ВІДОКРЕМЛЕННИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ

«ОПТИКО-МЕХАНІЧНИЙ ФАХОВИЙ КОЛЕДЖ КИЇВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ІМЕНІ ТАРАСА ШЕВЧЕНКА»

Циклова комісія спеціальності «Інженерія програмного забезпечення»

**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:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. С. Нікітенко  (оцінка) (підпис)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т. І. Лумпова  (оцінка) (підпис)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ М. С. Ходжаєв  (оцінка) (підпис) |  | Виконав: |
| студента 2-го курсу групи ІПЗ-21 |
| прізвище ім‘я по-батькові  залікова книжка № \_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис виконавця)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 року |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Степанюк  (оцінка) (підпис) |  |  |
| «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 року |  |  |

м. Київ — 2024

ЗАТВЕРДЖУЮ

Директор ОМФК КНУ

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

«17» травня 2024 р.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| на період | *навчальної* | | практики | |
|  | (навчальної, технологічної, виробничої) | |  | |
| *Дзюбенко Олександр Дмитрович* | | | | |
| (П.І.Б студента.) | | | | |
| Спеціальність | *121 Інженерія програмного забезпечення* | Група | | *ІПЗ-21* |

|  | **Питання для вивчення** | **Відмітки про виконання** |
| --- | --- | --- |
| **1** | **Вступне заняття.** Вивчення питань охорони праці. Ознайомлення з електронною бібліотекою коледжу. Вивчення правил оформлення звітної документації. |  |
| **2** | **Розділ "Інформатика**" |  |
| 2.1 | Засобами MS Word розробити документ складної структури та зберегти у різних форматах |  |
| 2.2 | Розв’язування рівнянь |  |
| 2.3 | Розв’язування оптимізаційних задач |  |
| 2.4 | Основи статистичного аналізу даних. Ряди даних. Кореляційний аналіз даних |  |
| 2.5 | Обчислення основних статистичних характеристик вибірки |  |
| 2.6 | Візуалізація рядів і трендів даних |  |
| 2.7 | Розв‘язок задач за допомогою засобів табличного редактора MS Excel |  |
| 2.8 | Проектування бази даних та створення бази даних *MS ACCESS* |  |
| 2.9 | Основні поняття реляційних баз даних |  |
| 2.10 | Введення, коригування, вилучення інформації з бази даних *MS ACCESS* |  |
| 2.11 | Вибірка, пошук, фільтрація даних в *MS ACCESS* |  |
| 2.12 | Виконання практичних завдань за допомогою засобів табличного редактора MS Access |  |

|  | **Питання для вивчення** | **Відмітки про виконання** |
| --- | --- | --- |
| **3** | **Розділ "Програмування С++"** |  |
| 3.1 | Алгоритмізація обчислювальних процесів. Розробка постановки задачі, визначення вхідних та вихідних даних, оформлення блок-схем. |  |
| 3.2 | Типи даних в мові С/С++, внутрішнє представлення даних, операції перетворення типів даних. |  |
| 3.3 | Оператор присвоювання. Операції введення- виведення даних |  |
| 3.4 | Програмування лінійних та розгалужених обчислювальних процесів |  |
| 3.5 | Програмування циклічних обчислювальних процесів |  |
| 3.6 | Складання програм |  |
| 3.7 | Виконання тестів |  |
| 4 | **Розділ "** **Практична робота з С++"** |  |
| 4.1 | Тема 1. Введення та виведення у базових консольних програмах |  |
| 4.2 | Тема 2. Структури даних для управління станом програми |  |
| 4.3 | Тема 3. Робота з файлами: запис та читання |  |
| 4.4 | Тема 4. Формати файлів та їх обробка |  |
| 4.5 | Тема 5. Системи контролю версій (VCS) |  |
| 4.6 | Тема 6. Мова Markdown для опису проекті |  |
| 4.7 | Розроблення ігрової програми |  |
| 5 | Розділ ”Веб-дизайн та HTML” |  |
| 5.1 | Мова розмітки HTML та 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 |  |
| 5.5 | Розробка та оформлення Постановки задачі |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Питання для вивчення** | | | | **Відмітки про виконання** | | |
| 5.9 | | Розробка HTML документу | | | |  | | |
| **6** | | Систематизація матеріалів, оформлення звіту з навчальної практики | | | |  | | |
| Голова циклової комісії | |  |  | Савєльєва І.В. | |  |  |
|  | | (підпис) |  | (ініціали, прізвище) | |  | (дата) |
| Керівники практики від коледжу | |  |  |  | |  |  |
|  | |  |  | Нікітенко А.С. | |  | 17.05.2024р. |
|  | | (підпис) |  | (ініціали, прізвище) | |  | (дата) |
|  | |  |  | Лумпова Т.І. | |  | 17.05.2024р. |
|  | | (підпис) |  | (ініціали, прізвище) | |  | (дата) |
|  | |  |  | Нікітенко А.С. | |  | 17.05.2024р. |
|  | | (підпис) |  | (ініціали, прізвище) | |  | (дата) |
|  | |  |  | Лумпова Т.І. | |  | 17.05.2024р. |
|  | | (підпис) |  | (ініціали, прізвище) | |  | (дата) |
|  | |  |  |  | |  |  |
|  | |  |  |  | |  |  |

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

4

Розроб.

Перевір.

