

**Чернівецький національний університет імені Юрія Федьковича**

(повне найменування закладу вищої освіти)

**факультет математики та інформатики**

(назва інституту/факультету)

**Кафедра**

**математичного моделювання**

(назва кафедри)

**СИЛАБУС**  
**навчальної дисципліни**

**Теорія алгоритмів**

(назва навчальної дисципліни)

**обов'язкова**

(вказати: обов'язкова / вибіркова )

**Освітньо-професійна програма** **Інформатика та математика**

(назва програми)

**Спеціальність** **014.09 – Середня освіта (інформатика)**

(вказати: код, назва)

**Галузь знань** **01 – Освіта/Педагогіка**

(вказати: шифр, назва)

**Рівень вищої освіти** **перший (бакалаврський)**

(вказати: перший бакалаврський/другий магістерський)

**факультет математики та інформатики**

(назва факультету/інституту, на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

**Мова навчання**

**українська**

(вказати: на якій мові читається дисципліна)

**Розробники:** **Караванова Тетяна Петрівна, асистент кафедри математичного моделювання**

(вказати авторів, їхні посади, наукові ступені, вчені звання)

**Профайл викладача** <http://matmod.fmi.org.ua/pro-kafedru/spivrobotnyky/karavanova-tetiana-petr%D1%96vna/>

**Контактний тел.**

0372584825

**E-mail:**

[t.karavanova@chnu.edu.ua](mailto:t.karavanova@chnu.edu.ua)

**Сторінка курсу в Moodle**

<https://moodle.chnu.edu.ua/course/view.php?id=3759>

**Консультації**

Консультації в аудиторно/дистанційному форматі

Очні/онлайн консультації: 10 годин

Онлайн-консультації: п'ятниця (другий навчальний тиждень) з 13.00 до 14.00.

### **1. Анотація дисципліни (призначення навчальної дисципліни).**

Теорія алгоритмів - один з фундаментальних розділів математики, що вивчає загальні властивості та закономірності алгоритмів і різноманітні формальні моделі їх подання.

Теорія алгоритмів розглядає такі задачі, як формалізація поняття «алгоритм» і дослідження формальних алгоритмічних систем; формальний доказ алгоритмічної нерозв'язності низки завдань; класифікацію задач, визначення і дослідження складності класів; асимптотичний аналіз складності алгоритмів; дослідження та аналіз рекурсивних алгоритмів; отримання явних функцій трудомісткості з метою порівняльного аналізу алгоритмів; розробка критеріїв порівняльної оцінки якості алгоритмів.

Коректний підбір структур даних є надзвичайно важливим для ефективного функціонування відповідних алгоритмів, що дозволяє оптимізувати використання машинного часу та пам'яті комп'ютера для виконання найкритичніших операцій.

Відомо, що ознайомившись з базовими алгоритмічними структурами — лінійними, розгалуженими та циклічними, можна побудувати алгоритм розв'язку практично будь-якої задачі. Частіше за все розв'язання складних алгоритмічних задач може бути реалізовано за допомогою повноперебірних алгоритмів.

Однак існують різні методи, які дозволяють оптимізувати такі задачі і розв'язати їх за більш короткий час. Методи розв'язування таких задач представлені в інформатиці різними розділами: теорія графів, лінійне та динамічне програмування тощо.

Метою даного курсу є як вироблення навичок коректного використання структур даних з подальшою оптимізацією розроблених алгоритмів, так і ознайомлення з базовими алгоритмами теорії графів, динамічного програмування, жадібними алгоритмами, алгоритмами обчислювальної геометрії. Ці теми зустрічаються як у математичних курсах, так і у розділах алгоритмізації. Але саме в алгоритмізації теоретичні базові знання набувають зовсім іншого «звучання»: їх ефективність можна оцінити на практиці у вигляді комп'ютерних програм.

Завданням курсу є опанування студентами базовими структурами даних та оптимізаційними алгоритмами, вміння реалізації їх у вигляді програм мовою програмування, набуття навичок тестування розроблених програмних кодів та оцінювання ефективності алгоритмів.

Завданням даного курсу є вироблення навичок коректного використання структур даних з подальшим використанням та оптимізацією розроблених алгоритмів.

### **2. Мета навчальної дисципліни:**

- розвиток логічного, аналітичного мислення та основних видів розумової діяльності: уміння використовувати індукцію, дедукцію, аналіз, синтез, робити висновки, узагальнення;
- формування теоретичної бази знань студентів щодо структур даних, побудови алгоритмів з використанням структур даних та їх реалізації мовою програмування;
- розвиток уміння розв'язувати алгоритмічні задачі, користуючись відомими теоретичними положеннями, математичним апаратом, літературою та комп'ютерною технікою;
- доведення вивчення курсу до творчого рівня;
- бачення студентами можливостей використання набутих знань у їх майбутній професії;
- інтеграція курсу з іншими дисциплінами, що викладаються в навчальному закладі.

