

**ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ**

**СИСТЕМА УПРАВЛІННЯ ЯКІСТЮ**

**Система забезпечення якості освітньої діяльності та якості вищої освіти**  
*сертифікована на відповідність ДСТУ ISO 9001:2015 / ISO 9001:2015*

**Кафедра комп'ютерних наук та інформаційних систем**

**ЗАТВЕРДЖЕНО**

вченою радою ДТЕУ

(пост. П. 9 від «28» 09 2022 р.)

Ректор



**Анатолій МАЗАРАКІ**

**АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ/  
ALGORITHMIZATION AND PROGRAMMING**

**ПРОГРАМА /  
COURSE SUMMARY**

**Київ 2022**

**Розповсюдження і тиражування без офіційного дозволу ДТЕУ  
заборонено**

Автор: Т.В. Томашевська, кан.тех.наук. доц.

Програму розглянуто і затверджено на засіданні кафедри комп'ютерних наук та інформаційних систем 17.05.2022р., протокол № 18

Рецензенти: О.І.Пурський док.фіз-мат наук проф.  
С.П. Кудрявцева, Провідний науковий співробітник  
Міжнародного науково-навчального центру інформаційних технологій та систем НАН та МОН України,  
канд. техн. наук.

**АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ/  
ALGORITHMIZATION AND PROGRAMMING**

**ПРОГРАМА /  
COURSE SUMMARY**

## ВСТУП

Програма дисципліни «Алгоритмізація та програмування» призначена для здобувачів першого рівня вищої освіти ОС «Бакалавр», галузі знань 12 «Інформаційні технології» спеціальності 122 "Комп'ютерні науки" спеціалізації "Комп'ютерні науки" та спеціальності 126 "Інформаційні системи та технології" спеціалізації "Інформаційні системи та технології".

Програму підготовлено з урахуванням вимог Стандартів вищої освіти України із зазначених спеціальностей та відповідних освітньо- професійних програм підготовки бакалаврів.

Розроблена програма складається з таких розділів:

1. Мета, завдання та предмет дисципліни.
2. Передумови вивчення дисципліни як вибіркової компоненти освітньої програми.
3. Результати вивчення дисципліни.
4. Зміст дисципліни.
5. Список рекомендованих джерел.

### **1. МЕТА, ЗАВДАННЯ ТА ПРЕДМЕТ ДИСЦИПЛІНИ**

*Метою* вивчення дисципліни “Алгоритмізація та програмування” є формування у студентів навичок оволодіння технологіями обробки простих та структурованих даних, опанування технологій структурного, модульного та об'єктно-орієнтованого програмування на базі мови програмування C++.

*Завданням* вивчення дисципліни є теоретична та практична підготовка студентів з таких питань:

- парадигми програмування (структурного, модульного, об'єктно-орієнтованого) та засоби сучасних мов програмування для реалізації різних концепцій;
- засоби мов програмування для реалізації розгалужених та циклічних алгоритмів;
- використання функцій та окремих модулів користувача;
- реалізація класичних інформаційних структур (списків, дерев) з використанням статичного та динамічного розподілу пам'яті;
- базові технології проектування процесів пошуку та побудови впорядкованих даних у лінійних списках та деревовидних структурах;
- використання класів для реалізації парадигми об'єктно-орієнтованого програмування;
- засоби створення ієрархічної об'єктної структури з використанням

базового об'єкта, подальше розширення його властивостей з використанням статичних та віртуальних методів;

- сучасні інструментальні засоби для створення прикладного програмного забезпечення. Автоматизація програмування: технологія RAD (Rapid Application Development);

- особливості програмування в операційних системах родини Windows. Система управління повідомленнями. Структура програми для Windows;

*Предметом* дисципліни є знання у сфері технологій програмування простих та більш складних алгоритмів реалізованих засобами мови програмування C++.

