

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
Кафедра моделювання та програмного забезпечення



ЗАТВЕРДЖУЮ

Перший проректор

Владислав ЧУБАРОВ

2024 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Алгоритмізація обчислювальних процесів

(шифр і назва навчальної дисципліни)

спеціальність

121 – «Інженерія програмного забезпечення»

(шифр і назва напрямку підготовки)

факультет

Інформаційних технологій

(назва інституту, факультету, відділення)

Форма навчання	Курс	Семестр	Всього годин за планом	Кількість кредитів	Всього аудиторних годин	Аудиторних годин, (у тому числі КЗ)		Самостійна робота (год.)	Контрольно-модульні роботи	Залік (сем.)	Екзамен (сем.)
						Лекції	Лабораторні				
Денна	1	1	120	4	48	16	32	72	1		*
Заочна	1	1	120	4	12	6	6	108	-		*

Кривий Ріг – 2024 рік

Робочу програму навчальної дисципліни «Алгоритмізація обчислювальних процесів» для здобувачів першого (бакалаврського) рівня вищої освіти за освітньою програмою «Інженерія програмного забезпечення» розроблено згідно з ОПП галузі знань 12 «Інформаційні технології» зі спеціальності 121 «Інженерія програмного забезпечення».

Розробник: старший викладач кафедри МПЗ Рибальченко О.Г.

Робоча програма затверджена на засіданні кафедри моделювання та програмного забезпечення

Протокол від "25" квітня 2024 року № 9

Завідувач кафедри МПЗ, доцент, к.п.н.



Андрій СТРЮК

Схвалено вченою радою факультету інформаційних технологій

Протокол від "11" червня 2024 року № 13

Голова вченої ради



Іван МУЗИКА

Схвалено групою забезпечення ОПП

Протокол від "25" 04 2024 року № 9

Гарант ОПП



Андрій СТРЮК

© Рибальченко О.Г., 2024 рік

**КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
Кафедра моделювання та програмного забезпечення

**ЗАТВЕРДЖУЮ**

Перший проректор

\_\_\_\_\_ Владислав ЧУБАРОВ  
“ \_\_\_\_\_ ” \_\_\_\_\_ 2024 р.

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Алгоритмізація обчислювальних процесів

(шифр і назва навчальної дисципліни)

спеціальність

121 – «Інженерія програмного забезпечення»

(шифр і назва напрямку підготовки)

факультет

Інформаційних технологій

(назва інституту, факультету, відділення)

Форма навчання	Курс	Семестр	Всього годин за планом	Кількість кредитів	Всього аудиторних годин	Аудиторних годин, (у тому числі КЗ)		Самостійна робота (год.)	Контрольно-модульні роботи	Залік (сем.)	Екзамен (сем.)
						Лекції	Лабораторні				
Денна	1	1	120	4	48	16	32	72	1		*
Заочна	1	1	120	4	12	6	6	108	-		*

Кривий Ріг – 2024 рік

Робочу програму навчальної дисципліни «Алгоритмізація обчислювальних процесів» для здобувачів першого (бакалаврського) рівня вищої освіти за освітньою програмою «Інженерія програмного забезпечення» розроблено згідно з ОПП галузі знань 12 «Інформаційні технології» зі спеціальності 121 «Інженерія програмного забезпечення».

Розробник: старший викладач кафедри МПЗ Рибальченко О.Г.

Робоча програма затверджена на засіданні кафедри моделювання та програмного забезпечення

Протокол від “ ” \_\_\_\_\_ 2024 року № \_\_\_\_\_

Завідувач кафедри МПЗ, доцент, к.п.н. \_\_\_\_\_ Андрій СТРЮК

Схвалено вченою радою факультету інформаційних технологій

Протокол від “ ” \_\_\_\_\_ 2024 року № \_\_\_\_\_

Голова вченої ради \_\_\_\_\_ Іван МУЗИКА

Схвалено групою забезпечення ОПП

Протокол від “ ” \_\_\_\_\_ 2024 року № \_\_\_\_\_

Гарант ОПП \_\_\_\_\_ Андрій СТРЮК

## ЗМІСТ

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	4
2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	5
3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	6
4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	7
5. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ.....	8
6. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ.....	8
7. САМОСТІЙНА РОБОТА.....	9
8. МЕТОДИ НАВЧАННЯ.....	10
9. МЕТОДИ КОНТРОЛЮ.....	11
10. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ.....	11
11. ПЕРЕЛІК ПИТАНЬ ДЛЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ.	15
12. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ З ДИСЦИПЛІНИ.....	17
13. ІНФОРМАЦІЙНІ РЕСУРСИ.....	17
14. ТЕРМІНОЛОГІЧНИЙ СЛОВНИК	18
Додаток до робочої програми. Робочий план.....	20

## 1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – <b>4</b>	<b>Галузь знань</b> <u>12 Інформаційні технології</u> (шифр і назва)	Нормативна	
Модулів – <b>1</b>	<b>Спеціальність</b> <u>121 Інженерія програмного забезпечення</u> (код та найменування спеціальності)	<b>Рік підготовки:</b>	
Змістових модулів – <b>1</b>		2024/2025 н.р.	2024/2025 н.р.
Загальна кількість годин - <b>120</b>		<b>Семестр</b>	
		1-й	1-й
		<b>Лекції</b>	
<b>Тижневих годин для денної форми навчання:</b> аудиторних – <b>3</b> самостійної роботи студента – <b>4,5</b>	Ступінь вищої освіти: <u>бакалавр</u>	16 год.	6 год.
		<b>Практичні, семінарські</b>	
		-	-
		<b>Лабораторні</b>	
		32 год.	6 год.
		<b>Самостійна робота</b>	
		72 год.	108 год.
<b>Вид контролю:</b> Екзамен			

