

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ
УНІВЕРСИТЕТ**
СИСТЕМА УПРАВЛІННЯ ЯКІСТЮ
Система забезпечення якості освітньої діяльності та якості вищої освіти
сертифікована на відповідність ДСТУ ISO 9001:2015 / ISO 9001:2015

Кафедра цифрової економіки та системного аналізу

ЗАТВЕРДЖЕНО

вченою радою
(поступило від «21 2021 р.)
Ректор


А. А. Мазаракі

**АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ/
ALGORITHMIZATION AND PROGRAMMING**

**РОБОЧА ПРОГРАМА/
COURSE OUTLINE**

освітній ступінь	бакалавр	/ bachelor
галузь знань	12 Інформаційні технології	/ Information Technologies
спеціальності	122 Комп'ютерні науки	/ Computer Sciences
	126 Інформаційні системи та технології	/ Information Systems And Technologies

КИЇВ-2020

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

Автори: В.В.Кулаженко канд.екон. наук доц.
О.Ф. Кузнєцов, ст. викл.

Робочу програму обговорено та схвалено на засіданні кафедри цифрової економіки та системного аналізу від «21» грудня 2020р., протокол № 5

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

**АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ/
ALGORITHMIZATION AND PROGRAMMING**

**РОБОЧА ПРОГРАМА/
COURSE OUTLINE**

освітній ступінь	бакалавр	/ bachelor
галузь знань	12 Інформаційні технології	/ Information Technologies
спеціальності	122 Комп'ютерні науки	/ Computer Sciences
	126 Інформаційні системи та технології	/ Information Systems And Technologies

1. Структура дисципліни та розподіл годин за темами (тематичний план)

№ ро-зді-лу	№ те-ми	Назва теми	Розподіл навчального часу				
			Всього Год/кред	Лекції	ЛР	СР С	Форми контр.
1	Основи програмування мовою С		124	24	42	58	
	1.1.	Введення у дисципліну «Алгоритмізація та програмування»	4	2		2	
	1.2.	Огляд можливостей мови програмування С/С++. Знайомство з середовищем розробки програм	8	2	2	4	К,Т
	1.3.	Основні поняття мови програмування. Базові типи, константи, змінні, оператори, вирази	12	2	4	6	К, Із
	1.4.	Керування виконанням програми	12	2	4	6	К,Із
	1.5.	Функції користувача та класи пам'яті	12	2	6	4	Із
	1.6.	Вказівники та адресна арифметика	14	2	4	8	Із
	1.7.	Складені типи даних: масиви	16	2	6	8	Із
	1.8.	Складені типи даних: структури, об'єднання, перерахування	14	2	4	8	Із
	1.9.	Алгоритми на масивах	12	2	4	6	
	1.10	Директиви препроцесора, макророзстановка, умовна компіляція	6	2	2	2	К,Т
	1.11	Функції стандартної бібліотеки	8	2	2	4	Із
1.12	Принципи аналізу алгоритмів	8	2	2	4	К	
2	Структури даних та алгоритми		56	10	26	20	
	2.1.	Огляд класичних алгоритмів для абстрактних типів даних	4	2		2	К
	2.2.	Організація списків та їх обробка	12	2	6	4	Із
	2.3.	Стеки	12	2	6	4	Із

№ ро-з-ді-лу	№ те-ми	Назва теми	Розподіл навчального часу				
			Всього Год/кред	Лекції	ЛР	СР С	Форми контр.
	2.4.	Черги	12	2	6	4	Кр.
	2.5.	Деревовидні структури	16	2	8	6	Із
	Разом за 1 семестр		180/3	34	68	78	
	Підсумковий контроль - екзамен						
3.	С++ та об'єктно-орієнтоване програмування		72	12	18	42	
	3.1.	Особливості програмування у С++	6	2		4	К,Т
	3.2.	Основні поняття та властивості об'єктно-орієнтованого програмування (ООП). Класи С++	10	2	2	6	К,Т
	3.3.	Структура та ієрархія класів, наслідування, поліморфізм	14	2	4	8	Із
	3.4.	Перевантаження операторів у С++. Шаблони та шаблонні функції.	14	2	4	8	Із
	3.5.	Створення власних класів на основі структур для реалізації списків, стеків та черг.	10	2	4	4	Із
	3.6.	Створення власних класів та рекурсивних функцій у віконних додатках оснований на проектах АРІ	18	2	4	12	Із
	3.1	Тема. Впорядкування масивів та списків	8	2	2	4	Із
	3.2.	Задачі пошуку	8	2	2	4	Із
	3.3.	Пошук та впорядкування у деревовидних структурах	8	2	2	4	Із
4.	Алгоритми впорядкування та пошуку		24	6	6	12	
	4.1.	Бінарні дерева та вирази	8	2	2	4	К,Із
	4.2.	Прості алгоритми впорядкування масивів та списків	12	2	4	6	Із
	4.3.	Алгоритми швидкого впорядкування	12	2	4	6	Із

№ ро-з-ді-лу	№ те-ми	Назва теми	Розподіл навчального часу				
			Всього Год/кред	Лекції	ЛР	СР С	Форми контр.
	4.4.	Черги по пріоритетах та пірамі- дальне сортування	16	2	6	8	Із
	4.5.	Задачі пошуку	14	2	4	8	Із
	4.6.	Пошук методом хешування	16	2	6	8	Із
	4.7.	Пошук та впорядкування у деревовидних структурах	14	2	4	6	Із
	4.8.	Зовнішній пошук у деревовидних структурах	8	2	4	4	Із
Разом за 2 семестр			180/3	28	56	96	
Підсумковий контроль - екзамен							
Разом по дисципліні			360/6	62	124	174	

К - конспект
Т- тестування

Із - індивідуальне завдання;
Кр - контрольна робота.

2. Тематика та зміст лекційних лабораторних занять та самостійної роботи студентів

Результати навчання	Навчальна діяльність	Кількість годин
Розділ 1. Основи програмування мовою С		
Знати: Основні поняття теорії алгоритмів. Вміти: Створити графічне відображення простих алгоритмів та дати оціночний аналіз кількості кроків для реалізації.	Тема 1.1. Введення у дисципліну «Алгоритмізація та програмування» <i>План лекції:</i> 1. Поняття алгоритму. 2. Алгоритм Евкліда. 3. Властивості алгоритмів. 4. Характеристики алгоритмів. 5. Формальне визначення алгоритму. Список рекомендованих джерел Основний: 2,3,4. Додатковий: 7,8. Інтернет джерела: 11,13,14.	2
	<u>Самостійна робота студента</u> 1. Вивчити засоби формального визначення алгоритмів. 2. Навчитися застосовувати графічні засоби відображення алгоритмів. 3. Вивчити шляхи формалізації алгоритмів для різних областей застосування та функції реалізації.	2
Знати: основні засоби створення та редагування програмного коду в середовищі MS Visual Studio Вміти:	Тема 1.2. Огляд можливостей мови програмування С/С++. Знайомство з середовищем розробки програм <i>План лекції:</i> 1. Класифікація мов програмування та напрями їх розвитку. 2. Види програмного забезпечення для розробки програмних продуктів мовою С++. Знайомство з інструментальними засобами Microsoft Visual Studio. 3. Знайомство з інтерфейсом інструментального засобу Microsoft Visual C++.	2

