

**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ  
«КИЇВСЬКИЙ ФАХОВИЙ КОЛЕДЖ МІСЬКОГО ГОСПОДАРСТВА  
ТАВРІЙСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ  
ІМЕНІ В. І. ВЕРНАДСЬКОГО»**

Циклова комісія комп'ютерно інтегрованих технологій

ЗАТВЕРДЖУЮ

Заступник директора коледжу з  
навчально-виховної роботи

Людмила ПУСТОВОЙТ

«29» серпня 2022 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

**ВНПП03.01.01 ПРОГРАМУВАННЯ**

**підготовки фахового молодшого бакалавра**

**освітньо-професійної програми Обслуговування комп'ютерних систем і мереж  
спеціальності 123 Комп'ютерна інженерія**

відділення Екології, комп'ютерних систем та автоматизації


Київ – 2022

Робоча програма з дисципліни Програмування для підготовки фахових молодших бакалаврів за освітньо-професійною програмою Обслуговування комп'ютерних систем і мереж студентів III курсу спеціальності 123 Комп'ютерна інженерія.

РОЗРОБНИКИ : Валерія ДРОМЕНКО, к.т.н., доц., спеціаліст вищої категорії

Робочу програму схвалено на засіданні циклової комісії комп'ютерно інтегрованих технологій

Протокол № 1 від «26» серпня 2022 р.

Голова циклової комісії \_\_\_\_\_  Людмила ГЛУШКО

Розглянуто і затверджено навчально-методичною радою коледжу

Протокол № 1 від «29» серпня 2022 р.

Голова НМР \_\_\_\_\_  Аліна ОДИНЕЦЬ

## Зміст

1. Пояснювальна записка	4
2. Навчально-тематичний план дисципліни	5
3. Календарно-тематичний план дисципліни	5
4. Теми і плани лекційних занять	9
5. Теми практичних занять	11
6. Теми і завдання для самостійної роботи студентів	13
7. Методи активізації навчального процесу	15
8. Система поточного і підсумкового контролю знань	15
9. Критерії оцінювання навчальних досягнень здобувачів освіти	17
10. Рекомендована література	18
Додатки	18

## **1. Пояснювальна записка**

### **Актуальність даної програми сьогодні.**

Дисципліна «Програмування» належить до вибірових компонентів ОПП за вибором здобувачів освіти.

Курс програмування є органічним поєднанням методів та способів опрацювання інформації, математичних моделей, алгоритмів обчислювальних процесів, інформаційних технологій та систем автоматизованого проектування.

**Мета дисципліни** програмування – здобуття студентами знань, вмінь та навичок у вирішуванні типових спеціалізованих задач в галузі інформаційних технологій в процесі професійної діяльності або навчання, що вимагає застосування методів і технологій комп'ютерної інженерії та може характеризуватися певною невизначеністю умов.

Основним **завданням** програмування є ознайомлення з принципами роботи в середовищі програмування високого рівня MICROSOFT VISUAL C++. Предметом дисципліни є формування у студентів вміння самостійно складати алгоритми і на їх основі створювати програмне забезпечення; розвивати у студентів вміння створювати оптимізоване, як по розміру коду, так і по швидкодії, програмне забезпечення.

Процес вивчення дисципліни ВНПП03.01.01 Програмування спрямований на формування елементів наступних компетентностей:

### **Загальні компетентності**

КЗ 3. Здатність до абстрактного мислення, аналізу та синтезу.

КЗ 4 Здатність спілкуватися державною мовою як усно, так і письмово.

КЗ 7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел та практичного її застосування.

КЗ 8. Здатність вчитися і бути сучасно навченим

### **Спеціальні (фахові) компетентності**

КФ 6. Здатність застосовувати засоби сучасних мов програмування, основи структур даних для розробки програмного забезпечення.

КФ 7. Здатність використовувати професійно-орієнтовані знання і практичні навички з дисциплін циклу професійної та практичної підготовки для проектування, побудови та обслуговування сучасних комп'ютерних мереж різного виду та призначення.

КФ 8. Здатність використовувати знання сучасних технологій та інструментальних засобів розробки складних програмних систем (інженерії програмного забезпечення), уміння їх застосовувати на всіх етапах життєвого циклу розробки.

КФ 9. Здатність брати участь в модернізації та реконструкції апаратних та програмних засобів комп'ютерної інженерії, зокрема з метою підвищення їх ефективності.

КФ 11. Здатність до ділових комунікацій у професійній сфері, знання основ ділового спілкування, навички роботи в команді.

КФ 14. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати прийняті рішення.

### **Програмні результати навчання**

РН 1. Знати способи аналізу, синтезу та подальшого сучасного навчання. Вміти проводити аналіз інформації, приймати обґрунтовані рішення, вміти придбати

сучасні знання. Встановлювати відповідні зв'язки для досягнення цілей. Нести відповідальність за своєчасне набуття сучасних знань.

РН 2. Мати спеціалізовані концептуальні знання, набуті у процесі навчання. Вміти розв'язувати складні задачі і проблеми, які виникають у професійній діяльності. Зрозуміле і недвозначне донесення власних висновків, знань та пояснень, що їх обґрунтовують, до фахівців та нефахівців. Відповідати за прийняття рішень у складних умовах.

РН 3. Мати глибокі знання із структури професійної діяльності. Вміти здійснювати професійну діяльність, що потребує оновлення та інтеграції знань. Здатність ефективно формувати комунікаційну стратегію у професійній діяльності. Нести відповідальність за професійний розвиток, здатність до подальшого професійного навчання з високим рівнем автономності.

