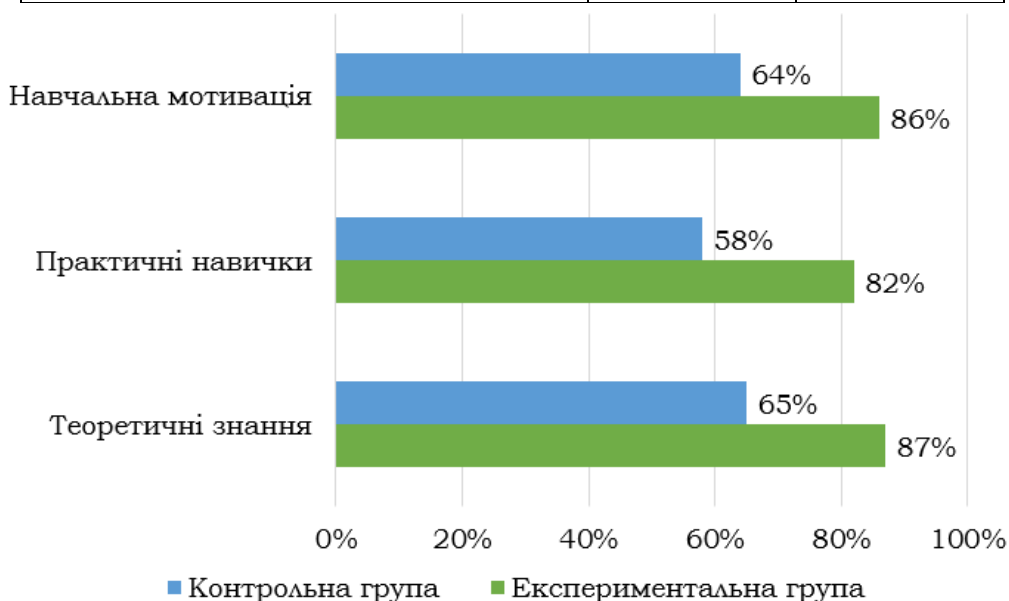


Рис.2.12. Діаграма результатів експерименту у відсотках.

Для аналізу підсумкових результатів застосовано ті ж критерії класифікації, що й на етапі пре-тесту: низький рівень відповідає значенням нижче 50%, середній – від 50% до 80%, а високий – показникам понад 80%. В цьому розподілі усі 12 учнів КГ мають результати у середньому діапазоні, а 13 з ЕГ продемонстрували високий рівень знань та практичних навичок.

Тепер, коли результати пост-тестування проаналізовано та проранговано (додатки Б та Г), їх суми дорівнюють $\sum R_1 = 247$ для ЕГ, а для КГ $\sum R_2 = 78$. Задля перевірки присвоєння рангів перевіряємо загальну їх суму, яка у пре- та пост-тестуваннях дорівнює теоретичній сумі $\frac{N(N+1)}{2} = 325$. Визначаємо мінімальне значення U серед двох груп за тими ж формулами, що і раніше:

$$U_1 = 156 + 91 - 247 = 0$$

$$U_2 = 156 + 78 - 78 = 156$$

Тепер мінімальне значення, що дозволяє відхилити нульову гіпотезу та констатувати наявність статистично значущої різниці між групами після впровадження експериментальної методики ($p < 0,05$). Обчислюючи нормальне наближення для інформативності аналізу за попередньо описаною формулою:

$$z = \frac{U - \mu}{\sigma} = \frac{0 - \frac{156}{2}}{\sqrt{\frac{156(13 + 12 + 1)}{12}}} = -4,243$$

Отримані результати підтверджують висновки, зроблені на основі табличних значень: $z_{\text{пре}} < 1,96$ (незначуще), $z_{\text{пост}} > 1,96$ (значуще). Тобто тепер величину ефекту за рекомендаціями Коена дорівнюватиме:

$$r = \frac{|z|}{\sqrt{N}} = \frac{4,243}{\sqrt{25}} \approx 0,849$$

На етапі пост-тесту розмір ефекту є великим ($r > 0,5$), що свідчить про значний вплив експериментальної методики на результативність навчання. Застосування U -критерію Манна-Уїтні дозволило об'єктивно продемонструвати відсутність статистично значущих відмінностей між групами до впровадження методики та наявність високо значущих відмінностей після її застосування, що

підтверджує ефективність використання ЕОС «МійКлас» у навчанні програмуванню.