### **Завдання:**

- розкриття значення структур даних для використання їх у задачах на графах, динамічного програмування, жадібних алгоритмів, обчислювальної геометрії;
- розуміння важливості коректного вибору структур даних для реалізації кожної окремої алгоритмічної задачі;

- отримання практичних навичок використання структур даних для побудови оптимізаційних алгоритмів на графах, динамічного програмування, жадібних алгоритмів, обчислювальної геометрії;
- отримання практичних навичок добору тестових даних для перевірки коректності роботи розроблених алгоритмів;
- забезпечення вивчення студентами змістової складової курсу;
- формування умінь щодо реалізації теоретичної бази знань під час розв'язування лабораторних завдань;
- вироблення навичок вивільнення навчального часу завдяки використанню можливостей середовищ програмування щодо налагодження програм;
- орієнтування студентів на можливість використання набутих практичних навичок у їх професійній діяльності;
- формування умінь та навичок здійснення самоконтролю.

### **3. Пререквізити.** «Програмування».

### **4. Результати навчання:**

#### **знати:**

- сутність поняття алгоритму, структур даних, пошуку інформації та її впорядкування, використання оптимізаційних методів для розв'язання задач на графах, динамічного програмування, жадібних алгоритмів, обчислювальної геометрії;
- базові структури даних;
- фундаментальні алгоритми з використанням структур даних: алгоритми на графах, динамічне програмування, жадібні алгоритми, обчислювальна геометрія;
- використання базових оптимізаційних алгоритмів;
- основи алгоритмізації та програмування;
- загальні принципи розв'язування алгоритмічних задач: постановка задачі, побудова алгоритму, реалізація алгоритму мовою програмування, тестування реалізованого алгоритму;
- методи розв'язання алгоритмічних задач.

#### **вміти:**

- застосовувати теоретичні знання щодо структур даних та фундаментальних оптимізаційних алгоритмів для розв'язування практичних завдань;
- використовувати навички роботи з інтегрованим середовищем програмування;
- аналізувати відомі методи побудови алгоритмів та визначати найоптимальніші з них для розв'язування конкретної задачі;
- реалізовувати побудовані алгоритми мовою програмування;
- розробляти власні тести для перевірки коректності розроблених алгоритмів;
- тестувати розроблені алгоритми;
- використовувати навички техніки програмування.

ЗК 3. Здатність оволодівати сучасними знаннями, розуміти предметну галузь та сфери професійної діяльності.

ЗК 4. Здатність до професійного використання інформаційно комунікаційних технологій.

ЗК 6. Здатність оцінювати та забезпечувати якість виконуваних робіт.

ФК 1. Здатність до виявлення, постановки та вирішення професійних проблем у поєднанні математичних інструментів для опису природних явищ.

ФК 5. Здатність застосовувати в професійній діяльності сучасні мови програмування і мови баз даних, системи автоматизації проектування, електронні бібліотеки, мережеві технології, бібліотеки і пакети програм, сучасні професійні стандарти.

ФК 6. Здатність професійно вирішувати завдання виробничої і науково-педагогічної діяльності з урахуванням сучасних досягнень науки і техніки, включаючи: розробку алгоритмічних і програмних рішень в області системного і прикладного програмування; розробку інформаційних моделей за тематикою виконуваних досліджень; створення інформаційних ресурсів глобальних мереж, освітнього контенту, прикладних баз даних; розробку тестів і засобів тестування.

ФК 7. Здатність самостійно здобувати за допомогою ІТ і використовувати в практичній діяльності нові знання і вміння, в тому числі в нових галузях знань, безпосередньо не пов'язаних зі сферою діяльності, розширювати і поглиблювати своє світосприйняття.

ПРН2. Здобувати систематичні знання в галузі освіти, аналізувати проблеми з точки зору сучасних наукових парадигм, осмислювати і робити обґрунтовані висновки з наукової і навчальної літератури та результатів експериментів

ПРН 3. Використовувати технології та інструментарії пошукових систем, методи інтелектуального аналізу даних і текстів, здійснювати опрацювання, інтерпретацію та узагальнення даних. Демонструвати уміння і навички роботи з науково-методичною літературою та періодичними виданнями з метою включення до занять інформації про новітні досягнення в галузі сучасних інформаційних технологій, методів і засобів навчання.