## **2. ПЕРЕДУМОВИ ВИВЧЕННЯ ДИСЦИПЛІНИ ЯК ВИБІРКОВОЇ КОМПОНЕНТИ ОСВІТНЬОЇ ПРОГРАМИ**

*Знання:*

- предмету і сутності комп'ютерних технологій обробки даних;
- особливостей роботи операційної системи Windows;
- основ вищої математики та дискретного аналізу;
- основ методології математичного моделювання економічних процесів;
- механізмів застосування теоретичних методів і моделей у відображенні економічних процесів;

*Вміння:*

- вільно працювати з файловою структурою операційної системи Windows;
- працювати з основними програмами пакету MS Office;
- розуміти математичні моделі структурованих даних;

## **3. РЕЗУЛЬТАТИ ВИВЧЕННЯ ДИСЦИПЛІНИ**

Дисципліна «Алгоритмізація та програмування», як обов'язкова компонента освітньої-професійної програми, забезпечує оволодіння студентами загальними та фаховими компетентностями і досягнення ними програмних результатів навчання за відповідними освітньо-професійними програмами:

*"Комп'ютерні науки" (ОС бакалавр)*

<b>Номер в освітній програмі</b>	<b>Зміст компетентності</b>	<b>Номер теми, що розкриває зміст компетентності</b>
----------------------------------	-----------------------------	--

<i>Загальні компетентності за освітньо-професійною програмою</i>		
ЗК 2	Здатність застосовувати знання у практичних ситуаціях.	1.1-1.7, 2.1-2.5
ЗК 3	Здатність до розуміння предметної області та професійної діяльності.	1.1,1.4,1.7,2.4,2.5
<i>Фахові компетентності за освітньо-професійною програмою</i>		
СК 3	Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.	1.2-1.6,2.1-2.3
СК 4	Здатність використовувати сучасні методи математичного моделювання об'єктів, процесів і явищ, розробляти моделі й алгоритми чисельного розв'язування задач математичного моделювання, враховувати похибки наближеного чисельного розв'язування професійних задач.	1.7,2.1,2.4
СК 8	Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.	1.6,2.4
<i>Програмні результати навчання за освітньо-професійною програмою</i>		
ПРН 5	Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.	1.2,1.3,2.2,2.3
ПРН 9	Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі	1.1,1.4,2.1-2.3

	комп'ютерних наук.	
ПРН 13	Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.	1.5,2.5
ПРН14	Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.	1.5,1.6,1.7,2.5
ПРН 17	Виконувати паралельні та розподілені обчислення, застосовувати чисельні методи та алгоритми для паралельних структур, мови паралельного програмування при розробці та експлуатації паралельного та розподіленого програмного забезпечення.	2.4

*"Інформаційні системи та технології" (ОС бакалавр)*

<b>Номер в освітній програмі</b>	<b>Зміст компетентності</b>	<b>Номер теми, що розкриває зміст компетентності</b>
<i>Загальні компетентності за освітньо-професійною програмою</i>		
КЗ 2	Здатність застосовувати знання у практичних ситуаціях.	1.1-1.7, 2.1-2.5
<i>Спеціальні (фахові) компетентності за освітньо-професійною програмою</i>		
КС 3	Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно-апаратного забезпечення інформаційних систем та технологій, Інтернету речей (ІоТ), комп'ютерно-інтегрованих систем та системної мережної структури, управління ними.	1.2-1.6,2.1-2.3

