Класичний приватний університет
Інститут управління
Кафедра інформаційних технологій та дизайну

# КУРСОВА РОБОТА
# з дисципліни Сучасні мови програмування

**Методичні вказівки для здобувачів другого (магістерського) рівня вищої освіти спеціальності 014 Середня освіта, спеціалізації 014.09 Інформатика, освітня програма «Інформатика»**

Запоріжжя – 2024

Методичні рекомендації до виконання курсових робіт з дисципліни “Сучасні мови програмування (HTML, CSS, JavaScript)” для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 014 Середня освіта, спеціалізації 014.09 Інформатика, освітня програма «Інформатика»/ Класичний приватний університет. – Запоріжжя : КПУ, 2024. – 20 с.

Розробник: проф., Хрипко С.Л., ст.викл. Бречко Н.А.

Рекомендовано кафедрою інформаційних технологій та дизайну, Протокол № 1 від «_28_» ___08___2024 року

# ВСТУП

Відповідно до навчального плану студенти спеціальності 014 Середня освіта (Інформатика) освітнього ступеня «бакалавр» виконують і захищають курсову роботу з дисципліни «Сучасні мови програмування» у 7 семестрі (для здобувачів скороченого терміну навчання – у 5 семестрі).

Методичні рекомендації мають за мету допомогти студентам при виконанні курсової роботи й визначають послідовність етапів її написання, містять рекомендації щодо підбору й опрацювання літературних джерел (технічної документації мов HTML, CSS, JavaScript та науково-методичних праць), збору й опрацювання фактичного матеріалу (програмного коду та результатів його апробації), визначають вимоги, які висуваються до змісту й оформлення курсової роботи, визначають порядок захисту й критерії оцінювання роботи.

Дотримання студентами усіх вимог щодо планування, розробки програмного продукту та оформлення курсових робіт сприяє вихованню у них належного професійного стилю роботи, формує навички до ведення науково-педагогічного дослідження в галузі сучасних інформаційних технологій та методики навчання програмуванню.

## 1. МЕТА ТА ЗАВДАННЯ КУРСОВОЇ РОБОТИ

Курсова робота з дисципліни «Сучасні мови програмування» є одним з основних видів самостійної роботи студентів і спрямована на закріплення, поглиблення й узагальнення знань з об’єктно-орієнтованого програмування, оволодіння сучасними методами розробки програмного забезпечення мовами HTML, CSS, JavaScript, а також формування навичок вирішення актуальних науково-методичних проблем навчання інформатики в закладах середньої освіти.

Основними завданнями курсової роботи є формування у студентів вмінь:

- сформулювати відповідно до теми роботи мету і завдання;
- зібрати й опрацювати теоретичні матеріали з обраної теми;
- критично проаналізувати отримані технічні та методичні матеріали;
- провести систематизацію й узагальнення зібраної інформації щодо використання сучасних мов програмування в освітньому процесі;
- самостійно вирішити поставлені творчі завдання пов’язані з написанням програмного коду, проектуванням архітектури додатку та реалізацією алгоритмів;
- розробити технологію розв’язання конкретної прикладної проблеми шляхом створення веб-орієнтованого програмного рішення з використанням інструментарію HTML, CSS, JavaScript;
- логічно обґрунтувати й сформулювати висновки, щодо ефективності створеного програмного рішення або запропонованої методики, надати практичні рекомендації щодо масштабування та подальшого використання.

Відповідно до освітньо-професійної програми підготовки бакалавра галузі знань 014 Середня освіта (освітня програма: Інформатика) написання курсової роботи з дисципліни «Сучасні мови програмування» сприяє формуванню компетентностей та програмних результатів навчання:

Інтегральна компетентність:

Здатність розв’язувати складні спеціалізовані задачі у галузі середньої освіти, що передбачає застосування теоретичних знань і практичних умінь з наук предметної спеціальності, педагогіки, психології, теорії та методики навчання і характеризується комплексністю та невизначеністю умов організації освітнього процесу в закладах загальної середньої освіти.

Загальні компетентності:

ЗК 2. Здатність до абстрактного мислення, аналізу та синтезу. Володіння методами логічного та системного аналізу, здатність виявляти закономірності в інформаційних процесах та розв’язувати складні завдання предметної області.

ЗК 3. Інформаційно-комунікаційна компетентність. Здатність ефективно використовувати сучасні ІКТ для пошуку, обробки, критичної оцінки та поширення інформації в освітньому та соціальному просторах.

ЗК 6. Здатність до навчання впродовж життя. Здатність самостійно визначати потреби власного розвитку, опановувати нові знання та технології, бути готовим до професійної адаптації.

ЗК 7. Здатність до креативності та інновацій. Здатність генерувати нові ідеї, виявляти гнучкість у професійних ситуаціях та застосовувати творчий підхід до розв'язання практичних проблем.

ЗК 8. Здатність спілкуватися іноземною мовою. Здатність до комунікації іноземною мовою в професійній діяльності; здатність розуміти іншомовні фахові тексти.