Результати навчання	Навчальна діяльність	Кількість годин
створити проект Console Application, користуватись редактором коду, засобами компіляції та налагодження програми.	<p>4. Поняття проекту. Огляд майстрів для створення проектів різних видів у Microsoft Visual C++.</p> <p>5. Приклад створення першого проекту для консольної прикладної програми.</p> <p>6. Основні структурні компоненти проекту Visual Studio, функція main(). Поняття програмного модуля.</p> <p>7. Компіляція та запуск програм на виконання. Етап налагодження програми.</p> <p>8. Основні види помилок. Виправлення помилок. Інтерактивні засоби Microsoft Visual C++ для налагодження програм</p> <p>Список рекомендованих джерел Основний: 1,2, 3. Додатковий: 7,8. Інтернет джерела: 10,13,14</p> <p><u>Лабораторне заняття №1 Ознайомлення з середовищем розробки програм</u></p> <p>1. Ознайомлення з інтегральним середовищем розробки програм Microsoft Visual C++.</p> <p>2. Знайомство з інструментальними засобами MS Visual Studio.</p> <p>3. Аналіз майстрів для створення проектів різних видів у MS Visual C++.</p> <p>4. Основні структурні компоненти програми C, функція main().</p> <p><u>Самостійна робота студента</u></p> <p>1. Знати класифікацію мов програмування та місце C++ і її особливості серед інших мов програмування.</p> <p>2. Вивчити види програмного забезпечення для розробки програмних продуктів на мові C/C++.</p> <p>3. Опанувати інтерфейс розробки програм: редактор коду, компіляція та налагодження.</p>	<p>2</p> <p>4</p>
Знати: Базові типи	<u>Тема 1.3. Основні поняття мови програмування. Базові типи, константи, змінні, операції, вирази.</u>	2

Результати навчання	Навчальна діяльність	Кількість годин
<p>даних, прості оператори та правила виклику системних функцій.</p> <p>Вміти: написати код для реалізації лінійного алгоритму. Розподіляти код реалізації алгоритму, щодо компонент проекту консольного додатку.</p>	<p><i>План лекції:</i></p> <ol style="list-style-type: none"> 1. Алфавіт мови програмування. 2. Поняття лексеми. Типи лексем. 3. Поняття типу. Категорії типів даних. Оператор sizeof. 4. Поняття змінної. Оголошення та ініціалізація змінних. 5. Операнди та операції. Види операцій. 6. Пріоритети операцій. Перетворення типів у виразах. 7. Створення першого проекту для консольної прикладної програми. 8. Огляд основних структурних компонентів проекту на С. <p>Список рекомендованих джерел</p> <p>Основний: 2,3,4.</p> <p>Додатковий: 7.</p> <p>Інтернет джерела: 12,13,15</p> <p><u>Лабораторне заняття №2 Програма з найпростішою структурою</u></p> <ol style="list-style-type: none"> 1. Створення першого проекту для консольної прикладної програми. 2. Додавання в проект програмного коду для реалізації алгоритму розв'язання задачі реалізації лінійних алгоритмів. 3. Ознайомлення зі структурними компонентами програми та лексичними конструкціями мови. 4. Освоєння технології компіляції програми, налагодження та виконання програми. <p><u>Самостійна робота студента</u></p> <ol style="list-style-type: none"> 1. Ознайомитись з інструментальними засобами MS Visual Studio. 2. Спробувати засоби налаштування вікон розробки програмного коду. 3. Знати призначення майстрів для створення проектів різних видів у MS Visual C ++. 4. Вивчити основні структурні компоненти про- 	<p>4</p> <p>6</p>

Результати навчання	Навчальна діяльність	Кількість годин
	грами C++, функція main() та інші. 5. Вміти створити файли проекту. 6. Знати основні види помилок та методи налагодження програми.	
<p>Знати: Структуру програмного модулю та складних операторів. Розуміти яким чином працюють оператори C++ та пріоритети їх виконання</p> <p>Вміти: Написати програму, що оснований на розгалуженнях з використанням операторів циклу.</p>	<p><u>Тема 1.4. Керування виконанням програми</u> <i>План лекції:</i></p> <ol style="list-style-type: none"> 1. Поняття складеного оператора. 2. Оператори розгалуження: if ... else ; if ... else if ... else; оператор-перемикач switch. 3. Організація циклів: оператор while ; оператор for ; оператор do while. 4. Використання операторів break та continue. Оператор переходу goto. 5. Побудова складених операторів розгалуження та циклів; <p>Список рекомендованих джерел Основний: 1,2, 3. Додатковий: 6, 7 Інтернет джерела: 12,13,15.</p> <p><u>Лабораторне заняття №3 Використання керуючих конструкцій</u></p> <ol style="list-style-type: none"> 1. Ознайомлення з конструкціями мови C++, що використовуються для реалізації розгалужених алгоритмів. 2. Ознайомлення з конструкціями мови C++, що призначені для реалізації циклічних процесів. 3. Вивчення синтаксису для інкрементних та декрементних операцій. 4. Використання операторів для дострокового виходу з циклів. 5. Реалізація алгоритмів для розв'язання елементарних задач про геометричні фігури, обчислення на основі ланцюгових дробів, рекурентних співвідношень, генератора псевдо випадкових чисел. 6. Запуск налагоджувача. Аналіз виконання програми по кроках. 	<p>2</p> <p>4</p>

Результати навчання	Навчальна діяльність	Кількість годин
у створені функції.	<p>3. Створення функцій зі змінним числом аргументів.</p> <p>4. Освоєння виклику функцій, передачі їм фактичних параметрів та одержання значень, що повертаються функціями.</p> <p>5. Ознайомлення з рекурсивними функціями та особливостями їх використання.</p> <p>6. Створення простих рекурсивних функцій.</p> <p><u>Самостійна робота студента</u></p> <p>1. Ознайомлення з класами задач, розв'язання яких потребує використання алгоритмів розгалуження та циклів.</p> <p>2. Подання розгалужених та циклічних алгоритмів у вигляді блок-схем.</p> <p>3. Синтаксис операторів для реалізації розгалужених та циклічних алгоритмів з застосуванням функцій реалізації окремих циклів.</p> <p>4. Оголошення та ініціалізація статичних та глобальних змінних.</p>	4
<p>Знати: правила застосування вказівників для створення даних в динамічній області пам'яті.</p> <p>Вміти: Застосовувати вказівники для передачі параметрів функціям та створю-</p>	<p><u>Тема 1.6. Вказівники та адресна арифметика</u></p> <p><i>План лекції:</i></p> <p>1. Поняття та оголошення вказівника. Використання модифікаторів в оголошеннях вказівників.</p> <p>2. Вказівники та адреси. Унарні операції одержання адреси (&) та розкриття посилання вказівника (*).</p> <p>3. Інші операції з вказівниками,</p> <p>4. Вказівники та аргументи функцій. Використання вказівників для передачі аргументів різного типу.</p> <p>5. Вказівники на вказівники. Вказівники на функції.</p> <p>Список рекомендованих джерел</p> <p>Основний: 1,2, 3.</p> <p>Додатковий:7.</p> <p>Інтернет джерела: 10,12.</p>	2