РН 4. Знати види та способи адаптації, принципи дії в новій ситуації. Вміти застосувати засоби саморегуляції, вміти пристосовуватися до нових ситуацій (обставин) життя та діяльності. Встановлювати відповідні зв'язки для досягнення результату. Нести відповідальність своєчасне використання методів саморегуляції.

РН 5. Знати тактики та стратегії спілкування, закони та способи комунікативної поведінки. Вміти приймати обґрунтоване рішення, обирати способи та стратегії спілкування для забезпечення ефективної командної роботи. Нести відповідальність за вибір та тактику способу комунікації.

РН 6. Мати досконалі знання державної мови та базові знання іноземної мови. Вміти застосовувати знання державної мови, як усно так і письмово, вміти спілкуватись іноземною мовою. Використовувати при фаховому та діловому спілкуванні та при підготовці документів державну мову. Використовувати іноземну мову у професійній діяльності.

РН 12. Вміти застосовувати базові знання стандартів в області інформаційних технологій при розробці та впровадженні інформаційних систем і технологій

РН 14. Володіти навиками аналізу навчальної і спеціальної літератури, нормативних положень, технічної документації для вирішення проблем, що виникають у професійній діяльності.

РН 17. Знати сучасні методи побудови та аналізу ефективних алгоритмів, основи теорії чисельних методів, вміти аналізувати, оцінювати та вибирати існуючі алгоритми, розробляти нові алгоритми, які пов'язані з проектуванням апаратних та програмних компонент КСМ

РН 19. Володіти засобами сучасних мов програмування для створення програмних продуктів.

РН 28. Вміти опановувати та розробляти документацію на системи, продукти і сервіси інформаційних технологій, професійно спілкуватись рідною та англійською мовою

РН 29. Вміти використовувати існуючі та розробляти нові математичні методи, використовувати відповідні закони фізики для вирішення задач, пов'язаних з проектуванням та використанням КСМ.

РН 34. Уміти застосовувати засоби сучасних мов програмування під час програмної реалізації алгоритмів професійних задач.

РН 42. Уміння застосовувати технології та інструментальні засоби проектування для створення програмних систем.

РН 45. Практично володіти рідною та однією з іноземних мов в обсязі тематики, зумовленої професійними потребами.

РН 46. Використовувати відповідну термінологію у власних дослідженнях та професійній діяльності державною мовою та/або іноземною; спілкуватися в діалоговому режимі в галузі професійної діяльності; вміти презентувати результати власних досліджень та описувати їх у фахових публікаціях, використовуючи сучасні інформаційні та комунікативні технології

РН 48. Вдосконалювати професійний та особистісний розвиток протягом усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

РН 49. Критично осмислювати основні теорії, принципи, методи і поняття у навчанні та професійній діяльності.

РН 50. Дотримуватися етичних норм, враховуючи авторське право та норми академічної доброчесності при проведенні досліджень, розробці програмних продуктів, проектів, презентацій результатів роботи.

## 2. Навчально-тематичний план дисципліни

№	Назва розділу	Кількість годин			
		Всього	Лекції	Практ.	Самост.
<b>V семестр</b>					
1.	Вступ до основ комп'ютерних технологій і програмування	16	4	8	6
2.	Базисні можливості мови програмування C++ і середовища MICROSOFT VISUAL C++	128	8	76	42
<b>VI семестр</b>					
3.	Розширені можливості мови програмування C++ і середовища MICROSOFT VISUAL C++.	96	18	46	32
	Всього	240	30	130	80

## 3. Календарно-тематичний план дисципліни

№ з/п	Назва розділу, теми, заняття	Кількість годин		
		Лекції	Практ.	Самост.
<b>V семестр</b>				
Розділ 1. Вступ до основ комп'ютерних технологій і програмування				
1.	Основні концепції комп'ютерних технологій і програмування.	2		
2.	Знайомство з програмуванням.		2	
3.	Загальний опис типового середовища програмування на C++		2	

4.	Основи алгоритмізації задач.	2		
5.	Способи запису алгоритмів.		2	
6.	Правила складання блок-схем. Основні структури алгоритмів.		2	3
Розділ 2. Базисні можливості мови програмування C++ і середовища MICROSOFT VISUAL C++				
7.	Вступ до програмування мовою C++.	2		
8.	Структура програми.		2	
9.	Проста програма: виведення тексту.		2	
10.	Проста програма: додавання чисел.		2	
11.	Алфавіт C++, коментарі.		2	
12.	Типи даних.	2		
13.	Приклади застосування типів <code>int</code> , <code>char</code> , <code>bool</code> .		2	
14.	Приклади застосування типів <code>float</code> , <code>double</code>		2	
15.	Змінні. Літерали (константи) C++.		2	
16.	Арифметичні операції в C++.		2	
17.	Постановка та розв'язування задач лінійної структури.		2	3
18.	Алгоритмізація задач лінійної структури.		2	3
19.	Реалізація програм лінійної структури.		2	3
20.	Інструкції керування для реалізації програм розгалуженої структури	2		
21.	Приклад застосування інструкції вибору за умовою <code>if else</code>		2	
22.	Приклад застосування інструкції безумовного переходу <code>goto</code> .		2	
23.	Приклад застосування інструкції множинного вибору <code>switch</code> .		2	
24.	Постановка та розв'язування задач простої розгалуженої структури.		2	3
25.	Алгоритмізація задач простої розгалуженої структури.		2	3
26.	Реалізація програм простої розгалуженої структури.		2	3
27.	Постановка та розв'язування задач складної розгалуженої структури.		2	3
28.	Алгоритмізація задач складної розгалуженої структури.		2	3
29.	Реалізація програм складної розгалуженої структури.		2	3
30.	Інструкції керування для реалізації програм циклічної структури	2		