Спеціальні (фахові) компетенції

СК 3. Програмно-технологічна компетентність. Здатність розробляти програмні продукти, використовуючи сучасні мови програмування, знання архітектури ПК, операційних систем та мереж.

Програмні результати навчання:

РН 1. Знає та розуміє фундаментальні засади математики та теоретичної інформатики, достатні для формування предметних компетентностей з інформатики.

РН 3. Використовує сучасні мови, методи та технології програмування для розв’язання практичних завдань і створення програмних продуктів.

РН 7. Будує інформаційні та математичні моделі об’єктів, процесів і явищ; використовує бази даних та технології візуалізації для опрацювання і представлення даних.

РН 15. Демонструє готовність до навчання впродовж життя, професійного самовдосконалення та адаптації до змін у цифровій галузі.

РН 16. Використовує іноземну мову для аналізу фахових джерел та технічної документації, а також для професійної комунікації у цифровому середовищі.

Курсова робота виконується на основі поглибленого вивчення нормативних документів (Державних стандартів загальної середньої освіти), навчальних програм та методичних рекомендацій щодо навчання інформатики у закладах загальної середньої освіти, технічної документації веб-технологій HTML, CSS та JavaScript (офіційна документація W3C, MDN Web Docs), педагогічної та методичної вітчизняної й закордонної літератури, відкритих Інтернет-ресурсів для розробників, а також результатів власних досліджень, які провів студент під час виконання практичних завдань курсу або проходження педагогічної практики.

Показником якісної підготовки майбутнього вчителя інформатики є самостійні програмні розробки студента (зокрема навчальні веб-ресурси, інтерактивні вправи, тести або прикладні веб-застосунки), виконані на матеріалах конкретного розділу шкільного курсу інформатики (наприклад, теми «Алгоритмізація та програмування»).

Висновки й рекомендації, представлені в роботі студента-бакалавра, можуть бути використані в практичній педагогічній діяльності, при створенні дидактичних матеріалів та впроваджені в практику роботи сучасної школи для підвищення ефективності навчання учнів програмуванню

## 2. ЗАГАЛЬНА СТРУКТУРА ТА ОБСЯГ КУРСОВОЇ РОБОТИ

Курсова робота повинна мати наступну структуру:

- Титульний аркуш
- Зміст
- Вступ
- Основна частина (як правило 2 розділи: теоретична частина, практична частина).
- Висновки.
- Список використаних джерел
- Додатки (за необхідності)

Рекомендований обсяг курсової роботи – 25-40 сторінок (при дотриманні норм оформлення тексту). Додатки до рекомендованого обсягу роботи не включаються.

Текст кожної із вказаних структурних частин курсової роботи повинен починатися із нової сторінки.

Зразок оформлення титульного аркуша наведено нижче у Додатку А. Зразок оформлення змісту (орієнтована структура) наведена у Додатку Б.

## ВСТУП

**ВСТУП** містить обґрунтування актуальності теми роботи й основних її характеристик: актуальність проблеми, мета і завдання, об'єкт і предмет дослідження, методи дослідження, теоретична значущість, практичне значення одержаних результатів, апробація результатів курсової роботи (вказується за наявності), структура та обсяг курсової роботи.

При обґрунтуванні **актуальності** обраної теми доцільно показати її значущість у методиці навчання інформатики, зокрема через активний розвиток веб-технологій та необхідність підготовки учнів до створення сучасних інтерактивних цифрових продуктів засобами HTML, CSS і JavaScript. Необхідність розв’язання обраної проблеми зумовлюється потребою підвищення ефективності навчального процесу в закладах загальної середньої освіти, зокрема під час вивчення розділів «Алгоритмізація та програмування», «Основи веб-програмування», «Інформаційні технології». Такі висновки робляться на основі аналітичного огляду досягнень сучасної педагогіки, комп’ютерних наук і методики навчання шкільної інформатики. Короткий огляд науково-методичної літератури, праць з веб-програмування та дидактичних матеріалів для вчителів інформатики має підвести до висновку, що обраний аспект використання HTML, CSS і JavaScript в освітньому процесі потребує подальшого теоретичного осмислення та практичної реалізації. Висвітлення актуальності повинно бути чітким і лаконічним (обсягом до однієї сторінки).

**Мета** – це запланований результат, який має бути отриманий у процесі виконання курсової роботи. У межах курсової роботи, присвяченої використанню HTML, CSS і JavaScript, мета спрямовується на вдосконалення цифрових компетентностей учнів, розробку ефективних веб-орієнтованих засобів навчання (інтерактивних вправ, навчальних веб-застосунків, візуалізаторів алгоритмів) або методики вивчення програмування у шкільному курсі інформатики. Мета формулюється одним-двома реченнями.

**Завдання** курсової роботи конкретизують і уточнюють її мету. Доцільно сформулювати 3-5 завдань, використовуючи такі формулювання:

- *проаналізувати* психолого-педагогічну, методичну та технічну літературу з проблеми використання HTML, CSS і JavaScript у навчанні інформатики;
- *обґрунтувати* доцільність вибору веб-технологій HTML, CSS і JavaScript для розв’язання навчальних завдань у шкільному курсі інформатики;
- *розробити* навчальний веб-ресурс або інтерактивні завдання для учнів;
- *підготувати* методичні рекомендації щодо використання розробленого веб-продукту в освітньому процесі;
- *оцінити* можливості впровадження результатів курсової роботи у практику викладання інформатики в закладах загальної середньої освіти.