Результати навчання	Навчальна діяльність	Кількість годин
<p>вати вказівники на функції.</p>	<p><u>Лабораторне заняття №5 Вказівники та адреса арифметика</u></p> <ol style="list-style-type: none"> 1.Засвоєння поняття вказівника, синтаксису та семантики операцій з вказівниками. 2.Освоєння технології передачі адрес змінних як фактичних параметрів функцій. 3.Засвоєння процесу створення вказівника та його ініціалізації. <p><u>Самостійна робота студента</u></p> <ol style="list-style-type: none"> 1.Вивчити поняття оголошення вказівника. 2.Операції з вказівниками. 3.Динамічний розподіл пам'яті. 4.Передача адреси у функції. 5.Вказівники типу void. 4.Вказівники на вказівники 	<p>4</p> <p>8</p>
<p>Знати: Як створювати нові статичні та динамічні масиви.</p> <p>Вміти: Створювати та обробляти масиви різних типів даних. Вміти застосовувати доступ до даних за допомогою індексів та вказівників. Передавати ма-</p>	<p><u>Тема 1.7.Складені типи даних: масиви</u></p> <p><u>План лекції:</u></p> <ol style="list-style-type: none"> 1.Поняття масиву. Оголошення та ініціалізація масивів. 2.Багатовимірні масиви. Розміщення в пам'яті. 3.Динамічний розподіл пам'яті. Функції malloc() та free(). Оператори new та delete. 4.Вказівники на багатовимірні масиви. Масиви вказівників. 5.Приклади програм для роботи масивами. <p>Список рекомендованих джерел</p> <p>Основний: 1,3,4.</p> <p>Додатковий: 7,8.</p> <p>Інтернет: 12,15.</p> <p><u>Лабораторне заняття №6 Складені типи даних масиви</u></p> <ol style="list-style-type: none"> 1.Одержання практичних навичок використання функцій для динамічного розподілу пам'яті. 2.Використання операторів циклів для обробки елементів масивів. 3.Реалізація операцій з алгебраїчними об'єктами: 	<p>2</p> <p>6</p>

Результати навчання	Навчальна діяльність	Кількість годин
	<p>та ->.</p> <p>3. Вивчення обробки масивів, елементами яких є структури.</p> <p>4. Освоєння основних операцій зі структурами та їх полями.</p> <p>5. Реалізація елементарних операцій обробки простої бази даних.</p> <p><u>Самостійна робота студента</u></p> <p>1. Описування структур.</p> <p>2. Доступ до полів структури.</p> <p>3. Множини (enum),</p> <p>4. Об'єднання (union).</p> <p>5. Рекурсивні оголошення структур.</p>	8
<p>Знати: Принципи створення масивів структур та правила обробки таких даних.</p> <p>Вміти: Самостійно створювати масиви структур в динамічній області пам'яті та обробляти масиви різного типу.</p>	<p><u>Тема 1.9. Алгоритми на масивах</u></p> <p><i>План лекції:</i></p> <p>1. Алгоритми створення простої БД з масиву структур.</p> <p>2. Обчислення відстані від точок з застосуванням структури Point.</p> <p>3. Алгоритм отримання простих чисел з числового ряду – «Решето Ератосфена».</p> <p>4. Імітація підкидання монети</p> <p>Список рекомендованих джерел</p> <p>Основний: 1,2,5.</p> <p>Додатковий: 6,7.</p> <p>Інтернет джерела: 10,14,15</p> <p><i>Лабораторне заняття №8 Алгоритми на масивах.</i></p> <p>1. Програмування алгоритму розрахунку геометричних даних різних фігур з допомогою структур типу .Point.</p> <p>2. Реалізація програми отримання простих чисел числового ряду за допомогою алгоритму «Решето Ератосфена».</p> <p>3. Протестувати програму для масиву на 10^3 10^4 та більше.</p> <p>4. Зробити аналіз часу, що необхідно для реалі-</p>	2 4

Результати навчання	Навчальна діяльність	Кількість годин
<p>Знати: Оператори умовної компіляції, правила їх застосування та дію. Вміти: Самостійно написати власні оператори управління компілятором та макropідстановки.</p>	<p>зації алгоритму. <u>Самостійна робота студента</u> 1.Змінити програму «Решето Ератосфена» для масиву символів. Протестувати програму для різної довжини масиву. 2.Створити програми реалізації розрахунку максимального, мінімального та середнього значення у одновимірному масиві. 3.Написати програму створення та ініціалізації двовимірною та тривимірною масиву випадковими числами. 4.Написати програми пошуку максимального, мінімального та середнього у двовимірному та тривимірному масиві.</p>	6
	<p><u>Тема 1.10. Директиви препроцесора, макropідстановка, умовна компіляція</u> <i>План лекції:</i> 1. Директиви препроцесора як засіб налаштування процесу компіляції. 2. Директиви розгалуження компіляції. 3. Макropідстановка та застосування власних макросів. 4. Засоби макropідстановки для запобігання повторній компіляції коду. Список рекомендованих джерел Основний: 2,3,4 Додатковий: 7. Інтернет джерела: 14,15. <i>Лабораторне заняття №9 Директиви препроцесора, макropідстановка, умовна компіляція</i> 1. В заготовочному файлі проекту розмістити прототипи функцій, що мають бути викликані у функції main. 2. Написати засобами команд пре процесора код, що запобігає повторній компіляції функцій та визначень проекту. 3. Навчитися створювати власні макроси та ско-</p>	2