Реценз.

В.Я

Н. Контр.

В.Я

Затверд.

Звіт з навчальної практики

Літ.

Аркушів

ІПЗ-21

ОМФК.0121 НП 20

**ЗМІСТ**

[ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 6](#_Toc167919156)

[ВСТУП 7](#_Toc167919157)

[1 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ПРИ РОБОТІ З КОМП’ЮТЕРОМ 8](#_Toc167919158)

[2 РОЗДІЛ "ІНФОРМАТИКА" 9](#_Toc167919159)

[**2.1** **Розроблення засобами MS Word документа складної структури** 9](#_Toc167919160)

[**2.2 Розв‘язок задач за допомогою засобів табличного редактора MS Excel** 9](#_Toc167919161)

[**2.3 Виконання** **практичних завдань за допомогою засобів табличного редактора MS Access** 9](#_Toc167919162)

[3 РОЗДІЛ "ПРОГРАМУВАННЯ С++" 10](#_Toc167919163)

[**3.1 Теоретичні відомості** 10](#_Toc167919164)

[3.1.1 Загальні поняття. Елементи мови С++ - константи, змінні, операції, перетворення типів 10](#_Toc167919165)

[3.1.2 Арифметичні операції 10](#_Toc167919166)

[3.1.3 Поняття вхідного та вихідного потоку, найпростіші математичні функції 10](#_Toc167919167)

[3.1.4 Програмування лінійних та розгалужених обчислювальних процесів 10](#_Toc167919168)

[3.1.5 Програмування циклічних обчислювальних процесів 10](#_Toc167919169)

[**3.2 Практичне завдання** 11](#_Toc167919170)

[3.2.1 Завдання 11](#_Toc167919171)

[3.2.2 Постановка задачі 11](#_Toc167919172)

[3.2.3 Код програми 11](#_Toc167919173)

[3.2.4 Скріншот виконання програми 11](#_Toc167919174)

[**3.3 Виконання тестів** 11](#_Toc167919175)

[4 РОЗДІЛ " ПРАКТИЧНА РОБОТА З С++" 12](#_Toc167919176)

[**4.1 Тема 1. Введення та виведення у базових консольних програмах** 12](#_Toc167919177)

[**4.2 Тема 2. Структури даних для управління станом програми** 12](#_Toc167919178)

[**4.3 Тема 3. Робота з файлами: запис та читання** 12](#_Toc167919179)

[**4.4 Тема 4. Формати файлів та їх обробка** 12](#_Toc167919180)

[**4..5 Тема 5. Системи контролю версій (VCS)** 13](#_Toc167919181)

[**4.6 Тема 6. Мова Markdown для опису проекті** 13](#_Toc167919182)

[**4.7 Розроблення ігрової програми** 13](#_Toc167919183)

[4.7.1 Код програми 13](#_Toc167919184)

[4.7.2 Скріншоти виконання програми 13](#_Toc167919185)

[5 РОЗДІЛ ”ВЕБ-ДИЗАЙН ТА HTML” 14](#_Toc167919186)

[**5.1 Мова розмітки HTML та структура Веб-сторінки** 14](#_Toc167919187)

[**5.2 Розмітка тексту за допомогою HTML. Зображення та посилання** 14](#_Toc167919188)

[**5.3 Каскадні таблиці стилів CSS та їх структура** 14](#_Toc167919189)

[**5.4 Спадковості і каскадування за допомогою CSS** 14](#_Toc167919190)

[**5.5 Практична частина** 15](#_Toc167919191)

[5.5.1 Постановки задачі 15](#_Toc167919192)

[5.5.2 HTML документ 15](#_Toc167919193)

[ВИСНОВКИ 16](#_Toc167919194)

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ 17](#_Toc167919195)

[ДОДАТКИ 18](#_Toc167919196)

ОМФК.0121 НП 20

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

# ВСТУП

Автор: Дзюбенко Олександр Дмитрович

Дата: 13.06.24

Мета:

Перелік вивчених тем:

1. Засобами MS Word розробити документ складної структури та зберегти у різних форматах

2. Розв’язування рівнянь

3. Розв’язування оптимізаційних задач

4. Основи статистичного аналізу даних. Ряди даних. Кореляційний аналіз даних

5. Обчислення основних статистичних характеристик вибірки

6. Візуалізація рядів і трендів даних

7. Розв‘язок задач за допомогою засобів табличного редактора MS Excel

8. Проектування бази даних та створення бази даних MS ACCESS

9. Основні поняття реляційних баз даних

10. Введення, коригування, вилучення інформації з бази даних MS ACCESS

11. Вибірка, пошук, фільтрація даних в MS ACCESS

12. Виконання практичних завдань за допомогою засобів табличного редактора MS Access

13. Алгоритмізація обчислювальних процесів. Розробка постановки задачі, визначення вхідних та вихідних даних, оформлення блок-схем.

14. Типи даних в мові С/С++, внутрішнє представлення даних, операції перетворення типів даних.