### Примітка.

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

- для денної форми навчання – 0,67
- для заочної форми навчання – 0,11

## 2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

2.1. Метою викладання навчальної дисципліни «Алгоритмізація обчислювальних процесів» є формування у студентів чіткого уявлення про методи розробки алгоритмів та модульний підхід до побудови алгоритмів, про методи структурного програмування, про використання професійних стандартів та інших нормативно-правових документів у галузі інженерії програмного забезпечення.

2.2. Основними завданнями вивчення дисципліни «Алгоритмізація обчислювальних процесів» є:

- ознайомити студентів з класичними методами побудови алгоритмів, з основами знаходження та дослідження оптимального обчислювального алгоритму;
- надати студентам знання про основи структурного програмування;
- навчити студентів розроблювати документацію, необхідну для розробки та супроводу програмного об'єкту (ПО), відповідно до вимог Єдиної системи програмної документації (ЄСПД), користуючись автоматизованими системами виготовлення програмної документації.

2.3. В результаті опанування дисципліни студенти освоюють наступні компетентності:

### *загальні*

- здатність до абстрактного мислення, аналізу та синтезу (ЗК01);
- здатність застосовувати знання у практичних ситуаціях (ЗК02);

### *фахові*

- здатність дотримуватись специфікацій, стандартів, правил і рекомендацій в професійній галузі (СК05);
- здатність застосовувати фундаментальні та міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення (СК08);
- здатність до алгоритмічного та логічного мислення (СК14).

В результаті вивчення дисципліни студенти повинні *знати*:

- класичні загальні методи розв'язання алгоритмів, недоліки і переваги кожного з них;
- принципи побудови рекурсивних алгоритмів;
- основні принципи структурного програмування;
- вимоги Єдиної системи програмної документації (ЄСПД) до оформлення алгоритмів;

### *вміти*:

- застосовувати професійні стандарти та інші нормативно-правові документи у галузі інженерії програмного забезпечення
- вибирати та використовувати відповідну задачі методологію створення програмного забезпечення;
- застосовувати відповідні методи розробки алгоритмів, структур даних і знань;
- документувати та презентувати результати розробки програмного забезпечення.

### 2.4. Міждисциплінарні зв'язки

При вивченні дисципліни використовуються знання здобувачів з дисциплін шкільних курсів «Інформатика», «Математика».

Знання, одержані здобувачами при вивченні дисципліни, використовуються при вивченні дисциплін «Основи програмування», «Алгоритми та структури даних».

Вимоги до знань та умінь визначаються галузевими стандартами вищої освіти України.

### 3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

#### Модуль 1 Алгоритмізація обчислювальних процесів

##### **Тема 1. Алгоритми та блок-схеми – 2 год.**

Поняття алгоритму. Вимоги до алгоритму. Способи запису алгоритму. Блок-схема. Система стандартизації ЕСПД. Елементарні базові конструкції алгоритмів: послідовність, розгалуження та повторення. Лінійні алгоритми.

##### **Тема 2. Елементарні базові структури розгалужень – 2 год.**

Види та типи даних. Константи та змінні. Операції відношення. Логічні операції та складні логічні вирази. Таблиці істинності.

##### **Тема 3. Типові розгалужені алгоритми – 2 год.**

Знаходження максимальних та мінімальних величин. Перевірка влучення в інтервал та його виключення. Класичні прийоми сортування величин.

##### **Тема 4. Вступ до циклічних алгоритмів – 2 год.**

Елементарні базові структури циклів. Організація простих циклів. Організація циклів з лічильником. Цикли з накопиченням. Ітераційні цикли.

##### **Тема 5. Поняття масиву – 2 год.**

Організація даних у масиві. Організація циклів з переадресацією.

##### **Тема 6. Типові алгоритми обробки одновимірного масиву – 2 год.**

Вибір за ознакою. Обробка частки масиву. Пошук екстремуму. Вставка та видалення елементів. Зсув та переміщення. Обробка векторів, заданих компонентами. Методи сортування масивів.

##### **Тема 7. Багатовимірні масиви. Типові алгоритми обробки двовимірних масивів – 2 год.**

Робота з матрицями. Типові алгоритми обробки матриць: вибір за ознакою, пошук екстремуму, сортування рядків та стовпців.

##### **Тема 8. Структурне програмування – 2 год.**

Поняття про модульний принцип програмування. Висхідне та низхідне програмування. Організація підпрограм, функції та процедури. Механізм формальних та фактичних параметрів. Рекурсія та рекурсивні алгоритми.