Результати навчання	Навчальна діяльність	Кількість годин
	<p>мпілювати їх, а результати передивитися з допомогою налагоджувача.</p> <p><u>Самостійна робота студента</u></p> <ol style="list-style-type: none"> 1. Вивчити директиви препроцесора які створені в проекті засобом Application Wizard. 2. Вивчити структуру та з'ясувати застосування файлу заголовків StdAfx.h. 3. Створити власний файл заголовків. Включити туди структури для функцій проекту. 4. Навчитися виконувати компіляцію з двома файлами заголовків. 	2
<p>Знати: Функції стандартної бібліотеки C. Правила застосування функцій стандартної бібліотеки обробки символної інформації.</p> <p>Вміти: Застосувати функції введення/виведення та стандартними вбудованими функціями</p>	<p><u>Тема 1.11. Функції стандартної бібліотеки, обробка символної інформації, функції введення/виведення</u></p> <p><i>План лекції:</i></p> <ol style="list-style-type: none"> 1. Рядки C++ як масив літер. 2. Масиви рядків і масиви вказівників на char. 3. Функції стандартної бібліотеки для роботи з рядками 4. Огляд функцій стандартної бібліотеки для організації введення/виведення. Головний файл stdio.h. 5. Функції getchar() та putchar()- Приклади функцій користувача для введення/виведення рядків. 6. Форматне виведення: функція printf(). 7. Форматне введення: функція scanf(). 8. Доступ до файлів <p>Список рекомендованих джерел</p> <p>Основний: 2,3, 5.</p> <p>Додатковий: 8</p> <p>Інтернет джерела: 9,11,14.</p> <p><u>Лабораторне заняття №10 Функції стандартної бібліотеки, обробка символної інформації, функції введення/виведення</u></p> <ol style="list-style-type: none"> 1. Одержання практичних навичок щодо використання вказівників для обробки рядків символів. 	2

Результати навчання	Навчальна діяльність	Кількість годин
	<p>2. Ознайомлення з функціями стандартної бібліотеки для: копіювання рядків; порівняння рядків; конкатенації рядків; пошуку символів та лексем у рядках.</p> <p>3. Виділення у текстах деяких лексем</p> <p>4. Вивчення основних функцій стандартної бібліотеки для організації введення/виведення інформації.</p> <p>5. Ознайомлення з функціями для одержання доступу до файлів.</p> <p>6. Використання функцій для відкриття та закриття файлів.</p> <p>7. Використання функцій для позиціонованого виведення</p> <p><u>Самостійна робота студента</u></p> <p>1. Функції для обробки рядків.</p> <p>2. Функції введення/виведення консольного терміналу.</p> <p>3. Форматне введення/виведення.</p> <p>4. Функції для перевірки класів символів.</p> <p>5. Математичні функції.</p> <p>6. Функції дати та часу.</p> <p>7. Функції загального призначення</p>	4
<p>Знати: Основні принципи аналізу алгоритмів та функції які визначають якість алгоритму.</p> <p>Вміти: Виконати емпіричний аналіз простих алго-</p>	<p><u>Тема 1.12. Принципи аналізу алгоритмів</u></p> <p><i>План лекції:</i></p> <p>1. Розробка та емпіричний аналіз.</p> <p>2. Ріст функцій. Q – нотація.</p> <p>3. Приклади алгоритмічного аналізу.</p> <p>4. Гарантії передбачення та обмеження.</p> <p>Список рекомендованих джерел</p> <p>Основний: 3, 5.</p> <p>Додатковий: 7,8.</p> <p>Інтернет джерела: 10,15.</p> <p><u>Лабораторне заняття №11 Принципи аналізу алгоритмів</u></p> <p>1. Створити програму що дозволяє проаналізувати час виконання деяких раніше створених</p>	2

Результати навчання	Навчальна діяльність	Кількість годин
<p>ритмів. Вибрати функцію що найкращим чином відповідає аналізу вибраного алгоритму.</p>	<p><i>програм для різних масивів даних.</i> <i>2. Написати програму переводу секунд в хвилини, години, дні, неділі і т.д.</i> <i>3. На основі аналізу часу виконання програми створити діаграму та визначити якою функцією можна найкраще описати вибраний вами алгоритм.</i> <u>Самостійна робота студента</u> 1. Розглянути програми послідовного та бінарного пошуку числа в масиві чисел. 2. Проаналізувати час виконання обох програм для великих масивів даних. 3. Написати програму що виконую перерахунок до 10^6, 10^7, 10^8. 4. Проаналізувати час виконання для різних множин.</p>	<p>4</p>
<p>Розділ 2. Структури даних та алгоритми</p>		
<p>Знати: Методи організації та зберігання абстрактних типів даних за допомогою структур. Вміти: Вибрати необхідний тип даних і структуру та алгоритм обробки виходячи з завдання, що поставле-</p>	<p><u>Тема 2.1. Огляд класичних алгоритмів для абстрактних типів даних</u> <i>План лекції:</i> 1. Алгоритми та структури. 2. Застосування алгоритмів для реалізації практичних завдань в різних сферах діяльності людини. 3. Задача зв'язності. Алгоритми об'єднання-пошуку. 4. Лінійні списки та черги різних типів. 5. Алгоритми сортування. Пошук даних та обробка строк. 6. Приклад реалізації алгоритму зв'язності. Оцінка алгоритмів. Список рекомендованих джерел Основний: 1,2,5. Додатковий: 6, 8. Інтернет джерела: 9,13. <u>Самостійна робота студента</u> 1. Вивчити теоретичні аспекти створення абст-</p>	<p>2</p> <p>2</p>

Результати навчання	Навчальна діяльність	Кількість годин
но.	<p>рактних типів даних.</p> <p>2.Протестувати та проаналізувати програму пошуку для задачі зв'язування, що подана на лекції.</p>	
<p>Знати: Методи організації та зберігання лінійних списків. Концептуальні задачі, що виникають при роботі зі списками різного типу.</p> <p>Вміти: Написати функції обробки списків з абстрактним типом даних для різних структур реалізації списку.</p>	<p><u>Тема 2.2. Організація лінійних списків та їх обробка</u></p> <p><i>План лекції:</i></p> <ol style="list-style-type: none"> 1. Методи організації та зберігання лінійних списків. 2. Концептуальне описання операцій, що мають місце при роботі з лінійними списками. 3. Реалізація лінійних списків за допомогою масивів. 4. Реалізація лінійних списків за допомогою вказівників. 5. Двічі пов'язані списки. 6. Циклічні списки. <p>Список рекомендованих джерел</p> <p>Основний: 2,4,5.</p> <p>Додатковий: 6, 7</p> <p>Інтернет джерела: 11,14.</p> <p><u>Лабораторне заняття №12 Організація списків та їх обробка</u></p> <ol style="list-style-type: none"> 1. Ознайомлення з абстрактним типом даних для описування списків. 2. Реалізація списків та операцій над ними за допомогою масивів. 3. Реалізація списків та операцій над ними з використанням вказівників. 4. Порівняння реалізацій. <p><u>Самостійна робота студента</u></p> <ol style="list-style-type: none"> 1. Реалізація дво зв'язних списків 2. Реалізація основних операцій для дво зв'язних списків. 3. Циклічні списки. 	<p>2</p> <p>6</p> <p>4</p>
<p>Знати:</p> <p>Принципи</p>	<p><u>Тема 2.3. Стеки</u></p> <p><i>План лекції:</i></p>	2