**Об'єкт** дослідження – це процес або явище, що досліджується. Наприклад: *процес навчання учнів старшої школи основам програмування та створення веб-орієнтованих застосунків.*

**Предмет** дослідження – це конкретні аспекти, властивості або відносини в межах об'єкта дослідження. Наприклад: *методика використання веб-технологій HTML, CSS і JavaScript для формування алгоритмічного та логічного мислення учнів 10-11 класів.* Предмет та об'єкт мають чітко корелювати з назвою курсової роботи.

**Методи дослідження.** Перераховують використані наукові методи:

- *теоретичні* (аналіз нормативних документів, навчальних програм, методичної літератури та офіційної документації HTML, CSS і JavaScript);
- *емпіричні* (педагогічне спостереження, аналіз результатів виконання учнями навчальних завдань, тестування розробленого веб-ресурсу);
- *моделювання* (проєктування структури веб-ресурсу, логіки взаємодії елементів інтерфейсу, сценаріїв роботи користувача).

**Теоретична значущість** дослідження полягає в узагальненні та систематизації знань про використання сучасних веб-технологій у шкільному курсі інформатики, уточненні підходів до навчання програмування та формування алгоритмічного мислення учнів засобами HTML, CSS і JavaScript.

**Практичне значення одержаних результатів** полягає у створенні навчального веб-ресурсу або комплексу інтерактивних завдань, а також методичних матеріалів щодо їх використання. Отримані результати можуть бути застосовані вчителями інформатики під час проведення уроків, факультативних занять або в позакласній діяльності (наприклад, гуртках з веб-програмування).

**Апробація результатів** (за наявності) передбачає представлення результатів курсової роботи під час студентських наукових конференцій, участь у методичних семінарах або використання розроблених матеріалів у процесі проходження педагогічної практики.

**Структура та обсяг курсової роботи.** У цьому пункті зазначається структура роботи (вступ, основні розділи, висновки, список використаних джерел, додатки), загальний обсяг сторінок, кількість ілюстрацій (схем, алгоритмів, скріншотів веб-ресурсу), таблиць та використаних інформаційних джерел.

## ОСНОВНА ЧАСТИНА

**Основна частина** курсової роботи складається, як правило, із двох розділів, які поділяються на підрозділи, пункти та підпункти. За необхідності допускається більша кількість розділів (до п'яти), що залежить від складності обраної теми та обсягу програмної розробки.

Заголовки розділів повинні точно відображати зміст викладеного в них наукового та технічного матеріалу, але не дублювати загальну назву курсової роботи. У структурі роботи має чітко простежуватися логічна послідовність розв'язання поставлених завдань дослідження: від теоретичного аналізу сучасних веб-технологій HTML, CSS і JavaScript та їх дидактичних можливостей до практичної реалізації навчального веб-ресурсу.

Типова структура основної частини:

# РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДИЧНІ ЗАСАДИ ВИКОРИСТАННЯ ВЕБ-ТЕХНОЛОГІЙ У КУРСІ ІНФОРМАТИКИ.

У цьому розділі здійснюється аналіз сучасного стану розвитку веб-програмування, розкриваються теоретичні основи використання HTML, CSS і JavaScript у навчальному процесі, а також розглядаються методичні підходи до їх вивчення у закладах загальної середньої освіти. Студент аналізує чинні шкільні навчальні програми, підручники та навчально-методичні матеріали з інформатики з позицій доцільності впровадження веб-технологій у процес навчання програмування.

*Наприклад:* аналіз ролі алгоритмічного підходу у JavaScript; огляд можливостей HTML і CSS для формування уявлень про структуру та візуальне подання інформації; розгляд освітнього потенціалу інтерактивних веб-елементів і середовищ браузерів для навчання учнів основ програмування.

# РОЗДІЛ 2. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА МЕТОДИКА ВИВЧЕННЯ [ТЕМИ] ЗАСОБАМИ HTML, CSS ТА JAVASCRIPT

Цей розділ присвячений практичній частині курсової роботи й містить опис розробленого веб-орієнтованого програмного продукту, структури та логіки його реалізації, алгоритмів розв’язання навчальних завдань, а також методичних рекомендацій щодо використання створених матеріалів у шкільному курсі інформатики. У розділі висвітлюються особливості побудови інтерфейсу користувача засобами HTML і CSS та реалізація інтерактивної й алгоритмічної складової за допомогою JavaScript.

*Наприклад:* розробка навчальної веб-гри або інтерактивного тренажера; опис системи онлайн-тестів або практичних завдань з автоматичною перевіркою результатів; розробка та методичне обґрунтування циклу уроків з теми «Алгоритмізація та програмування» або «Основи веб-програмування» з використанням HTML, CSS і JavaScript.