Висновки до другого розділу

Відмітимо, що особливості використання платформи полягають у її здатності ефективно адресувати ключові методичні виклики через цілісну цифрову інфраструктуру. Платформа забезпечує системне впровадження принципів диференційованого навчання та формувального оцінювання шляхом автоматизованої генерації персоналізованих завдань, надання миттєвого об'єктивного зворотного зв'язку та детальної аналітики результатів. Це дозволяє організувати практико-орієнтовану роботу з кодом, розвиваючи алгоритмічну культуру та операційні навички програмування, при цьому значно оптимізуючи час педагога на рутинні операції. Проаналізовано можливості платформи та обґрунтовано доцільність її використання як інструменту підтримки навчального процесу з інформатики.

У ході дослідження було визначено, що стандартний курс інформатики не забезпечує достатньої кількості часу та практичних завдань для якісного засвоєння основ програмування, зокрема об'єктно-орієнтованого підходу. У зв'язку з цим розроблено авторський 35-годинний курс «Програмування мовою Python» для учнів 7 класу, який має чітку модульну структуру з шести тематичних блоків, поєднує теоретичний матеріал із практичними та проєктними завданнями та враховує попередній рівень підготовки учнів. Описано процес створення та наповнення навчального курсу в системі «МійКлас», зокрема додавання теоретичних матеріалів, тестових завдань і практичних вправ, що дозволяє організувати диференційоване навчання, здійснювати контроль знань та відстежувати індивідуальний прогрес учнів.

Результати педагогічного експерименту, проведеного в умовах Бродівської спеціалізованої школи №2, показали статистично значущу ефективність запропонованої методики. У експериментальній групі, що навчалася з використанням розробленого курсу в середовищі «МійКлас», спостерігався

суттєвий приріст у всіх діагностованих аспектах: рівень теоретичних знань зріс з 48% до 82%, практичних навичок – з 42% до 86%, а мотиваційний компонент покращився з 56% до 87%. Контрольна група, яка навчалася за традиційною методикою, продемонструвала значно скромнішу динаміку (від 51% до 65%, 45% до 58%, та 58% до 64% відповідно). Статистична обробка даних за допомогою U-критерію Манна-Уїтні підтвердила наявність високо значущих відмінностей між групами після експерименту (розмір ефекту $r = 0,849$). Це підтверджує ефективність запропонованої методики та доцільність її впровадження у практику навчання інформатики в закладах загальної середньої освіти.

ВИСНОВКИ

На основі виконання поставлених завдань кваліфікаційної роботи отримано низку ключових результатів. У теоретичній частині підтверджено, що мова програмування Python є пріоритетним інструментом для впровадження у шкільний курс інформатики, що обумовлено її синтаксичною простотою, універсальністю та відповідністю вимогам компетентнісного підходу Нової української школи. Одночасно виявлено суттєві методичні прогалини у сучасній практиці викладання, зокрема обмеженість навчального часу для формування практичних навичок, недостатній рівень індивідуалізації навчального процесу та складність оперативного та об'єктивного контролю результатів великої кількості учнів.

Обґрунтовано необхідність та ефективність залучення сучасних електронних освітніх систем для подолання зазначених обмежень. Зокрема, встановлено, що платформа «МійКлас» володіє значним дидактичним потенціалом для навчання програмування, що реалізується через автоматизовану перевірку програмного коду, надання миттєвого зворотного зв'язку, можливість створення завдань різного рівня складності та наявність інструментів для самостійної роботи учнів. Ці особливості роблять платформу ефективним засобом для тренувальної роботи, формувального оцінювання та організації індивідуальних освітніх траєкторій.

