

**ЗАТВЕРДЖУЮ**
 Декан факультету інформаційних технологій  
 проф. Тетяна ГОВОРУЩЕНКО

«05» вересня 2024 р.

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ****Алгоритмізація та програмування**

Назва дисципліни

**Галузь знань** – 12 Інформаційні технології**Спеціальність** – 122 Комп'ютерні науки**Рівень вищої освіти** – Перший бакалаврський**Освітньо-професійна програма** – Комп'ютерні науки**Обсяг дисципліни** – 11 кредитів ЄКТС**Шифр дисципліни** – ОЗП.05**Мова навчання** – українська**Статус дисципліни** – обов'язкова (цикл загальної підготовки)**Факультет** – Інформаційних технологій**Кафедра** – Комп'ютерних наук

Курс	Семестр	Обсяг дисципліни		Кількість годин						Форма семестрового контролю			
				Аудиторні заняття				Індивідуальна робота студента	Самостійна робота, у т.ч. ІРС	Курсовий проєкт	Курсова робота	Залік	Іспит
		Разом	Лекції	Лабораторні роботи	Практичні заняття								
1	1	5	150	68	34	34			82				+
1	2	4	120	54	18	36			66			+	
1	2	2	60						60	+			
<b>Разом</b>		<b>11</b>	<b>330</b>	<b>122</b>	<b>52</b>	<b>70</b>			<b>208</b>	<b>1</b>		<b>1</b>	<b>1</b>

Робоча програма складена на основі освітньо-професійної програми «Комп'ютерні науки» та стандарту вищої освіти зі спеціальності 122 – Комп'ютерні науки.

Програма складена

\_\_\_\_\_ проф., д.т.н. Олександр БАРМАК

\_\_\_\_\_ доц., к.т.н. Олександр МАЗУРЕЦЬ

\_\_\_\_\_ асистент каф. КН Олена СОБКО

Схвалена на засіданні кафедри *комп'ютерних наук*Протокол від 30 серпня 2024 р. № 1. Зав. кафедри \_\_\_\_\_ проф. Олександр БАРМАК

Робоча програма розглянута та схвалена вченою радою факультету інформаційних технологій

Голова вченої ради факультету

\_\_\_\_\_ проф. Тетяна ГОВОРУЩЕНКО

# АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ

<b>Тип (статус) дисципліни</b>	Обов'язкова загальної підготовки
<b>Рівень вищої освіти</b>	бакалаврський
<b>Мова викладання</b>	українська
<b>Семестр</b>	1-й, 2-й
<b>Кількість призначених кредитів ЄКТС</b>	11
<b>Форми здобуття освіти, для яких викладається дисципліна</b>	Очна денна

**Результати навчання.** Студент, який успішно завершив вивчення дисципліни, повинен: *знати* – основні обчислювальні та пошукові алгоритми, програмні засоби та способи відображення даних, основні інструменти програмування та їх застосування; *уміти* – читати та пояснювати алгоритми; обговорювати алгоритмічну коректність; використовувати та адаптувати стандартні алгоритми; розробляти алгоритми; читати та пояснювати значення програмних об'єктів; створювати, використовувати та змінювати програми, читати та пояснювати результати операцій; реалізовувати та описувати операції; створювати програми для реалізації стандартних алгоритмів; виявляти та виправляти логічні та синтаксичні помилки в програмному коді; описувати процес трансляції програми; визначати зображення різних типів даних за допомогою різних форматів; успішно використовувати інструменти програмування під час розробки програмного забезпечення; ефективно застосовувати стандартні комп'ютерні застосування.

**Зміст навчальної дисципліни.** Вступ до фаху. Алгоритмізація та проектування алгоритмів. Вступ до програмування. Мови програмування та їх елементи. Мова програмування C#. Константи, змінні. Типи даних. Оператори. Вирази. Методи. Алгоритмічна структура розгалуження мовою програмування. Алгоритмічна структура повторення мовою програмування. Парадигми програмування. Масиви. Одновимірні масиви. Багатовимірні масиви. Символьний та рядковий типи даних. Стандартні класи для роботи з масивами та рядками. Користувачькі інтерфейси .NET Framework. Windows Forms. Обробка виняткових ситуацій. Робота з файловими структурами. Робота з спеціалізованими файлами. Сучасні та перспективні задачі у галузі комп'ютерних наук. Курсове проектування з дисципліни. Робота з інформаційними джерелами при вирішенні практичних задач в галузі комп'ютерних наук. Робота з інформаційними джерелами для вибору засобів розробки програмного забезпечення. Етапи розроблення та впровадження програмного забезпечення. Основи тестування програмного забезпечення. Використання структур даних при розробці програмного забезпечення. Основи багатопоточності. Основи Windows Presentation Foundation. Засоби Windows Presentation Foundation. Сучасні та перспективні засоби та середовища програмування.

**Пререквізити** – вихідна.

**Кореквізити** – Об'єктно-орієнтоване проектування, Теорія алгоритмів, Основи програмної інженерії та тестування програмного забезпечення, Інтелектуальний аналіз даних, Методи та системи штучного інтелекту, Системне та мережеве програмне забезпечення.

**Запланована навчальна діяльність:** лекції – 52 год., лабораторні заняття – 70 год., самостійна робота – 208 год., курсове проектування 60 год, разом – 330 год.

**Методи навчання:** лекції (з використанням наочних методів (слайдів), пояснення); лабораторні заняття (з використанням тренінгів, майстер-класів), курсове проектування, самостійна робота.

**Форми оцінювання результатів навчання:** захист лабораторної роботи, поточна контрольна робота, підсумкова контрольна робота, курсовий проект.

**Вид семестрового контролю** іспит (1 семестр), залік (2 семестр), КП (2 семестр).

## Навчальні ресурси:

1. Собко О. В., Молчанова М. О., Бармак О. В., Мазурець О. В. Алгоритмізація та програмування: методичні вказівки до лабораторних робіт для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 122 «Комп'ютерні науки». Хмельницький, ХНУ. 2022. 108с.
2. Собко О. В., Молчанова М. О., Мазурець О. В., Бармак О. В. Алгоритмізація та програмування : методичні рекомендації до курсового проектування для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 122 «Комп'ютерні науки». Хмельницький : ХНУ, 2024. 36 с.
3. Коноваленко І. В. Платформа .NET та мова програмування C# 8.0: навчальний посібник. Тернопіль, 2020. 320 с. URL: <https://elartu.tntu.edu.ua/bitstream/lib/32825/1/Konovalenko%20I.%20NET-C%23.pdf>
4. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування: Конспект лекцій. Київ, КПІ ім. І.Сікорського. 2020. 117 с. URL: [https://ela.kpi.ua/bitstream/123456789/39004/1/OTaPr\\_KonspLek\\_1.pdf](https://ela.kpi.ua/bitstream/123456789/39004/1/OTaPr_KonspLek_1.pdf)
5. Модульне середовище. URL: <https://msn.khnu.km.ua/>
6. Електронна бібліотека університету. URL: [http://lib.khmnu.edu.ua/asp/php\\_fplage\\_lib.php](http://lib.khmnu.edu.ua/asp/php_fplage_lib.php)
7. Репозитарій ХНУ. URL: <https://library.khmnu.edu.ua/#>

**Викладачі:** доктор технічних наук, професор Олександр БАРМАК; кандидат технічних наук, доцент кафедри комп'ютерних наук Олександр МАЗУРЕЦЬ; асистент кафедри комп'ютерних наук Олена СОБКО.