#### 4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин							
	Денна форма				Заочна форма			
	усього	у тому числі			усього	у тому числі		
		лекції	лабораторні	самостійна робота		лекції	лабораторні	самостійна робота
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
<b>Модуль 1</b>								
Тема 1. Алгоритми та блок-схеми	14	2	4	8	14	0,5	-	13,5
Тема 2. Елементарні базові структури розгалужень	15	2	4	9	15	0,5	0,5	14
Тема 3. Типові розгалужені алгоритми	15	2	4	9	15	1	1	13
Тема 4. Вступ до циклічних алгоритмів	14	2	4	8	14	1	0,5	12,5
Тема 5. Поняття масиву	15	2	4	9	15	1	1	13
Тема 6. Типові алгоритми обробки одновимірного масиву	15	2	4	9	15	1	1	13
Тема 7. Багатовимірні масиви. Типові алгоритми обробки двовимірних масивів	15	2	4	9	15	1	1	13
Тема 8. Структурне програмування	17	2	4	11	17	-	1	16
<b>Усього годин за модуль 1:</b>	<b>120</b>	<b>16</b>	<b>32</b>	<b>72</b>	<b>120</b>	<b>6</b>	<b>6</b>	<b>108</b>

## 5. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ

Не передбачено.

## 6. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ з/п	Назва теми	Кількість годин	
		Денна ф. н.	Заочна ф. н.
1.	Лінійні алгоритми	4	-
2.	Прості розгалужені алгоритми	4	0,5
3.	Складні розгалужені алгоритми	4	1
4.	Прості цикли	4	0,5
5.	Цикли з переадресацією	4	1
6.	Типові алгоритми обробки одновимірного масиву	4	1
7.	Типові алгоритми обробки двовимірних масивів	4	1
8.	Структурне програмування	4	1
<b>РАЗОМ</b>		<b>32</b>	<b>6</b>

## 7. САМОСТІЙНА РОБОТА

На самостійну роботу студентам денної форми навчання відведено 72 години, заочної – 108 годин.

Самостійна робота студентів при вивченні дисципліни «Алгоритмізація обчислювальних процесів» залучає такі складові:

- опрацювання теоретичних основ прослуханого лекційного матеріалу;
- вивчення окремих тем або питань, що передбачені для самостійного опрацювання;
- підготовка до виконання, а також до захисту лабораторних робіт;
- підготовка до проведення контрольних заходів.

При виконанні самостійної роботи студент денної форми навчання повинен:

- скласти алгоритми рішення індивідуальних задач;
- оформити звіт з лабораторних робіт;

При виконанні самостійної роботи студент заочної форми навчання повинен:

- опрацювати матеріал, що винесено на самостійне вивчення;
- скласти алгоритми рішення індивідуальних задач;
- оформити звіт з лабораторних робіт

### Питання для самостійного опрацювання

№	Назва теми	Кількість годин	
		денна ф.н.	заочна ф.н.
1.	Різні форми запису елементарних базових структур усіх типів	6	6
2.	Переваги і недоліки різних алгоритмів ранжування засобами розгалужених алгоритмів	6	6
3.	Організація циклу зі збереженням результатів обчислень	6	6
4.	Алгоритм ранжування елементів одномірного масиву методом прямого включення.	6	6
5.	Типові алгоритми обробки матриць: складання двох матриць, множення матриці на вектор, множення матриці на матрицю	6	6
6.	Особливості організації та зображення рекурсивних алгоритмів	6	6
7.	Опрацювання матеріалу лекцій, робота з рекомендованою літературою	18	48
8.	Виконання лабораторних робіт	8	16
9.	Підготовка звітів з лабораторних робіт	8	8
10.	Підготовка до контрольних заходів	2	-
	<b>Разом</b>	<b>72</b>	<b>108</b>

## 8. МЕТОДИ НАВЧАННЯ

Використовуються наступні методи навчання: лекції, лабораторні заняття, самостійна робота.

Навчальна лекція – це логічне, послідовне викладання змісту навчання, яке характеризується судженнями, висновками, підсумком. Вона охоплює основний теоретичний матеріал однієї або кількох тем навчальної дисципліни. Призначенням лекції є формування у здобувачів фундаментальних знань з дисципліни, а також визначає основний зміст і характер усіх інших навчальних занять та самостійної роботи здобувачів із цієї дисципліни.

Лабораторне заняття - форма організації навчання, яку проводять за завданням і під керівництвом НПП. Основні дидактичні цілі – експериментальне підтвердження вивчених теоретичних положень навчальної дисципліни та формування вмінь й навичок їх практичного застосування. Проведення лабораторного заняття ґрунтується на попередньо підготовлених наборах завдань різної складності для розв'язання на занятті. Лабораторне заняття проводиться у навчальних лабораторіях з використанням пристосованого до умов навчального процесу устаткування.

Самостійна робота здобувача є основним способом оволодіння навчальним матеріалом у час, вільний від обов'язкових аудиторних занять. Мета виконання самостійної роботи – поглиблення, узагальнення й закріплення теоретичних знань і практичних умінь здобувачів із дисципліни шляхом вироблення вмінь самостійної роботи з навчальною і фаховою літературою та інформацією в мережі Інтернет.

Самостійна робота здобувачів здійснюється у формі: підготовки до лекцій і лабораторних занять, виконанні самостійних проєктів. Самостійну роботу здобувач може виконувати у бібліотеці, комп'ютерних класах, а також у домашніх умовах.

Підготовка до лекцій передбачає самостійне опрацювання теоретичного матеріалу. При цьому необхідно звернути увагу на необхідність чіткого засвоєння основних термінів та визначень, розуміння їх змісту, обов'язкового аналізу використання теоретичних положень для розв'язання наданих прикладів.