КС 4	Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).	1.1,1.7, 2.4,2.5
<i>Програмні результати навчання за освітньо-професійною програмою</i>		
ПРН 2	Застосовувати знання фундаментальних і природничих наук, системного аналізу та технологій моделювання, стандартних алгоритмів та дискретного аналізу при розв'язанні задач проектування і використання інформаційних систем та технологій.	1.4,1.6,2.4
ПРН 3	Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.	1.2,1.3,1.7,2.3
ПРН 4	Проводити системний аналіз об'єктів проектування та обґрунтовувати вибір структури, алгоритмів та способів передачі інформації в інформаційних системах та технологіях.	1.1,1.3,1.4,2.1,2.2
ПР 5	Аргументувати вибір програмних та технічних засобів для створення інформаційних систем та технологій на основі аналізу їх властивостей, призначення і технічних характеристик з урахуванням вимог до системи і експлуатаційних умов; мати навички налагодження та тестування програмних і технічних засобів інформаційних систем та технологій.	1.2,1.6,1.7,2.2,2.4
ПР 6	Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх	1.5,1.6,1.7,2.5

	запровадження у професійній діяльності.	
ПРН 7	Обґрунтовувати вибір технічної структури та розробляти відповідне програмне забезпечення, що входить до складу інформаційних систем та технологій.	1.1,1.6,2.2,2.5

## 4. ЗМІСТ ДИСЦИПЛІНИ

### РОЗДІЛ 1. ОСНОВИ ПРОГРАМУВАННЯ МОВОЮ C++

#### Тема 1.1. Введення у дисципліну «Алгоритмізація та програмування».

##### Огляд можливостей мови програмування C++

Вступ. Мета та завдання дисципліни, її місце у освітньому процесі. Парадигма та основні ідеї, покладені у сучасні алгоритмічні мови програмування. Введення до мови програмування C++, її місце серед інших мов програмування. Види програмного забезпечення для розробки програмних продуктів мовою C++. Знайомство з інструментальними засобами Microsoft Visual Studio. Приклад створення першого проекту для консольної прикладної програми. Основні структурні компоненти програми C++, функція **main()**. Компіляція та запуск програм на виконання. Етап налагодження програми. Інтерактивні засоби Microsoft Visual C++ для налагодження програм: покрокове виконання програми, перегляд значень змінних, встановлення точок зупинок.

##### *Список рекомендованих джерел*

*Основний: 1[13-16],3[1-47],4[1-15].*

*Додатковий: 6[21-25],9[12-48].*

*Інтернет джерела: 10,12,14*

#### Тема 1.2. Основні поняття мови програмування. Базові типи, константи, змінні, операції, вирази

Алфавіт мови програмування. Поняття лексеми. Лексична структура мови програмування. Типи лексем: константи, літерали, ідентифікатори, ключові слова, операції, розділові знаки. Числові константи у різних системах числення, літерні константи, символні константи, використання спеціальних символів для визначення недрукованих символів. Призначення та використання коментарів у програмі. Поняття синтакси- су та семантики мови програмування. Концепція типу даних. Базові типи. Кваліфікатори типів. Співвідношення між розмірами даних базових типів. Поняття змінної. Оголошення та ініціалізація змінних. Поняття оператора. Простий та складений оператор. Порожній



оператор. Оператор присвоєння. Поняття виразу. Операнди та операції. Унарні та бінарні операції. Арифметичні операції. Операції відношення. Логічні операції. Побітові операції. Операції інкременту та декременту. Приклади використання операцій. Пріоритет виконання операцій у виразах. Перетворення типів у виразах.

*Список рекомендованих джерел*

*Основний: 1[16-29],2[48-125],4[15-143].*

*Додатковий: 9[11-35].*

*Інтернет джерела: 10,11,13*

### **Тема 1.3. Керування виконанням програми**

Поняття розгалужених та циклічних алгоритмів. Приклади задач, розв'язання яких зводиться до застосування розгалужених та циклічних алгоритмів. Рекурентні співвідношення, циклічні ітераційні процеси, табуляція функцій, наближене обчислення функцій. Реалізація розгалужених алгоритмів за допомогою операторів: **if ... else ; if ... else if ... else; switch**. Порівняння операторів розгалуження. Реалізація циклічних процесів за допомогою операторів: **while ; for ; do while**. Порівняння операторів циклів. Особливості операторів циклів з лічильником та з умовою. Відмінність між синтаксисом та семантикою циклічних конструкцій з пост- та передумовою. Продовження та достроковий вихід із циклів: оператори **break** та **continue**. Оператор переходу **goto**. Правила побудови вкладених операторів циклів та розгалуження.

