ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ  
«ОПТИКО-МЕХАНІЧНИЙ ФАХОВИЙ КОЛЕДЖ КИЇВСЬКОГО  
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ІМЕНІ ТАРАСА ШЕВЧЕНКА»  
Циклова комісія програмування та інформаційних технологій

**З В І Т**

**З НАВЧАЛЬНОЇ ПРАКТИКИ**

спеціальність 121 Інженерія програмного забезпечення  
освітньо-кваліфікаційний рівень «Фаховий молодший бакалавр»

|  |  |
| --- | --- |
| ПІДСУМКОВА ОЦІНКА  (оцінка цифрою та прописом)  Керівники практики від коледжу:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) | Виконав:  студенту 2-го курсу групи ІПЗ-23  *Столбуна Богдана Леонідовича*  залікова книжка № \_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис виконавця)  « \_» 20 року |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис) |  |

« \_»\_ 2024 року

КИЇВ

ЗАТВЕРДЖУЮ

Директор ОМФК КНУ

імені Тараса Шевченка  
 \_\_\_\_\_\_\_\_\_\_ Борис ГАПРІНДАШВІЛІ

« » 20 р.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

на період *Навчальної практики*

Столбуна Богдана Леонідовича

Спеціальність *121 Інженерія програмного забезпечення* Група *ІПЗ-23*

|  |  |  |
| --- | --- | --- |
|  | **Питання до вивчення** | **Відмітки про**  **виконання** |
| 4 | Розділ "Практична робота з С++" |  |
| 4.1 | Введення та виведення у базових консольних програмах |  |
| 4.2 | Робота з файлами: запис та читання |  |
| 4.3 | Формати файлів та їх обробка |  |
| 4.4 | Системи контролю версій (VCS) |  |
| 4.5 | Мова Markdown для опису проектів |  |
| 4.6 | Створення ігрової програми |  |
| 5 | Розділ ”Веб-дизайн та HTML” |  |
| 5.1 | Мова розмітки HTML і структура Веб-сторінки |  |
| 5.2 | Розмітка тексту за допомогою HTML. Зображення та посилання |  |
| 5.3 | Каскадні таблиці стилів CSS та їх структура |  |
| 5.4 | Спадковості і каскадування за допомогою CSS |  |
| 6 | Систематизація матеріалів, оформлення звітів і залік з навчальної практики |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Голова циклової комісії |  |  |  |
|  | (підпис) | (ініціали, прізвище) | (дата) |
| Керівники практики від коледжу |  |  |  |
|  | (підпис) | (ініціали, прізвище) | (дата) |
|  |  |  |  |
|  | (підпис) | (ініціали, прізвище) | (дата) |
|  |  |  |  |
|  | (підпис) | (ініціали, прізвище) | (дата) |

**ЗМІСТ**

[ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 5](#_Toc167990294)

[ВСТУП 7](#_Toc167990295)

БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ПРИ РОБОТІ З КОМП’ЮТЕРОМ [9](#_Toc167990296)

[**Тема 1 Робота з файлами: запис та читання** 10](#_Toc167990298)

[**Тема 2 Системи контролю версій (VCS)** 13](#_Toc167990300)

[**Тема 3 Створення ігрової програми** 16](#_Toc167990302)

[***Тема 4 Мова розмітки HTML та структура Веб-сторінки*** 18](#_Toc167990303)

[Опис постановки задачі 23](#_Toc167990307)

[Висновки 24](#_Toc167990308)

[***До 3 теми:*** 24](#_Toc167990309)

[***До 4 теми:*** 25](#_Toc167990310)

[Список використаних джерел 26](#_Toc167990311)

# ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

* <html> – позначає початок і кінець HTML-документа.
* <head> – містить метадані веб-сторінки.
* <title> – задає заголовок веб-сторінки, який відображається в заголовку браузера.
* <meta> – використовується для вказання метаданих, таких як кодування символів.
* <style> – містить CSS-стилі для веб-сторінки.
* <body> – визначає основний вміст веб-сторінки.
* <header> – визначає верхній колонтитул документа або розділу.
* <nav> – містить навігаційне меню.
* <ul> – створює ненумерований список.
* <li> – визначає елемент списку.
* <a> – створює посилання на іншу веб-сторінку або документ.
* <main> – визначає головний вміст документа.
* <section> – визначає розділ у документі.
* <h1> до <h6> – визначає заголовки різного рівня на веб-сторінці.
* <p> – створює абзац тексту.
* <br> – вставляє розрив рядка.
* <hr> – вставляє горизонтальну лінію (роздільник).
* <footer> – визначає нижній колонтитул документа або розділу.
* <table> – створює таблицю.
* <thead> – задає заголовок таблиці.
* <tbody> – задає тіло таблиці.
* <tfoot> – задає підвал таблиці.
* <tr> – створює рядок таблиці.
* <th> – створює заголовок стовпця таблиці.
* <td> – створює клітинку таблиці.
* <form> – створює форму для введення даних користувача.
* <input> – створює поле для введення даних.
* <label> – прив'язує мітку до елемента введення.
* <em> – виділяє текст курсивом.
* <b> – виділяє текст жирним.
* <i> – виділяє текст курсивом.
* <u> – підкреслює текст.
* <strong> – виділяє текст жирним шрифтом, підкреслюючи його важливість.
* <small> – відображає текст меншим шрифтом.
* <blockquote> – визначає цитату з відступом.
* <code> – відображає текст у вигляді коду.
* <pre> – відображає попередньо відформатований текст, зберігаючи пробіли та розриви рядків.
* <textarea> – створює багаторядкове поле введення тексту.
* <button> – створює кнопку.
* <select> – створює випадаючий список.
* <option> – створює елемент списку в <select>.
* <article> – визначає самостійний фрагмент вмісту.
* <aside> – визначає побічний вміст, наприклад, бокову панель.
* <img> – вбудовує зображення в HTML-документ.
* <audio> – додає аудіо-контент.
* <source> – вказує джерело мультимедіа для <audio> або <video>.
* <video> – додає відео-контент.

# 

# 

# ВСТУП

HTML (HyperText Markup Language - «мова гіпертекстової розмітки») є стандартною мовою розмітки для створення веб-сторінок та веб-додатків. HTML описує структуру веб-сторінок за допомогою текстових елементів.

Мова XHTML є більш суворим варіантом HTML, вона являється додатком синтаксису XML і є додатком мови XML в області розмітки гіпертексту. У всесвітній павутині HTML-сторінки, як правило, передаються браузерам від сервера по протоколах HTTP або HTTPS, у вигляді простого тексту або з використанням шифрування.

Мова HTML була розроблена в кінці 1980-х - на початку 1990-х років британським вченим Тімом Бернерсом-Лі (Tim Berners-Lee). У 1989 році Тім Бернерс-Лі, працюючи в Європейській організації з ядерних досліджень (CERN), запропонував проект глобальної гіпертекстової системи для полегшення обміну інформацією між вченими. Цей проект став відомим як Всесвітня павутина (World Wide Web).

Кабінет інформатики та інформаційно-комунікаційних технологій на­сичений різноманітною електронною технікою, яка живиться від елект­ричної мережі з напругою змінного струму 220 В. Це потребує дотриман­ня додаткових правил безпеки під час проведення навчальних занять і позаурочних заходів.

**До початку роботи:**

* **з дозволу вчителя займіть своє робоче місце за комп’ютером, налаш­туйте висоту стола, стільця, підставки для ніг, кут нахилу монітора так**
* **середина екрана монітора знаходилася трохи нижче горизонтальної лінії зору;**
* **відстань від очей до поверхні екрана становила 40-80 см, залежно від розмірів об’єктів на екрані;**
* **лінія від очей до центра екрана монітора була перпендикулярна до площини екрана;**
* **пальці рук вільно лежали на клавіатурі;**
* **руки утворювали в ліктьовому суглобі кут, близький до 90;**
* **клавіатура лежала на поверхні стола або на спеціальній полиці на відстані 10-30 см від краю і була нахилена під кутом 5-15;**
* **спина опиралася на спинку стільця;**
* **ноги опиралися на підлогу або на спеціальну підставку;**
* **наведіть лад на робочому столі, приберіть з нього предмети, які не потрібні для роботи;**
* **перевірте чистоту своїх рук, за потреби вимийте їх і витріть насухо;**
* **за потреби, з дозволу вчителя і тільки спеціальною серветкою протріть екран монітора, клавіатуру, килимок і мишу;**
* **перевірте відсутність зовнішніх пошкоджень комп’ютерного облад­нання;**
* **з дозволу вчителя ввімкніть ком­п’ютер.**