15. Оператор присвоювання. Операції введення- виведення даних

16. Програмування лінійних та розгалужених обчислювальних процесів

17. Програмування циклічних обчислювальних процесів

18. Складання програм

19. Виконання тестів

20. Введення та виведення у базових консольних програмах

21. Структури даних для управління станом програми

22. Робота з файлами: запис та читання

23. Формати файлів та їх обробка

24. Системи контролю версій (VCS)

25. Мова Markdown для опису проекті

26. Розроблення ігрової програми

27. Мова розмітки HTML та структура Веб-сторінки

28. Розмітка тексту за допомогою HTML. Зображення та посилання

29. Каскадні таблиці стилів CSS та їх структура

30. Спадковості і каскадування за допомогою CSS

31. Розробка та оформлення Постановки задачі

32. Розробка HTML документу

33. Систематизація матеріалів, оформлення звіту з навчальної практики

# 1 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ПРИ РОБОТІ З КОМП’ЮТЕРОМ

* вмикати і вимикати апаратуру без вказівки вчителя;
* торкатися роз'ємів з'єднувальних кабелів та самих кабелів;
* торкатися екрана та тильної частини монітора;
* переміщувати увімкнені складові обчислювальної системи (системний блок, монітор тощо);
* класти будь-які предмети на системний блок, монітор, клавіатуру;
* приносити та використовувати носії даних (флешки, компакт-диски) без дозволу вчителя.
* знаходитись у кабінеті і виконувати будь-які роботи без присутності відповідальної особи — завідувача кабінету, вчителя, лаборанта.
* працювати з клавіатурою та маніпулятором «миша» брудними або вологими руками.

# 2 РОЗДІЛ "ІНФОРМАТИКА"

## **2.1 Розроблення засобами MS Word документа складної структури**

Розробка документа складної структури в MS Word включає в себе використання різних інструментів форматування, організації тексту та додавання графічних елементів. Ось кілька кроків, які можуть бути корисними для створення такого документа:

1. \*\*Використання заголовків та стилів\*\*: Використовуйте заголовки (Heading 1, Heading 2 і т.д.) для ієрархії розділів та підрозділів документа. Вони дозволяють автоматично створювати зміст документа і полегшують навігацію.

2. \*\*Форматування абзаців і списків\*\*: Використовуйте різні рівні вкладеності списків (нумеровані та марковані), а також налаштуйте відступи та міжрядкові інтервали для кращого зручності читання.

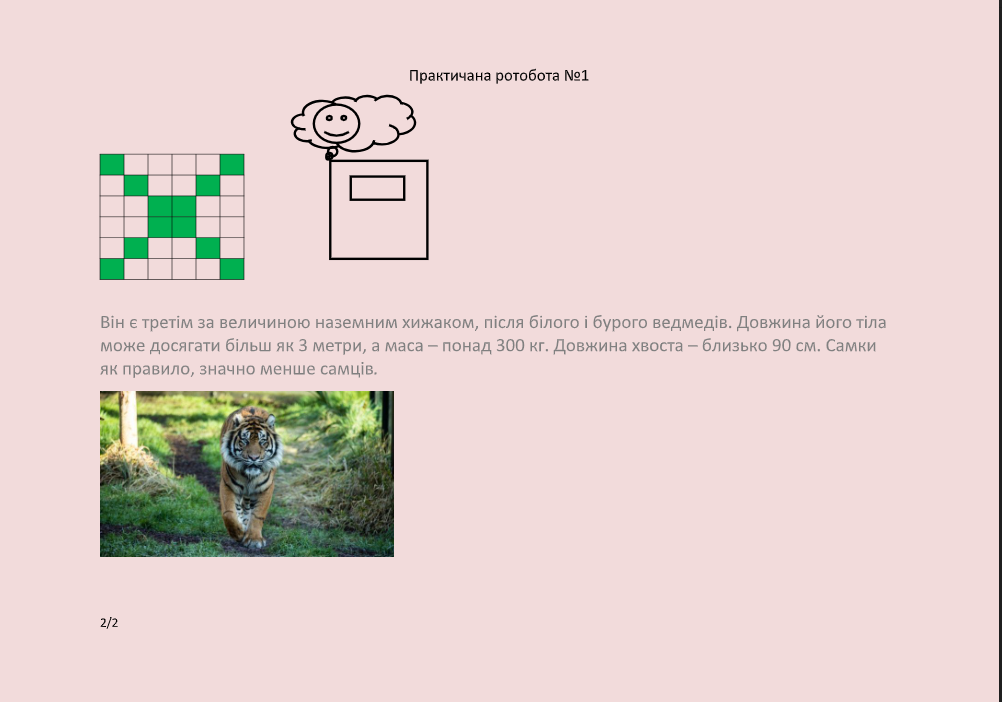
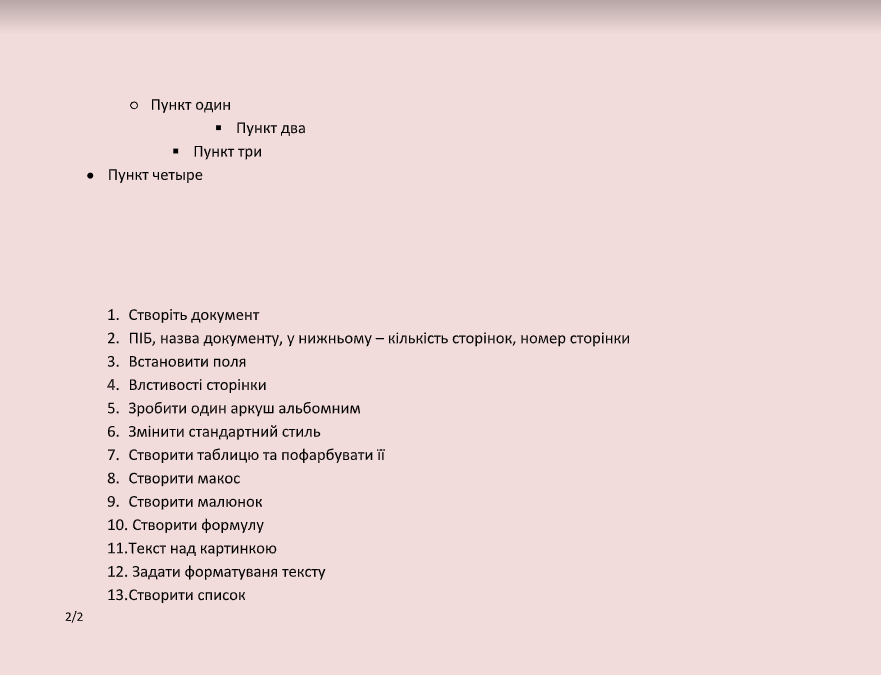
3. \*\*Вставка таблиць\*\*: Використовуйте таблиці для організації табличних даних. MS Word дозволяє швидко створювати, форматувати та редагувати таблиці зі зручними інструментами.