## Пояснювальна записка

Навчальна дисципліна «Алгоритмізація та програмування» є обов'язковою освітньою компонентою загальної підготовки здобувачів першого (бакалаврського) рівня вищої освіти, які навчаються за освітньо-професійними програмами в межах спеціальності 122 «Комп'ютерні науки». Двосеместрова дисципліна обсягом 11 кредитів ЄКТС, складається із теоретичного курсу, лабораторних робіт, індивідуальних завдань та курсового проєкту. Курсовий проєкт обсягом 2 кредити ЄКТС.

**Пререквізити** – вихідна.

**Кореквізити** – Об'єктно-орієнтоване проєктування, Теорія алгоритмів, Основи програмної інженерії та тестування програмного забезпечення, Інтелектуальний аналіз даних, Методи та системи штучного інтелекту, Системне та мережеве програмне забезпечення.

Відповідно до Стандарту вищої освіти із зазначеної спеціальності та освітньої програми дисципліна має забезпечити:

– **компетентності:**

**ЗК03.** Знання та розуміння предметної області та розуміння професійної діяльності.

**ЗК06.** Здатність вчитися й оволодівати сучасними знаннями.

**ЗК07.** Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

**ЗК08.** Здатність генерувати нові ідеї (креативність).

**ЗК10.** Здатність бути критичним і самокритичним.

**ФК 3.** Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проєктування, 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.

**ФК 7.** Здатність застосовувати теоретичні та практичні основи методології та технології моделювання для дослідження характеристик і поведінки складних об'єктів і систем, проводити обчислювальні експерименти з обробкою й аналізом результатів.

– **компетентності (частково):**

**ЗК02.** Здатність застосовувати знання у практичних ситуаціях.

**ЗК11.** Здатність приймати обґрунтовані рішення.

**ЗК12.** Здатність оцінювати та забезпечувати якість виконуваних робіт.

**ЗК15.** Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя.

**ФК 1.** Здатність до математичного формулювання та досліджування неперервних та дискретних математичних моделей, обґрунтування вибору методів і підходів для розв'язування теоретичних і прикладних задач у галузі комп'ютерних наук, аналізу та інтерпретування.

**ФК 8.** Здатність проєктувати 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.

– **програмні результати навчання:**

**ПРН 5.** Проєктувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

– **програмні результати навчання (частково):**

**ПРН 1.** Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

**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 комп'ютерних наук.

**ПРН 17.** Професійно розвиватися, опрацьовувати україномовні та англomовні джерела предметної області, усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань у галузі комп'ютерних наук, адаптуватися до роботи за конкретною професією, пропагувати ведення активного та здорового способу життя як ефективної складової професійного розвитку.

**ПРН 18.** Взаємодіяти з колегами й працювати у складі команди, ставитись відповідально до роботи, асоціювати себе як члена громадянського суспільства та наукової спільноти, здійснювати україномовну та англomовну комунікацію з професійних питань у галузі комп'ютерних наук.

**Мета дисципліни** – формування знань, вмінь та навичок, необхідних для розробки алгоритмів та програм для задач оптимізації, розробки алгоритмів та програм аналізу даних, розробки прикладного програмного забезпечення інформаційних систем і технологій, створення та налагодження програмного інтерфейсу, візуальних компонент програмного забезпечення інформаційних систем і технологій для вирішення задач у галузі комп'ютерних наук.

**Предмет дисципліни.** Теоретичні принципи та практичні прийоми структурного, процедурного, модульного програмування; технологія розробки алгоритмів вирішення прикладних задач, кодування обраною мовою програмування, налагодження програми.

**Завдання дисципліни.** надати студентам знання технології розробки алгоритмів вирішення прикладних задач у галузі комп'ютерних наук, мов програмування та їх елементів, алгоритмічних структур; та *навики* застосування мови програмування C#, типів даних, констант, операторів, виразів, алгоритмічних структур при розробці програмного забезпечення у галузі комп'ютерних наук, що характеризуються комплексністю та невизначеністю умов.

#### СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Тема	Кількість годин відведених на		
	Денна форма навчання		
	Лекції	Лабораторні роботи	СРС
<b>1-й семестр</b>			
Тема 1. Вступ до фаху. Алгоритмізація та проектування алгоритмів.	2	4	6
Тема 2. Вступ до програмування	2	4	6
Тема 3. Мови програмування та їх елементи. Мова програмування C#.	2	4	6
Тема 4. Константи, змінні. Типи даних. Оператори. Вирази. Методи.	2	4	6
Тема 5. Алгоритмічна структура розгалуження мовою програмування.	2	4	6
Тема 6. Алгоритмічна структура повторення мовою програмування.	2	4	6
Тема 7. Парадигми програмування.	2	4	6
Тема 8. Масиви. Одновимірні масиви.	2	4	6
Тема 9. Багатовимірні масиви.	2	4	6
Тема 10. Символьний та рядковий типи даних.	2	4	6
Тема 11. Стандартні класи для роботи з масивами та рядками.	2	4	6
Тема 12. Користувацькі інтерфейси .NET Framework.	2	4	6
Тема 13. Windows Forms.	2	4	6
Тема 14. Обробка виняткових ситуацій.	2	4	6

Тема 15. Робота з файловими структурами.	2	4	6
Тема 16. Робота з спеціалізованими файлами.	2	8	6
Тема 17. Сучасні та перспективні задачі у галузі комп'ютерних наук. Курсове проектування з дисципліни	2		12
<b>2-й семестр</b>			
Тема 18. Робота з інформаційними джерелами при вирішенні практичних задач в галузі комп'ютерних наук.	2	4	12
Тема 19. Робота з інформаційними джерелами для вибору засобів розробки програмного забезпечення.	2	4	12
Тема 20. Етапи розроблення та впровадження програмного забезпечення.	2	4	12
Тема 21. Основи тестування програмного забезпечення.	2	8	12
Тема 22. Використання структур даних при розробці програмного забезпечення.	2		12
Тема 23. Основи багатопоточності.	2	4	14
Тема 24. Основи Windows Presentation Foundation.	2	4	14
Тема 25. Засоби Windows Presentation Foundation.	2	8	18
Тема 26. Сучасні та перспективні засоби та середовища програмування.	2		20
<b>Разом</b>	<b>52</b>	<b>70</b>	<b>208</b>

### Програма навчальної дисципліни

#### Зміст лекційного курсу\*

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
	<i>1-ий семестр</i>	
1	<p><i>Лекція 1. Вступ до фаху. Алгоритмізація та проєктування алгоритмів.</i> Комп'ютерні науки – вступ до спеціальності. Чинники успішного професійного розвитку та діяльності. Історія розвитку предметної області комп'ютерних наук. Алгоритм як основне поняття програмування. Властивості алгоритмів. Типи алгоритмічних структур: лінійна структура, алгоритмічна структура розгалуження, алгоритмічна конструкція повторення. Ефективність алгоритмів. Способи подання алгоритмів. Блок-схеми. Засоби побудови блок-схем та текстові редактори. Літ.: [1] с. 10-19, [4] с. 1-9</p>	2
2	<p><i>Лекція 2. Вступ до програмування</i> Архітектура комп'ютера. Програмне забезпечення комп'ютера. Етапи розробки програмного забезпечення. Огляд парадигм програмування. Комп'ютерні платформи. Платформа .NET Framework. Огляд сучасних інтегрованих систем програмування. Середовище програмування Visual Studio. Транслятори, редактори, компонувальники та налагоджувачі. Процедурне програмування. Основи синтаксису С# (типи даних, константи, змінні, ініціалізація даних, введення й виведення даних). Літ.: [1] с. 4-9, [3] с. 7-27</p>	2
3	<p><i>Лекція 3. Мови програмування та їх елементи. Мова програмування С#.</i> Поняття про технології та мови програмування. Класифікація мов програмування. Елементи мови програмування. Алфавіт, синтаксис та семантика мови програмування С#. Літ.: [1] с. 20-25, [3] с. 28-39</p>	2
4	<p><i>Лекція 4. Константи, змінні. Типи даних. Оператори. Вирази. Методи.</i> Константи та змінні. Типи даних С#. Ініціалізація даних. Вирази (унарні, бінарні та присвоювання). Операції послідовного присвоювання, умовного та явного перетворення типів. Пріоритет та асоціативність операцій. Методи та стандартні бібліотеки. Організація введення-виведення. Клас Math як основа роботи з математичними виразами. Літ.: [1] с. 26-31, [3] с. 40-57, [4] с. 24-29</p>	2