*Список рекомендованих джерел*

*Основний: 1[16-29],2[48-125],4[15-143].*

*Додатковий: 7[114-150].*

*Інтернет джерела: 10,11,13*

### **Тема 1.4. Складені типи даних: масиви. Алгоритми на масивах**

Концепція складених типів даних. Поняття масиву, індексу, елемента. Оголошення та ініціалізація одновимірних масивів. Методи доступу до елементів масивів. Індексний доступ до елементів масиву. Використання вказівників для посилання на елементи масиву. Багатовимірні масиви. Розміщення елементів багатовимірних масивів у пам'яті. Вказівники на багатовимірні масиви. Розрізи багатовимірних масивів. Доступ до рядків та стовпців двовимірних масивів. Використання циклів для обробки масивів. Алгоритм вибору простих чисел за допомогою масиву.

*Список рекомендованих джерел*

*Основний: 1[150-191].*

*Додатковий: 5[122-159],6[122-138],8[73-77].*

*Інтернет джерела: 12,15.*

## Тема 1.5. Вказівники та адресна арифметика

Поняття та оголошення вказівника. Використання модифікаторів в оголошеннях вказівників. Вказівники типу **void**. Вказівники та адреси. Операції зі вказівниками. Унарні операції одержання адреси та розкриття посилання вказівника. Динамічний розподіл пам'яті. Операції **new**, **delete**. Функції **malloc()** та **free()**. Вказівники та аргументи функцій. Використання вказівників для передачі фактичних аргументів за посиланнями. Вказівникина функції.

Масиви символів. Використання вказівників для доступу до символів рядків. Масиви вказівників. Ініціалізація масивів вказівників. Вказівники на вказівники. Правила інтерпретації складних оголошень. Приклади складених оголошень.

### *Список рекомендованих джерел*

*Основний: 2[167-191],3[149-198].*

*Додатковий: 8[70-103].*

*Інтернет джерела: 10,13.*

## Тема 1.6. Функції користувача та класи пам'яті

Парадигма структурного та модульного програмування. Загальні відомості про функції користувача. Синтаксис для оголошення та визначення функцій користувача. Формальні та фактичні параметри функцій. Виклик функцій. Підстановка фактичних параметрів за значеннями та посиланнями: основна різниця між двома способами передачі аргументів. Приклади створення функцій користувача та їх виклику. Функції зі змінним числом аргументів. Поняття про рекурсію. Труднощі, що виникають під час реалізації рекурсії. Приклади класичних рекурсивних алгоритмів. Поняття області видимості та "часу життя" програмних об'єктів. Зовнішні змінні та функції. Локальні імена та локальні змінні, автоматична пам'ять. Оголошення та ініціалізація статичних змінних. Регістрові змінні. Змінні класу **volatile**. Передача аргументів у функцію **main()**.

### *Список рекомендованих джерел*

*Основний: 1[211-236],2[125-167].*

*Додатковий: 5[119-121],6[86-102], 7[197-219], 9[215-243]*

*Інтернет джерела:10,14.*

## Тема 1.7. Складені типи даних: структури, об'єднання, перелічення

Концепція складеного типу даних. Основні відомості про структури. Тег та елементи структури. Приклади структур. Описання та ініціалізація структури. Використання декларації **typedef** для створення нових типів. Операції зі структурами. Доступ до елементів структури за допомогою операції ".". Використання вказівників для одержання доступу до елементів структури: операція "->". Масиви структур. Поняття запису. Рекурсивні оголошення структур

з використанням вказівників. Приклади рекурсивних структур: лінійний список, бінарне дерево. Об'єднання (**union**) як спосіб збереження даних типів даних в одній області пам'яті. Приклади об'єднань. Створення списків-переліків (множин). Декларація **enum**. Зміна нумерації елементів множин.