31.	Приклад застосування інструкції повторення <code>for</code> .		2	
32.	Приклад застосування інструкції повторення <code>while</code> .		2	
33.	Приклад застосування інструкції повторення <code>do-while</code>		2	
34.	Приклад застосування операторів <code>continue, break</code> .		2	
35.	Форматування виведення даних.		2	
36.	Постановка та розв'язування задач циклічної структури.		2	3
37.	Алгоритмізація задач циклічної структури.		2	3
38.	Реалізація програм циклічної структури.		2	3
39.	Постановка та розв'язування задач ітераційних циклічних процесів.		2	3
40.	Алгоритмізація задач ітераційних циклічних процесів.		2	3
41.	Реалізація програм ітераційних циклічних процесів.		2	3
42.	Функції в C++. Створення і виклик функцій.		2	
43.	Прототипи функцій і заголовні файли.		2	
44.	Аргументи функцій по замовчуванню.		2	
45.	Області видимості. Локальні і глобальні змінні.		2	
46.	Рекурсія.		2	
47.	Посилання.		2	
48.	Семестровий контроль.		2	
	Всього за V семестр	12	84	48
<b>VI семестр</b>				
Розділ 3. Розширені можливості мови програмування C++ і середовища MICROSOFT VISUAL C++.				
49.	Масиви.	2		
50.	Приклади використання одновимірних масивів.		2	
51.	Сортування одновимірних масивів.	2		2
52.	Пошук в одновимірних масивах.	2		
53.	Перетворення двох масивів в один.		2	
54.	Постановка та розв'язування задач обробки одновимірних масивів.		2	2
55.	Алгоритмізація задач обробки одновимірних масивів.		2	2



56.	Реалізація програм обробки одновимірних масивів.		2	2
57.	Реалізація програм обробки одновимірних масивів.		2	
58.	Передача одновимірних масивів у функції.	2		
59.	Постановка та розв'язування задач з передачею одновимірних масивів у функції.		2	2
60.	Алгоритмізація задач з передачею одновимірних масивів у функції.		2	2
61.	Реалізація програм з передачею одновимірних масивів у функції.		2	2
62.	Реалізація програм з передачею одновимірних масивів у функції.		2	
63.	Двовимірні і багатовимірні масиви.	2		
64.	Приклади використання двовимірних масивів.	2		
65.	Приклади використання двовимірних масивів.		2	
66.	Постановка та розв'язування задач обробки двовимірних масивів.		2	2
67.	Алгоритмізація задач обробки двовимірних масивів.		2	2
68.	Реалізація програм обробки двовимірних масивів.		2	2
69.	Реалізація програм обробки двовимірних масивів.		2	
70.	Передача двовимірних масивів у функції.	2		
71.	Постановка та розв'язування задач з передачею двовимірних масивів у функції.		2	2
72.	Алгоритмізація задач з передачею двовимірних масивів у функції.		2	2
73.	Реалізація програм з передачею двовимірних масивів у функції.		2	2
74.	Реалізація програм з передачею двовимірних масивів у функції.		2	
75.	Обробка символічних масивів.	2		
76.	Стандартні бібліотечні функції для обробки символічних масивів.	2		

77.	Постановка та розв'язування задач обробки символьних масивів.		2	2
78.	Алгоритмізація задач обробки символьних масивів.		2	2
79.	Реалізація програм обробки символьних масивів.		2	2
80.	Семестровий контроль.		2	
Всього за VI семестр		18	46	32
Всього		30	130	80

#### 4. Теми і плани лекційних занять

##### V семестр

##### Розділ 1 . Вступ до основ комп'ютерних технологій і програмування

##### Лекція 1. Основні концепції комп'ютерних технологій і програмування.

1. Мета і задачі курсу.
2. Основні концепції комп'ютерних технологій.
3. Знайомство з програмуванням.
4. Мови програмування.
5. Загальний опис типового середовища програмування на C++.

*Література: [1, с. 5-15].*

##### Лекція 2. Основи алгоритмізації задач

1. Поняття алгоритму.
2. Властивості алгоритмів.
3. Способи запису алгоритмів.
4. Правила складання блок-схем.
5. Основні структури алгоритмів.

*Література: [1, с. 15-19].*

##### Розділ 2. Базисні можливості мови програмування C++ і середовища MICROSOFT VISUAL C++.

##### Лекція 3. Вступ до програмування на C++.

1. Структура програми.
2. Алфавіт C++, коментарі.
3. Концепції пам'яті.

*Література: [1, с. 19-26].*

##### Лекція 4. Типи даних. Літерали (константи) C++.

1. Основні типи даних.
2. Змінні.
3. Літерали.

*Література: [1, с. 26-32].*

##### Лекція 5. Інструкції керування для реалізації програм лінійної та розгалуженої структур.

1. Керуючі структури.
2. Інструкція вибору за умовою `if else` (скорочена та повна форма).
3. Інструкція безумовного переходу `goto`.
4. Інструкція множинного вибору `switch`.

5. Приклад застосування розгалуженої структури алгоритму.  
*Література: [1, с. 40-49].*

**Лекція 6.** Інструкції керування для реалізації програм циклічної структури.

1. Поняття циклічного алгоритму.
2. Арифметичні цикли (з відомим числом повторень).
3. Ітераційні цикли.
4. Операції інкременту та декременту.
5. Інструкція повторення `for`.
6. Інструкція повторення `while` (цикл з передумовою).
7. Інструкція повторення `do-while` (цикл з пост-умовою).
8. Оператор `continue, break`.
9. Форматування виведення даних.