**Вимоги до викладу:**

**Логічність:** кожен наступний підрозділ повинен послідовно випливати з попереднього, від опису ідеї та структури веб-продукту до реалізації функціональності й методики його використання.

**Аргументованість:** усі технічні рішення (вибір структури HTML-документа, способів стилізації, методів обробки подій, алгоритмів на JavaScript) мають бути обґрунтовані з позицій ефективності, наочності та доступності для учнів.

**Методична спрямованість**: опис програмної реалізації повинен супроводжуватися поясненнями щодо її дидактичного призначення та можливостей використання в навчальному процесі.

**Висновки до розділів:** У кінці кожного розділу обов’язково формулюються висновки обсягом 0,5-1 сторінка, у яких подається стислий виклад теоретичних результатів (що проаналізовано та узагальнено) і практичних досягнень (що розроблено, які алгоритми та інтерактивні елементи реалізовано).

Дотримання такої структури дає змогу студенту продемонструвати не лише рівень володіння сучасними веб-технологіями, а й методичну готовність до професійної діяльності вчителя інформатики.

## ВИСНОВКИ

**ВИСНОВКИ** – це короткий виклад результатів виконаної роботи, в якому характеризується рівень досягнення поставленої мети та ступінь виконання завдань дослідження.

Висновки не повинні мати характер стислого переказу змісту всієї курсової роботи. Ця частина є узагальненням і систематизацією теоретичних положень та практичних результатів, отриманих у процесі дослідження. У висновках подається послідовний і логічно виважений виклад здобутих результатів як технічного характеру (пов'язаних із розробкою програмного продукту засобами HTML, CSS і JavaScript), так і методичного (щодо організації та реалізації навчання інформатики), а також їх співвіднесення із загальною метою та завданнями, сформульованими у вступі.

Під час формулювання висновків рекомендується узагальнити основні результати дослідження в розрізі окремих розділів роботи:

**За результатами теоретичного розділу:** підбиваються підсумки аналізу можливостей сучасних веб-технологій HTML, CSS і JavaScript, їх ролі у формуванні алгоритмічного мислення та особливостей використання в шкільному курсі інформатики. Зазначається, яким чином обрані технології сприяють розвитку цифрових компетентностей учнів та підвищенню мотивації до вивчення програмування.

**За результатами практичного розділу:** висвітлюються досягнення, пов'язані з розробкою веб-орієнтованого програмного продукту (його функціональні можливості, структура, реалізовані алгоритми, інтерактивні елементи), а також методичні результати, зокрема створені дидактичні матеріали, навчальні завдання, плани-конспекти уроків або рекомендації щодо використання розробки в освітньому процесі. Окремо акцентується увага на практичній цінності отриманих результатів для вчителя інформатики.

Сформульовані у такий спосіб висновки можуть бути використані як основа для підготовки доповіді на захисті курсової роботи та демонструють рівень фахової підготовки студента у сфері веб-програмування й методики навчання інформатики.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** містить найменування робіт, які були безпосередньо використані автором при роботі над курсовою роботою. Кількість використаних джерел в курсовій роботі як правило має бути не менше 10.

До списку використаних джерел включають літературу наступного характеру: наукову, навчально-методичну літературу, статті із періодичних фахових видань, електронні ресурси. Особливо цінується використання нових

публікацій з теми, які видавалися та були опубліковані в поточному році, оскільки саме вони відбивають сучасний рівень досягнень в галузі методики навчання інформатики.

**ДОДАТОК А**

До **ДОДАТКІВ** доцільно включати допоміжні матеріали, необхідні для повного сприйняття дослідження та підтвердження практичних результатів розробки веб-орієнтованого програмного продукту на основі HTML, CSS і JavaScript. Зокрема, у додатках оформлюють:

- таблиці, схеми та ілюстрації допоміжного характеру (скріншоти інтерфейсу веб-ресурсу);
- лістинги розробленого програмного коду або окремих модулів;
- UML-діаграми класів, блок-схеми алгоритмів та інші графічні моделі структури та логіки роботи веб-ресурсу;
- конспекти уроків інформатики з тем веб-програмування, алгоритмізації та програмування;
- інструкції до лабораторних або практичних занять для учнів, зокрема щодо використання веб-технологій та налаштування середовищ розробки (браузер, редактор коду, локальний сервер) для освітніх цілей;
- зразки розроблених дидактичних матеріалів (картки завдань, тести, інтерактивні вправи, презентації до уроків, методичні рекомендації), які були використані або створені студентом під час виконання курсової роботи.

Розділ ДОДАТКИ розташовується після висновків та списку використаних джерел і нумерується згідно із загальним порядком нумерації курсової роботи

## 3.ВИМОГИ ДО НАПИСАННЯ КУРСОВОЇ РОБОТИ

Підготовка курсової роботи з сучасних мов програмування потребує ретельного планування часу, оскільки вона поєднує в собі як складну технічну розробку (написання коду на HTML, CSS і JavaScript), так і методичне обґрунтування. Тому доцільно виділити наступні етапи роботи:

1.Вибір та затвердження теми (Див. Додаток В).

2.Складання плану виконання роботи (структури проекту та логіки програми).