**Під час роботи:**

* **тримайте робоче місце охайним, не розміщуйте на ньому сторонніх ре­чей;**
* **витримуйте правильну поставу - не нахиляйтеся близько до поверхні ек­рана, не згинайтеся, тримайте руки без напруження;**
* **після 15-20 хв роботи або при від­чутті втоми виконайте комплекс вправ для очей і для зняття м’язово­го напруження;**
* **не торкайтеся задніх стінок монітора та системного блока, не чіпай­те дротів живлення;**
* **не торкайтеся екрана монітора руками (це забруднює його і порушує антиблікове покриття);**
* **категорично забороняється знімати кришки корпусів пристроїв комп’ютера, самостійно без дозволу вчителя приєднувати і від’єдну­вати пристрої комп’ютера;**
* **акуратно вставляйте і виймайте змінні носії.**

**Після закінчення роботи:**

* **приберіть своє робоче місце;**
* **з дозволу вчителя вимкніть комп’ютер або закінчіть сеанс роботи.**

## ***Тема 2 Робота з файлами: запис та читання***

JSON (JavaScript Object Notation) - це легкий формат обміну даними, який легко читається та пишеться як людьми, так і машинами. JSON використовується для зберігання та передачі структурованих даних у текстовому вигляді.

**Робота з файлами JSON:**

**Плюси**:

1. Легкість читання та редагування: Формат JSON має просту та легку для сприйняття структуру, що робить файли JSON легкими для читання та редагування як людиною, так і машинами.
2. Підтримка мов програмування: Більшість сучасних мов програмування мають бібліотеки або пакети, які дозволяють легко читати та записувати дані у форматі JSON, що робить його досить універсальним для використання.
3. Гнучкість: JSON дозволяє представляти різноманітні дані, включаючи тексти, числа, масиви, об'єкти та вкладені структури даних.
4. Популярність: JSON є одним з найбільш поширених форматів обміну даними в сучасному програмуванні, що означає, що для нього існують багато засобів підтримки та бібліотек для різних мов програмування.

**Мінуси:**

1. Великий обсяг даних: JSON може виявитися неефективним для великих обсягів даних, оскільки він зазвичай використовує багато зайвих символів для форматування.
2. Брак сховищ даних: JSON не надає вбудованих механізмів для роботи з великими обсягами даних, такими як індексація або пошук, що може ускладнити обробку великих файлів.

## ***Тема 3 Системи контролю версій (VCS)***

Системи контролю версій (VCS) - це програмні засоби, які використовуються для відстеження змін у файлів і коду програм під час розробки програмного забезпечення.

Основна мета VCS полягає в тому, щоб зберігати історію змін, дозволяти співробітникам працювати одночасно над одним проектом та відновлювати попередні версії коду у випадку помилок або потреби у відновленні попередньої функціональності.

Основні плюси систем контролю версій включають:

1. ***Історія змін****:* Всі зміни до коду або файлів зберігаються і можуть бути переглянуті, зроблені коментарі до змін, включаючи інформацію про того, хто та коли зробив зміну.
2. ***Робота з різними версіями****:* Ви можете відновити попередню версію файлу або навіть порівняти дві версії, щоб побачити, що було змінено.
3. ***Гілки розробки****:* Ви можете створювати окремі гілки розробки для роботи над різними функціональностями або різними аспектами проекту паралельно.
4. ***Співпраця****:* Декілька розробників може одночасно працювати над одним проектом, а VCS автоматично об'єднає їхні зміни.

На сьогодні існує розподіл СКВ на три головні категорії:

-  **Локальні;**

-  **клієнт-серверні (централізовані);**

-  **розподілені (децентралізовані).**

Усі системи контролю версій по суті вирішують 4 задачі.

1. **Доступ до коду.** Вихідники коду зберігаються у віддаленому репозиторії (сховищі даних), куди звертаються розробники, щоб забрати актуальну версію файлів або внести зміни. Так вибудовується командна технологія.

2. **log-ування змін у коді.** Відстеження **commit-**ів (внесення змін до коду) допомагає знайти хто, що і коли змінював, вирішити конфлікти при модифікуванні одних і тих же файлів, відкотитися на будь-який попередній стан.

3. **Розгалуження розробки.** Програмісти паралельно ведуть розробку нового функціоналу у окремих гілках, не торкаючись працездатності старого.