*Література: [1, с. 49-67].*

## **VI семестр.**

### **Розділ 3. Розширені можливості мови програмування C++ і середовища MICROSOFT VISUAL C++.**

**Лекція 7.** Масиви.

1. Поняття масиву.
2. Оголошення і ініціалізація.
3. Звернення до елементів масиву.
4. Приклади використання одновимірних масивів.

*Література: [1, с. 84-90].*

**Лекція 8.** Сортування одновимірних масивів.

1. Метод обміну (бульбашки).
2. Метод вибору.
3. Метод включення.

*Література: [1, с. 90-94].*

**Лекція 9.** Пошук в одновимірних масивах.

1. Ключ пошуку.
2. Перетворення двох масивів в один.

*Література: [1, с. 94-101].*

**Лекція 10.** Передача одновимірних масивів у функції.

1. Виклик по посиланню.
2. Показчик (або адреса).
3. Способи оголошення параметра-масиву.

*Література: [1, с. 101-106].*

**Лекція 11.** Двовимірні і багатовимірні масиви.

1. Поняття двовимірних масивів.
2. Поняття багатовимірних масивів.

*Література: [1, с. 106-107].*

**Лекція 12.** Приклади використання двовимірних масивів.

1. Приклади пошуку елементів масиву.
2. Приклади визначення місця елемента в масиві.

*Література: [1, с. 107-110].*

**Лекція 13.** Передача двовимірних масивів у функції.

1. Способи оголошення.

2. Приклад використання.

*Література: [1, с. 110-114].*

**Лекція 14.** Обробка символьних масивів.

1. Символи і рядки в C++.
2. Оголошення і ініціалізація символьних масивів.
3. Введення рядків з клавіатури.

*Література: [1, с. 114-116].*

**Лекція 15.** Стандартні бібліотечні функції для обробки символьних масивів.

1. Функції копіювання.
2. Функції об'єднання (конкатенації).
3. Функції пошуку.
4. Приклади роботи з символьними масивами.

*Література: [1, с. 116-123].*

**5. Теми практичних занять**

№ з/п	Назва теми	К-ть годин
<b>V семестр</b>		
Розділ 1. Вступ до основ комп'ютерних технологій і програмування		
1.	Знайомство з програмуванням.	2
2.	Загальний опис типового середовища програмування на C++	2
3.	Способи запису алгоритмів.	2
4.	Правила складання блок-схем. Основні структури алгоритмів.	2
Розділ 2. Базисні можливості мови програмування C++ і середовища MICROSOFT VISUAL C++		
5.	Структура програми.	2
6.	Проста програма: виведення тексту.	2
7.	Проста програма: додавання чисел.	2
8.	Алфавіт C++, коментарі.	2
9.	Приклади застосування типів <code>int</code> , <code>char</code> , <code>bool</code> .	2
10.	Приклади застосування типів <code>float</code> , <code>double</code>	2
11.	Змінні. Літерали (константи) C++.	2
12.	Арифметичні операції в C++.	2
13.	Постановка та розв'язування задач лінійної структури.	2
14.	Алгоритмізація задач лінійної структури.	2
15.	Реалізація програм лінійної структури.	2
16.	Приклад застосування інструкції вибору за умовою <code>if else</code>	2
17.	Приклад застосування інструкції безумовного переходу <code>goto</code> .	2
18.	Приклад застосування інструкції множинного вибору <code>switch</code> .	2

19.	Постановка та розв'язування задач простої розгалуженої структури.	2
20.	Алгоритмізація задач простої розгалуженої структури.	2
21.	Реалізація програм простої розгалуженої структури.	2
22.	Постановка та розв'язування задач складної розгалуженої структури.	2
23.	Алгоритмізація задач складної розгалуженої структури.	2
24.	Реалізація програм складної розгалуженої структури.	2
25.	Приклад застосування інструкції повторення <code>for</code> .	2
26.	Приклад застосування інструкції повторення <code>while</code> .	2
27.	Приклад застосування інструкції повторення <code>do-while</code>	2
28.	Приклад застосування операторів <code>continue, break</code> .	2
29.	Форматування виведення даних.	2
30.	Постановка та розв'язування задач циклічної структури.	2
31.	Алгоритмізація задач циклічної структури.	2
32.	Реалізація програм циклічної структури.	2
33.	Постановка та розв'язування задач ітераційних циклічних процесів.	2
34.	Алгоритмізація задач ітераційних циклічних процесів.	2
35.	Реалізація програм ітераційних циклічних процесів.	2
36.	Функції в C++. Створення і виклик функцій.	2
37.	Прототипи функцій і заголовні файли.	2
38.	Аргументи функцій по замовчуванню.	2
39.	Області видимості. Локальні і глобальні змінні.	2
40.	Рекурсія.	2
41.	Посилання.	2
42.	Семестровий контроль.	2
	Всього за V семестр	84
<b>VI семестр</b>		
Розділ 3. Розширені можливості мови програмування C++ і середовища MICROSOFT VISUAL C++.		
43.	Приклади використання одновимірних масивів.	2
44.	Перетворення двох масивів в один.	2
45.	Постановка та розв'язування задач обробки одновимірних масивів.	2
46.	Алгоритмізація задач обробки одновимірних масивів.	2
47.	Реалізація програм обробки одновимірних масивів.	2
48.	Реалізація програм обробки одновимірних масивів.	2
49.	Постановка та розв'язування задач з передачею одновимірних масивів у функції.	2
50.	Алгоритмізація задач з передачею одновимірних масивів у функції.	2