ПРН 4. Володіти сучасними методами ефективного доступу до інформації, її збору, систематизації та збереження, використовувати методи ідентифікації та класифікації інформації на базі нових інформаційних технологій за допомогою програмних технічних засобів, локальних і глобальних комп'ютерних мереж.

ПРН 5. Вміти розв'язувати задачі з інформаційних технологій та програмування різного рівня складності та формувати відповідні уміння користуючись відомими теоретичними положеннями, математичним апаратом, літературою та комп'ютерною технікою в здобувачів освіти.

ПРН 6. Формувати необхідні вміння та навички підготовки учнів для участі в олімпіадах, конкурсах, турнірах, науково-практичних конференціях, конкурсах-захистах науково-дослідницьких робіт різного рівня та інших інтелектуальних змаганнях.

ПРН 7. Демонструвати вміння та навички розробки інтерактивних вебсторінок для локальних комп'ютерних мереж та мережі Internet, використовуючи текстові, графічні та html-редактори. Володіти знаннями про основні види інформаційних систем та інструментальні засоби їх розробки.

## 5. Опис навчальної дисципліни

### 5.1. Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	2(3)	4(6)	4	120	15	–	–	30	75	–	Екзамен
Заочна	3	6	4	120	6	4	–	10	100	–	Екзамен

## 5.2. Дидактична карта навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
<b>Змістовий модуль 1. Базові структури даних та алгоритми обробки їх елементів</b>												
Тема 1. Теорія алгоритмів. Основні поняття.	6	1				5	6					6
Тема 2. Базові структури даних. Проста змінна. Масив. Стек. Черга.	11	2		4		5	11	1	2	2		6
Тема 3. Динамічна структура даних зв'язний список	9	2		2		5	9	1		2		6
Тема 4. Древа. Бінарні древа	11	2		4		5	11	1		1		9
Тема 5. Хеш-функції та хеш-таблиці	5					5	5					5
Тема 6. Пошукові алгоритми на деревах та у мережах.	9			4		5	9					9
Разом за ЗМ 1	51	7		14		30	51	3	2	5		41
<b>Змістовий модуль 2. Оптимізаційні алгоритми з використанням структур даних</b>												
Тема 7. Основні поняття теорії графів. Пошук в глибину та в ширину.	9	2		2		5	9	1		2		6
Тема 8. Ейлерів та Гамільтонів графи.	10	1		4		5	10					10
Тема 9. Остовне дерево. Мінімальне остовне дерево.	5					5	5					5
Тема 10. Пошук найкоротшого шляху між двома вершинами у зваженому графі.	10	1		4		5	10	1		2		7
Тема 11. Топологічне сортування.	5					5	5					5

Тема 12. Основи динамічного програмування. Базові задачі динамічного програмування.	7	2			5	7	1				6
Тема 13. Жадібні алгоритми.	7			2	5	7			1		6
Тема 14. Основні поняття обчислювальної геометрії. Базові алгоритми обчислювальної геометрії.	7	2			5	7		2			5
Тема 15. Побудова опуклої оболонки.	9			4	5	9					9
Разом за ЗМ2	69	8		16	45	69	3	2	5		59
<b>Усього годин</b>	120	15		30	75	120	6	4	10		100

### 5.3. Зміст завдань для самостійної роботи

№ з/п	Назва теми
1.	Створення алгоритму. Оцінка складності та аналіз ефективності алгоритмів
2.	Поняття про структури даних, обробка інформації. Стек. Черга.
3.	Динамічний розподіл пам'яті. Зв'язний список.
4.	Дерево як структура даних.
5.	Пошукові алгоритми на деревах та у мережах.
6.	Основні поняття теорії графів. Застосування алгоритмів на графах для розв'язування алгоритмічних задач.
7.	Основні поняття динамічного програмування та жадібних алгоритмів. Застосування базових алгоритмів динамічного програмування та жадібних алгоритмів для розв'язування алгоритмічних задач.
8.	Основні поняття обчислювальної геометрії. Застосування базових алгоритмів обчислювальної геометрії для розв'язування алгоритмічних задач.

### Тематика індивідуальних завдань

1. Структура даних дек: опис структури, алгоритми опрацювання елементів структури, реалізація мовою програмування.
2. Використання структури даних «зв'язний список» та «бінарне дерево» для розв'язування алгоритмічних задач.
3. Використання структур даних «хеш-таблиця» для розв'язування алгоритмічних задач.
4. Використання графових алгоритмів для розв'язування алгоритмічних задач.
5. Використання алгоритмів динамічного програмування та жадібних алгоритмів для розв'язування алгоритмічних задач.
6. Використання алгоритмів обчислювальної геометрії для розв'язування алгоритмічних задач.