Самоперевірку засвоєння навчального матеріалу здобувач здійснює за контрольними запитаннями, що надано після кожної теми у конспекті лекцій та іншій літературі, та після кожного лабораторного заняття у відповідних методичних вказівках. Якщо на деякі запитання здобувач не може надати відповіді, то необхідно повторити вивчення навчального матеріалу, або визначити правильну відповідь з викладачем на консультації.

Під час вивчення даної дисципліни використовуються:

- мультимедійні освітні технології: інтерактивні лекції (презентації) із використанням програми MS Power Point у поєднанні з анімацією та звуковим супроводом; перегляд відеороликів за окремими пунктами тем занять, використання електронних посібників;
- діалогові технології: організація групових обговорень, використання «мозкового штурму».

Лекції проводяться з використанням технічних засобів навчання й супроводжуються демонстрацією презентацій за допомогою проєктора.

У разі виникнення необхідності забезпечення навчального процесу в дистанційному режимі супровід та контроль знань реалізовується за допомогою дистанційного курсу, розробленого в Google Classroom. Онлайн лекції, консультації та усні відповіді на питання, захист проєктів проводиться за допомогою Google Meet або Zoom.

## 9. МЕТОДИ КОНТРОЛЮ

Основними завданнями контролю знань здобувачів вищої освіти з дисципліни є оцінювання засвоєння теоретичних знань і практичних навичок, отриманих під час навчання.

Контрольні заходи мають виконувати наступні функції:

- стимулювати систематичну самостійну роботу над навчальним матеріалом;
- забезпечувати закріплення та реалізацію набутих теоретичних знань при підготовці до практичних занять;
- прищеплювати навички відповідального ставлення до своїх обов'язків, самостійного цілеспрямованого пошуку потрібної інформації, чіткої організації свого робочого дня.

Оцінювання знань здобувачів складається з поточного та підсумкового контролю.

Поточний контроль знань здобувачів вищої освіти передбачає оцінювання за наступними основними напрямками:

- перевірка теоретичних знань;
- перевірка підготовки до лабораторних занять.

З даних компонентів складаються загальні бали, які фіксуються в журналі викладача.

Оцінювання рівня засвоєння теоретичних знань здобувачів вищої освіти проводиться під час усної співбесіди зі здобувачами по теоретичним матеріалам, за результатами захисту лабораторних робіт й виконання самостійних робіт. Підсумковим контролем є екзамен.

## 10. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ

Використовується модульно-рейтингова система оцінювання, яка передбачає розподіл балів за виконання всіх запланованих видів робіт. При цьому максимальна кількість балів за модуль при умові його бездоганного виконання дорівнює 100. Ця сума складається з балів отриманих за контрольню-модульну роботу (КМР), балів, що їх накопичив студент у ході поточного контролю, та балів за виконання додаткового завдання.

КМР відображує теоретичні знання і відповідає 10-ти відсоткам його ваги, тобто вона може дати максимально 10 балів при найвищій якості виконання. При зниженні якості КМР знижується і сума балів відповідно до шкали, що наводиться у таблиці:

### Шкала оцінювання контрольню-модульних робіт

Відсоток вірних компонентів КМР	0 – 30	31 – 60	61 – 75	76 – 85	86 – 94	95 – 100
Сума балів за КМР	0	2	4	6	8	10

Успішність студентів-заочників оцінюється аналогічним чином, за 100-бальною шкалою. Контрольна робота (КР), яку виконують студенти-заочники, може дати максимально 80 балів при найвищій якості виконання. При зниженні якості виконання контрольної роботи знижується і сума балів відповідно до шкали, що наводиться в таблиці:

### Шкала оцінювання контрольного завдання студентів заочної форми навчання

Відсоток вірних компонентів КР	0 – 30	31 – 50	51 – 60	61 – 80	81 – 94	95 – 100
Сума балів за КР	0	30	50	75	90	100

### Шкала оцінювання лабораторних робіт

№ модуля	№ зан.	Вид роботи	Тема	Кількість задач	Максимальна кількість балів		
					денна ф.н.	заочна ф.н.	
1	1	Лабораторна робота № 1	Лінійні алгоритми	2	10	10	
	2	Лабораторна робота № 2	Прості розгалужені алгоритми	2	10	10	
	3	Лабораторна робота № 3	Складні розгалужені алгоритми	2	10	10	
	4	Лабораторна робота № 4	Прості цикли	2	10	10	
	5	Лабораторна робота № 5	Цикли з переадресацією	2	10	10	
	6	Лабораторна робота № 6	Типові алгоритми обробки одновимірного масиву	2	10	20	
	7	Лабораторна робота № 7	Типові алгоритми обробки двовимірних масивів	2	10	20	
	8	Лабораторна робота №8	Структурне програмування	2	10	10	
	<b><i>Разом за лабораторні роботи / індивідуальне завдання</i></b>					<b>80</b>	<b>100</b>
	<b><i>Контрольно-модульна робота</i></b>					<b>10</b>	<b>-</b>
	<b><i>Екзамен / додаткове завдання</i></b>					<b>10</b>	
<b><i>Разом за модуль</i></b>					<b>100</b>	<b>100</b>	

Лабораторні роботи у модулі відображують оволодіння навичками та вміння застосовувати знання на практиці і сукупно відповідають 80-ти відсоткам ваги модуля для денної форми навчання. Таким чином всі лабораторні роботи модуля при бездоганному їх виконанні можуть дати 80 балів. Ці бали розподіляються поміж лабораторними роботами модуля у відповідності до їх відносної складності. При зниженні якості виконання тієї чи іншої лабораторної роботи, знижується і кількість балів, якою вона оцінюється.