3.Пошук та вивчення науково-методичної літератури, технічної документації та Інтернет-джерел.

4.Добір та опрацювання теоретичного матеріалу, проектування архітектури програми та написання коду.

5.Написання тексту роботи (пояснювальної записки).

6.Публічний захист роботи з демонстрацією програмного продукту.

Тему курсової роботи студенти обирають самостійно із запропонованого кафедрою переліку. Студент має право запропонувати власну тему, якщо вона пов’язана з його інтересами у сфері розробки ПЗ або об’єктом дослідження під час педагогічної практики. Остаточний варіант теми закріплюється рішенням засідання кафедри.

**Тема курсової роботи повинна бути спрямована на:**

**- поглиблене вивчення розділів програмування**, що входять до шкільних програм, наприклад: *«Методика вивчення об'єктно-орієнтованої парадигми на прикладі JavaScript у старшій школі»; «Використання HTML, CSS та JavaScript для створення навчальних веб-проєктів»;*

**- розв'язання дослідницьких та прикладних завдань**, наприклад: *«Розробка інтерактивного веб-тренажера для підготовки учнів до ЗНО з інформатики»; «Методика використання веб-технологій для візуалізації алгоритмів сортування у 8-9 класах».*

З числа штатних викладачів кафедри назначається науковий керівник курсової роботи.

**Призначений науковий керівник:**

- разом із студентом складає календарний план виконання роботи;
- консультує студента щодо вибору технологій для реалізації навчальних веб-ресурсів (браузерів, редакторів коду, бібліотек HTML, CSS і JavaScript);
- надає допомогу у підборі сучасної літератури (як методичної, так і технічної);
- контролює хід виконання роботи відповідно до календарного плану та перевіряє правильність оформлення тексту;
- складає письмовий відгук на роботу.

Узгоджений з керівником **план роботи** студента має логічно поєднувати теоретичний аналіз веб-технологій та методики їх використання з практичною реалізацією навчального веб-ресурсу.

Після вибору технологій студент вивчає літературу. Порядок роботи з джерелами рекомендується такий: фундаментальні праці з педагогіки, психології та загальної методики навчання інформатики; вузькоспеціалізовані технічні джерела (наприклад, офіційна документація HTML, CSS, JavaScript, навчальні ресурси MDN, книги авторів з веб-програмування); методичні матеріали та розробки щодо навчання алгоритмізації та створення інтерактивних навчальних завдань для учнів.

Для отримання найсучасніших відомостей варто звертатися до:

- науково-методичних журналів: *«Комп'ютер у школі та сім'ї»*, *«Інформатика та інформаційні технології в навчальних закладах»*;
- технічних ресурсів: *офіційні довідники HTML, CSS та JavaScript (MDN Web Docs), професійні блоги та ресурси для веб-програмування (наприклад, Stack Overflow, W3Schools, CSS-Tricks);*
- інституційних репозиторіїв та професійних спільнот: *онлайн-спільноти вчителів інформатики, GitHub, освітні платформи з веб-програмування.*

Під час опрацювання джерел необхідно виділяти основні поняття та терміни (елемент, атрибут, селектор, подія, функція, DOM, маніпуляція стилями), аналізувати існуючі підходи до навчання веб-програмуванню та створення інтерактивних веб-ресурсів та фіксувати недоліки або обмеження існуючих методик, які може виправити власне дослідження. При цитуванні або переказі ідей обов'язково слід зазначати джерело інформації.

Курсова робота має продемонструвати вміння студента-бакалавра самостійно створювати якісний програмний код на HTML, CSS та JavaScript та

знаходити інноваційні шляхи розв’язання педагогічних проблем за допомогою сучасних ІТ-технологій.

**Вимоги до написання пояснювальної записки.**

Текст кожної із вказаних структурних частин курсової роботи повинен починатися із нової сторінки.

Зразок оформлення титульного аркуша наведено нижче у Додатку А. Зразок оформлення змісту (орієнтована структура) наведена у Додатку Б.

Рекомендований обсяг курсової роботи – 25-40 сторінок (при дотриманні норм оформлення тексту). Додатки до рекомендованого обсягу роботи не включаються.

Робота повинна бути виконана комп'ютерним набором; формат – А4 (210 х 297мм), розмір шрифту – 14 пт, друк тільки з одного боку, інтервал – 1,5, рівномірна щільність, відступ в абзацах – 1,25 см; поля: ліве – 30 мм, праве – 15 мм, верхнє – 20 мм, нижнє – 20 мм.

**Заголовки** структурних частин роботи друкують великими літерами симетрично до тексту. Заголовки друкують з абзацу звичайним шрифтом. Крапка в кінці заголовків не ставиться. Якщо заголовок складається з двох або більше речень, їх розділяють крапками. Кожну структурну частину курсової роботи слід починати з нової сторінки (підрозділи – через рядок від попереднього).

Обов’язкова **нумерація сторінок**. Нумерацію сторінок, розділів, рисунків, таблиць, формул подають арабськими цифрами без знаку «№».