5	<p><i>Лекція 5. Алгоритмічна структура розгалуження мовою програмування.</i> Умовні конструкції розгалуження «if» та «if-else». Оператор вибору «switch». Переривання виконання алгоритму. Вкладені умови. Літ.: [1] с. 32-39, [3] с. 58-63, [4] с. 48-52</p>	2
6	<p><i>Лекція 6. Алгоритмічна структура повторення мовою програмування.</i> Оператор циклу з умовою «for», поняття лічильника. Оператор «foreach». Оператори циклу «while», «do-while». Переривання виконання циклу. Вкладені цикли. Літ.: [1] с. 40-47, [3] с. 64-70, [4] с. 53-67</p>	2
7	<p><i>Лекція 7. Парадигми програмування.</i> Проектування та розробка програмного забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного. Методи (функції). Приклади програм із застосуванням різних парадигм програмування. Введення в об'єктно-орієнтоване програмування. Літ.: [1] с.80-84, [3] с. 71-141</p>	2
8	<p><i>Лекція 8. Масиви. Одновимірні масиви.</i> Одновимірні масиви. Способи заповнення масивів. Алгоритми пошуку в масивах (послідовностях). Літ.: [1] с.48-55, [3] с. 165-167, [4] с. 77-96</p>	2
9	<p><i>Лекція 9. Багатовимірні масиви.</i> Двовимірні масиви. Багатовимірні масиви. Ступінчаті масиви, масиви масивів. Алгоритми пошуку в масивах (послідовностях). Приклади програм, що використовують багатовимірні масиви. Літ.: [1] с. 56-62, [3] с. 168-177, [4] с. 105-116</p>	2
10	<p><i>Лекція 10. Символьний та рядковий типи даних.</i> Символьний тип даних, символічна константа. Рядки, константа-рядок. Введення-виведення рядків та масивів символів. Розташування у пам'яті. Обробка рядків: вставлення та видалення символів, аналіз слів, копіювання та поєднання рядків. Алгоритми пошуку підрядків у рядках. Алгоритми та демонстраційні програми роботи з текстами. Літ.: [1] с.63-68, [3] с. 42-44, [4] с. 39-47</p>	2
11	<p><i>Лекція 11. Стандартні класи для роботи з масивами та рядками.</i> Методи класу Array для роботи з масивами. Методи класу StringBuilder для роботи з масивами та рядками. Форматування рядків. Літ.: [1] с. 69-74, [3] с. 176-177, [4] с. 87-92</p>	2
12	<p><i>Лекція 12. Користувацькі інтерфейси .NET Framework.</i> Поняття користувацького інтерфейсу. Типи користувацького інтерфейсу. Загальні відомості про Windows Forms, Windows Presentation Foundation, Universal Windows Platform. Літ.: [3] с. 208-222</p>	2
13	<p><i>Лекція 13. Windows Forms.</i> Введення у Windows Forms, Структура програми Windows Forms. Спадкування класу Form. Робота з формами та елементами управління. Програмне створення елементів керування вікна. Елементи керування з панелі Toolbox Літ.: [1] с. 101-105, [3] с. 223-236</p>	2
14	<p><i>Лекція 14. Обробка виняткових ситуацій.</i> Відлагодження програмного забезпечення. Поняття виняткової ситуації. Обробка винятків у мові C#. Принципи обробки винятків. Послідовність обробки винятків. Блоки «try» та «catch». Оператор «throw». Створення власних винятків. Літ.: [3] с. 143-156</p>	2
15	<p><i>Лекція 15. Робота з файловими структурами.</i> Файл. Формати файлів. Створення файлів за допомогою середовища програмування. Алгоритми читання та запису інформації в текстовий файл. Літ.: [1] с.75-79, [3] с. 281-284</p>	2
16	<p><i>Лекція 16. Робота з спеціалізованими файлами.</i> Запис та обробка інформації на прикладі програм Microsoft Word, Excel та створення xml-документів Літ.: [5], [6], [7]</p>	2
17	<p><i>Лекція 17. Сучасні та перспективні задачі у галузі комп'ютерних наук. Курсове</i></p>	2

	<b>проектування з дисципліни</b> Практичні задачі у галузі комп'ютерних наук. Тематика КП. Етапи виконання КП. Структура пояснювальної записки КП. Підходи до дослідження предметної області. Приклади КП. Літ.: [2]	
	<b>Разом за семестр:</b>	<b>34</b>

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
	<b>2-ий семестр</b>	
1	<b>Лекція 18. Робота з інформаційними джерелами при вирішенні практичних задач в галузі комп'ютерних наук</b> Підходи до пошуку, оброблення та аналізу інформації з різних джерел для прийняття обґрунтованих рішень при розв'язанні задач у галузі комп'ютерних наук. Вимоги до формулювання постановки завдання. Оформлення результатів вишукувань у текстових редакторах. Приклади оформлення аналізу інформаційних джерел та постановки завдання. Літ.: [9] с.45-150, [10] с. 86-90	2
2	<b>Лекція 19. Робота з інформаційними джерелами для вибору засобів розробки програмного забезпечення</b> Особливості пошуку, оброблення та аналізу інформації з різних джерел для прийняття обґрунтованих рішень щодо вибору засобів розробки програмного забезпечення. Критерії вибору зручної парадигми програмування за поставленого завдання. Документування процесу та результатів проектування програмного забезпечення. Приклади документування результатів аналізу засобів розробки програмного забезпечення. Літ.: [10] с. 15-23	2
3	<b>Лекція 20. Етапи розроблення та впровадження програмного забезпечення</b> Основні етапи розроблення програмного забезпечення. Етапи проектування та розробки. Використання ескізів для проектування інтерфейсу програмного забезпечення. Розроблення алгоритму методом покрокового уточнення. Вимоги до програмного коду. Поняття рефакторингу коду. Інструменти для рефакторингу коду. Літ.: [10] с. 24-50, [11] с. 1-65	2
4	<b>Лекція 21. Основи тестування програмного забезпечення.</b> Проблема оцінювання якості створеного програмного забезпечення. Поняття тестування програмного забезпечення. Роль тестування в процесі розробки програмного забезпечення. Види тестування. Функціональне тестування. Визначення вимог до використання програмного забезпечення. Оцінювання відповідності програмного забезпечення вимогам та самокритичність. Приклади документування результатів функціонального тестування програмного забезпечення. Літ.: [12] с. 24-59, [13] с. 88-108	2
5	<b>Лекція 22. Використання структур даних при розробці програмного забезпечення.</b> Колекції. Список (клас List). Масиви (клас ArrayList). Стек (клас Stack). Черга (клас Queue). Хештаблиці (клас Hashtable). Літ.: [1] с. 91-95, [3] с. 192-207	2
6	<b>Лекція 23. Основи багатопоточності.</b> Однопоточність та багатопоточність. Багатозадачність з орієнтацією на процеси і потоки. Процес. Поток. Синхронізація. Клас Thread. Використання основного потоку. Стан потоку. Створення декількох потоків. Пріоритети потоків. Блокування. Клас Monitor. Приклади реалізації багатопоточності. Літ.: [1] с. 96-100, [3] с. 288-308	2
7	<b>Лекція 24. Основи Windows Presentation Foundation.</b> Огляд технології WPF. Структура WPF-застосунків. Мова розмітки XAML. Компоновка та побудова WPF-застосунків. Найпростіший WPF-застосунок. Літ.: [3] с. 237-246	2
8	<b>Лекція 25. Засоби Windows Presentation Foundation.</b> Клас Application. Базові елементи управління. Макет вікна. Панелі. Події. Стили і шаблони. Прив'язка даних. Шаблон MVVM	2