51.	Реалізація програм з передачею одновимірних масивів у функції.	2
52.	Реалізація програм з передачею одновимірних масивів у функції.	2
53.	Приклади використання двовимірних масивів.	2
54.	Постановка та розв'язування задач обробки двовимірних масивів.	2
55.	Алгоритмізація задач обробки двовимірних масивів.	2
56.	Реалізація програм обробки двовимірних масивів.	2
57.	Реалізація програм обробки двовимірних масивів.	2
58.	Постановка та розв'язування задач з передачею двовимірних масивів у функції.	2
59.	Алгоритмізація задач з передачею двовимірних масивів у функції.	2
60.	Реалізація програм з передачею двовимірних масивів у функції.	2
61.	Реалізація програм з передачею двовимірних масивів у функції.	2
62.	Постановка та розв'язування задач обробки символьних масивів.	2
63.	Алгоритмізація задач обробки символьних масивів.	2
64.	Реалізація програм обробки символьних масивів.	2
65.	Семестровий контроль.	2
Всього за VI семестр		46
Всього		130

## 6. Теми і завдання для самостійної роботи студентів

Самостійна робота полягає в індивідуальному опрацюванні матеріалів дисципліни та у виконанні кожним студентом індивідуальних завдань з використанням навчальної та спеціальної літератури під керівництвом викладача.

*Форма контролю:* звіт з виконання індивідуальних завдань, проведення тестів.

№ з/п	Назва теми	К-ть годин
<b>V семестр</b>		
Розділ 1. Вступ до основ комп'ютерних технологій і програмування		
1.	Правила складання блок-схем. Основні структури алгоритмів.	3
Розділ 2. Базисні можливості мови програмування C++ і середовища MICROSOFT VISUAL C++		
2.	Постановка та розв'язування задач лінійної структури.	3

3.	Алгоритмізація задач лінійної структури.	3
4.	Реалізація програм лінійної структури.	3
5.	Постановка та розв'язування задач простої розгалуженої структури.	3
6.	Алгоритмізація задач простої розгалуженої структури.	3
7.	Реалізація програм простої розгалуженої структури.	3
8.	Постановка та розв'язування задач складної розгалуженої структури.	3
9.	Алгоритмізація задач складної розгалуженої структури.	3
10.	Реалізація програм складної розгалуженої структури.	3
11.	Постановка та розв'язування задач циклічної структури.	3
12.	Алгоритмізація задач циклічної структури.	3
13.	Реалізація програм циклічної структури.	3
14.	Постановка та розв'язування задач ітераційних циклічних процесів.	3
15.	Алгоритмізація задач ітераційних циклічних процесів.	3
16.	Реалізація програм ітераційних циклічних процесів.	3
	Всього за V семестр	48
<b>VI семестр</b>		
Розділ 3. Розширені можливості мови програмування C++ і середовища MICROSOFT VISUAL C++.		
17.	Сортування одновимірних масивів.	2
18.	Постановка та розв'язування задач обробки одновимірних масивів.	2
19.	Алгоритмізація задач обробки одновимірних масивів.	2
20.	Реалізація програм обробки одновимірних масивів.	2
21.	Постановка та розв'язування задач з передачею одновимірних масивів у функції.	2
22.	Алгоритмізація задач з передачею одновимірних масивів у функції.	2
23.	Реалізація програм з передачею одновимірних масивів у функції.	2
24.	Постановка та розв'язування задач обробки двовимірних масивів.	2
25.	Алгоритмізація задач обробки двовимірних масивів.	2
26.	Реалізація програм обробки двовимірних масивів.	2
27.	Постановка та розв'язування задач з передачею двовимірних масивів у функції.	2
28.	Алгоритмізація задач з передачею двовимірних масивів у функції.	2
29.	Реалізація програм з передачею двовимірних масивів у функції.	2

30.	Постановка та розв'язування задач обробки символьних масивів.	2
31.	Алгоритмізація задач обробки символьних масивів.	2
32.	Реалізація програм обробки символьних масивів.	2
Всього за VI семестр		32
Всього		80

### 7. Методи активізації навчального процесу

Для активізації навчально-пізнавальної діяльності студентів при вивченні дисципліни застосовуються наступні методики:

- 1) класичні лекції з усним викладенням навчального матеріалу, що відрізняється великою ємністю матеріалу, складністю логічних побудов, інтелектуальних образів, доказів і узагальнень;
- 2) лекції-бесіди з розгорнутим діалогом викладача зі студентами, що організовується за допомогою ретельно продуманої системи питань, що поступово підводять студентів до засвоєння нового поняття чи закономірності;
- 3) метод обговорення навчального матеріалу та дискусії для поглиблення і систематизації знань, усвідомлення розуміння тієї чи іншої проблеми, перевірки підстави висновків, до яких прийшли студенти в ході вивчення конкретної теми;
- 4) виконання студентами самостійних завдань з метою повторення розумових і практичних дій в умовах послідовного ускладнення задачі з метою формування і закріплення навичок і вмінь;
- 5) ілюстрація і показ (демонстрація) з метою переключення форм сприйняття аудиторії й активізації інтересу студентів.

При будь-якому виді навчальних занять можна застосовуються декілька методів навчання в різних поєднаннях. Використання конкретних методів навчання залежить від цільової установки і мети проведення заняття.

### 8. Система поточного і підсумкового контролю знань

Поточний контроль: усне/письмове опитування, самостійна робота, презентація (представлення виконаного завдання); тестові завдання, розв'язання аналітично-розрахункових задач; перевірка конспектів тощо.

Підсумковий контроль: екзамен.