Результати навчання	Навчальна діяльність	Кількість годин
<p>реалізації стеку на масивах та вказівниках</p> <p>Вміти: Створити програмну реалізацію стеків різних типів та програму тестування роботи стеку.</p> <p>Знати: принципи реалізації черг та деків на циклічних масивах та за допомогою ланцюгів вказівників.</p>	<ol style="list-style-type: none"> 1. Поняття стека. Способи зберігання стеків. 2. Концептуальний опис операцій, які виникають при роботі з стеками. 3. Реалізація основних операцій за допомогою масивів. 4. Реалізація основних операцій за допомогою вказівників. 5. Приклади застосування стеків. <p>Список рекомендованих джерел Основний: 1,2,4. Додатковий: 9. Інтернет джерела: 12,13,14.</p> <p><u>Лабораторне заняття №13 Стеки</u> <i>1. Ознайомлення зі спеціальним типом списку - стеком.</i> <i>2. Реалізація стека з використанням масиву та динамічних структур.</i> <i>3. Реалізація основних операцій для роботи зі стеком.</i></p> <p><u>Самостійна робота студента</u> 1. Структури LIFO (Last In First Out) та FIFO (First In First Out). 2. Реалізація стеків з використанням вказівників. 3. Перегляд елементів стеків. 4. Освоєння технології реалізації рекурсивних функцій з використанням стеків.</p> <p><u>Тема 2.4. Черги</u> <i>План лекції:</i> 1. Поняття черги. Двостороння черга. 2. Концептуальне описання операцій з чергами. 3. Реалізація черги на циклічному масиві. 4. Реалізація черги з використанням вказівників.</p> <p>Список рекомендованих джерел Основний: 1,3,5. Додатковий: 6,7.</p>	<p>6</p> <p>4</p> <p>2</p>

Результати навчання	Навчальна діяльність	Кількість годин
<p>Вміти: Створити програмну реалізацію черги або дека на циклічному масиві та з застосуванням ланцюга вказівників.</p>	<p>Інтернет джерела: 10,13,15.</p> <p><u>Лабораторне заняття №14 Черги</u> 1. Ознайомлення зі спеціальним типом списку - чергою. 2. Реалізація черги з використанням динамічних структур. 3. Ознайомлення зі спеціальним типом списку - деком. Подвійний дек та обмежений дек 4. Реалізація основних операцій для роботи з чергою.</p> <p><u>Самостійна робота студента</u> 1. Реалізація операції додавання, вилучення елементів черги. 2. Створення циклічної черги. 3. Реалізація операції додавання, вилучення елементів деку.</p>	<p>6</p> <p>4</p>
<p>Знати: Проблемні питання створення дерев та реалізацію функцій для роботи з деревами. Вміти: Застосовувати функції попередніх проектів для реалізації дерев різного ступеня. Реалізувати функції</p>	<p><u>Тема 2.5.Деревовидні структури.</u> <u>План лекції:</u> 1. Основні поняття та визначення: Способи подання дерев. 2. Поняття бінарного дерева. 3. Способи обходів дерев: прямий, симетричний, зворотний. 4. Концептуальний опис операцій, які виникають при роботі з деревами. 5. Способи зберігання дерев. 6. Реалізація алгоритмів обходів дерев і деяких операцій. 7. Відповідність між послідовністю звичайних дерев і бінарним деревом.</p> <p>Список рекомендованих джерел Основний: 1,3,5. Додатковий: 6,8. Інтернет джерела: 12,14.</p> <p><u>Лабораторне заняття №15 Деревовидні структури</u></p>	<p>2</p> <p>8</p>

Результати навчання	Навчальна діяльність	Кількість годин
побудови дерев та обходу.	<p>1.Реалізація дерева 3 ступеня у вигляді динамічних структур.</p> <p>2.Застосування функцій побудови списків для реалізації дво зв'язних дерев</p> <p>3.Реалізація рекурсивних функцій для різних способів обходів дерев.</p> <p><u>Самостійна робота студента</u></p> <p>1. Ознайомлення з абстрактними типами даних для описування ієрархічної структури елементів.</p> <p>2. Реалізація дерева 3 ступеня у вигляді рекурсивних структур,</p> <p>3. Реалізація основних операцій з деревами.</p> <p>4. Запропонувати алгоритм створення дерева, де ступінь не зафіксовано.</p>	6
	Разом 1 семестр	180
	Підсумковий контроль — екзамен	
Розділ 3. С++ та об'єктно-орієнтоване програмування		
<p>Знати: синтаксис мови С++ відмінний від С. Особливості оголошення змінних та передачі параметрів функціям.</p> <p>Вміти: застосувати нові оператори та синтаксис для розробки власних про-</p>	<p><u>Тема 3.1. Особливості програмування у С++</u></p> <p><u>План лекції:</u></p> <p>1.Порівняння деяких інструкцій С та С++.</p> <p>2.Потокові операції вводу (виводу) – cout (cin).</p> <p>3.Застосування аргументів по замовчуванню.</p> <p>4.Аргументи за посиланням (&).</p> <p>5.Перевантаження імен функцій.</p> <p>Список рекомендованих джерел</p> <p>Основний: 2, 3,4.</p> <p>Додатковий: 6.</p> <p>Інтернет джерела: 12,13.</p> <p><u>Самостійна робота студента</u></p> <p>1.Оператор розширення області бачення змінної (::).</p> <p>2.Оператори new, delete.</p> <p>3.Inline - функції. Порівняння застосування inline-функцій та макровизначення (#define).</p>	2

Результати навчання	Навчальна діяльність	Кількість годин
грам.		
<p>Знати: структуру класів користувача, розуміти призначення всіх членів класу та місце кожного в створеному класі.</p> <p>Вміти: створити власний клас розмістивши в ньому функції та змінні. Написати функції класу в створеному просторі імен.</p>	<p><u>Тема 3.2. Основні поняття та властивості об'єктно-орієнтованого програмування (ООП). Класи С++</u> <i>План лекції:</i></p> <ol style="list-style-type: none"> 1.Парадигми програмування 2.Компоненти об'єктно орієнтованої парадигми: об'єкт, повідомлення, клас, властивість, метод. 3.Обробка виключень 4.Вказівники на константи та на змінні типу void. 5.Конструктори і ініціалізація 6.Перевантаження імен функцій <p>Список рекомендованих джерел Основний: 1, 4,5. Додатковий: 6,7. Інтернет джерела: 12,14.</p> <p><u>Лабораторне заняття №1 Основні поняття та властивості об'єктно-орієнтованого програмування (ООП). Класи С++</u></p> <ol style="list-style-type: none"> 1.Ознайомлення з синтаксисом, що використовується для визначення класів. 2.Засвоєння основних понять ООП. 3.Знайомство з технологією інкапсуляції даних. 4.Створення базового класу Ознайомлення з синтаксисом, що використовується для визначення похідних класів. 5.Створення ієрархічної структури класів, що успадковують методи базового класу. <p><u>Самостійна робота студента</u></p> <ol style="list-style-type: none"> 1.Парадигма ООП. Інтерпретація основних понять ООП у термінах С++; 2.Призначення розділів описування класу. 3.Створення конструкторів та деструктора класу. 4.Доступ до полів даних та функцій-членів об'є- 	<p>2</p> <p>2</p> <p>6</p>