4. **Підтримка версії продуктів.** При випуску оновлень програмних продуктів ми позначаємо релізні версії, наприклад, за допомогою **tag**-ів, щоб зафіксувати їх у цьому стані, для дебагу або ретроспективи.

## ***Тема 4.5 Мова Markdown для опису проектів***

**Markdown** — полегшена мова розмітки даних, яку створено з ухилом на прочитність та зручність у публікації з подальшим перетворенням її на structurally valid XHTML або HTML. Чимало ідей для мови позичено з існуючих домовленостей у розмітці тексту в електронних листах. Першу реалізацію Markdown написано Грубером на Perl, але з часом з'явились багато реалізацій від сторонніх розробників. Реалізація на Perl розповсюджується за ліцензією BSD

1. ***Простота використання****:* Markdown - це дуже легка мова розмітки, яка проста у вивченні та використанні. Вона використовує прості та зрозумілі синтаксичні правила, такі як використання символів "\*" або "\_" для виділення тексту.
2. ***Читабельність****:* Текст, написаний у форматі Markdown, легко читати у вихідному коді, оскільки він є простим текстом. Він також має чистий вигляд при перегляді на веб-сторінках або інших середовищах.
3. ***Підтримка великої кількості платформ****:* Багато веб-сервісів та платформ, такі як GitHub, GitLab, Bitbucket, Reddit, а також багато блог-систем, підтримують формат Markdown для форматування тексту.
4. ***Можливості розширення****:* Існують різні інструменти та розширення, які дозволяють розширити функціональність Markdown, додаючи підтримку для таблиць, зображень, посилань та інших елементів.
5. ***Інтеграція з іншими мовами програмування****:* Markdown легко інтегрується з іншими мовами програмування, такими як HTML, CSS, а також з системами контролю версій, що дозволяє використовувати його в різних середовищах розробки.
6. ***Підтримка структурованих документів***: Markdown дозволяє легко створювати структуровані документи за допомогою заголовків, списків, блоків коду та інших елементів. Це робить опис проектів більш організованим та зрозумілим для читачів.
7. ***Легкість інтеграції з іншими інструментами****:* Markdown може бути легко інтегрований з різними іншими інструментами та платформами, такими як Jekyll, Hugo, Pelican для створення блогів, або Jupyter Notebook для створення наукових звітів та досліджень.
8. ***Можливості розмітки****:* Markdown підтримує різноманітні можливості розмітки, такі як створення посилань, вставка зображень, таблиць, форматування коду, списків та багато іншого. Це дозволяє створювати документи з різними типами контенту, що підвищує їхню інформативність та привабливість.
9. ***Підтримка спеціалізованих розширень****:* Деякі платформи та середовища розробки підтримують спеціалізовані розширення Markdown, які дозволяють вставляти відео, аудіо, математичні формули та інші складні елементи безпосередньо у документ.
10. ***Крос-платформеність та універсальність****:* Markdown-документи можуть бути відкриті та переглянуті на будь-якому пристрої з будь-яким операційним середовищем, що робить їх універсальними для спільної роботи та обміну інформацією.

**Markdown Extra** - це легка мова розмітки на основі Markdown реалізована в PHP (початково), Python та Ruby. Вона містить функції недоступні в звичайному синтаксисі Markdown. Markdown Extra підтримується системами керування вмістом, наприклад Drupal, TYPO3 та MediaWiki.

Додає наступні функції до Markdown:

• Markdown всередині блоків HTML

• Елементи з атрибутами id/class

• "обгороджені" блоки коду (англ. Fenced code blocks)

• Таблиці

• Списки означень (dl)