У практичному аспекті дослідження спроектовано та детально розроблено методичну систему інтеграції ЕОС «МійКлас» у процес навчання Python. Розроблено авторський 35-годинний курс для учнів 7-х класів, що має чітку модульну структуру з шести тематичних блоків, поєднує теоретичне викладення основ об'єктно-орієнтованого програмування, роботу з графічним інтерфейсом Tkinter та проєктну діяльність. Методика включає конкретні рекомендації щодо планування уроків, створення навчальних матеріалів у середовищі платформи та алгоритму роботи учня, що забезпечує послідовний перехід від вивчення теорії до практичного кодування, аналізу помилок та корекції розв'язків.

Експериментальна перевірка ефективності запропонованої методики, проведена в умовах педагогічного експерименту, засвідчила позитивну динаміку в усіх діагностованих аспектах. У експериментальній групі спостерігалось статистично значуще підвищення рівня теоретичних знань з 48% до 87%, практичних навичок з 42% до 82% та мотиваційного компонента з 56% до 86%. Результати контрольної групи, яка навчалася за традиційною методикою, були значно скромнішими. Обробка даних за допомогою U-критерію Манна-Уїтні підтвердила високий розмір ефекту ($r = 0,849$), що свідчить про практичну значущість та ефективність запропонованого методичного підходу з використанням ЕОС «МійКлас» для покращення якості навчання програмуванню мовою Python у закладах загальної середньої освіти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Вапнічний С. Д. Навчання початківців програмування на уроках інформатики *Вісник Національного університету «Чернігівський колегіум» імені Т. Г. Шевченка. Серія: Педагогічні науки*. 2021. Вип. 4. С. 128–136.
2. Вовкодав О.В. Інноваційні аспекти застосування теорії об'єктно-орієнтованого програмування в умовах дистанційного навчання. *Сучасні цифрові технології та інноваційні методики навчання: досвід, тенденції, перспективи: матеріали XIV Міжнародної науково-практичної інтернет-конференції, 7–8 листопада, 2024 р. м. Тернопіль*. Тернопіль: ТНПУ ім. В. Гнатюка, 2024. С. 32-33.
3. Глушко О. Компетентнісний підхід в освіті: європейський досвід. *Науково-педагогічні студії*. 2021. Вип. 5. С. 8–21.
4. Жалдак М.І. Проблеми інформатизації навчального процесу в середніх і вищих навчальних закладах. *Комп'ютер в школі та сім'ї*. Вип. 3. 2013. С. 8-15.
5. Квак П.А. Методи навчання школярів програмуванню на основі створення комп'ютерної гри. *Педагогічні науки та освіта*. 2024. Вип. XLVI–XLVII. С. 89–96.
6. Кобильник Т.П., Когут У.П., Жидик В.Б. Методичні аспекти вивчення основ алгоритмізації і програмування мовою Python у шкільному курсі інформатики у старших класах. *Фізико-математична освіта*, 2021. Випуск 5(31). С. 36-44.
7. Ковальчук В., Винницька Н., Жигайло О. Використання цифрових інструментів, дієвих практик в процесі формування математичної та інформаційно-комунікаційної компетентностей учнів. *Молодь і ринок*. 2025. № 4 (236). С. 17–21.
8. Ковтанюк М.С., Тітова Л.О. Використання ігрових симуляторів під час вивчення програмування. *Комп'ютерні технології: інновації, проблеми, рішення: матеріали. IV Всеукраїнської науково-технічної конференції, (м. Житомир, 18–20 листопада 2021 р.)*. Житомир: Державний університет «Житомирська політехніка», 2021. С. 95–96.

9. МійКлас: інформація для вчителів. URL: miyklas.com.ua (дата звернення: 14.04.2025).