Результати навчання	Навчальна діяльність	Кількість годин
	кта. 5.Реалізація поліморфізму віртуальними функціями-членами класу.	
Знати: правила застосування базових та похідних класів а також правила обробки виключень.	<u>Тема 3.3. Структура та ієрархія класів, наслідування та поліморфізм</u> <i>План лекції:</i> 1.Створення класів на основі базового класу. 2.Наслідування в похідних класах. 3.Обробка виключень 4.Поліморфізм та віртуальні функції. Список рекомендованих джерел Основний: 1, 2, 4. Додатковий: 6,7. Інтернет джерела: 10,15.	2
Вміти: створити власний клас для фігур на площині та у 3D просторі.	<u>Лабораторне заняття №2 Наслідування та поліморфізм</u> 1. Створення класу для реалізації структури, що описує фігури з Теми 1.8. 2. Розглянути варіант з реалізацією фігур у 3D. 3. Створення власних класів для реалізації проєктів роботи зі списками. 4. Протестувати створені класи у варіантах розрахунку в конструкторі класу та у вигляді окремих функціях – методах класу.	4
	<u>Самостійна робота студента</u> 1. Створення ієрархічної структури класів, що успадковують методи базового класу. 2. Реалізація дерева 3 ступеня засобами створення власного класу та методів класу.	8
Знати: правила створення різних типів функцій, що працюють з класом.	<u>Тема 3.4. Перевантаження операторів C++. Шаблони та шаблонні функції</u> <i>План лекції:</i> 1. Створення функцій, що не належать класу, для перевантаження операторів. 2. Застосування функцій класу для перевантаження операторів. 3. Деякі обмеження.	2

Результати навчання	Навчальна діяльність	Кількість годин
<p>Вміти: створити власний клас та функції перевантаження операторів. Вміти застосувати віртуальні функції.</p>	<p>4. Приклад класів з перевантаженими операторами. 5. Друзі класу та спеціальні випадки перевантаження операторів. 6. Застосування шаблонів та шаблонних функцій. Список рекомендованих джерел Основний: 1,3, 5. Додатковий: 8,9. Інтернет джерела: 13,14,15. <i>Лабораторне заняття №3 Перевантаження операторів C++. Шаблони та шаблонні функції</i> 1. Стандартні контейнери 2. Функції над контейнерами 3. Створення проекту з застосуванням власного класу. Застосування методу перевантаження операторів для C++. <u>Самостійна робота студента</u> 1. Віртуальні функції 2. Шаблони та шаблонні функції 3. Створення власного класу для реалізації масиву екземплярів класу. 4. Застосувати метод «Бульбашки» для сортування абстрактних типів створеного масиву.</p>	<p>4</p> <p>8</p>
<p>Знати: правила створення класів користувача та функцій – методів класу. Вміти: створити класи простих кла-</p>	<p><u>Тема 3.4. Створення власних класів на основі структур для реалізації списків, стеків та черг.</u> <i>План лекції:</i> 1. Створення конструкторів та деструкторів класів. 2. Застосування шаблонів. 3. Створення функцій – членів класу, функцій друзів та помічників класу. 4. Функції обробки виключень. 5. Тестування створених класів. Список рекомендованих джерел Основний: 1,2, 4.</p>	<p>2</p>

Результати навчання	Навчальна діяльність	Кількість годин
<p>сичних структур, що розглядаються у темах 2.1. – 2.4.</p> <p>Створити функції обробки виключень класів.</p>	<p>Додатковий: 6,7.</p> <p>Інтернет джерела: 12,13,15.</p> <p><u>Лабораторне заняття №4 Створення власних класів на основі структур для реалізації списків, стеків та черг.</u></p> <p>1. Створення власних класів для реалізації списків на вказівниках.</p> <p>2. Розробка функцій класу як методів реалізації операцій з даними.</p> <p>3. Обробка виключень з допомогою глобальних змінних.</p> <p><u>Самостійна робота студента</u></p> <p>1. Тестування функції – методів створених класів для різних абстрактних типів даних.</p> <p>2. Розробка алгоритму диференціювання поліному та алгоритму додавання поліномів.</p> <p>3. Розробка класів для «деків» обмеженого вводу та виводу.</p>	<p>4</p> <p>4</p>
<p>Знати:</p> <p>структуру програмного коду додатку основаного на API а правила обробки основних повідомлень. Створити власні класи та рекурсивну функцію малювання.</p>	<p><u>Тема 3.6. Створення власних класів у віконних додатках основаних на проектах API</u></p> <p><u>План лекції:</u></p> <p>1. Структура проекту основаного на «Application Programming Interface» API.</p> <p>2. Запуск додатку, створення вікна.</p> <p>3. Обробка повідомлень.</p> <p>4. Структура повідомлення та типи повідомлень.</p> <p>5. Створення власного повідомлення через пункт меню.</p> <p>Список рекомендованих джерел</p> <p>Основний: 2, 3, 5.</p> <p>Додатковий: 8.</p> <p>Інтернет джерела: 11,12,14.</p> <p><u>Лабораторне заняття №5 Створення власних класів у віконних додатках основаних на проектах API</u></p> <p>1. Створення проекту з застосуванням API.</p> <p>2. Структура проекту у вигляді FileView.</p>	<p>2</p> <p>4</p>

Результати навчання	Навчальна діяльність	Кількість годин
	<p>3. Створення власного пункту меню.</p> <p>4. Створення вікна діалогу та полів для вводу змінних.</p> <p>5. Створення класу для збереження параметрів малювання та рекурсивної функції малювання.</p> <p><u>Самостійна робота студента</u></p> <p>1. Створення функції обробки повідомлення від пункту меню та елементів інтерфейсу.</p> <p>2. Створення рекурсивної функції малювання з застосуванням контексту пристрою.</p> <p>3. Створення класу для отримання параметрів малювання.</p>	12
Розділ 4. Алгоритми впорядкування та пошуку		
<p>Знати: способи реалізації бінарних дерев та алгоритм перетворення дерева деякого ступеня в бінарне дерево.</p> <p>Вміти: створити бінарне дерево та реалізувати програму перетворення дерева 3 ступеня в бінарне дерево.</p>	<p><u>Тема 4.1. Бінарні дерева та вирази.</u></p> <p><i>План лекції:</i></p> <p>1. Поняття бінарного дерева.</p> <p>2. Відповідність між послідовністю звичайних дерев та бінарним деревом.</p> <p>3. Прошите бінарне дерево.</p> <p>4. Послідовне представлення бінарного дерева.</p> <p>Список рекомендованих джерел</p> <p>Основний: 2,3,5.</p> <p>Додатковий: 8.</p> <p>Інтернет джерела: 12,15</p> <p><u>Лабораторне заняття №6 Бінарні дерева та вирази</u></p> <p>1. Реалізація бінарних дерев у вигляді динамічних структур.</p> <p>2. Реалізація рекурсивних функцій для різних способів обходів бінарних дерев.</p> <p>3. Застосування бінарних дерев для побудови синтаксичних аналізаторів арифметичних виразів.</p> <p>4. Реалізація калькулятора за допомогою деревовидної структури.</p> <p><u>Самостійна робота студента</u></p> <p>1. Реалізація бінарного дерева у вигляді рекур-</p>	<p>2</p> <p>6</p> <p>8</p>