	(Model-View-ViewModel). Літ.: [3] с. 247-257	
9	<i>Лекція 26. Сучасні та перспективні засоби та середовища програмування.</i> Перспективні напрямки програмування. Засоби автоматизації програмування. Сучасні середовища програмування (навчальне, інтегроване, візуальне). Онлайн-компілятори. Приклади використання сучасних засобів програмування для вирішення прикладних задач. Літ.: [3] с. 7-12, [10] с. 9-14	2
<b>Разом за семестр:</b>		<b>18</b>

### Зміст лабораторних занять

#### Перелік лабораторних занять для студентів денної форми навчання

№ з/п	Тема лабораторного заняття	Кількість годин
<b>1-й семестр</b>		
1	<i>Лабораторна робота № 1. Проектування алгоритмів вирішення задач.</i> Ознайомлення з програмними засобами побудови блок-схем та текстовими редакторами. Розробка, проектування алгоритмів вирішення задач відповідно до завдання. Подання алгоритмів у вигляді опису, псевдокоду, блок-схеми. Оформлення звіту з виконання завдання. Літ.: [1] с. 10-19, [1], с. 20-25, [3] с. 28-39, [4] с. 1-9	4
2	<i>Лабораторна робота № 2. Створення простої програми на мові C#.</i> Ознайомлення з середовищем програмування Visual Studio та мовою програмування C#. Створення програми на мові C#, компіляція програмного коду. Оформлення звіту з виконання поставленого завдання. Літ.: [1] с. 4-9, [3] с. 7-27	4
3	<i>Лабораторна робота № 3. Робота з типами даних, виразами, операціями та математичними функціями на мові C#.</i> Використання виразів і операцій при написанні програмного коду. Робота з типами даних C#. Використання методів бібліотек стандартних математичних функцій. Створення програм з використанням стандартних математичних функцій відповідно до завдання. Оформлення звіту з виконання поставленого завдання. Літ.: [1] с. 26-31, [3] с. 40-57	4
4	<i>Лабораторна робота № 4. Робота з операторами розгалуження, вибору та циклу в програмних реалізаціях на мові C#</i> Проектування алгоритмів із використанням розгалуження й вибору, циклів. Використання операторів розгалуження й вибору, циклів при написанні програмного коду. Створення програм відповідно до завдання. Оформлення звіту з виконання поставленого завдання. Літ.: [1] с. 32-47, [3] с. 60-71, [4] с. 48-67	4
5	<i>Лабораторна робота № 5. Робота з одновимірними масивами.</i> Програмна реалізація одновимірних масивів, алгоритми пошуку мінімальних та максимальних значень, сортування в одновимірних масивах. Оформлення звіту з виконання поставленого завдання. Літ.: [1] с.48-55, [3] с. 65-168, [4] с. 77-96	4
6	<i>Лабораторна робота № 6. Робота з багатовимірними масивами.</i> Програмна реалізація двовимірних масивів, алгоритми пошуку мінімальних та максимальних значень, сортування в двовимірних масивах. Оформлення звіту з виконання поставленого завдання. Літ.: [1] с. 56-62, [3] с. 168-177, [4] с. 105-116	4
7	<i>Лабораторна робота № 7. Робота з символьним та рядковим типом даних</i> Програмна реалізація алгоритмів пошуку та сортування символьних величин і	4



	рядків. Програмна реалізація методів сортування та пошуку в масивах та рядках за допомогою вбудованих класів Array та StringBuilder. Оформлення звіту з виконання поставленого завдання. Літ.: [1] с.63-64, [3] с. 42-44, [3] с. 176-178	
8	<i>Лабораторна робота № 8. Розробка програм з використанням Windows Forms.</i> Створення Windows Forms додатка відповідно до завдання. Зчитування інформації з текстових полів, програмування кнопок, робота з компонентом Chart. Оформлення звіту з виконання поставленого завдання. Літ.: [1] с. 101-105, [3] с. 208-236	6
<b>Разом за семестр:</b>		<b>34</b>

№ з/п	Тема лабораторного заняття	Кількість годин
<i>2-й семестр</i>		
1	<i>Лабораторна робота № 9. Робота з текстовими та спеціалізованими файлами</i> Використання Visual Studio C# для читання та запису інформації в текстовий файл. Програмне створення спеціалізованих файлів та наповнення їх інформацією. Робота з компонентом dataGridView у Windows Forms. Оформлення звіту з виконання поставленого завдання. Літ.: [1] с.75-79, [3] с. 253-259, [5], [6], [7]	4
2	<i>Лабораторна робота № 10. Пошук та аналіз інформаційних джерел для вирішення практичної задачі в галузі комп'ютерних наук</i> Проведення дослідження предметної області згідно отриманої від викладача теми. Пошук, оброблення та аналіз інформації з різних джерел для прийняття обґрунтованих рішень щодо розв'язання задачі у галузі комп'ютерних наук. Оформлення результатів вишукувань і постановка завдання. Літ.: [8] с.45-150, [9] с. 86-90	4
3	<i>Лабораторна робота № 11. Проектування програмного продукту для вирішення практичної задачі.</i> Розробка, проектування алгоритмів для вирішення прикладної задач відповідно до поставленого завдання. Вибір парадигми програмування та засобів розробки програмного продукту. Оформлення результатів проектування програмного продукту. Літ.: [2] с. 20-28, [9] с. 15-23	4
4	<i>Лабораторна робота №12. Створення програмного продукту для вирішення практичної задачі згідно проекту</i> Проектування інтерфейсу програмного продукту згідно проекту. Створення компонентів інтерфейсу. Програмування подій для взаємодії користувача з компонентами інтерфейсу. Літ.: [9] с. 24-50, [10] с. 1-65	4
5	<i>Лабораторна робота № 13. Документування результатів розроблення програмного продукту</i> Візуальне подання програмної архітектури. Опис програмних складових. Функціональне тестування програмного продукту та визначення вимог до використання. Підсумок результатів розроблення програмного продукту. Літ.: [2] с. 11-21, [11] с. 24-59, [12] с. 88-108	4
6	<i>Лабораторна робота № 14. Робота зі структурами даних.</i> Програмна реалізація простого списку List, стеку та хеш-таблиці відповідно до завдання. Оформлення звіту з виконання поставленого завдання. Літ.: [1] с. 91-95, [3] с. 192-207	4
7	<i>Лабораторна робота № 15. Робота з потоками.</i> Програмна реалізація читання та запису текстової інформації в файл за	4

	допомогою багатопоточності відповідно до завдання. Оформлення звіту з виконання поставленого завдання. Літ.: [1] с. 96-100, [3] с. 266-287	
8	<i>Лабораторна робота № 16. Створення WPF-застосунка.</i> Створення WPF-застосунка. Створення програми відповідно до завдання. Оформлення звіту з виконання поставленого завдання. Літ.: [3] с. 237-257	8
<b>Разом за семестр:</b>		<b>36</b>

### **Зарахування результатів неформальної освіти**

Окремі результати вивчення курсу можуть бути зараховані у випадку отримання студентом результатів навчання у неформальній освіті, що підтверджені відповідним документом (сертифікат, свідоцтво, освітня програма тощо):

- Як результат виконання ЛР №1, 3, 4, зараховується онлайн курс <https://www.coursera.org/learn/c-sharp-for-dot-net?action=enroll#testimonials>
- Як результат виконання ЛР №1, 3, 4, 7, зараховується онлайн курс <https://www.freecodecamp.org/learn/foundational-c-sharp-with-microsoft/foundational-c-sharp-with-microsoft-certification-exam/foundational-c-sharp-with-microsoft-certification-exam>

### **Зміст самостійної (у т.ч. індивідуальної) роботи**

Самостійна робота студентів денної форми навчання полягає у систематичному опрацюванні теоретичного та методичного матеріалу, підготовці до виконання і захисту лабораторних робіт, виконанні індивідуальних завдань, курсовому проектуванні тощо.