4. \*\*Вставка зображень і діаграм\*\*: Додавайте зображення, графіки або інші графічні елементи для ілюстрації вашого тексту. Ви можете вставляти зображення з файлів або використовувати вбудовані функції для створення діаграм.

5. \*\*Використання сторінкового макету і відступів\*\*: Налаштуйте сторінковий макет, встановлюйте поля та відступи, щоб ваш документ виглядав професійно.

Якщо ви шукаєте більше конкретних порад або детальніше розглядати функціонал MS Word, відвідайте [офіційний сайт Microsoft Word](https://www.microsoft.com/uk-ua/microsoft-365/word).

Ці кроки допоможуть вам створити документ складної структури за допомогою MS Word.



## **2.2 Розв‘язок задач за допомогою засобів табличного редактора MS**

Розв'язок задач за допомогою засобів табличного редактора MS Excel включає в себе використання різноманітних функцій, формул та інструментів для обробки даних, аналізу та візуалізації результатів. Ось кілька ключових кроків для розв'язання задач у MS Excel:

1. \*\*Введення даних\*\*: Введіть потрібні дані у відповідні комірки аркуша Excel.

2. \*\*Використання формул\*\*: Використовуйте вбудовані формули для виконання різноманітних обчислень, таких як сума, середнє значення, мінімум, максимум тощо.

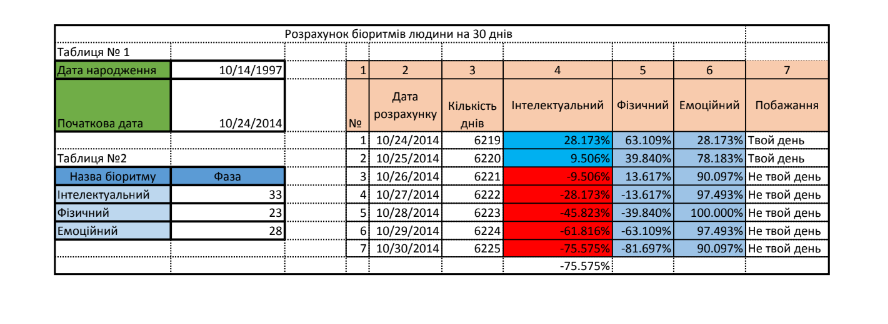
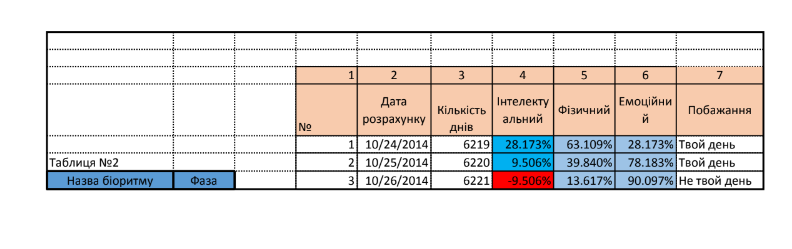
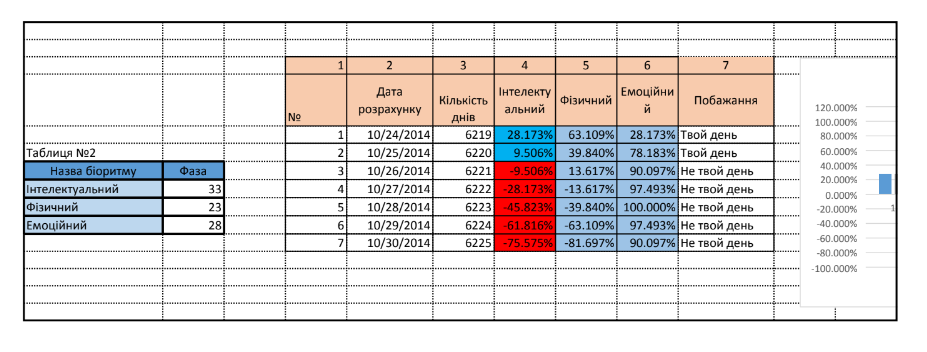
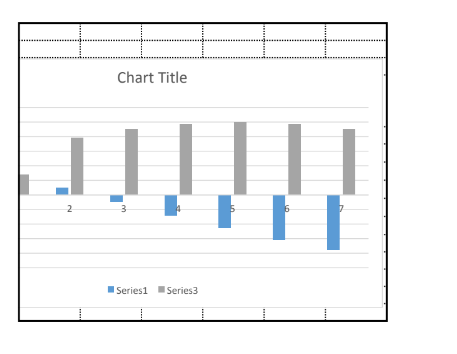
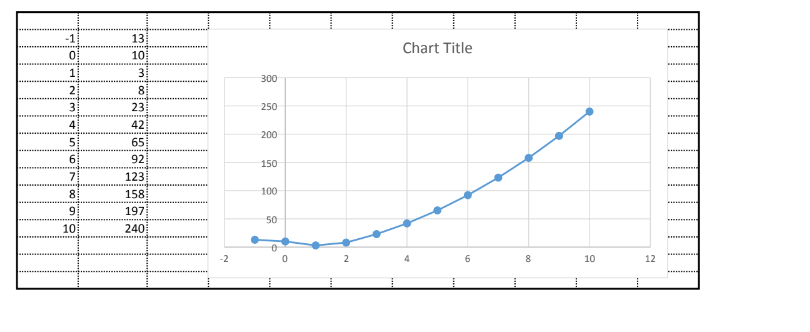
3. \*\*Фільтрація та сортування\*\*: Використовуйте функції фільтрації і сортування для організації даних та вибору потрібної інформації.