Першою сторінкою роботи є титульний аркуш, який зараховують до загальної нумерації сторінок. На титульному аркуші номер сторінки не ставлять, на наступних сторінках номер проставляють у правому верхньому куті без крапки в кінці.

На другій сторінці міститься зміст – остаточно відредагований план курсової роботи з вказівкою сторінок кожного розділу і підрозділу в тексті.

Якщо в тексті роботи використовуються **ілюстрації** (рисунки, графіки, схеми) і таблиці, то вони подаються безпосередньо після тексту, де згадуються вперше. Якщо ілюстрація чи таблиця мають формат більший за А4, то її можна розмістити в додатках.

Ілюстрації позначають словом "Рис." і нумерують послідовно в межах розділу (за винятком ілюстрацій, поданих у додатках). Номер ілюстрації складається з номера розділу й порядкового номера ілюстрації, між якими ставиться крапка (Приклад Рис. 2.1). Номер ілюстрації, її назва й пояснювальний підпис розміщується безпосередньо під ілюстрацією.

Таблиці нумеруються послідовно (за винятком таблиць, поданих у додатках) у межах розділу. У правому верхньому куті над відповідним заголовком таблиці розміщується напис Таблиця із зазначенням її номера. Номер таблиці повинен складатися з номера розділу й порядкового номера таблиці, між якими ставиться крапка (Приклад: Таблиця 1.1).

Якщо таблиця переноситься на наступну сторінку, то над таблицею пишеться «Продовження таблиці» і вказується її номер. Назви граф у таблиці пишуться з великої літери, підзаголовки – з маленької, якщо вони становлять

одне речення із заголовком. Якщо вони самостійні, то пишуться з великої літери.

Формули в курсовій роботі (якщо їх більше однієї) нумерують у межах розділу. Номер формули складається з номера розділу й порядкового номера формули в розділі, між якими ставиться крапка. Номери формул пишуть біля правого поля аркуша на рівні відповідної формули в круглих дужках.

Для підтвердження власних аргументів посиланням на авторитетне джерело або для критичного аналізу того чи іншого тексту слід наводити цитати. Кожна цитата обов'язково супроводжується посиланням на джерело. Посилання в тексті роботи на джерела слід зазначати у квадратних дужках порядковим номером цього джерела за бібліографічним списком, наприклад: [5]. У посиланні на кілька джерел, вони відокремлюються точкою з крапкою, наприклад: [15; 21; 27]

Заголовок будь-якої зі структурних частин курсової роботи не може знаходитися в кінці аркуша, після нього мусять бути принаймні два рядки тексту, що знаходяться на відстані 5 мм від заголовка. Жоден заголовок не може закінчуватися крапкою. Якщо він довший одного рядка, то не слід вживати перенесення (поділу) слів.

Список використаних джерел формується відповідно до порядку появі в тексті та повинен містити такий порядок (автори, назва, журнал, рік, сторінки, DOI); він повинен бути оформлений згідно до вимог ДСТУ 8302:2015. **Наводяться лише ті джерела, на які зроблено посилання в курсовій роботі**.

До списку використаних джерел рекомендується включати не менше 10 найменувань. Усі джерела подаються тією мовою, якою вони видані. Неприпустимим є переклад назв видань українською мовою.

Правила оформлення додатків:

Кожний додаток слід починати з нової сторінки із зазначенням угорі з правого боку сторінки слова **«Додаток»** і його літерного позначення.

Позначають додатки послідовно великими літерами української абетки, за винятком літер **Ґ, Є, І, Ї, Й, О, Ч, Ь**. Наприклад: **Додаток А**, **Додаток Б** і т.д. Якщо в роботі тільки один додаток, він позначається як Додаток А.

Додаток повинен мати **заголовок**, який друкують нижче з великої літери посередині строки (наприклад: *Програмний код для візуалізації алгоритмів сортування на JavaScript*).

Текст кожного додатка за потреби може бути поділений на розділи й підрозділи, які нумерують у межах кожного додатка. У цьому разі перед кожним номером ставлять позначення додатка (літеру) і крапку, наприклад: **А.2** – другий розділ додатка А.

Наявність якісно оформлених додатків (особливо авторського програмного коду та методичних розробок для школи) є важливим показником практичної значущості курсової роботи майбутнього вчителя інформатики.

# 4.ЗАХИСТ КУРСОВОЇ РОБОТИ ТА КРИТЕРІЇ ОЦІНЮВАННЯ

Під час захисту курсової роботи з дисципліни «Сучасні мови програмування» студент повинен доповісти про результати дослідження за обраною темою, продемонструвати роботу створеного програмного продукту (застосунку, тренажера або моделі), виявити вміння чітко викладати власні думки, використовувати ілюстративний матеріал (UML-діаграми, схеми алгоритмів, скріншоти інтерфейсу) та аргументовано відповідати на запитання членів комісії щодо технічної реалізації коду та методики його використання в школі.

У процесі захисту курсової роботи оцінюється глибина знань студента в досліджуваній галузі (знання синтаксису HTML, CSS, JavaScript, принципів DOM та алгоритмізації), вміння вести дискусію, обґрунтовувати вибір технологічних рішень та відстоювати власну точку зору, здатність логічно пояснити технічні та методичні аспекти створеного веб-ресурсу.