Оцінювання задач лабораторних робіт проводиться за 5-тибальною шкалою. Найвищий бал можна отримати тільки при успішному захисті лабораторної роботи вчасно (на наступному занятті після видачі завдання). За несвоєчасне виконання завдання оцінка зменшується на 1 бал за кожну задачу.

Оцінювання кожної задачі лабораторної роботи ведеться за наступними показниками:

1. Блок-схема складена правильно, оформлена відповідно до вимог ЕСКД (1 бал).
2. Тестування блок-схеми виконано вірно (1 бал).
3. Аналіз блок-схеми детальний та правильний (1 бал).
4. При захисті роботи студент демонструє достатній рівень знань (1 бал).
5. Завдання виконано та захищено вчасно відповідно до графіку виконання робіт (1 бал).

Для студентів-заочників показники оцінюються наступним чином:

1. Блок-схема складена правильно, оформлена відповідно до вимог ЕСКД (2 бали).
2. Тестування блок-схеми виконано вірно (1 бал).
3. Аналіз блок-схеми детальний та правильний (1 бал).
4. При захисті роботи студент демонструє достатній рівень знань (1 бал).

Оцінка всієї лабораторної роботи знаходиться підсумовуванням балів за кожний з показників.

Модульний контроль здійснюється 1 раз за семестр.

Поточний контроль за модулем 1 проводиться на 16-ому тижні першого семестру.

Контрольно-модульна робота складається з контрольних питань та завдання.

Оцінювання *контрольних питань* розподіляється пропорційно їх кількості.

Оцінювання *контрольного завдання* виконується за принципами оцінювання лабораторних робіт (вірно виконана та оформлені блок-схема, тестування та аналіз задачі).

Для допуску до підсумкового контролю студент повинен виконати графік навчального процесу, усі види запланованих завдань і протягом семестру отримати в сумі не менше 50 балів.

Для підвищення рейтингового балу студенти, які стабільно показують максимально можливі бали за результатами поточного контролю, за бажанням отримують *додаткове завдання*. Для виконання завдання потрібно вивчити теоретичний матеріал, скласти алгоритм та програму, підготувати презентацію і доповідь про один зі складних методів сортування масивів. Результат роботи максимально оцінюється *10 балами*.

Семестровий контроль здійснюється у формі екзамену в першому семестрі для всіх форм навчання.

На екзамені студент-заочник повинен мати при собі оформлений звіт та змінний носій зі звітами до всіх лабораторних робіт.

Результати екзамену оцінюються за 10-бальною шкалою. Підсумкова оцінка з дисципліни розраховується як середня арифметична чи зважена з оцінок за модуль, додаткове завдання та екзамен.

У відомість оцінка проставляється як у балах національної шкали, так і за шкалою ECTS:

При наявності у здобувачів **результатів неформального навчання** за освітнім компонентом «Алгоритмізація обчислювальних процесів» у повному обсязі, визнання та оцінювання результатів здійснюється відповідно до «Положення про порядок визнання у Криворізькому національному університеті результатів навчання, отриманих в умовах неформальної освіти». У випадку, якщо за підсумками визнання результатів неформального навчання визнається тільки частина результатів навчання, заявнику зараховуються окремі види навчальної роботи за освітнім компонентом «Алгоритмізація обчислювальних процесів».

Нижче наведені окремі види навчальної роботи, які можуть бути зараховані здобувачеві при наявності сертифікату про успішне проходження рекомендованих онлайн курсів.

Тема	Посилання на рекомендовані курси
Сортування. Алгоритми сортування масивів	<a href="https://pro.foxminded.ua/algorithms-data-structures-1/">https://pro.foxminded.ua/algorithms-data-structures-1/</a> <a href="https://www.coursera.org/learn/algorithms-searching-sorting-indexing">https://www.coursera.org/learn/algorithms-searching-sorting-indexing</a> <a href="https://www.coursera.org/specializations/data-structures-algorithms">https://www.coursera.org/specializations/data-structures-algorithms</a> <a href="https://www.coursera.org/learn/algorithms-part1#syllabus">https://www.coursera.org/learn/algorithms-part1#syllabus</a>

Пошук. Алгоритми пошуку даних	<a href="https://pro.foxminded.ua/algorithms-data-structures-1/">https://pro.foxminded.ua/algorithms-data-structures-1/</a> <a href="https://www.coursera.org/learn/algorithms-searching-sorting-indexing">https://www.coursera.org/learn/algorithms-searching-sorting-indexing</a> <a href="https://www.coursera.org/specializations/data-structures-algorithms">https://www.coursera.org/specializations/data-structures-algorithms</a> <a href="https://www.coursera.org/learn/algorithms-part1#syllabus">https://www.coursera.org/learn/algorithms-part1#syllabus</a>
Рекурсія	<a href="https://pro.foxminded.ua/algorithms-data-structures-1/">https://pro.foxminded.ua/algorithms-data-structures-1/</a> <a href="https://courses.prometheus.org.ua/courses/course-v1:KPI+Algorithms101+2015_Spring/about">https://courses.prometheus.org.ua/courses/course-v1:KPI+Algorithms101+2015_Spring/about</a> <a href="https://www.coursera.org/specializations/data-structures-algorithms">https://www.coursera.org/specializations/data-structures-algorithms</a> <a href="https://www.coursera.org/learn/algorithms-part1#syllabus">https://www.coursera.org/learn/algorithms-part1#syllabus</a>