Самостійна робота студентів денної форми навчання полягає у систематичному опрацюванні програмного матеріалу, підготовці до виконання і захисту лабораторних робіт, поточних контрольних робіт з теоретичного матеріалу, курсовому проектуванні тощо. Керівництво самостійною роботою та контроль за виконанням індивідуального завдання здійснює викладач згідно з розкладом консультацій у позаурочний час.

### **Зміст самостійної роботи студентів денної форми навчання**

№ тижня	Зміст самостійної роботи <i>1-й семестр</i>	Год
1-2	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи.	6
3-4	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи.	10
5-6	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи.	10
7-8	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи. Підготовка до проміжного контрольного заходу.	10
9-10	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи.	10
11-12	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи.	10
13-14	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи. Підготовка звіту з самостійної роботи.	10
15-17	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи. Підготовка до проміжного контрольного заходу. Підготовка до підсумкового контрольного заходу.	16
<b>Разом годин:</b>		<b>82</b>

<b>№ тижня</b>	<b>Зміст самостійної роботи 2-й семестр</b>	<b>Год</b>
1-2	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи.	12
3-4	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи.	12
5-6	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи. Виконання курсового проєкту.	12
7-8	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи. Виконання курсового проєкту. Підготовка до проміжного контрольного заходу.	12
9-10	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи. Виконання курсового проєкту.	12
11-12	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи. Виконання курсового проєкту.	14
13-14	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи. Виконання курсового проєкту. Підготовка звіту з самостійної роботи.	14
15-16	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи. Підготовка до захисту курсового проєкту.	18
17-18	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи. Підготовка до захисту лабораторної роботи. Захист курсового проєкту. Підготовка до проміжного контрольного заходу.	20
<b>Разом годин:</b>		<b>126</b>

### **Технології та методи навчання**

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних технологій, зокрема: лекції (з використанням методів візуалізації); лабораторні заняття (з використанням тренінгів, майстер-класів), самостійна робота (індивідуальні завдання).

Необхідні інструменти, обладнання, програмне забезпечення: комп'ютер (надається для використання в лабораторіях кафедри КН), Visual Studio (ліцензія ХНУ), Dia, Flying Logic, Edraw MAX, Algorithm Flowcharts Editor (виробниками дозволяється безкоштовне некомерційне використання), текстовий редактор (пропонується використання безкоштовних онлайн-сервісів).

### **Методи контролю**

Поточний контроль здійснюється під час лекційних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочою програмою і графіком навчального процесу. При цьому використовуються такі методи поточного контролю:

- усне опитування перед допуском до лабораторного заняття;
- захист лабораторних робіт;
- проміжний контроль теоретичного матеріалу з теми.

При виведенні підсумкової семестрової оцінки враховуються результати як поточного контролю так і захисту лабораторних робіт та поточних контрольних робіт.

### **Оцінювання результатів навчання студентів у семестрі**

Оцінювання академічних досягнень здобувача вищої освіти здійснюється відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ». Кожний вид роботи з дисципліни оцінюється за інституційною чотирибальною шкалою і виставляється в електронному журналі обліку успішності.

Засвоєння студентом теоретичного матеріалу з дисципліни оцінюється під час захисту лабораторних робіт, на поточних контрольних роботах та підсумковому контрольному заході.

Поточний контроль передбачає захист лабораторних робіт. Поточний контроль здійснюється під час лабораторних занять та поточних контрольних робіт, а також у дні проведення контрольних заходів, встановлених робочою програмою і графіком навчального процесу.

Політика проведення лабораторних робіт з навчальної дисципліни «Об'єктно-орієнтоване проектування»:

1. Студент має бути присутнім на кожному занятті.

2. Студент може бути відсутнім на занятті, якщо здав відповідну лабораторну роботу або з поважної причини: хвороба, тощо.

3. В разі потреби при виконанні лабораторної роботи студент може скористатися засобами зв'язку для пошуку інформації в інтернеті, тощо.

4. На одному занятті захищається тільки одна лабораторна робота. В разі бажання студента здати ще одну роботу, студент має стати в кінець черги. За одне заняття можна здати не більше 2 робіт.

5. Перескладання лабораторної роботи можливе на наступному занятті.

6. Можливе перезарахування деяких лабораторних робіт елементами неформальної освіти згідно Робочої програми дисципліни.

При виконанні лабораторних робіт студент має дотримуватись положень академічної доброчесності; використання напрацювань інших студентів (компоненти програмного забезпечення та програмні коди, графічні матеріали лабораторних робіт та фрагменти звітів з лабораторних робіт) є підставою для зниження оцінки чи перескладання лабораторної роботи.

Студент до виконання лабораторної роботи має бути підготовлений, зокрема: ознайомлений із темою, порядком виконання та відповідними теоретичними відомостями. Для встановлення рівня готовності студента до виконання роботи викладач здійснює опитування.

Для захисту виконаної лабораторної роботи студенту необхідно підготувати звіт. У залежності від завдання, в ході виконання лабораторної роботи виконується прикладна розробка програмного забезпечення мовою C# або проектування програмного забезпечення засобами візуального проектування діаграм. Відповідно одержані результати в звіті мають містити програмні коди та світлини екрана інтерфейсу з поясненнями або спроектовані діаграми з їх детальним описом та обґрунтуванням.

Пропущене лабораторне заняття студент зобов'язаний відпрацювати не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

*Семестрова підсумкова оцінка* визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих студентом позитивно, з урахуванням коефіцієнта вагомості і розраховується в автоматизованому режимі за відповідною програмою. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих видів її робіт. При виведенні підсумкової семестрової оцінки враховуються результати як поточного контролю, так і підсумкового контрольного заходу, який проводиться письмово з усього матеріалу дисципліни. Студент, який набрав позитивний середньозважений бал за поточну роботу і не здав підсумковий контрольний захід, вважається невстигаючим.

Оцінювання знань студентів здійснюється за такими критеріями:

Оцінка за інституційною шкалою	Узагальнений критерій
Відмінно	Студент глибоко і у повному обсязі опанував зміст навчального матеріалу, легко в ньому орієнтується і вміло використовує понятійний апарат; уміє пов'язувати теорію з практикою, вирішувати практичні завдання, впевнено висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає, логічний виклад відповіді державною мовою (в усній або у письмовій формі), демонструє якісне оформлення роботи і володіння спеціальними інструментами. Студент не вагається при видозміні запитання, вміє робити детальні та узагальнюючі висновки. При відповіді допустив дві–три несуттєві <i>похибки</i> .