10. Освітня програма Бродівської спеціалізованої загальноосвітньої школи I-III ступенів №2 з вивченням англійської мови Бродівської міської ради Львівської області на 2024/2025 навчальний рік. URL: <https://brody2.school.org.ua/osvitnya-programa-12-01-42-14-01-2023/> (дата звернення 18.08.2025)

11. Павелків О. Особливості вивчення дисциплін математичного циклу в закладі вищої освіти в умовах змішаного навчання . *Інноватика у вихованні*. 2022. Вип. 16. С. 190–201.

12. Підручник з інформатики для 7 кл. закладів загальної середньої освіти / О. О. Бондаренко, В. В. Ластовецький, О. П. Пилипчук, Є. А. Шестопапов. Харків: Вид-во «Ранок», 2020. 159 с.

13. Пометун О. І., Жиляєв І. В. Гейміфікація у навчанні: методологія і практика : монографія. Київ : Педагогічна думка, 2020. 215 с.

14. Про оновлені навчальні програми для учнів 5-9 класів загальноосвітніх навчальних закладів від 07.06.2017 № 804. URL: <https://zakon.rada.gov.ua/rada/show/v0804729-17#Text> (дата звернення: 16.08.2025)

15. Ракута В.М. Python у шкільному курсі інформатики. Основи програмування: навчальний посібник. Чернігів : ЧОППО ім. К.Д. Ушинського, 2024. 160 с.

16. Семеніхіна О.В., Руденко Ю.О. Проблеми навчання програмувати учнів старших класів та шляхи їх подолання. *Інформаційні технології і засоби навчання*. 2018. Том 66, № 4. С. 54-61.

17. Смолин Я. В. Особливості викладання інформаційних дисциплін. Академічні візії. 2025. Випуск 44.

18. Струк О.О., Струк О.С. Особливості вивчення PYTHON в різних формах навчання. *Сучасні інформаційні технології та інноваційні методики навчання: досвід, тенденції, перспективи: матеріали V Міжнародної науково-*

практичної інтернет-конференції (м. Тернопіль, 30 квітня, 2020). Тернопіль: ТНПУ ім. В. Гнатюка, 2020. С. 16-18

19. Теорія та методика навчання математики, фізики, інформатики : збірник наукових праць. Випуск XI : в 3-х томах. Кривий Ріг: Видавничий відділ КМІ, 2013. 200 с.

20. Цифровізація освіти та використання сучасних інструментів цифрового навчання на прикладі міжнародної освітньої платформи «МійКлас» URL: <https://miyklas.com.ua/> (дата звернення 05.04.2025)

21. Ящик О.Б. Компетентнісний підхід у навчанні об'єктно-орієнтованого програмування як основа підготовки учнів старших класів. *Наукові записки Тернопільського національного педагогічного університету ім. В. Гнатюка*. Серія: Педагогіка. 2011. Вип. 1. – С. 103-107.

22. Korotun O.V., Vakaliuk T.A., Makhno A.M. Tools for Teaching the R Programming Language to Bachelors of Computer Science in the Period of Distance Learning. *E-Learning and Enhancing Soft Skills: Contemporary Models of Education in the Era of Artificial Intelligence*. (October 14–15, 2024). Cieszyn: Faculty of Arts and Educational Sciences, 2024. P.309-330

23. Morze N.V., Barna O.V. The Concept of Teaching Secondary Students Programming Using Artificial Intelligence. *E-Learning and Enhancing Soft Skills: Contemporary Models of Education in the Era of Artificial Intelligence*. (October 14–15, 2024). Cieszyn: Faculty of Arts and Educational Sciences, 2024. P.37-59.

24. Mukundan G. Google Classroom Basics URL: <https://hiverhq.com/blog/google-classroom-basics> (дата звернення: 24.04.2025)

25. Pramudita D.A., Afiyanto A. Block-Based Visual Programming as a Tool for Learning the Concepts of Programming for Novices. *International Journal of Information and Education Technology*. 2022. Vol. 12, Iss. 5. P. 365-371.