4. \*\*Графіки і діаграми\*\*: Побудуйте графіки і діаграми для візуалізації даних і аналізу їх розподілу.

5. \*\*Аналіз даних\*\*: Використовуйте інструменти аналізу даних, такі як таблиці Pivot (зведені таблиці), для отримання зведеної інформації та висновків з набору даних.

Якщо вам потрібна додаткова інформація про використання MS Excel або підтримка, відвідайте [офіційний сайт Microsoft Excel](https://www.microsoft.com/uk-ua/microsoft-365/excel).

Ці інструкції допоможуть вам ефективно використовувати MS Excel для розв'язання різноманітних завдань і задач.



## **2.3 Виконання** **практичних завдань за допомогою засобів табличного редактора MS Access**

MS Access — це база даних, яка включає в себе можливості для створення таблиць, запитів, форм і звітів. Для виконання практичних завдань за допомогою MS Access вам знадобиться знання основних концепцій баз даних і можливостей, які надає ця програма.

Основні кроки для виконання практичних завдань в MS Access можуть включати:

1. \*\*Створення таблиць\*\*: Визначення структури даних і створення таблиць для зберігання інформації.

2. \*\*Створення запитів\*\*: Використання запитів для витягування, оновлення або видалення даних з таблиць.

3. \*\*Створення форм\*\*: Розробка форм для введення даних користувачами або для їх відображення.

4. \*\*Створення звітів\*\*: Виготовлення звітів для аналізу даних і їх представлення.

5. \*\*Імпорт і експорт даних\*\*: Робота з імпортом і експортом даних для обміну інформацією з іншими додатками.

Для отримання більш детальної інформації про MS Access і його функціональні можливості рекомендую відвідати [офіційний сайт Microsoft Access](https://www.microsoft.com/uk-ua/microsoft-365/access).

Цей ресурс містить актуальну інформацію, документацію та ресурси для роботи з MS Access.

# 3 РОЗДІЛ "ПРОГРАМУВАННЯ С++"

## **3.1 Теоретичні відомості**

### 3.1.1 Загальні поняття. Елементи мови С++ - константи, змінні, операції, перетворення типів

* Константи
* Змінні
* Операції
* Логічні операції
* Порівняльні операції
* Побітові операції
* Перетворення типів
* Неявне перетворення
* Явне перетворення

### 

### 3.1.2 Арифметичні операції

Арифметичні операції в C++ дозволяють виконувати базові математичні обчислення. Вони працюють з числовими типами даних, такими як int, float, double, тощо. Нижче наведено основні арифметичні операції та приклади їх використання.

### 3.1.3 Поняття вхідного та вихідного потоку, найпростіші математичні функції

### Поняття вхідного та вихідного потоку

1. \*\*Вхідний потік (Input Stream)\*\*:

- Це послідовність даних, які надходять у програму або процес з зовнішнього джерела. Наприклад, вхідний потік може бути текстовим рядком, що користувач вводить з клавіатури, дані, що приходять через мережу, і так далі.

2. \*\*Вихідний потік (Output Stream)\*\*:

- Це послідовність даних, які програма або процес виводить у зовнішнє середовище. Наприклад, вихідний потік може бути текстовим рядком, який виводиться на екран для користувача, або дані, які записуються у файл, відправляються через мережу тощо.