#### Питання до екзамену:

1. Що таке алгоритм? Визначення алгоритму.
2. Властивості алгоритму.
3. Способи відображення алгоритму.
4. Базові конструкції схем алгоритмів.
5. Представлення даних у мові C++.
6. Що таке оператор присвоювання?
7. Чим відрізняється оператор обходу від оператора розгалуження?
8. Визначте основні етапи розробки алгоритмів.
9. Який вплив мають типи даних на процес розробки алгоритму?
10. Які типи даних вважають сумісними?
11. Що розуміють під структурним підходом до розробки алгоритмів?



12. Що треба враховувати при виборі типу даних?
13. Як розуміти таку властивість алгоритмів, як результативність?
14. Як розуміти таку властивість алгоритмів, як дискретність?
15. Як розуміти таку властивість алгоритмів, як масовість?
16. Дайте стисло характеристику алфавіту мови C++.
17. Яку роль відіграють спеціальні символи?
18. Назвати правила оголошення змінної в програмі.
19. Які символи можна використовувати в імені змінної.
20. Назвати основні числові типи мови C++.
21. У програмі записано оператор:  $x=x+dx$ . Яким іншим способом можна записати цей оператор?
22. Синтаксис оператора **cout**?
23. Синтаксис оператора **cin**?
24. Запишіть оператор виводу на екран повідомлення: «Введіть початкові дані для **a** і **b**».
25. Запишіть оператор виводу значень змінних **a** і **b**.
26. Чим відрізняється мова програмування від загальноновживаної?
27. Які основні типи даних підтримують сучасні мови програмування?
28. Чи можна суміщати в арифметичному виразі різні типи?
29. Як змінити пріоритет виконання операцій арифметичного виразу?
30. Чи може в одному файлі бути декілька разів визначена змінна з однаковим іменем?
31. Як перервати нескінчений цикл?
32. Чому небезпечно виконувати цілочисельне ділення?
33. Чи можна в тілі циклу примусово змінювати значення змінної циклу?
34. Чи може у C++ використовуватись глобальна функція?
35. Що забезпечує ефективність реалізації алгоритму?
36. В чому відмінність **break** від **continue**?
37. Для чого потрібне коментування тексту програми?
38. Масиви задаються статичним чи динамічним чином?
39. Як відбувається створення проекту C++?
40. Поняття лінійних алгоритмів.
41. Яку структуру має програма мовою C++?
42. Які математичні функції мови C++?
43. Формати операторів:
  - а) введення;
  - б) присвоєння;
  - в) виведення;
  - г) закінчення роботи;
  - д) коментар.
44. Як позначається розгалуження на блок-схемах?
45. Що розуміється під логічними виразами та логічними відношеннями у мові C++?
46. Формат операторів: **if** (повна та скорочена форми оператора); **goto**.
47. Для чого застосовується оператор **switch**? Формат запису оператора.
48. Наведіть приклади застосування операторів розгалуження.

49. Як позначається розгалуження на блок-схемах?
50. Чим відрізняються цикли з пост- і передумовою?
51. Дайте визначення поняття «циклічний обчислювальний процес».
52. Дайте визначення поняття «цикл».
53. Дайте визначення поняття «параметр циклу».
54. Який склад алгоритму циклічної структури?
55. Які різновиди циклічних алгоритмів?
56. Які дії необхідно виконати для організації арифметичного циклу?
57. Перерахуйте оператори керування, зазначте їхній формат і умови застосування.
58. Приведіть приклади використання операторів керування.
59. Дайте визначення поняття «ітераційний обчислювальний процес».
60. Які умови збіжності методу ітерацій?
61. Як організований вихід із циклу в ітераційному алгоритмі?
62. Які дії необхідно виконати для організації ітераційного циклу?
63. Які етапи розробки алгоритмів Ви знаєте?
64. Яку роль відіграє процес тестування алгоритму?
65. Який зв'язок між інформацією та кодом?

## **9. Критерії оцінювання навчальних досягнень здобувачів освіти**

Оцінювання знань за 4-бальною системою:

**Оцінка "5" (відмінно, високий рівень)** ставиться тоді, коли студент: дає повні відповіді на запитання щодо ефективних алгоритмів, сучасних методів і мови програмування C++; дає правильні визначення понять щодо розроблення, тестування, впровадження, експлуатації програмного забезпечення; при вирішенні завдань використовує сучасні методи та технології для розв'язання прикладних задач комп'ютерної інженерії; використовує обчислювальну техніку для здійснення пошуку інформації з різних джерел для розв'язання задач комп'ютерної інженерії; демонструє глибоке розуміння матеріалу; точно формулює свої думки і обґрунтовує їх; послідовно, зв'язно викладає матеріал, логічно міркує; виявляє вміння ілюструвати теоретичні знання, наводить приклади, аналізує; виявляє глибокі систематизовані знання теоретичного та практичного матеріалу на рівні творчого використання; самостійно виконує 95-85% від загальної кількості тестів поточного контролю.

**Оцінка «4» (добре, достатній рівень)** ставиться тоді, коли відповідь в основному задовольняє ті самі вимоги, що й на відповідь на оцінку «5», але робить незначні помилки при відповіді на теоретичні питання, допускає незначні неточності і виконує завдання на рівні аналогічного відтворення, самостійно виконує **75-85%** від загальної кількості тестів поточного контролю.