Добре	Студент виявив повне засвоєння навчального матеріалу, володіє понятійним апаратом і фаховою термінологією, орієнтується у вивченому матеріалі; свідомо використовує теоретичні знання для вирішення практичних завдань; виклад відповіді грамотний, але у змісті і формі відповіді можуть мати місце окремі неточності, нечіткі формулювання закономірностей тощо. Відповідь студента будується на основі самостійного мислення. Студент у відповіді допустив дві–три <b>несуттєві помилки</b> .
Задовільно	Студент виявив знання основного програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь студента будується на рівні репродуктивного мислення, студент має слабкі знання структури курсу, допускає неточності і <b>суттєві помилки</b> у відповіді, вагається при відповіді на видозмінене запитання. Разом з тим, набув навичок, необхідних для виконання нескладних практичних завдань, які відповідають мінімальним критеріям оцінювання і володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.
Незадовільно	Студент виявив розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати теоретичні знання при вирішенні практичних завдань. Як правило, оцінка "незадовільно" виставляється студенту, який не може продовжити навчання без додаткової роботи з вивчення дисципліни.

**Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми навчання у 1 семестрі (іспит) за ваговими коефіцієнтами:**

Аудиторна робота, самостійна, індивідуальна робота		Семестровий контроль, залік
<i>Захист лабораторних робіт</i> № 1-8	<i>Поточні контрольні роботи</i> № 1-2	<i>Підсумковий контрольний захід</i>
ВК: 0,4*	ВК: 0,2	ВК: 0,4

\*Умовні позначення: ВК – ваговий коефіцієнт

**Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми навчання у 2 семестрі (залік) за ваговими коефіцієнтами:**

Аудиторна робота, самостійна, індивідуальна робота	
<i>Захист лабораторних робіт</i> № 9-16	<i>Поточні контрольні роботи</i> № 3-4
ВК: 0,8	ВК: 0,2

**Курсове проєктування**

Індивідуальна робота студентів у другому семестрі відповідно до навчального плану передбачає виконання курсового проєкту.

Метою курсового проєктування з дисципліни є закріплення, поглиблення та узагальнення здобувачами одержаних під час навчання знань і практичних навичок з розроблення алгоритмів та використання мов програмування, та їх застосування до комплексного вирішення конкретних спеціалізованих завдань та практичних проблем у галузі комп'ютерних наук і в процесі навчання.

На початку семестру керівник курсового проєкту проводить загальну консультацію, на якій висвітлюються мета та завдання курсового проєктування, етапи і терміни його виконання, календарний план, напрямки тематики, форми та критерії оцінювання. Розглядаються загальні

вимоги щодо написання й оформлення курсового проєкту, вимоги до розробки програмного застосунку, розклад індивідуальних консультацій, список рекомендованої навчальної й довідкової літератури для виконання завдань курсового проєкту. Оголошується перелік запропонованих тем курсових проєктів.

Після чого, згідно з розкладом індивідуальних консультацій, із здобувачами проводиться обговорення, коригування, уточнення або заміна тем курсових проєктів, формується титульний аркуш. Завдання на курсовий проєкт відповідним чином коригується викладачем, й не пізніше четвертого тижня семестру видається здобувачу, підписується здобувачем та керівником. Завдання на курсовий проєкт містить Календарний план виконання у вигляді таблиці, де зазначені назви розділів (етапів) та терміни їх виконання.

Після одержання завдання і ознайомлення зі змістом та обсягом роботи із курсового проєкту, здобувач за погодженням із керівником проєкту формулює орієнтовні назви розділів та пунктів пояснювальної записки курсового проєкту. У подальшому, здобувач згідно з календарним планом поетапно розробляє та оформляє пояснювальну записку і презентаційний матеріал, створює програмний застосунок і своєчасно звітує перед керівником курсового проєкту про виконання кожного етапу відповідно до графіка. Звітність здобувачів керівникові курсового проєкту стосовно ходу його виконання є обов'язковою.

Пояснювальна записка є текстовим документом, що містить аналіз, обґрунтування й опис прийнятих у курсовому проєкті рішень щодо розробки програмного застосунку, а також документування процесу програмного проєктування та реалізації програмного застосунку. Всі розділи пояснювальної записки повинні бути логічно взаємопов'язані і переконливо аргументовані. Матеріал її має бути цілком присвячений темі роботи, досягненню мети, вирішенню поставлених завдань. Неприпустимі будь-які відступи, що не мають відношення до завдання. Рекомендована кількість сторінок пояснювальної записки курсового проєкту з дисципліни «Алгоритмізація та програмування» (без завдання, анотації та додатків) 30–35 с. Максимальна сукупна кількість сторінок (із додатками) – не більше 100 с.

Консультації з курсового проєкту здійснює керівник проєкту – викладач кафедри відповідно до графіку, затвердженого деканом факультету. Захист курсового проєкту здійснюється після його перевірки керівником перед комісією, яка складається з 2–3 викладачів кафедри. До складу комісії обов'язково входить керівник проєкту.

### **Орієнтовна тематика курсових проєктів**

- Застосунок для вивчення синтаксису мови C# для початківців з використанням інтервального повторення матеріалу на платформі .NET.
- Застосунок-менеджер для планування розпорядку дня мовою C#.
- Програмний застосунок на мові C# для визначення енергетичної цінності раціонів здорового харчування.
- Застосунок для перевірки рівня знань з української мови на платформі .NET.
- Програмне забезпечення на платформі .NET для оцінювання ігрового потенціалу сформованих футбольних команд за характеристиками гравців.
- Застосунок для визначення якості сну за опитуванням користувача на платформі .NET.
- Застосунок для обліку витрачених калорій за виконаними видами спортивної діяльності з метою індивідуального регулювання навантаження засобами .NET.
- Застосунок-тренажер для покращення набору тексту методом сліпого друку засобами мови програмування C#.
- Застосунок для планування виду та обсягу тренувань в спортзалі згідно цілей користувача мовою C#.
- Програмний застосунок для підбору комплектуючих персональних комп'ютерів за параметрами користувача на платформі .NET.
- Програмний продукт для визначення рівня емпатії шляхом тестування психологічних властивостей та уподобань мовою C#.
- Програмний застосунок для відтворення аудіофайлів з елементами візуалізації та

персоналізованих налаштувань на платформі .NET.

- Програмний застосунок-щоденник для планування особистого навчального процесу студентами вищих навчальних закладів мовою C#.
- Комп'ютерна програма для формування білетів з підсумкових заходів дисциплін на платформі .NET.
- Графічний редактор для створення двовимірних зображень засобами .NET.
- Програма на платформі .NET для автоматизованого обрахунку потреб замовлень продуктового магазину.
- Застосунок для читання електронних книг з можливістю персоналізованих налаштувань засобами .NET.
- Програма-калькулятор на мові C# для розрахунку вартості комунальних послуг у багатоквартирних будинках міста Хмельницький.

### **Критерії оцінювання курсового проєкту**

Оцінювання курсового проєкту здійснюється за інституційною чотирибальною шкалою «відмінно», «добре», «задовільно», «незадовільно» і шкалою ЄКТС.

Критерії оцінювання курсового проєкту:

– відмінно – розроблений програмний застосунок виконує всі необхідні функції, що були поставлені у завданні, використано сучасні засоби розробки, пояснювальна записка повністю відповідає вимогам оформлення та розкриває всі положення курсового проєкту, на всі запитання на захисті дано вичерпні й коректні відповіді;

– добре – розроблений програмний застосунок виконує основні необхідні функції, що були поставлені у завданні, використано сучасні засоби розробки, пояснювальна записка у достатній мірі відповідає вимогам оформлення і розкриває ключові положення проєкту, на всі запитання на захисті дано коректні відповіді;

– задовільно – розроблений програмний застосунок виконує більшість необхідних функцій, що були поставлені у завданні, пояснювальна записка переважно відповідає вимогам оформлення й розкриває більшість положень курсового проєкту, на більшість запитань на захисті дано коректні відповіді;

– незадовільно – розроблений програмний застосунок виконує недостатню кількість функцій, що були поставлені у завданні, або не в повній мірі відповідає темі курсового проєкту, пояснювальна записка не відповідає вимогам оформлення або недостатньо розкриває положення курсового проєкту, на більшість запитань на захисті дано некоректні відповіді.