### Найпростіші математичні функції

1. \*\*Додавання (+)\*\*:

- Операція, що додає два числа разом. Наприклад:

[ 2 + 3 = 5]

2. \*\*Віднімання (-)\*\*:

- Операція, що віднімає одне число від іншого. Наприклад:

[ 5 - 3 = 2]

3. \*\*Множення (\*)\*\*:

- Операція, що перемножує два числа. Наприклад:

[ 4 \times 2 = 8]

4. \*\*Ділення (/)\*\*:

- Операція, що ділить одне число на інше (за умови, що дільник не дорівнює нулю). Наприклад:

[10 / 2 = 5]

5. \*\*Модуль (%)\*\*:

- Операція, що повертає залишок від ділення одного числа на інше. Наприклад:

[10 \% 3 = 1 \quad (\text{залишок від ділення 10 на 3})]

6. \*\*Піднесення до степеня (\^)\*\* або \*\*піднесення до степеня (Math.pow)\*\*

### 3.1.4 Програмування лінійних та розгалужених обчислювальних процесів

Програмування лінійних та розгалужених обчислювальних процесів відноситься до структурування програмного коду для виконання послідовності дій або управління в залежності від умов. Розглянемо їхні основні аспекти:

### Лінійні обчислювальні процеси

Лінійні обчислювальні процеси означають послідовне виконання інструкцій без умовного вибору або циклів. Це типовий сценарій для простих обчислень або виконання послідовності операцій.

### 3.1.5 Програмування циклічних обчислювальних процесів

Програмування циклічних обчислювальних процесів включає в себе використання циклів для повторення виконання певного блоку коду доти, доки виконується певна умова. Це дозволяє ефективно обробляти повторювані завдання і маніпулювати даними в програмах. Розглянемо основні типи циклів і їх використання.

## **3.2 Практичне завдання**

### 3.2.1 Завдання

### 3.2.2 Постановка задачі

Вхідні дані

Вихідні дані

Блок-схема алгоритму роботи програми представлена на рис. ххх

Блок-схема

Рисунок хх - Блок-схема алгоритму роботи програми

### 3.2.3 Код програми

### 3.2.4 Скріншот виконання програми

Скріншот виконання програми представлена на рис. ххх

Скріншот

Рисунок хх - Скріншот виконання програми

## **3.3 Виконання тестів**

# 4 РОЗДІЛ " ПРАКТИЧНА РОБОТА З С++"

## **4.1 Тема 1. Введення та виведення у базових консольних програмах**

### Поняття вхідного та вихідного потоку

У контексті програмування вхідний потік (input stream) і вихідний потік (output stream) використовуються для обміну даними між програмою і зовнішнім середовищем, таким як консоль або файли.

1. **Вхідний потік** (istream): Використовується для отримання даних з зовнішнього середовища в програму. Наприклад, зчитування введених користувачем даних з клавіатури або з файлу.
2. **Вихідний потік** (ostream): Використовується для виведення даних з програми до зовнішнього середовища. Наприклад, виведення результатів обчислень на екран консолі або у файл.

В C++ стандартна бібліотека iostream надає класи для роботи з вхідними та вихідними потоками: cin для вхідного потоку (стандартний ввід з клавіатури) і cout для вихідного потоку (стандартний вивід на консоль).

### Найпростіші математичні функції

Найпростіші математичні функції включають базові операції, такі як додавання, віднімання, множення та ділення. У мові програмування C++ ці функції використовуються для обробки числових даних.

.

## **4.2 Тема 2. Структури даних для управління станом програми**

### Поняття вхідного та вихідного потоку

У контексті програмування вхідний потік (input stream) і вихідний потік (output stream) використовуються для обміну даними між програмою і зовнішнім середовищем, таким як консоль або файли.

1. **Вхідний потік** (istream): Використовується для отримання даних з зовнішнього середовища в програму. Наприклад, зчитування введених користувачем даних з клавіатури або з файлу.
2. **Вихідний потік** (ostream): Використовується для виведення даних з програми до зовнішнього середовища. Наприклад, виведення результатів обчислень на екран консолі або у файл.

В C++ стандартна бібліотека iostream надає класи для роботи з вхідними та вихідними потоками: cin для вхідного потоку (стандартний ввід з клавіатури) і cout для вихідного потоку (стандартний вивід на консоль).

### Найпростіші математичні функції

Найпростіші математичні функції включають базові операції, такі як додавання, віднімання, множення та ділення. У мові програмування C++ ці функції використовуються для обробки числових даних.

## **4.3 Тема 3. Робота з файлами: запис та читання**

У мові програмування C++ робота з файлами включає в себе можливість запису і читання даних з файлової системи комп'ютера. Для цього використовуються стандартні потоки введення-виведення (iostream) та класи для роботи з файлами (fstream).

## **4.4 Тема 4. Формати файлів та їх обробка**

Ось короткий огляд форматів файлів та їх обробки у програмуванні:

1. \*\*Текстові файли\*\*: Містять людиночитабельний текст і використовуються для збереження даних у вигляді символів. Для їх обробки використовуються різні формати, такі як CSV, JSON, XML. Більше інформації можна знайти на [сторінці про текстові файли](https://en.wikipedia.org/wiki/Text\_file).

2. \*\*Бінарні файли\*\*: Зберігають дані у вигляді бінарного коду, який може представляти числа, зображення, аудіофайли і т.д. Для роботи з ними необхідно знання структури даних. [Детальніше про бінарні файли](https://en.wikipedia.org/wiki/Binary\_file).

Для більш детального ознайомлення з форматами файлів і їх обробкою в програмуванні рекомендую відвідати відповідні сторінки у Вікіпедії, які наведені вище.