Результати навчання	Навчальна діяльність	Кількість годин
	<p>сивних структур.</p> <p>2.Реалізація основних операцій з бінарним деревом.</p> <p>3.Обчислення виразів та лексичний аналіз виразів.</p> <p>4.Створення рекурсивних функцій для аналітичного обчислення похідних.</p>	
<p>Знати: основні поняття процесу сортування масивів та списків. Прості способи реалізації сортування.</p> <p>Вміти: створити програмний код для реалізації запропонованих методів впорядкування.</p>	<p><u>Тема 4.2. Прості алгоритми впорядкування масивів та списків</u></p> <p><i>План лекції:</i></p> <ol style="list-style-type: none"> 1. Постановка завдання і термінологія. 2. Обмінне сортування. 3. Сортування за допомогою вибору. 4. Сортування вставкою. 5. Впорядкування методом вибору максимального чи мінімального елемента. 6. Оцінка складності та швидкості алгоритмів. Порівняння простих методів упорядкування. 7. Сортування пов'язаних списків. <p>Список рекомендованих джерел</p> <p>Основний: 1, 2, 5.</p> <p>Додатковий: 6,8.</p> <p>Інтернет джерела: 14,15.</p> <p><u>Лабораторне заняття №7 Прості алгоритми впорядкування масивів та списків</u></p> <ol style="list-style-type: none"> 1. Вивчення простих алгоритмів для впорядкування масивів та списків. 2. Реалізація деяких простих алгоритмів впорядкування. 3. Реалізація алгоритму впорядкування списків методом вибору та вставки. 4. Аналіз та порівняння простих алгоритмів впорядкування. <p><u>Самостійна робота студента</u></p> <ol style="list-style-type: none"> 1. Вивчення алгоритмів; впорядкування лінійних структур. 2. Реалізація алгоритмів простого впорядку- 	<p>2</p> <p>4</p> <p>6</p>

Результати навчання	Навчальна діяльність	Кількість годин
	вання. 3.Впорядкування методом вставки. 4.Порівняння та аналіз розглянутих методів впорядкування	
Знати: принципи роботи швидких методів впорядкування та методи аналізу розглянутих алгоритмів.	<u>Тема 4.3. Алгоритми швидкого впорядкування</u> <i>План лекції:</i> 1.Модифікація алгоритмів впорядкування. 2.Алгоритми швидкого впорядкування. 3.Впорядкування методом розподілу. 4.Бітове впорядкування. 5.Бінарне впорядкування. 6.Метод Шелла. Список рекомендованих джерел Основний: 2, 3, 5. Додатковий: 6.	2
Вміти: створити програмний код для реалізації запропонованих методів впорядкування.	Інтернет джерела: 13,15 <u>Лабораторне заняття №8 Алгоритми швидкого впорядкування</u> <i>1.Характеристики різних методів впорядкування.</i> <i>2.Злиття двох списків.</i> <i>3.Впорядкування методом злиття.</i> <i>4.Порівняння та аналіз розглянутих методів впорядкування</i> <u>Самостійна робота студента</u> 1.Вивчення модифікованих алгоритмів; впорядкування лінійних структур. 2.Реалізація алгоритмів швидкого впорядкування. 3.Впорядкування методом розподілу. 4.Бітове впорядкування. 5.Бінарне впорядкування. 6.Метод Шелла.	4
		6
Знати: принципи пірамідального со-	<u>Тема 4.4. Черги по пріоритетах та пірамідальне сортування</u> <i>План лекції:</i> 1.Пірамідальна структура даних сортування.	2

Результати навчання	Навчальна діяльність	Кількість годин
<p>ртування абстрактних типів даних.</p> <p>Вміти: застосувати програмний код реалізації алгоритмів дерев сортування.</p>	<p>2. Абстрактний тип даних черги по пріоритетах.</p> <p>3. Алгоритми дерев сортування.</p> <p>4. Черга по пріоритетах для індексних елементів.</p> <p>Список рекомендованих джерел</p> <p>Основний: 2, 3, 5.</p> <p>Додатковий: 6,7.</p> <p>Інтернет джерела: 13,15</p> <p><u>Лабораторне заняття №9 Черги по пріоритетах та пірамідальне сортування</u></p> <p>1. Базовий тип абстрактних черги по пріоритетах.</p> <p>2. Прості реалізації.</p> <p>3. Реалізація черги по пріоритетах з використанням масивів.</p> <p>4. Пірамідальна структура даних.</p> <p>5. Алгоритми для дерев сортування.</p> <p><u>Самостійна робота студента</u></p> <p>1. Оцінка алгоритмів для черги по пріоритетах.</p> <p>2. Аналіз повного бінарного дерева у вигляді пірамідального впорядкованого масиву.</p> <p>3. Висхідна та низхідна установка структури дерева сортування.</p> <p>4. Тестування функцій.</p>	<p>6</p> <p>8</p>
<p>Знати:</p> <p>принципи застосування пошуку в динамічних масивах.</p> <p>Принципи роботи пошуку по різних алгоритмах.</p> <p>Вміти:</p>	<p><u>Тема 4.5 Задачі пошуку</u></p> <p><u>План лекції:</u></p> <p>1. Постановка завдання і термінологія</p> <p>2. Пошук з застосуванням індексації по ключах.</p> <p>3. Послідовний пошук.</p> <p>4. Бінарний пошук.</p> <p>Список рекомендованих джерел</p> <p>Основний: 2, 4, 5.</p> <p>Додатковий: 6, 8.</p> <p>Інтернет джерела: 13,15.</p> <p><u>Лабораторне заняття №10 Задачі пошуку</u></p> <p>1. Застосування абстрактного типу даних.</p> <p>2. Реалізація пошуку для таблиці символів.</p>	<p>2</p> <p>4</p>