ДОДАТКИ

Таблиця індивідуальних результатів вхідної діагностики у % (пре-тест)

Група	Учень	Мотивація	Теоретичні знання	Практичні навички	Середнє Значення
ЕГ	1.	60	50	45	51,67
	2.	55	48	40	47,67
	3.	58	52	42	50,67
	4.	52	46	38	45,33
	5.	62	54	48	54,67
	6.	56	49	43	49,33
	7.	54	47	41	47,33
	8.	59	51	44	51,33
	9.	57	50	42	49,67
	10.	53	45	37	45
	11.	61	53	47	53,67
	12.	55	48	40	47,67
	13.	58	50	43	50,33
КГ	14.	62	54	58	54,67
	15.	57	50	44	50,33
	16.	60	52	46	52,67
	17.	55	49	42	48,67
	18.	58	51	45	51,33
	19.	56	48	41	48,33
	20.	61	53	47	53,67
	21.	54	47	40	47
	22.	59	52	45	52
	23.	57	50	43	50
	24.	56	49	42	49
	25.	58	51	44	51

Таблиця індивідуальних результатів вхідної діагностики у % (пост-тест)

Група	Учень	Мотивація	Теоретичні знання	Практичні навички	Середнє Значення
ЕГ	1.	90	88	89	89
	2.	85	80	82	82,33
	3.	88	85	87	86,67
	4.	82	78	80	80
	5.	92	90	93	91,67
	6.	86	83	84	84,33
	7.	84	81	83	82,67
	8.	89	87	88	88
	9.	89	85	86	86
	10.	83	79	81	81
	11.	91	89	92	90,67
	12.	85	82	84	83,67
КГ	13.	88	86	88	87,33
	14.	68	60	66	64,67
	15.	64	56	62	60,67
	16.	70	63	68	67
	17.	62	54	60	58,67
	18.	67	59	65	63,67
	19.	65	57	63	61,67
	20.	69	62	67	66
	21.	61	53	59	57,67
	22.	66	58	64	62,67
	23.	63	55	61	59,67
	24.	64	56	62	60,67
	25.	67	59	65	63,67

**Присвоєння рангів за результатами пре-тесту для обчислення критерію U
Манна-Уїтні**

Група	Учень	Середнє значення у %	Ранг
ЕГ	10	45,00	1
ЕГ	4	45,33	2
КГ	21	47,00	3
ЕГ	7	47,33	4
ЕГ	2	47,67	5,5
ЕГ	12	47,67	5,5
КГ	19	48,33	7
КГ	17	48,67	8
КГ	24	49,00	9
ЕГ	6	49,33	10
ЕГ	9	49,67	11
КГ	23	50,00	12
ЕГ	13	50,33	13,5
КГ	15	50,33	13,5
ЕГ	3	50,67	15
КГ	25	51,00	16
ЕГ	8	51,33	17,5
КГ	18	51,33	17,5
ЕГ	1	51,67	19
КГ	22	52,00	20
КГ	16	52,67	21
ЕГ	11	53,67	22,5
КГ	20	53,67	22,5
ЕГ	5	54,67	24,5
КГ	14	54,67	24,5

Присвоєння рангів за результатами пост-тесту для обчислення критерію U

Манна-Уїтні

Група	Учень	Середнє зн. у %	Ранг
КГ	21	57,67	1
КГ	17	58,67	2
КГ	23	59,67	3
КГ	15	60,67	4,5
КГ	24	60,67	4,5
КГ	19	61,67	6
КГ	22	62,67	7
КГ	18	63,67	8,5
КГ	25	63,67	8,5
КГ	14	64,67	10
КГ	20	66,00	11
КГ	16	67,00	12
ЕГ	4	80,00	13
ЕГ	10	81,00	14
ЕГ	2	82,33	15
ЕГ	7	82,67	16
ЕГ	12	83,67	17
ЕГ	6	84,33	18
ЕГ	9	86,00	19
ЕГ	3	86,67	20
ЕГ	13	87,33	21
ЕГ	8	88,00	22
ЕГ	1	89,00	23
ЕГ	11	90,67	24
ЕГ	5	91,67	25