## **4..5 Тема 5. Системи контролю версій (VCS)**

Система контролю версій (VCS) — це програмне забезпечення, яке дозволяє відстежувати зміни в файловій системі, фіксувати відмінності між версіями файлів та керувати доступом до них. Ось ключові аспекти систем контролю версій:

### Основні поняття

1. \*\*Репозиторій\*\*: Це централізоване сховище, де зберігаються всі версії файлів і історія їх змін.

2. \*\*Коміт\*\*: Дія збереження змін у файлі чи наборі файлів у репозиторії. Коміти зберігають інформацію про те, як змінюються файли з часом.

3. \*\*Гілка\*\*: Відгалуження в репозиторії, яке дозволяє незалежно розробляти нові функції або експериментувати з кодом, не впливаючи на основну версію.

4. \*\*Злиття (Merge)\*\*: Об'єднання змін з однієї гілки в іншу. Відбувається для збереження синхронності між гілками.

### Види систем контролю версій

1. \*\*Централізовані (Centralized VCS)\*\*: У цих системах є один центральний сервер, на якому зберігається вся інформація про репозиторій. Приклади: Subversion (SVN).

2. \*\*Розподілені (Distributed VCS)\*\*: Кожен робочий примірник має повний копію репозиторію, що дозволяє працювати офлайн і швидко вносити зміни. Приклади: Git, Mercurial.

### Переваги використання систем контролю версій

- \*\*Історія змін\*\*: Збереження всіх версій файлів і історії їх змін.

- \*\*Колаборація\*\*: Можливість одночасної роботи декількох розробників над одним проектом.

- \*\*Відгалуження і злиття\*\*: Можливість розробки нових функцій відокремлено і потім об'єднання змін.

- \*\*Відновлення інтегрітету коду\*\*: Можливість відновлення попередніх версій коду в разі помилок.

### Застосування систем контролю версій

Системи контролю версій широко використовуються в розробці програмного забезпечення, веб-розробці, управлінні проектами, написанні документації тощо. Вони є необхідним інструментом для організації та управління процесом розробки, забезпечуючи ефективне співпрацю і контроль над змінами.

## **4.6 Тема 6. Мова Markdown для опису проекті**

Markdown - це легкий форматувальний стандарт, який дозволяє легко створювати структуровані документи, такі як README файли для проектів, веб-сторінки, документацію і т.д. Markdown використовується для написання простих текстових документів з елементами форматування, таких як заголовки, списки, посилання, код і т.д., що легко читаються і зрозумілі для людини, і можуть бути перетворені в HTML або інші формати для публікації в Інтернеті.

## **4.7 Розроблення ігрової програми**

### 4.7.1 Код програми

#include <iostream>

using namespace std;

char board[3][3] = { {'1','2','3'}, {'4','5','6'}, {'7','8','9'} };

char currentPlayer = 'X';

void drawBoard() {

cout << " Tic-Tac-Toe" << endl;

cout << "--------------" << endl;

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

cout << " | " << board[i][j];

}

cout << " | " << endl << "--------------" << endl;

}

}

bool isMoveValid(int choice) {

if (choice < 1 , choice > 9) {

cout << "Invalid move! Choose a number between 1 and 9." << endl;

return false;

}

int row = (choice - 1) / 3;

int col = (choice - 1) % 3;

if (board[row][col] == 'X' , board[row][col] == 'O') {

cout << "Cell already occupied! Choose another cell." << endl;

return false;

}

return true;

}

bool checkWin() {

for (int i = 0; i < 3; i++) {

if (board[i][0] == board[i][1] && board[i][1] == board[i][2])

return true;

if (board[0][i] == board[1][i] && board[1][i] == board[2][i])

return true;

}

if (board[0][0] == board[1][1] && board[1][1] == board[2][2])

return true;

if (board[0][2] == board[1][1] && board[1][1] == board[2][0])

return true;

return false;

}

void switchPlayer() {

currentPlayer = (currentPlayer == 'X') ? 'O' : 'X';

}

int main() {

int choice;

bool gameOver = false;

while (!gameOver) {

drawBoard();

cout << "Player " << currentPlayer << ", enter your move (1-9): ";

cin >> choice;

if (isMoveValid(choice)) {

int row = (choice - 1) / 3;

int col = (choice - 1) % 3;

board[row][col] = currentPlayer;

if (checkWin()) {

drawBoard();

cout << "Player " << currentPlayer << " wins!" << endl;

gameOver = true;

}

else {

switchPlayer();

}

}

}

return 0;

}

### 

### 4.7.2 Скріншоти виконання програми

# 5 РОЗДІЛ ”ВЕБ-ДИЗАЙН ТА HTML”

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

Мова розмітки HTML (HyperText Markup Language) використовується для створення структури веб-сторінок. Ось основні аспекти структури веб-сторінки в HTML:

1. **Елемент <html>**: Це кореневий елемент, що охоплює всю веб-сторінку.
2. **Елемент <head>**: Включає мета-інформацію про сторінку, таку як заголовок (<title>), мета-теги, посилання на зовнішні ресурси (стилі, скрипти) і т. д.
3. **Елемент <body>**: Містить основний вміст сторінки, такий як текст, зображення, відео, таблиці тощо.