### 6. Система контролю та оцінювання

Поточний контроль знань відбувається впродовж семестру шляхом усного та письмового опитування теоретичних основ теми, виконання лабораторних завдань, а також у вигляді контрольних робіт, тестів та індивідуальних завдань.

Студенти можуть отримати додаткові бали (до 10) після виконання індивідуальних навчально-дослідницьких завдань. Ці завдання видає та перевіряє лектор. Додаткові бали враховуватимуться під час здачі іспиту.

Теоретичне опитування може здійснюватись на лекціях. Студент, який повно й вичерпно відповів на питання лектора, отримує додатково 1 бал.

На підсумковому модулі (іспиті) студентам пропонуються білети, які містять три завдання. Два з них – теоретичні (кожне оцінюється в 10 балів), третє завдання передбачає розв'язання задачі на одну з тем «Структури даних», «Алгоритми на графах», «Алгоритми динамічного програмування», «Жадібні алгоритми», «Алгоритми обчислювальної геометрії» яке також оцінюється у 10 балів.

Диференціація оцінювання відповідей студентів здійснюється за таким алгоритмом: 100% балів - студент обирає усі правильні відповіді; 80% балів - студент обирає більшість правильних відповідей; 60% балів - студент обирає біля половини правильних відповідей; 40% балів - студент обирає менше половини правильних відповідей; 20% балів - студент не обирає практично жодної правильної відповіді.

### Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	<b>A</b>	відмінно	зараховано
80-89	<b>B</b>	добре	
70-79	<b>C</b>		
60-79	<b>D</b>	задовільно	
50-59	<b>E</b>		
35-49	<b>FX</b>	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	<b>F</b>	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

### Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота								Підсумковий тест (іспит)	Сума
Змістовий модуль 1				Змістовий модуль 2					
T1	T2	T3	T4-T6	T7	T8-T10	T12-T13	T14-T15	30	100
6	10	9	10	6	10	9	10		

### Методичне забезпечення

1. Караванова Т.П. Основи алгоритмізації та програмування: 777 задач з рекомендаціями та прикладами: Навч. посіб. Доп. та випр. – К.: Генеза, 2012. – 288 с.: іл.
2. Караванова Т.П. Методика розв'язування алгоритмічних задач. Основи алгоритмізації та програмування: Навчально-методичний посібник для вчителів/Т.П.Караванова. – Кам'янець-Подільський: Аксіома, 2013. – 460 с.

### 7. Рекомендована література Основа

1. Ахо Альфред В., Хопкрофт Джон, Ульман Джеффри Д. Структуры данных и алгоритмы. : Пер. с англ. : Уч. пос. — М.: Издательский дом «Вильямс», 2000. — 384 с.
2. Вирт Н. Алгоритмы и структуры данных: Пер. с англ. — 2-е изд., испр. — СПб.: Невский Диалект, 2001. — 352 с.: ил.
3. Кнут Дональд Эрвин Искусство программирования, т.3. Сортировка и поиск., 2-е изд.: Пер. с англ. — М.: Издательский дом «Вильямс», 2004. — 832 с.
4. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. — М.: МЦНМО, 2001. — 960 с.
5. Макконел Дж. Анализ алгоритмов. Вводный курс. — М.: Техносфера, 2002. — 304 с.

### Допоміжна

1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов.— М.: Мир, 1979.
2. Караванова Т.П. Методипобудовиалгоритмів та їханаліз: необчислювальніалгоритми: Навч. посіб. – К.: Генеза, 2007. – 224 с.: іл.
3. Караванова Т.П. Методипобудовиалгоритмів та їханаліз: обчислювальніалгоритми: Навч. посіб. – К.: Генеза, 2008. – 336 с.: іл.
4. Окулов С.М. Основы программирования. — М.: ЮНИМЕДИАСТАЙЛ, 2002. — 424 с.

### 7. Інформаційні ресурси

1. Структура даних [http://uk.wikipedia.org/wiki/Структура\\_даних](http://uk.wikipedia.org/wiki/Структура_даних)
2. АксёноваЕ.А., Соколов А.В. Алгоритмы и структурыданных на С++<http://window.edu.ru/library/pdf2txt/576/63576/33770>
3. Авдошин С. М. Алгоритмы и структурыданных <http://se.hse.ru/11003945/aisd>
4. Алгоритмы и структурыданных<http://squadette.ru/blog/2012/01/19/high-scalability-2/>