**Оцінка «3» (задовільно, середній рівень)** ставиться тоді, коли студент знає і розуміє основні положення теми, але: не досить глибоко володіє матеріалом, допускає помилки при визначенні понять щодо розроблення, тестування, впровадження, експлуатації програмного забезпечення; у відповіді немає

послідовності, чіткості; при виконанні завдання не може ефективно використовувати наявний наочний матеріал; виявляє основні знання в обсязі, необхідному для подальшого навчання і роботи на рівні репродуктивного відтворення; самостійно виконує **55-65%** від загальної кількості тестів поточного контролю.

**Оцінка «2» (незадовільно, початковий рівень) ставиться** тоді, коли студент не орієнтується в матеріалі, допускає істотні помилки, що спотворюють зміст вивченого розділу, виявляє серйозні пробіли в знаннях основного матеріалу, допускає грубі помилки на рівні нижче репродуктивного відтворення, самостійно виконує **20-45%** від загальної кількості тестів поточного контролю.

## 10. Рекомендована література

### Основна

1. Програмування : конспект лекцій для студентів денної та заочної форм навчання фахового передвищого рівня освіти галузі знань 12 Інформаційні технології спеціальності 123 Комп'ютерна інженерія. / упор.: В. Б. Дроменко. – К. : Таврійський національний університет ім. В.І. Вернадського, 2023. – 124 с.
2. C++. Основи програмування. Теорія та практика : підручник / [О.Г. Трофименко, Ю.В. Прокоп, І.Г. Швайко, Л.М. Буката та ін.] ; за ред. О.Г. Трофименко. – Одеса: Фенікс, 2020. – 544 с.
3. Грицюк Ю.І., Рак Т.Є. Програмування мовою C++ : навчальний посібник. – Львів : Вид-во Львівського ДУ БЖД, 2021. – 292 с. – Статистика: іл. 10, табл. 18, бібліогр. 31. ISBN 978-966-3466-85-9
4. Lippman, Stanley B.; Lajoie, Josée; Moo, Barbara E. (2021). C++ Primer (Fifth ed.). Addison-Wesley. ISBN 978-0-321-71411-4.
5. Stroustrup, Bjarne (2019). Programming: Principles and Practice Using C++ (Second ed.). Addison-Wesley. ISBN 978-0-321-99278-9.
6. Stroustrup, Bjarne (2018). The C++ Programming Language (Fourth ed.). Addison-Wesley. ISBN 978-0-321-56384-2.
7. Josuttis, Nicolai M. (2021). The C++ Standard Library, A Tutorial and Reference (Second ed.). Addison-Wesley. ISBN 978-0-321-62321-8.
8. Steven Prata (2019). C++ Primer Plus. (6th ed.). Addison-Wesley. ISBN: 978-0-321-77640-2.
9. Scott Meyers (2020). Effective C++, Third Edition: 55 Specific Ways to Improve Your Programs and Designs, (3rd Edition).

### Додаткова

1. Васильєв О. Програмування C++ в прикладах і задачах / Васильєв О. - Київ : Лира, 2019. – 382 с.
2. Янг Майкл Дж. Visual C++ 6 [Текст] : полное руководство : пер. с англ. / М. Дж. Янг. - К. : BHV, 2018. - 1056 с.
3. Andrei Alexandrescu "Modern C++ Design: Generic Programming and Design Patterns Applied", 2018.

### Інформаційні ресурси

1. Середовище програмування **MICROSOFT VISUAL Studio 2019** для виконання практичних робіт.

## Додатки

(зразки можливих видів контролю отриманих знань)

### Зразок питань для самоконтролю

Тема 6. Інструкції керування для реалізації програм циклічної структури.

1. Чим відрізняються цикли з пост- і передумовою?
2. Які відмінності дій операторів **break** і **continue**?
3. Дайте визначення поняття «циклічний обчислювальний процес».
4. Дайте визначення поняття «цикл».
5. Дайте визначення поняття «параметр циклу».
6. Який склад алгоритму циклічної структури?
7. Які різновиди циклічних алгоритмів?
8. Які дії необхідно виконати для організації арифметичного циклу?
9. Перерахуйте оператори керування, зазначте їхній формат і умови застосування.
10. Наведіть приклади використання операторів керування.
11. Дайте визначення поняття ітераційний обчислювальний процес.
12. Які умови збіжності методу ітерацій?
13. Як організований вихід із циклу в ітераційному алгоритмі?
14. Які дії необхідно виконати для організації ітераційного циклу?

### Зразок тестових завдань

Прості типи даних в C++. {

- = цілі – **int**, дійсні, – **float** або **double**, символічні, – **char**
- ~ цілі – **int**, дійсні, – **float** або **real**, символічні, – **char**
- ~ цілі – **bool**, дійсні, – **float** або **double**, символічні, – **string**
- ~ цілі – **int**, дійсні, – **float** або **double**, символічні, – **string**}

Якому зарезервованому слову програма передає управління у випадку, якщо значення змінної або виразу оператора **switch** не співпадає ні з одним константним виразом? {

- = default
- ~ all
- ~ contingency
- ~ other}

Який з наступних записів – правильний коментар в C++? {

- = /\* коментар \*/
- ~ { коментар }
- ~ /\* Коментарі \*/
- ~ \*\* Коментар \*\*}

До яких пір виконуватимуться оператори в тілі циклу **while (X < 100)**? {

- = Поки X строго менше ста
- ~ Поки X менше або рівний ста
- ~ Поки X рівний ста
- ~ Поки X більше ста}

Який з перерахованих типів даних НЕ є типом даних в C++? {

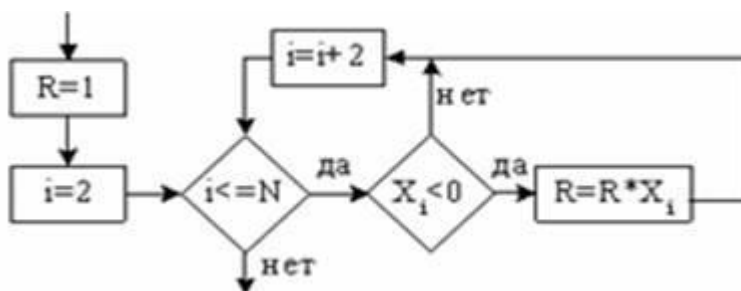
- = real

- ~ float
- ~ double
- ~ int}

Цикл з передумовою? {  
 = while  
 ~ do while  
 ~ for}

Цикл з постумовою ? {  
 = do while  
 ~ for  
 ~ while}

Заданий одновимірний масив  $X_1, X_2, X_3, \dots, X_n$ .