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

## **5.3 Каскадні таблиці стилів CSS та їх структура**

аскадні таблиці стилів CSS (Cascading Style Sheets) використовуються для оформлення веб-сторінок, щоб надати їм вигляд, який ви бажаєте. Ось основні аспекти структури CSS:

### 1. Використання CSS

CSS може бути використаний на веб-сторінці наступними способами:

* **Вбудований стиль**: Стилі можуть бути вбудовані безпосередньо в HTML-документ за допомогою атрибуту style у тегах HTML.

**Внутрішній стиль**: CSS може бути розміщений всередині блоку <style> у <head> секції HTML-документа.

**Зовнішній файл стилів**: Стилі можуть бути збережені у окремому файлі з розширенням .css, який потім підключається до HTML-документа за допомогою тега <link> у <head> секції.

### Структура CSS

CSS складається з набору правил, кожне з яких включає селектор і блок оголошень стилів

### 3. Каскадність

Каскадність у CSS означає, що стилі можуть бути успадковані і перевизначені в залежності від їх контексту і пріоритету. Наприклад, стилі, задані в зовнішньому файлі CSS, мають більший пріоритет, ніж вбудовані стилі.

### 4. Розширені можливості CSS

CSS також має різноманітність властивостей для керування виглядом і поведінкою елементів сторінки, таких як розміри, позиціонування, фон, границі, анімація і т. д.

### Загальні принципи використання CSS

* **Селектори**: Використовуйте селектори для точного вибору елементів, які ви хочете стилізувати.
* **Властивості і значення**: Використовуйте властивості для задання конкретних аспектів вигляду (наприклад, колір, шрифт, розмір).
* **Каскадність і спадкування**: Розумійте принцип каскадності і як він впливає на стилі на сторінці.

## **5.4 Спадковості і каскадування за допомогою CSS**

CSS (Cascading Style Sheets) — це мова стилів, яка використовується для опису вигляду HTML-документів. Одними з ключових концепцій у CSS є спадковість (inheritance) і каскадування (cascading). Вони визначають, як стилі застосовуються до елементів на веб-сторінці.

Спадковість у CSS означає, що деякі властивості стилів можуть успадковуватися від батьківських елементів до дочірніх. Це означає, що стилі, застосовані до батьківського елемента, можуть автоматично застосовуватися до його дочірніх елементів, якщо інше не визначено.

## **5.5 Практична частина**

### 5.5.1 Постановки задачі

### 5.5.2 HTML документ

# ВИСНОВКИ

Висновок по HDML: Під час вивчення HTML я отримав великий досвід у створенні структурованих веб-сторінок. Я освоїв основні елементи HTML і їх властивості, що дозволило мені ефективно організовувати та форматувати контент для кращої читабельності та зручності користувачів. Використання різноманітних тегів для тексту, таблиць, списків і форм дало мені змогу логічно структурувати інформацію на сторінці. Опанування основ CSS дозволило мені застосовувати різноманітні стилі для створення привабливого зовнішнього вигляду сторінки, включаючи зміну кольорів, розмірів шрифтів та розташування елементів. Підсумовуючи мою практику, вона дала мені не лише технічні навички у роботі з HTML і CSS, але й глибше розуміння важливості організації контенту для досягнення оптимальної ефективності і задоволення користувачів. цей досвід став важливим кроком у моєму професійному розвитку в сфері веб-розробки.

Висновок по С++: Під час вивчення HTML я отримав великий досвід у створенні структурованих веб-сторінок. Я освоїв основні елементи HTML і їх властивості, що дозволило мені ефективно організовувати та форматувати контент для кращої читабельності та зручності користувачів. Використання різноманітних тегів для тексту, таблиць, списків і форм дало мені змогу логічно структурувати інформацію на сторінці. Опанування основ CSS дозволило мені застосовувати різноманітні стилі для створення привабливого зовнішнього вигляду сторінки, включаючи зміну кольорів, розмірів шрифтів та розташування елементів. Підсумовуючи мою практику, вона дала мені не лише технічні навички у роботі з HTML і CSS, але й глибше розуміння важливості організації контенту для досягнення оптимальної ефективності і задоволення користувачів. цей досвід став важливим кроком у моєму професійному розвитку в сфері веб-розробки.

Всновки по Word, Exel: Під час вивчення Microsoft Word і Excel я отримав значні навички і знання, які стали важливими у моєму робочому процесі та професійному розвитку:

1. **Microsoft Word**:
   * Освоїв основи створення і форматування документів, включаючи вставку тексту, зображень, таблиць та інших об'єктів.
   * Навчився використовувати стилі, нумерацію та маркування для структурування документів і полегшення їх читабельності.
   * Здобув навички редагування і форматування тексту, включаючи вирівнювання, вибір шрифтів, розмірів та колірів.
2. **Microsoft Excel**:
   * Освоїв створення аркушів Excel, роботу з клітинками, формулами та функціями для обробки даних.
   * Навчився створювати і форматувати таблиці, включаючи сортування, фільтрацію і групування даних.
   * Здобув навички використання базових і складних функцій Excel для обчислень, аналізу даних і побудови звітів.

Ці навички в Microsoft Word і Excel допомогли мені підвищити продуктивність і ефективність у роботі з документами і даними, що є важливими для різних аспектів професійної діяльності, включаючи адміністрування, звітність та аналіз інформації.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ

1. ГОСТ 2.105-95 ЕСКД Общие требования к текстовым документам.

2. ДСТУ 1.5-93 Загальні вимоги до побудови, викладу, оформлення та змісту.

# ДОДАТКИ