*Список рекомендованих джерел*

*Основний: 2[281-300].*

*Додатковий: 6[155-160].*

*Інтернет джерела: 10,12,14.*

## РОЗДІЛ 2. C++ ТА ОБ'ЄКТНООРІЄНТОВАНЕ ПРОГРАМУВАННЯ

### Тема 2.1. Основні поняття та властивості об'єктно-орієнтованого програмування (ООП). Класи C++

Концепція об'єктно-орієнтованого програмування, її відмінність від концепції структурного програмування. Компоненти об'єктно орієнтованої парадигми: об'єкт, повідомлення, клас, властивість, метод. Основні властивості ООП: абстракція, інкапсуляція, наслідування, поліморфізм. Інтерпретація основних понять ООП в термінах C++. Клас як розширення структурного типу. Синтаксис описання класу. Ідентифікатори доступу **private**, **protected**, **public**. Інкапсуляція даних та функцій. Призначення та створення конструкторів класу. Параметри конструктора. Призначення та створення деструктора класу. Приклади опису класів. Об'єкт як екземпляр деякого класу. Створення об'єкта. Доступ до властивостей та методів об'єкта. Використання вказівника **this** для одержання доступу до поточного об'єкта. Опис класу в двох файлах: інтерфейс класу (файл \*.h), визначення функцій класу (файл \*.cpp).

*Список рекомендованих джерел*

*Основний: 3[246-268],4[407-430].*

*Додатковий: 8[163-171].*

*Інтернет джерела: 11,12,13.*

### Тема 2.2. Структура та ієрархія класів, наслідування, поліморфізм

Поняття простого наслідування, базового класу, похідного класу, ієрархії класів. Синтаксис визначення похідного класу на основі базового. Відкриті та закриті похідні класи. Правила доступу до полів даних та функцій базових класів. Правила доступу до похідних класів та їх об'єктів. Створення ієрархії класів для простого наслідування. Приклади простого наслідування. Поняття поліморфізму та інтерпретація його в термінах віртуальних функцій C++. Зв'язування методів з об'єктами на етапі виконання програми – "пізнє зв'язування". Зв'язування методів з об'єктами на етапі компіляції – "раннє зв'язування".

зв'язування". Відмінності між двома способами зв'язування. Поняття поліморфного кластера. Реалізація віртуальних функцій. Приклади віртуальних функцій. Поняття множинного наслідування. Подання множинного наслідування у вигляді графів. Приклади множинного наслідування. Конфлікти, що можуть виникати під час множинного наслідування. Розв'язання конфліктів.

*Список рекомендованих джерел*

*Основний: 3[295-343],4[495-540].*

*Додатковий: 8[172-206].*

*Інтернет джерела: 10,11,12.*

### **Тема 2.3. Перевантаження операторів у C++. Шаблони та шаблонні функції.**

Способи перевантаження операцій у C++. Обмеження, що виникають під час перевантаження операцій. Використання класів для перевантаження операцій. Приклад класу для перевантаження операцій " + ", " \* ". Перевантаження операцій new та delete. Перевантаження оператора присвоєння. Шаблони та шаблонні функції.

*Список рекомендованих джерел*

*Основний: 3[295-343],4[495-540].*

*Додатковий: 8[254-296].*

*Інтернет джерела: 10,12,13*

### **Тема 2.4. Організація абстрактних структур даних**

Поняття лінійного списку. Методи організації та збереження списків. Використання масивів для збереження елементів списків. Спосіб зв'язаного збереження списків з використанням рекурсивно визначених структур. Огляд операцій, що виникають під час обробки списків. Реалізація операцій для послідовного збереження списків. Реалізація операцій для зв'язаного збереження списків. Двозв'язані списки. Реалізація основних операцій для двозв'язаних списків. Циклічні списки.