### Шкала оцінювання

Національна шкала успішності	Оцінка ECTS	Визначення ECTS	100-бальна система оцінювання
відмінно/ зараховано	<b>A</b>	ВІДМІННО - відмінне виконання лише з незначними помилками	90...100
добре/ зараховано	<b>B</b>	ДУЖЕ ДОБРЕ - вище середнього рівня з кількома помилками	80...89
	<b>C</b>	ДОБРЕ - у цілому правильно робота з певною кількістю помилок і недоліків	71...79
задовільно/ зараховано	<b>D</b>	ЗАДОВІЛЬНО - непогано, але зі значною кількістю грубих помилок	61...70
	<b>E</b>	ДОСТАТНЬО - виконання задовольняє мінімальні потреби	50...60
незадовільно/ не зараховано	<b>FX</b>	НЕЗАДОВІЛЬНО - із можливістю повторного складання	30...49
	<b>F</b>	НЕЗАДОВІЛЬНО - з обов'язковим повторним вивчення дисципліни	0...29



## 11. ПЕРЕЛІК ПИТАНЬ ДЛЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ, УМІНЬ, НАВИЧОК

### Модуль 1

1. Етапи розробки програмного забезпечення.
2. Поняття алгоритму. Властивості алгоритмів. Поняття ступеня деталізації алгоритму.
3. Правила оформлення блок-схем алгоритмів.
4. Форми запису алгоритмів.
5. Приклади елементарних базових структур усіх типів у різних формах запису.
6. Основні принципи структурного програмування.
7. Класифікація алгоритмів. Визначення кожного типу алгоритму та його призначення.
8. Організація лінійного алгоритму. Приклади використання.
9. Алгоритми обміну перемінних значеннями.
10. Типові блок-схеми розгалужень.
11. Алгоритми пошуку найбільшого (найменшого) із трьох та чотирьох чисел.
12. Порівняльний аналіз алгоритмів пошуку екстремуму.
13. Ранжування трьох та чотирьох чисел засобами розгалужених алгоритмів.
14. Переваги і недоліки різних алгоритмів ранжування засобами розгалужених алгоритмів.
15. Алгоритми перевірки влучення в інтервал та його виключення.
16. Організація простого циклу за допомогою різних циклічних структур.
17. Класифікаційні ознаки простого циклу.
18. Простий цикл із лічильником.
19. Ітераційний цикл, його ознаки.
20. Поняття масиву. Характеристики масивів, різновиди масивів.
20. Організація циклу з переадресацією.
21. Класифікаційні ознаки циклу з переадресацією.
22. Організація циклу зі збереженням результатів обчислень.
22. Алгоритми заповнення одномірного масиву значеннями функції.
23. Алгоритм пошуку найбільшого (найменшого) елемента одномірного масиву.
24. Алгоритми циклічного зсуву елементів масиву вправо і вліво.
25. Алгоритм ранжування елементів одномірного масиву методом прямого вибору.
26. Алгоритм ранжування елементів одномірного масиву методом прямого обміну.
27. Алгоритм ранжування елементів одномірного масиву методом прямого включення.
28. Типові алгоритми обробки масиву: видалення, додавання і перестановка елементів.
29. Типові алгоритми обробки масиву: вибірка елементів за ознакою та обробка частини масиву.

30. Типові алгоритми обробки вектора: множення вектора на число; складання двох векторів; скалярний добуток двох векторів.
31. Організація циклів з накопиченням. Приклади використання для усіх типів циклічних алгоритмів.
32. Організація ітераційного циклу. Визначення суми нескінченного ряду.
33. Алгоритми обробки частини матриці щодо осей симетрії і щодо діагоналей.
34. Алгоритм транспонування матриці.
35. Типові алгоритми обробки матриць: складання двох матриць, множення матриці на вектор, множення матриці на матрицю.
36. Висхідне та низхідне структурне програмування.
37. Поняття модуля та підпрограми. Алгоритми підпрограм.
38. Процедури та функції.
39. Формальні та фактичні, глобальні та локальні параметри підпрограм.
40. Поняття про рекурсію.
41. Особливості організації та зображення рекурсивних алгоритмів.

### **Приклад екзаменаційного білету**

1. Алгоритми пошуку найбільшого (найменшого) із трьох та чотирьох чисел.
2. Ітераційний цикл, його ознаки.
3. Скласти таблицю імен змінних, алгоритм розв'язання задачі та виконати перевірку правильності алгоритму на конкретних значеннях.  
Обчислити значення квадратів парних чисел натурального ряду від  $K$  до  $M$ .