При оцінюванні комісія враховує фахову підготовку й теоретичні знання здобувача в межах дисципліни, вільне володіння фаховими термінами та поняттями, правильність і точність відповідей на питання, оригінальність та правильність вирішення поставлених завдань, дотримання графіку виконання та звітності, ступінь самостійності роботи здобувача, правильність вибору методів вирішення завдань та якість документування одержаних результатів у пояснювальній записці. Оцінюванню підлягає виключно особистий внесок розробника, тому здобувачу слід звернути особливу увагу та чітко вказати, де особисті наробки, а де використано загальнодоступні елементи та компоненти сторонніх розробників, як то програмні бібліотеки, автоматизовані засоби генерації програмних компонентів, існуючі математичні моделі та алгоритми тощо.

### **Структурування оцінювання курсового проєкту за компонентами та ваговими коефіцієнтами**

<b>№</b>	<b>Компоненти оцінювання</b>	<b>Коеф.</b>
1	Документування аналізу предметної області, існуючих технологій та аналогів. Постановка задачі, визначення вимог до системи.	0,1
2	Документування розробки блок-схем функцій програмного застосунку та UML-моделювання	0,2
3	Документування розробки комбінації засобів розробки, програмної реалізації застосунку та аналіз його функціональності	0,2

4	Якість оформлення пояснювальної записки (нормоконтроль)	0,1
5	Програмний застосунок – функціонал, коректність результатів, складність реалізації, використані технології.	0,2
6	Захист курсового проєкту (доповідь, відповіді на питання)	0,2

### Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Інституційна інтервальна шкала балів	Інституційна оцінка, критерії оцінювання		
A	4,75–5,00	5	Зараховано	<b>Відмінно</b> – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навичок
B	4,25–4,74	4		<b>Добре</b> – повне знання навчального матеріалу з кількома незначними помилками
C	3,75–4,24	4		<b>Добре</b> – в загальному правильна відповідь з двома-трьома суттєвими помилками
D	3,25–3,74	3		<b>Задовільно</b> – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00–3,24	3		<b>Задовільно</b> – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00–2,99	2	Незараховано	<b>Незадовільно</b> – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00–1,99	2		<b>Незадовільно</b> – необхідна серйозна подальша робота і повторне вивчення дисципліни

### Перелік питань для самоконтролю

1. Яким чином можна створити консольний додаток на мові C#?
2. Які парадигми програмування ви знаєте? Розкрийте коротко їх суть.
3. Що таке .NET Framework?
4. Які компоненти містить .NET Framework, що визначають його структуру?
5. Що таке транслятор, компілятор та інтерпретатор?
6. Що таке алгоритм?
7. Які властивості алгоритмів ви знаєте?
8. Які типи алгоритмів ви знаєте? Дайте їм визначення.
9. Що таке складність алгоритму? Які складності ви знаєте?
10. Які існують способи подання алгоритмів? Наведіть приклади.
11. Як позначаються основні блоки і що вони значать в блок-схемах?
12. Що таке мова програмування? Класифікації мов програмування.
13. З яких складових складається будь-яка мова програмування?
14. Що таке ключове слово? Наведіть приклад декількох ключових слів та що вони значать.
15. Що таке синтаксис мови програмування? Наведіть кілька правил синтаксису C#.
16. Що таке коментар? Які види коментарів ви знаєте?
17. Який метод в C# використовується для зчитування рядка символів, введеного з клавіатури в консольному вікні?
18. Яка різниця між методами Console.Write(), та Console.WriteLine ()?
19. Що таке змінна? Які види змінних існують у C#?
20. Які правила створення ідентифікатора змінної?



21. Що таке оголошення та ініціалізація змінної?
22. Що таке стек та купа? Яка між ними різниця?
23. Що таке константа? Характерні ознаки константи. Приклад константи.
24. Якими типами даних оперує мова C#?
25. В чому полягає різниця в оголошенні типу в C# і в CLR?
26. Які типи даних визначає користувач?
27. Як розміщуються дані у пам'яті під час роботи програми?
28. Яким чином здійснюється приведення типів змінних?
29. Основні логічні оператори.
30. Що таке унарний, бінарний та тернарний оператор? Наведіть приклади кожного з

них.

31. Що таке інкремент та декремент?
32. Які методи класу Math ви знаєте? Для чого вони використовуються?
33. Що таке умовна конструкція? Які типи умовних конструкцій ви знаєте?
34. Наведіть синтаксис оператора вибору – if.
35. Наведіть синтаксис оператора вибору – if...else.
36. Наведіть синтаксис оператора вибору switch.
37. Які існують оператори для створення циклів?
38. Оператор циклу for. Призначення, синтаксис.
39. Основна відмінність між циклами While і do While.
40. Що таке лічильник? З яким кроком може змінюватись лічильник?
41. Перерахуйте оператори переходу.
42. Дайте означення масиву.
43. Які види масивів ви знаєте?
44. Скільки вимірів може мати масив?
45. Як оголошуються одновимірні масиви?
46. Що таке коваріантність масивів?
47. Яким чином можна заповнювати масиви?
48. Яким чином описуються багатовимірні масиви?
49. Що таке зубчатий масив?
50. Як створити та ініціалізувати двовимірний зубчатий масив?
51. Що таке головна діагональ в двовимірному масиві? В якому масиві вона існує?
52. Для чого використовують масиви масивів?
53. Що таке символна інформація? Який тип даних в .NET Framework представляє

символи?

54. Які види символів може представляти тип даних char?
55. Наведіть декілька прикладів оголошення змінної типу char.
56. Що таке символ? Який тип даних використовується для представлення символів?
57. Які методи класу Char ви знаєте, що вони роблять?
58. Що таке символна константа? Наведіть приклади символних констант.
59. Що таке рядок? Який тип даних в .NET Framework представляє рядок?
60. Наведіть декілька прикладів оголошення змінної типу string.
61. Як розташовуються у пам'яті рядки? Який максимальний розмір об'єкта string?
62. Які методи класу String ви знаєте, що вони роблять?
63. За допомогою якого методу можна порівняти два рядки? Запишіть у вигляді коду.
64. За допомогою якого методу можна об'єднати два рядки? Запишіть у вигляді коду.
65. За допомогою якого методу можна скопіювати один рядок в інший? Запишіть у

вигляді коду.

66. За допомогою якого методу можна вставити підрядок з заданого індексу в інший рядок? Запишіть у вигляді коду.