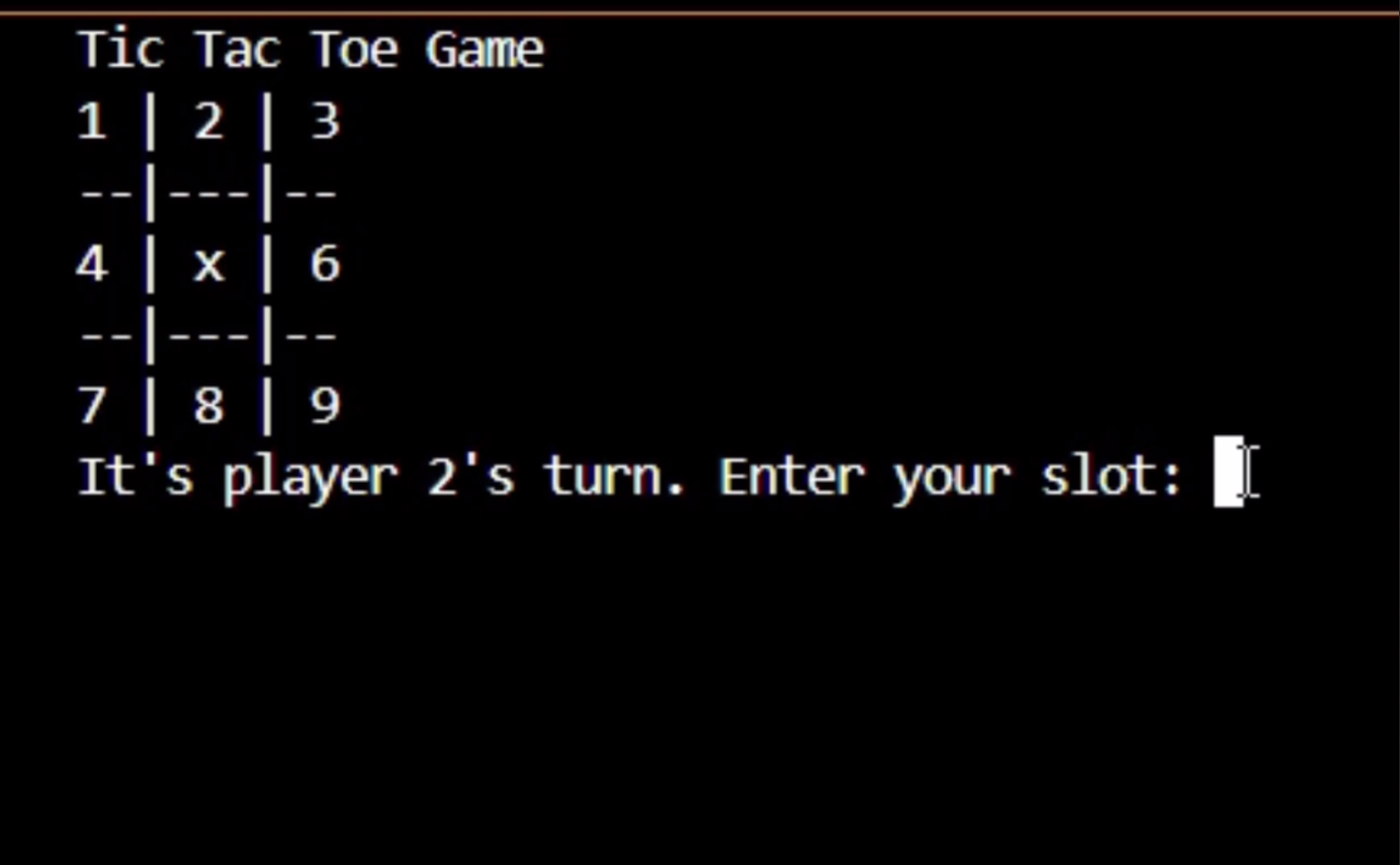
• Зноски

• Абревіатури

## ***Тема 4.6 Створення ігрової програми***

Мій код є реалізацією простої гри "Крестики Нулики" на мові програмування C++. Гра проводиться на квадратному ігровому полі розміром 3 на 3 клітинки або будь-якому іншому полі будь-якого розміру. Участь завжди бере два гравці, які грають один проти одного. Для перемоги, одному з учасників, потрібно першим скласти ряд із трьох або більше хрестиків чи нуліків. Ряд може бути вертикальним, горизонтальним або розташованим по діагоналі ігрового поля. Гравець, який першим зібрав цілий ряд, одразу перемагає, а суперник програє. Кожен ігровий матч триває доти, доки не переможе один із гравців, або ж закінчиться нічиєю.

Код демонструє базовий приклад використання мови програмування C++ для створення простої консольної гри. Він показує, як можна використовувати цикли, умовні оператори та змінні для створення ігрового середовища. Цей код може бути використаний як основа для розширення та вдосконалення гри, додавання нових функцій та покращення інтерфейсу користувача.



## ***Тема 5.1 Мова розмітки HTML та структура Веб-сторінки***

HTML (HyperText Markup Language — мова гіпертекстової розмітки) — це стандартна мова розмітки для створення веб-сторінок і веб-додатків. Вона визначає структуру веб-документів за допомогою різних елементів та тегів. Основні аспекти HTML включають:

Елементи та Теги:

HTML використовує теги для створення елементів веб-сторінок.