## 12. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ З ДИСЦИПЛІНИ

### 12.1 Основна навчальна література

1. Алгоритмізація обчислювальних процесів: навч. посіб. / Саїтгарєєв Н. Х., Котов І. А., Доценко І. О., Шаповалова Н. Н., Рибальченко О. Г., Білашенко С.В. Кривий Ріг, 2018. 380 с.
2. Основи алгоритмізації та програмування мовами C++, Visual Basic, Turbo Pascal: навч. посіб. / Азарян А. А., Карабут Н. О., Козикова Т. П., Рибальченко О. Г., Трачук А. А., Шаповалова Н. Н. Кривий Ріг, 2014. 308 с.

### 12.2 Додаткова навчальна література

1. Cormen Th.H., Leiserson Ch.E., Rivest R.L., Stein Cl. Introduction to Algorithms. 4th Edition. The MIT Press, 2022. 1312 p.
2. Aho A.V., Hopcroft J.E., Ullman J.D. Data Structures and Algorithms [Електронний ресурс] – Режим доступу до ресурсу: [http://lib.yasu.am/disciplines\\_bk/f095abf306249866fdbf8194ea788d2c.pdf](http://lib.yasu.am/disciplines_bk/f095abf306249866fdbf8194ea788d2c.pdf). 372 p.
3. Wirth N. Algorithms and Data Structures. [Електронний ресурс] – Режим доступу до ресурсу: [https://github.com/aforarup/interview/blob/master/Data Structures and Algorithm/Algorithm Books/Algorithms and Data Structures - Thin Book- Niklaus Wirth.pdf](https://github.com/aforarup/interview/blob/master/Data%20Structures%20and%20Algorithm/Algorithm%20Books/Algorithms%20and%20Data%20Structures%20-%20Thin%20Book-%20Niklaus%20Wirth.pdf). 288 p.

### 12.3. Нормативна та інструктивна література

1. ДСТУ ISO 5807:2016 Оброблення інформації. Символи та угоди щодо документації стосовно даних, програм та системних блок-схем, схем мережевих програм та схем системних ресурсів. Чинний від 10.10.2016. Вид. офіц. Київ : УкрНДНЦ, 2016. 43 с.
2. ДСТУ ISO/IEC/IEEE 26531:2015 Розробка систем і програмного забезпечення. Керування контентом для документування керування життєвим циклом продуктів, користувачів і послуг. Чинний від 01.01.2016. Вид. офіц. Київ : УкрНДНЦ, 2016. 64 с.

### 12.4. Методична література

1. Рибальченко О.Г., Білашенко С.В. Методичні вказівки до виконання лабораторних робіт з дисципліни «Алгоритмізація обчислювальних процесів» для студ. всіх форм навч. за спец. 121 «Інженерія програмного забезпечення». Кривий Ріг, 2023. 72 с.

## 13. ІНФОРМАЦІЙНІ РЕСУРСИ

До складу інформаційних ресурсів навчальної дисципліни входять:

1. Бібліотека Криворізького національного університету. URL: <http://lib.knu.edu.ua/> (дата звернення 10.06.2024).  
Internet-ресурси:
  1. Рибальченко О.Г. Алгоритмізація обчислювальних процесів: консп. лекцій. URL: <https://classroom.google.com/w/MTYzMzMxMDIwNTk3/t/all> (дата звернення 08.06.2024).
  2. MilanJovanovicTech. URL: <https://youtube.com/@MilanJovanovicTech> (дата звернення 10.06.2024).
  3. Raw Coding. URL: <https://youtube.com/@RawCoding> (дата звернення 10.06.2024).

## 14. ТЕРМІНОЛОГІЧНИЙ СЛОВНИК

**Алгоритм (Algorithm)** - is a set of instructions that describe the order of actions of the performer to achieve the result of solving the problem in a finite number of actions; a system of rules for the execution of a discrete process that achieves the set goal in a finite time. Block diagrams are often used to visualize algorithms .

**Блок-схема (Block scheme)** - Representation of an algorithm for solving or analyzing a problem using geometric elements (blocks) that denote operations, flow, data, etc. It is customary to mark the input and output data block with a parallelogram , the data calculation (processing) block with a rectangle , the decision-making block with a rhombus , and the beginning and end of the algorithm with an ellipse .

**Математична модель (Mathematical model)** - is a system of mathematical relationships that describe the process or phenomenon being studied . The mathematical model is important for such sciences as: economics , ecology , sociology , physics , chemistry , mechanics , computer science , biology , etc. Any mathematical tools can be used to create mathematical models - differential or integral equations, set theory , abstract algebra , mathematical logic , probability theory , graphs , etc. The process of creating a mathematical model is called mathematical modeling. This is the most general and most widely used research method in science, in particular, in cybernetics .

**Типи значень (Value types)** - a value type is a data type that directly stores its value in memory, rather than storing a reference to an object that contains the value. Examples of value types in C# include integral types (such as int, long, and byte), floating-point types (such as float and double), and the bool, char, and decimal types. Value types are typically smaller and more efficient than reference types, as they do not require memory allocation on the heap and do not need to be garbage collected. In C#, value types are passed by value, meaning that a copy of the value is passed to a method or assigned to a variable, rather than a reference to the original value. However, value types can also be boxed, which means that they can be wrapped in a reference type and treated as objects.

**Структура даних (Data structure)** - in programming and computer science, a data structure is a way of organizing data in computers . Often, a specific list of operations that can be performed on data organized into such a structure is associated with the data structure .The correct selection of data structures is extremely important for the effective functioning of the relevant algorithms for their processing. Well-constructed data structures allow you to optimize the use of machine time and computer memory to perform the most critical operations.