67. За допомогою якого методу можна видалити задану кількість символів з рядка?  
Запишіть у вигляді коду.
68. За допомогою якого методу можна замінити символи в рядку? Запишіть у вигляді коду.
69. За допомогою якого методу можна виділити у рядку підрядок? Запишіть у вигляді коду.
70. За допомогою якого методу можна розділити рядок на масив підрядків? Запишіть у вигляді коду.
71. Які інтерфейси успадковує клас System.Array?
72. Перерахуйте та опишіть декілька методів класу System.Array.
73. Який недолік класу String компенсує клас StringBuilder?
74. Якими методами володіє клас StringBuilder?
75. Що таке буфер класу StringBuilder? Що таке ємність буфера? Чи можна її збільшувати та зменшувати?
76. Дайте визначення, що таке файлова система?
77. Що таке файл?
78. Які операції можна проводити з файлами?
79. Що таке формат файлу?
80. Що таке розширення файлу? Як воно записується?
81. Що таке каталог (папка)?
82. Які класи для роботи з файловою системою ви знаєте?
83. Які класи для роботи з потоками даних ви знаєте?
84. Які класи для роботи з каталогами (папками) ви знаєте?
85. Які класи для роботи з файлами ви знаєте?
86. Опишіть основні методи класу Directory.
87. Які операції з каталогами (папками) дозволяє робити клас DirectoryInfo?
88. Опишіть основні методи класу File.
89. Які операції з файлами дозволяє робити клас FileInfo?
90. Що буде, якщо не закрити потік при роботі з файлом за допомогою методу Close класу FileInfo?
91. Для чого призначений клас Path? Які методи цього класу ви знаєте?
92. Для чого призначені таблиці розміщення файлів FAT?
93. Яку інформацію може надавати клас Environment через свої статичні члени?
94. Які каталоги (папки) називаються спеціальними? Перерахуйте їх.
95. Які можливості надає клас FileStream?
96. Що таке парадигма програмування?
97. Які основні парадигми програмування ви знаєте?
98. Що таке імперативне програмування?
99. Що таке декларативне програмування?
100. Яка мова називається мультипарадигмальною? Наведіть приклади таких мов.
101. Що таке узагальнене програмування?
102. Що таке об'єктно-орієнтоване програмування?
103. Які переваги об'єктно-орієнтованого програмування?
104. Що таке функціональне програмування?
105. Що таке логічне програмування?
106. Що таке інкапсуляція?
107. Що таке поліморфізм?
108. Що таке наслідування?
109. Що таке клас?
110. Що таке екземпляр класу?
111. Наведіть синтаксис створення класу та екземпляру класу.
112. Дайте визначення поля та методу класу?
113. Для чого в методах використовується інструкція return? Чи у всіх випадках вона використовується?

- 114.Що таке перевантаження метода? Наведіть приклад.
- 115.Які модифікатори доступу до полів та методів класів ви знаєте?
- 116.Що таке конструктор класу? Чи завжди клас має хоча б один конструктор класу?
- 117.Що таке деструктор класу?
- 118.Який клас називається базовим, а який класом-нащадком?
- 119.Що таке заміщення метода батьківського класу?
- 120.Що таке колекція у C#?
- 121.Головна перевага використання колекцій у C#.
- 122.Які існують типи колекцій?
- 123.До якого типу колекцій належить клас List? Чому?
- 124.Чим відрізняються узагальнені колекції від неузагальнених?
- 125.Які інтерфейси використовуються для реалізації колекцій у C#?
- 126.Які основні класи колекцій ви знаєте?
- 127.Чим відрізняється клас Array від класу ArrayList?
- 128.Що таке стек, його принцип роботи?
- 129.Що таке черга, її принцип роботи?
- 130.Яка суть процесу хешування?

## **МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ**

Навчальний процес з дисципліни «Алгоритмізація та програмування» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою. Зокрема, викладачами кафедри підготовлено і видано роботи:

1. Собко О. В., Молчанова М. О., Бармак О. В., Мазурець О. В. Алгоритмізація та програмування: методичні вказівки до лабораторних робіт для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 122 «Комп'ютерні науки». Хмельницький, ХНУ. 2022. 108с.
2. Собко О. В., Молчанова М. О., Мазурець О. В., Бармак О. В. Алгоритмізація та програмування : методичні рекомендації до курсового проєктування для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 122 «Комп'ютерні науки» / О. В. Собко, М. О. Молчанова, О. В. Мазурець, О. В. Бармак. Хмельницький : ХНУ, 2024. 36 с.

## **РЕКОМЕНДОВАНА ЛІТЕРАТУРА**

### **Основна**

- 1.Собко О. В., Молчанова М. О., Бармак О. В., Мазурець О. В. Алгоритмізація та програмування: методичні вказівки до лабораторних робіт для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 122 «Комп'ютерні науки». Хмельницький, ХНУ. 2022. 108с.
2. Собко О. В., Молчанова М. О., Мазурець О. В., Бармак О. В. Алгоритмізація та програмування : методичні рекомендації до курсового проєктування для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 122 «Комп'ютерні науки». Хмельницький : ХНУ, 2024. 36 с.
- 3.Коноваленко І. В. Платформа .NET та мова програмування C# 8.0: навчальний посібник. Тернопіль, 2020. 320 с. URL: <https://elartu.tntu.edu.ua/bitstream/lib/32825/1/Konovalenko%20I.%20NET-C%23.pdf>
- 4.Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування: Конспект лекцій. Київ, КПІ ім. І.Сікорського. 2020. 117 с. URL: [https://ela.kpi.ua/bitstream/123456789/39004/1/OTtaPr\\_KonspLek\\_1.pdf](https://ela.kpi.ua/bitstream/123456789/39004/1/OTtaPr_KonspLek_1.pdf)

### **Допоміжна**

5.How to automate Microsoft Excel from Microsoft Visual C# .NET. URL: <https://learn.microsoft.com/uk-ua/previous-versions/office/troubleshoot/office-developer/automate-excel-from-visual-c>

6.How to automate Microsoft Word to create a new document by using Visual C#. URL: <https://learn.microsoft.com/en-us/previous-versions/office/troubleshoot/office-developer/automate-word-create-file-using-visual-c>

7.Edit Word Documents using OpenXML and C#. URL: <https://www.codingame.com/playgrounds/11047/edit-word-documents-using-openxml-and-c-without-automationinterop>

8. Свістельник, І. Інформаційна культура студента : навч. посіб. Київ, Кондор, 2018. – 182 с.

9. Цибульник, С. О. Технології розроблення програмного забезпечення. Частина 1. Життєвий цикл програмного. Київ, КПІ ім. Ігоря Сікорського, 2022. – 270 с.

10. Eland M. Refactoring with C#: Safely Improve .NET Applications and Pay Down Technical Debt with Visual Studio, .NET 8, and C# 12. Packt Publishing, 2023 – 434 pp.

11. Трофименко О. Г., Дика А. І. Тестування та забезпечення якості програмних систем : навч. посіб.Нац. ун-т «Одес. юрид.академія», Фенікс, 2024. – 195 с

12. Бармак О. В., Кліменко В. І., Мазурець О. В., Молчанова М. О. Собко О. В.. Основи програмної інженерії та тестування програмного забезпечення. Теоретичний курс та лабораторний практикум: навчальний посібник – Хмельницький. ХНУ, 2023 – 236 с.

13. Бармак О.В., Радюк П.М., Молчанова М.О., Собко О.В. Підходи до практичного аналізу обчислювальних алгоритмів. Вісник Хмельницького національного університету, № 6, 2021 (303), 2021. С.102-105.

14. Молчанова М.О., Мазурець О.В., Собко О.В., Віт. Р.В., Назаров В.В. Алгоритм виявлення аб'юзивного вмісту в україномовному аудіоконтенті для імплементації в об'єктно-орієнтовану інформаційну систему. Науковий журнал «Вісник Хмельницького національного університету» серія: Технічні науки. Хмельницький, 2024. №1 (331). С. 101-106.

#### **Інформаційні ресурси:**

1. Модульне середовище. URL: <https://msn.khnu.km.ua/>
2. Електронна бібліотека університету. URL: [http://lib.khmnu.edu.ua/asp/php\\_f/p1age\\_lib.php](http://lib.khmnu.edu.ua/asp/php_f/p1age_lib.php)
3. Репозитарій ХНУ. URL: <https://library.khmnu.edu.ua/#>