Теги зазвичай парні: відкриваючий тег (<tag>) і закриваючий тег (</tag>). Наприклад, <p> і </p> для абзаців.

*Структура Документа:*

*Стандартний HTML документ починається з декларації типу документа <!DOCTYPE html>, що вказує браузерам, що використовується HTML5.*

Основні елементи HTML документа включають:

*<html>: кореневий елемент, що обгортає весь контент сторінки.*

*<head>*: містить метаінформацію про документ, таку як заголовок сторінки

*<title>), посилання на CSS (<link>), метадані (<meta>), і сценарії (<script>).*

<body>: містить основний вміст сторінки, який відображається користувачам, включаючи текст, зображення, посилання, таблиці, форми тощо.

*Заголовок (Head):*

Елемент *<head>* містить інформацію про документ, яка не відображається безпосередньо на сторінці. Приклади:

*<title>: заголовок сторінки, який відображається на вкладці браузера.*

<meta>: метаінформація, така як кодування символів або опис сторінки.

<link>: посилання на зовнішні ресурси, наприклад, таблиці стилів.

*Тіло (Body):*

Елемент *<body>* містить весь вміст, що відображається на веб-сторінці:

*Заголовки: <h1> - <h6> для заголовків різних рівнів.*

*Абзаци: <p> для текстових абзаців.*

*Посилання: <a href="URL"> для створення гіперпосилань.*

*Зображення: <img src="URL" alt="description"> для відображення зображень.*

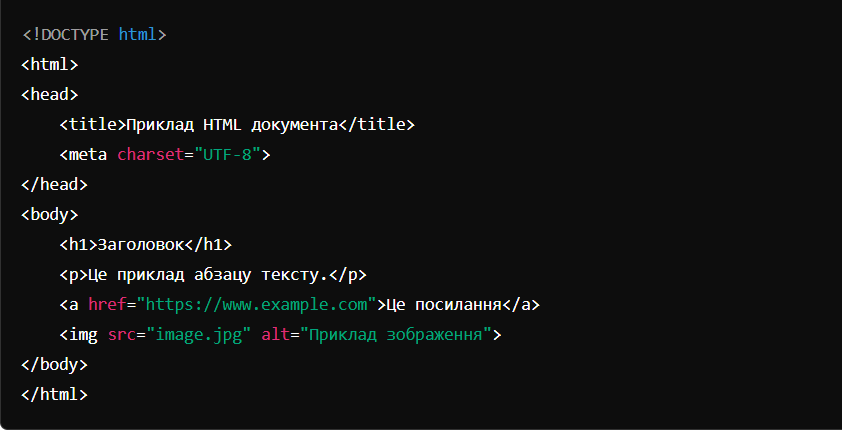
*Списки: <ul> для ненумерованих списків, <ol> для нумерованих списків, <li> для елементів списку.*

*Таблиці: <table>, <tr> (рядок), <td> (комірка), <th> (заголовок комірки).*

*Атрибути:*

*Теги можуть мати атрибути, що надають додаткову інформацію про елемент. Наприклад, атрибут href у тегу <a> вказує адресу посилання.*

*Приклад HTML документа:*



## ***Тема 5.2 Розмітка тексту за допомогою HTML. Зображення та посилання***

Щоб реалізувати семантичну розмітку, HTML надає спеціальні теги, які можна використовувати для представлення таких розділів, наприклад:

* заголовок: <header> .
* панель навігації: <nav> .
* основний вміст:з різноманітними підрозділами вмісту, <main>представленими елементами , і .<article><section><div>
* бічна панель: <aside> ; часто розміщують всередині <main>
* колонтитул: <footer>

Абзаци можна утворити завдяки тегу <p>. За замовчуванням абзаци починаються з нового рядка і мають вертикальні відступи, якими можна управляти за допомогою стилів. Але використовувати їх можна тільки всередині тега <body>.

Для заголовків в мові HTML для виділення заголовків передбачено ціле сімейство тегів: від <h1> до <h6>. Тег <h1> позначає найважливіший заголовок (заголовок верхнього рівня), а тег <h6> позначає підзаголовок самого нижнього рівня.