Фрагмент алгоритму визначає {  
 = Добуток від'ємних елементів з парними номерами  
 ~ Добуток додатних елементів з парними номерами  
 ~ Добуток від'ємних елементів  
 ~ Кількість додатних елементів з парними номерами}

У якому з варіантів відповідей оголошений двовимірний масив? {  
 = int anarray[20][20];  
 ~ array anarray[20][20];  
 ~ char array[20];  
 ~ int array[20, 20];}

### Приклад різнорівневого завдання.

*Початковий рівень – по 0,5 бала*

Що буде виведене, після виконання цієї коди: `cout << 5 << 3; ?` {  
 = 53  
 ~ 35  
 ~ 40}

Якщо умова оператора вибору хибна, то: {  
 = виконується оператор, наступний відразу після оператора if  
 ~ програма завершує роботу  
 ~ виконується тіло оператора вибору}

Алгоритм, в якому дії виконуються послідовно у природному і єдиному порядку дотримання, називається ... {  
 = лінійною  
 ~ підпрограмою  
 ~ циклічною

~ що розгалужується}

Оголошена змінна char a; Який з наступних виразів НЕ ВІРНИЙ?

= a = 3;

~ a = "3";

~ a = '3';

**Середній рівень – по 0,5 балів.**

Оператор виведення **cout** може друкувати декілька значень або змінних в одній команді, використовуючи наступний синтаксис: {

= cout << "Привіт" << name << "n";

~ cout << "Привіт", name, "n";

~ cout << "Привіт" + name + "n";

~ cout << ("Привіт" & name & "n");}

Алгоритм, в якому використовується структура, що містять дію вибору напрямку обчислювального процесу, називається ... {

= розгалуженим

~ підпрограмою

~ циклічним

~ лінійним}

Що з'явиться на екрані, після виконання цього фрагмента коду?

```
int a = 1, b = 2;
```

```
if (a == b);
```

```
cout << a << " = " << b << endl;
```

= 1 = 2

~ виведення на екран не виконається

~ a = b

~ синтаксична помилка

**Достатній рівень – 1 бал**

Який результат роботи наступного фрагмента коду?

```
1 int x = 0;
```

```
2
```

```
3 switch(x)
```

```
4
```

```
5
```

```
6 case 1: cout << "Один";
```

```
7
```

```
8 case 0: cout << "Нуль";
```

```
9
```

```
10 case 2: cout << "Привіт мир";
```

```
11
```

```
12
```

= НульПривіт мир

~ Нуль

~ Привіт мир

~ Один

Вкажіть правильну форму запису циклу **do while**

// форма запису оператора циклу do while:

```
do // початок циклу do while
{
/*блок операторов*/;
}
while {/*умова виконання цикла*/} // кінець циклу do while
```

```
// форма запису оператора циклу do while:
do // початок циклу do while
{
/*блок операторов*/;
}
while (/*умова виходу з цикла*/); // кінець циклу do while
```

```
// форма запису оператора циклу do while:
do while (/*умова виконання цикла*/) // початок циклу do while
{
/*блок операторов*/;
} // кінець циклу do while
```

Представлений алгоритм є



- = Циклічним
- ~ Лінійним
- ~ Розгалуженим

**Високий рівень – по 1 бал**

Властивість алгоритму «результативність» означає: {  
 = при точного виконання всіх команд алгоритму процес повинен припинитися за кінцеве число кроків, привівши до певного результату;  
 ~ алгоритм повинен бути розбитий на послідовність окремих кроків;  
 ~ записуючи алгоритм для конкретного виконавця, можна використовувати лише ті команди, що входять в систему його команд;  
 ~ алгоритм повинен забезпечувати вирішення деякого класу завдань даного типу для різних значень даних;  
 ~ виконавець алгоритму не повинен ухвалювати рішення, не передбачені укладачем алгоритму;}

Яке значення буде надруковано, в результаті виконання програми?

```
#include <iostream>
int main()
{
int x = 3;

switch(x)
```

```
case 0:
int x = 1;
std::cout << x << std::endl;
break;
case 3:
std::cout << x << std::endl;
break;
default:
x = 2;
std::cout << x << std::endl;
return 0;
}
```

= 3

~ 0

~ 2

~ 1

~ нічого не друкуватиметься, програма взагалі не працюватиме

У приведеному коді змініть або додайте один символ щоб код надрукував 20 зірочок – \*.

```
int i, N = 20;
for(i = 0; i < N; N--)
cout << "*";
```

```
int i, N = 20;
for(i = 19; i < N; i--)
cout << "*";
```

```
int i, N = 20;
for(i = 0; i < N; i--)
cout << "*";
```

```
int i, N = 40;
for(i = 0; i < N; i--)
cout << "*";
```

```
int i, N = 20;
for(i = 20; i < N; i--)
cout << "*";
```