**Інструкція (Instruction)** - in computer programming, an instruction is a syntactic unit of an imperative programming language that indicates a certain action to be performed. A program written in this language is a sequence of instructions. An instruction can have internal components (for example, expressions).

**Умовний перехід (Conditional transition)** – is a construction of the programming language that allows you to perform various actions depending on the boolean value of the condition specified by the programmer. Most often, a conditional transition has two stages: in the first, some values that determine the transition condition are compared, and in the second, the transition itself is performed.

**Безумовний перехід (Unconditional transition)** - transition to a given point of the program without checking the fulfillment of any conditions. In many programming languages, this transition corresponds to a special goto instruction.

**Багатоваріантний вибір (Switch statement)** - also known as the selection instruction and the switch operator is a special type of programming language instruction that provides multidirectional (multiple) branching in the program. The name of the instruction may vary in different languages. This selection mechanism exists in most imperative programming languages.

**Цикл (Cycle)** - is a type of control structure in high-level programming languages, designed to organize repeated execution of a set of instructions (commands). A loop can also be any repeatedly executed sequence of commands, organized in any way (for example, using a conditional transition).

**Ітерація (Iteration)** - is a multi-meaning term that, depending on the context, can mean: repeated use of a mathematical operation (with changed data) when solving computational problems, which makes it possible to gradually approach the correct result or the result of multiple repetition of some mathematical operation.

**Ітераційний цикл (Iterative cycle)** - a loop operator for which the number of iterations of the loop body is unknown in advance. In iterative cycles, at each step of the calculations, the condition for achieving the desired result is successively approximated and checked. Exit from the iteration cycle is carried out in case of fulfillment of the given condition.

**Цикл з передумовою (A loop with a precondition)** - is a loop that is executed until some condition specified before its start is true. This condition is checked before the execution of the loop body begins, so the body may not be executed once (if the condition is initially false). In most procedural languages, programming is carried out using the instruction while, hence its second name is the while loop.

**Цикл з післяумовою (A loop with a postcondition)** - is a loop in which the condition is checked after the loop body is executed. It follows that the loop body is always executed at least once.

**Цикл з лічильником (A cycle with a counter)** - is a loop in which some variable changes its value from a given initial value to a final value in some increments, and for each value of that variable the body of the loop is executed once. In most procedural languages, programming is implemented with a statement that specifies a counter (called a "loop variable"), the required number of passes (or the limit value of the counter), and possibly the step by which the counter changes.

**Вкладені цикли (Nested loops)** - It is possible to create a loop inside the body of the second loop. Such a loop is called a nested loop. A nested loop relative to the loop in whose body it is nested will be called an inner loop, and conversely, a loop in the body of which there is a nested loop will be called outer relative to the nested loop. Inside a nested loop can be the next nested loop, forming the next level of nesting and so on. The number of nesting levels, as a rule, is not limited.

**Масив (Array)** - is an ordered set of a fixed number of elements of the same type, stored in sequentially located RAM cells, having a sequence number and a common name provided by the user.

**Характеристики масиву (Characteristics of the array)** - dimensionality — the number of element indices (one-dimensional, two-dimensional, ..., multidimensional); size is the total number of elements in the array; arrays can be numeric, character and other types; each language has its own rules for describing arrays.

**Рекурсія (Recursion)** - in computer science, recursion is a method of solving a computational problem in which the solution relies on the solutions of smaller cases of the same problem. Recursion solves such recursive problems using functions or procedures that call themselves. This approach can be applied to many problems, and recursion is one of the central ideas of computer science.

*Алгоритм сортування (A sorting algorithm)* - is an algorithm that solves the sorting problem , that is, arranges a linear list ( array ) of elements. For the sorting algorithm (as for any other modern algorithm), the main characteristics are: the time required to arrange an n-element array, the need for additional memory for sorting, stability — stable sorting does not change the relative arrangement of elements with the same keys.

*Алгоритм пошуку (A search algorithm)* - is an algorithm that solves a search problem , that is, finds information that is stored in a certain data structure . Data structures can be implemented using linked lists , arrays , search trees , hash tables , or other information storage methods. The direct search algorithm depends on the data structure for which it is implemented. Very often, the search algorithm includes special commands that specify the data structure.

## Додаток до робочої програми Робочий план

з дисципліни «Алгоритмізація обчислювальних процесів»  
для здобувачів вищої освіти першого (бакалаврського) рівня вищої освіти  
за спеціальністю 121 «Інженерія програмного забезпечення»

Вид навчальної роботи	Годин у семестрі/кредитів	Розподіл годин по тижнях																Вид підсумкового контролю
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Лекційні заняття	16	2 ПК	-	2 ПК	-	2 ПК	-	2 ПК	-	2 ПК	-	2 ПК	-	2 ПК	-	2 ПК	-	екзамен
Лабораторні заняття	32	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ЗМ	
Самостійна робота	72	4	5	4	5	4	5	4	5	4	5	4	5	4	5	4	5	
Всього годин/ кредитів	120/4	8	7	8	7	8	7	8	7	8	7	8	7	8	7	8	7	

Позначки:

ПК                   - поточний контроль  
ЗМ                   - складання змістових модулів