Основні елементи для переносу та розділювання рядка: <br>, <hr>,

<br> створює розрив рядка в абзаці; це єдиний спосіб створити жорстку структуру в ситуації, коли вам потрібна серія фіксованих коротких рядків, наприклад, у поштовій адресі чи вірші. Без <br>елементів абзац просто відображатиметься в одному довгому рядку.

<hr>елементи створюють горизонтальну лінію в документі, яка позначає тематичну зміну в тексті (наприклад, зміну теми чи сцени). Візуально це просто виглядає як горизонтальна лінія.

Впорядкований список створюється за допомогою тега <ol> (ordered list), який може містити всередині себе теги <li>. Якщо елементи неупорядкованого списку за замовчуванням позначаються маркерами, то елементи впорядкованого списку - нумеруються.

Невпорядковані (або марковані) списки створюються за допомогою тега <ul>, який може містити всередині себе теги <li>, що позначають «елемент списку». Єдине, що дозволено в межах розмітки маркованого списку (тобто між відкриваючим і закриває тегами ul), - це один або кілька елементів списку. Ви не можете вставити туди інші елементи, і там не може бути нерозміченого тексту.

Посилання створюються за допомогою тегу <a>

Щоб перетворити виділений текст на посилання, помістіть його між відкриваючим і закриваючим тегами <а> та </а> і за допомогою атрибуту href вкажіть URL-адресу сторінки, на яку буде здійснюватися перехід. Вміст елемента якоря стає гіпертекстовим посиланням. Тег <a> без адреси не використовується, оскільки він описує посилання яке нікуди не веде. Також Посилання можна робити не лише за допомогою тексту, але і за допомогою зображень. Для цього потрібно обернути тег <img> в тег <a>.

Зображення додаються за допомогою тегу <img>

Щоб додати на сторінку зображення, потрібно використовувати одиночний тег <img> (image) з атрибутом src, в якому вказана адреса картинки. Щоб управляти шириною або висотою зображення, потрібно використовувати атрибути width і height. Найпоширенішими форматами зображень у мережі є JPEG і PNG.

# Висновки

## ***До 4 теми:***

Під час навчальної практики за розділом "Практична робота з С++" я значно розширив свої знання та здобулв цінні практичні навички. Ось ключові аспекти, які були охоплені під час практики:

1. Введення та виведення у базових консольних програмах

Я зрозумів, що введення та виведення є ключовими елементами базових консольних програм, що дозволяють здійснювати інтерактивну взаємодію між користувачем і програмою. Знання того, як правильно зчитувати дані від користувача та відображати результати, є важливим навиком для розробників. Ці операції є фундаментальними в програмуванні і використовуються у всіх основних мовах програмування.

1. Робота з файлами: запис та читання

Робота з файлами — це важливий аспект програмування, що дозволяє ефективно зберігати та обробляти дані. Запис у файл дозволяє зберігати результати роботи програм, а читання з файлу — отримувати збережені дані для подальшої обробки. Розуміння основних операцій з файлами та вміння їх використовувати є необхідним навиком для розробників, незалежно від мови програмування, якою вони користуються.

1. Формати файлів та їх обробка

Робота з файлами різних форматів є невід'ємною частиною програмування та обробки даних. Знання особливостей кожного формату та відповідних методів їх обробки дозволяє ефективно використовувати інформацію, зберігати її, передавати та аналізувати. Вміння працювати з різними форматами файлів розширює можливості програміста і дозволяє вирішувати широкий спектр завдань, пов'язаних з обробкою даних.

1. Системи контролю версій (VCS)

Системи контролю версій є важливим інструментом для управління змінами в проектах розробки програмного забезпечення. Вони забезпечують надійність, спільну роботу та контроль якості коду. Знання та вміння використовувати VCS, такі як Git, є критично важливими навичками для розробників, оскільки вони сприяють ефективності роботи команди, забезпечують безпеку даних і дозволяють легко відстежувати та управляти змінами в коді.

1. Мова Markdown для опису проектів

Markdown є потужним інструментом для створення структурованих текстових документів з форматуванням, що робить його ідеальним для опису проектів у сфері програмного забезпечення. Завдяки простоті використання, чіткості та сумісності з багатьма платформами, Markdown забезпечує ефективне представлення інформації, що сприяє кращій взаємодії в командах розробників та зручному доступу до важливих даних про проекти. Використання Markdown допомагає розробникам легко створювати та підтримувати документацію, що є критично важливим для успішного управління проектами.

# 