При оцінюванні курсових робіт беруть до уваги те, що студент повинен **уміти:**

- формулювати мету та завдання дослідження у контексті розробки веб-ресурсу та навчання інформатики;
- проводити бібліографічний та технічний пошук з використанням сучасних інформаційних технологій, зокрема офіційної документації HTML, CSS, JavaScript та ресурсів MDN Web Docs;
- застосовувати сучасні методи наукового дослідження та проєктування веб-систем, включно з моделюванням структури коду та алгоритмів;
- обробляти отримані дані, аналізувати програмний код, тестувати функціональність веб-ресурсу та систематизувати результати;
- узагальнювати результати дослідження та обґрунтовувати їх достовірність, оцінюючи працездатність коду та методичну доцільність створених матеріалів;
- оформлювати результати роботи, лістинги програм та всю курсову відповідно до сучасних академічних і технічних вимог.

**У процесі визначення оцінки враховується низка важливих показників роботи:**

- актуальність обраної теми для сучасної шкільної інформатики;
- чіткість формулювання мети і завдань дослідження;
- структура і логіка побудови змісту роботи, відповідність коду поставленим задачам;
- якість теоретико-методологічного аналізу об’єктно-орієнтованого підходу;
- наявність огляду наукових та технічних джерел;
- правильність і повнота виконання поставлених завдань розробки;
- змістовність зроблених узагальнень і висновків;
- дотримання вимог до оформлення (включаючи оформлення програмного коду);
- наявність та інформаційна повнота ілюстративних матеріалів (діаграм, графіків, таблиць);
- змістовність доповіді та якість демонстрації програмного ресурсу;
- правильність та чіткість відповідей на запитання членів комісії.

## Критерії оцінювання курсових робіт

Оцінка курсової роботи здійснюється за 100-бальною шкалою, яка складається з двох частин:

**- виконання курсової роботи та створення веб-продукту** (до 70 балів);

**- захист роботи та демонстрація коду** (до 30 балів).

**Оцінка «відмінно» (90-100 балів)** ставиться, якщо студент:

- показав глибокі теоретичні знання HTML, CSS, JavaScript та принципів веб-програмування;
- опанував навички дослідницької та практичної роботи: розробив якісний веб-продукт, проаналізував методику його застосування у навчальному процесі;
- надав обґрунтовані пропозиції щодо використання розробки в освітньому процесі;
- виконав роботу чіткою та зрозумілою мовою, дотримуючись академічних і технічних стандартів;
- оформив роботу відповідно до вимог та подав її у встановлений термін;
- на захисті впевнено володів темою та бездоганно продемонстрував роботу веб-продукту.

**Оцінка «добре» (75-89 балів)** ставиться, якщо студент:

- показав досить високі теоретичні знання з дисципліни;
- розробив працездатний веб-продукт, але не завжди критично оцінює технічні обмеження обраних методів;
- дає рекомендації щодо предмета дослідження, проте має труднощі з детальним методичним обґрунтуванням;
- виконав роботу грамотно, але допустив незначні помилки в оформленні коду або тексту;
- подав роботу вчасно;
- на захисті продемонстрував добрі знання та відповів на більшість запитань комісії.

**Оцінка «задовільно» (60-74 балів)** ставиться, якщо студент:

- показав достатні теоретичні знання, але поверхово володіє мовами HTML, CSS та JavaScript;
- в основному виконав практичне завдання (веб-продукт працює з обмеженнями), але допускає порушення логіки викладу;
- не може чітко сформулювати методичні рекомендації для використання розробки у школі;
- допускає помилки в оформленні роботи, ілюстративного матеріалу;
- на захисті продемонстрував задовільні знання, але не зміг впевнено відповісти на додаткові запитання.

**Оцінка «незадовільно» (менше 60 балів)** ставиться, якщо студент виявив повне незнання досліджуваної теми, веб-продукт не працює; виявлено ознаки несамостійного виконання роботи (плагіат коду чи тексту).

У випадку, якщо студент не захистив курсову роботу, він вважається не атестованим з даної дисципліни. Захищені курсові роботи та розроблені програмні модулі передаються на кафедру для зберігання згідно з встановленими вимогами.

| **100-бальна шкала** | **Національна шкала** | **Шкала ECTS** |
|---|---|---|
| 90-100 | відмінно | A |
| 82-89 | добре | B |
| 75-81 | | C |
| 67-74 | задовільно | D |
| 60-66 | | E |
| 35-59 | незадовільно | FX |
| 0-34 | | F |

Додаток А

КЛАСИЧНИЙ ПРИВАТНИЙ УНІВЕРСИТЕТ

(повне найменування вищого навчального закладу)

ІНСТИТУТ УПРАВЛІННЯ

(повне найменування інституту, назва факультету (відділення))

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

(повна назва кафедри (предметної, циклової комісії))