Стек (структура LIFO - Last In First Out) як спеціальний тип лінійного списку. Поняття вершини стека. Організація та збереження елементів стека. Основні операції для роботи зі стеками.

Черга (структура FIFO - First In First Out) як спеціальний тип лінійного списку. Поняття початку та кінця черги. Основні операції для роботи з чергами.

Основні поняття та визначення: дерево, корінь, вузол, листок, дочірній вузол, рівень вузла, праве та ліве під дерево, глибина дерева. Способи збереження дерев. Алгоритми для проходження дерева: прямий, симетричний, обернений. Основні операції для роботи з деревами.

*Список рекомендованих джерел*

*Основний: 2[328-371].*

*Додатковий: 5[201-214],6[199-203].  
Інтернет джерела: 11,13.*

## **Тема 2.5. Управління компіляцією. Стандартна бібліотека STL**

Призначення C++ препроцесора. Поняття та використання макропідстановки. Умовна компіляція. Основні компоненти STL. STL-рядки. Вектори. Ітератори. Алгоритми.

*Список рекомендованих джерел  
Основний: 3[28-39].  
Додатковий: 7[23-37].  
Інтернет джерела: 10,14.*

## **5. СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ**

### ***Основний:\****

1. Рудий Т. В., Паранчук Я. С., Сенік В. В. Алгоритмізація та програмування. Частина 1. Структурне програмування : навчальний посібник. Львів : Львівський державний університет внутрішніх справ, 2022. 240 с.
2. Ковалюк Т. В. Алгоритмізація та програмування: підручник для студентів вищих навчальних закладів, що навчаються за напрямками "Комп'ютерні науки", "Комп'ютерна інженерія", "Програмна інженерія". Міністерство освіти і науки, молоді та спорту України. Львів :Видавництво "Магнолія 2006". 2019. – 401 с.
3. Grimes R. Beginning C++ Programming. Packt Publishing Ltd. 2017. 516 p.
4. Halterman R.L. Fundamentals of Programming C++. Southern Adventist University, 2018. 766 p.

### ***Додатковий***

5. Трофименко О. Г., Прокоп Ю. В., Задерейко О. В. Алгоритмізація та програмування : навчально-методичний посібник. Одеса : Фенікс, 2020. 310 с.
6. Кублій Л.І. Алгоритмізація та програмування: Практикум [Електронний ресурс]: навч.посіб. для здобувачів ступеня бакалавра за спеціальністю 122 “Комп'ютерні науки”. КПІ ім. Ігоря Сікорського. Електронні текстові дані (1 файл: 28,15 Мбайт). Київ: КПІ ім. Ігоря Сікорського, 2019. 209 с.
7. Тверитникова О.Є, Крилова В.А., Васильченков О.Г. Базові алгоритми та основи програмування. Теорія і практика : навч.-метод. посіб. Харків : НТУ «ХПІ», 2020. 264 с.

8. Васильєв О. Програмування на С++ в прикладах і задачах : навчальний посібник. Київ :Видавництво Ліра-К, 2020.

9. Козак Л. І., Костюк І.В., Стасевич С.П. Основи програмування: навчальний посібник. Львів : Новий Світ. 2000. 2019.

### **Інтернет ресурси**

10. Уроки програмування на С++. Режим доступу: <https://acode.com.ua/uroki-po-cpp/>

11. С++. Теорія. Плейліст. Режим доступу: <https://www.youtube.com/playlist?list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g>

12. News, Status & Discussion about Standard C++. Режим доступу: <https://isocpp.org/>

13. С++ Tutorial. Уроки з мови С++. Режим доступу: <https://www.w3schools.com/cpp/default.asp>

14. Довідник по IDE VisualStudio Режим доступу: <https://msdn.microsoft.com/>.

\*Курсивом позначені видання, що присутні у бібліотеці ДТЕУ