# КУРСОВА РОБОТА

з дисципліни Сучасні мови програмування

**на тему РОЗРОБКА ІНТЕРАКТИВНИХ МЕТОДИЧНИХ МАТЕРІАЛІВ ДЛЯ ВЧИТЕЛІВ ІНФОРМАТИКИ ЗА ДОПОМОГОЮ HTML, CSS ТА JAVASCRIPT**

Студент (ки) 2 курсу
спеціальності 014 Середня освіта
освітня програма Інформатика
(шифр і назва спеціальності, освітньої програми)

___Прізвище І.Б._____
(прізвище та ініціали)

Керівник проф., д.т.н. Прізвище І.Б.__
посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____

Кількість балів: _____. Оцінка: ECTS ____

м. Запоріжжя – 2026 рік

Додаток Б

ЗМІСТ

ВСТУП …… 3

Розділ. 1 Назва розділу 1 …….. 5

1.1. Назва підрозділу …….. 10

1.2. Назва підрозділу …….. 15

Розділ. 2 Назва розділу 1 ……. 16

2.1. Назва підрозділу …… 18

2.2. Назва підрозділу ……..... 21

ВИСНОВКИ……..... 23

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ……... 25

ДОДАТКИ ……..... 26

Додаток В

# ОРІЄНТОВНІ ТЕМИ КУРСОВИХ РОБІТ

**Напрям 1: Методика вивчення конкретних розділів програмування (веб-технології)**

1. Методика вивчення основ веб-програмування на прикладі HTML, CSS та JavaScript у старшій школі.
2. Особливості викладання теми «Алгоритми сортування» з використанням інтерактивних веб-застосунків на JavaScript.
3. Методика вивчення структурних типів даних (масиви, об’єкти, списки) у JavaScript у класах технологічного профілю.
4. Використання HTML, CSS та JavaScript для вивчення теми «Рекурсивні алгоритми» у поглибленому курсі інформатики.
5. Методика формування навичок роботи з текстовою інформацією (робота зі строками, регулярні вирази) засобами HTML, CSS та JavaScript.
6. Організація вивчення теми «Обробка винятків та помилок у веб-програмах» на HTML, CSS та JavaScript.
7. Методика викладання концепцій наслідування та поліморфізму через створення інтерактивних ігрових проектів на HTML, CSS та JavaScript
8. Використання середовищ розробки (Visual Studio Code, CodeSandbox) як засобу візуалізації структури веб-застосунку при навчанні програмуванню.
9. Методика навчання учнів основам побудови графічного інтерфейсу користувача (GUI) за допомогою HTML, CSS та JavaScript.

**Напрям 2: Розробка навчального та контролюючого веб-програмного забезпечення**

10. Розробка інтерактивного тренажера на HTML, CSS та JavaScript для вивчення синтаксису мов програмування.
11. Проектування та реалізація автоматизованої системи тестування знань учнів з інформатики за допомогою веб-технологій.
12. Створення веб-середовища для візуалізації роботи класичних алгоритмів (пошук, сортування) на JavaScript.
13. Розробка мультимедійного електронного підручника «Основи веб-програмування для початківців».
14. Побудова моделюючого веб-середовища для проведення віртуальних лабораторних робіт з фізики/математики.
15. Розробка прикладного веб-застосунку «Електронний щоденник» для учнів 10-11 класів.
16. Створення гри-головоломки на JavaScript як засобу розвитку логічного мислення учнів середньої школи.
17. Розробка системи автоматичної перевірки веб-програмних кодів учнів на відповідність стандартам HTML, CSS та JavaScript.

18. Проектування веб-додатку для підтримки навчання інформатики в школі (наприклад, навчальні вправи, інтерактивні уроки).

19. Розробка чат-бота на JavaScript для інформаційної та методичної підтримки навчального процесу в закладі освіти.

**Напрям 3: Інноваційні підходи та міжпредметні зв'язки у веб-програмуванні**

20. Проектна технологія навчання веб-програмуванню на HTML, CSS та JavaScript при створенні прикладних систем автоматизації (наприклад, шкільна бібліотека онлайн).

21. Використання ігрових механік (Gamification) у процесі вивчення JavaScript в позакласній роботі з інформатики.

22. Методика підготовки обдарованих учнів до участі в олімпіадах з програмування з використанням веб-технологій.

23. Реалізація міжпредметних зв’язків інформатики та математики через моделювання геометричних об’єктів на JavaScript.

24. Використання JavaScript для навчання учнів основам тестування та верифікації веб-програм (наприклад, з Jest або іншими бібліотеками для тестування).

25. Методика використання хмарних сервісів та онлайн-середовищ розробки (CodeSandbox, Replit) при вивченні веб-технологій в умовах дистанційного навчання.

26. Розробка інтерактивних дидактичних матеріалів для вивчення веб-програмування в інклюзивному освітньому середовищі.

27. Методика організації групової роботи учнів над веб-проектами з використанням Git та GitHub.

28. Формування алгоритмічної культури учнів засобами об’єктно-орієнтованого моделювання на JavaScript (класи, об’єкти, наслідування).

29. Розробка веб-модуля для автоматизації складання розкладу занять на основі алгоритмів оптимізації (наприклад, генетичних) на JavaScript.