Результати навчання	Навчальна діяльність	Кількість годин
створити та протестувати програми пошуку для послідовних масивів абстрактного типу даних.	<p><i>3. Створення програми пошуку для індексовано-го по ключам масиву.</i></p> <p><u>Самостійна робота студента</u></p> <p>1. Реалізація послідовного пошуку на впорядкованих та неупорядкованих масивах.</p> <p>2. Реалізація пошуку в таблиці символів, що побудована на пов'язаному ланцюгу.</p>	8
<p>Знати: принципи створення хеш - таблиць для різних типів ключів та функцій хешування.</p> <p>Вміти: вибрати оптимальний алгоритм хешування та протестувати запропоновані на лекції функції хешування.</p>	<p><u>Тема 4.6. Пошук методом хешування</u></p> <p><i>План лекції:</i></p> <p>1. Постановка завдання і термінологія</p> <p>2. Хеш – функції та проблеми їх застосування.</p> <p>3. Хеш – функції для строкових ключів.</p> <p>4. Роздільне пов'язування та хешування.</p> <p>5. Лінійне зондування</p> <p>Список рекомендованих джерел</p> <p>Основний: 1, 2, 5.</p> <p>Додатковий: 8.</p> <p>Інтернет джерела: 12,14.</p> <p><u>Лабораторне заняття №11 Пошук методом хешування.</u></p> <p>1. Створення програмного коду функцій хешування для ключів з цілих чисел.</p> <p>2. Створення та тестування хеш - функції для строкових ключів. Тестування хеш – таблиці.</p> <p>3. Хешування за допомогою роздільного пов'язування.</p> <p><u>Самостійна робота студента</u></p> <p>1. Хешування методом лінійного зондування.</p> <p>2. Подвійне хешування.</p> <p>3. Динамічні хеш - таблиці.</p> <p>4. Аналіз ефективності алгоритмів хешування.</p>	2
		6
		8
<p>Знати: принципи створення</p>	<p><u>Тема 4.7. Пошук та впорядкування у деревовидних структурах</u></p> <p><i>План лекції:</i></p>	2

Результати навчання	Навчальна діяльність	Кількість годин
<p>та роботи з деревами бінарного пошуку.</p> <p>Вміти: працювати з пошуковими деревами. Застосовувати алгоритми концептуальних операцій з BST деревами.</p> <p>Створити реалізацію функцій основних операцій з BST деревами.</p>	<p>1. Постановка завдання і термінологія</p> <p>2. Дерева бінарного пошуку (BST дерево)</p> <p>3. Реалізація функцій пошуку для BST дерева.</p> <p>4. Збалансовані дерева.</p> <p>5. Характеристики продуктивності дерев бінарного пошуку.</p> <p>Список рекомендованих джерел</p> <p>Основний: 2, 5.</p> <p>Додатковий: 7, 8.</p> <p>Інтернет джерела: 13,15.</p> <p><u>Лабораторне заняття №12 Пошук та впорядкування у деревовидних структурах</u></p> <p>1. Таблиця символів на базі дерева бінарного пошуку.</p> <p>2. Сортування на основі BST дерева.</p> <p>3. Вставка елемента до BST дерева.</p> <p>4. Алгоритми для додавання та видалення вузла у впорядкованому дереві.</p> <p>5. Балансування дерева.</p> <p>6. Балансування дерева після видалення вузла.</p> <p><u>Самостійна робота студента</u></p> <p>1. Реалізація ротацій в BST деревах.</p> <p>2. Вставка в корінь пошукового дерева.</p> <p>3. Вибір та видалення вузла BST дерева.</p> <p>4. Задача про розміщення ферзів та пошук підмножини.</p>	<p>4</p> <p>6</p>
<p>Знати: принципи створення В – дерев та алгоритми операцій з В - деревами.</p> <p>Вміти: застосовувати програм-</p>	<p><u>Тема 4.8. Зовнішній пошук у деревовидних структурах</u></p> <p><u>План лекції:</u></p> <p>1. Постановка задачі пошуку для великих масивів даних.</p> <p>2. Індексований послідовний доступ.</p> <p>3. Структура та побудова В- дерева.</p> <p>4. Алгоритми для додавання вузла у В-дерево, розділення вузлів.</p> <p>5. пошуку даних у В-деревах.</p> <p>Список рекомендованих джерел</p>	<p>2</p>

Результати навчання	Навчальна діяльність	Кількість годин
ний код операцій з В - деревами та хеш таблиць що розширюються як альтернативи В - деревам.	Основний: 3, 4, 5. Додатковий: 6,8. Інтернет джерела: 11,13,15. <u>Лабораторне заняття №13 Зовнішній пошук у деревовидних структурах</u> 1. Побудова В – дерева. 2. Визначення типів вузлів В – дерева. 3. Пошук у В – дереві. 4. Вставка та поділ дерева.	4
	<u>Самостійна робота студента</u> 1. Розширене хешування як альтернатива В - деревам. 2. Структури для розширеного хешування. 3. Створення хеш таблиці, що розширюється. 4. Операції з хеш таблицею та пошук.	4
	<u>Разом за 2 семестр</u>	180
	<u>Підсумковий контроль — екзамен</u>	
<u>Разом по курсу</u>		360

* +20% інтерактиву – зазначені курсивом

3.Список рекомендованих джерел

Основний:¹

1. Белов Ю. Вступ до програмування мовою C++. Організація обчислень: навч. посіб. / Ю.Белов, Т.Карнаух, Ю.Коваль, А.Ставровський. – К. : Видавничо-поліграфічний центр "Київський університет", 2012. – 175 с. с.: іл. (укр.)
2. Л. М. Клакович, С. М. Левицька. Теорія алгоритмів: Навчальний посібник. — Друге видання, доповнене. — Львів :Видавничий центр ЛНУ ім. Івана Франка, 2015. — 161 с.
3. Зацерковний В. І. Алгоритмізація та програмування: навчальний посібник / В. І. Зацерковний, В. І. Гур'єв, І. В. Фірсова. – Ніжин: НДУ ім. М. Гоголя, 2013. – 302 с.
4. Матвієнко М.П. Теорія алгоритмів: підручник / Матвієнко М.П.

¹Курсивом позначені видання, що наявні у бібліотеці КНТЕУ

–Київ: Ліра-К, 2017. – 344 с.

5. Sedgewick R., Algorithms in C++ :Fundamentals, data structur, sorting, serching/ Princeton University 2011.– 1056 с.

Додатковий

6. Programming: Principles and Practice Using C++ (2nd Edition) / by Bjarne Stroustrup. – Addison-Wesley Professional, May 25, 2014. – 1312 p.

7. Грицюк Ю Програмування мовою C++ / Ю.Грицюк, Т.Рак навчальний посібник. – Львів : Вид-во Львівського ДУ БЖД, 2011. – 292 с.

8. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14 1st Edition / by Scott Meyers. - O'Reilly Media, Incorporated, December 5, 2014. – 334 p.

Інтернет ресурси

9. Український сайт, присвячений використанню та українізації операційної системи Linux. - Режим доступу : <http://linux.org.ua>

10. Книжки з методик програмування на C++. - Режим доступу : <http://progbook.net/ss>

11. Довідник мови програмування C++ [Електронний ресурс]. – Режим доступу: <https://msdn.microsoft.com/uk-ua/library/3bstk3k5.aspx>.

12. Сайт про мову програмування C++ [Електронний ресурс]. – Режим доступу: <http://www.cplusplus.com/>.

13. Довідник по IDE VisualStudio [Електронний ресурс]. – Режим доступу: <https://msdn.microsoft.com/>.

14. Сайт підтримки бібліотеки NumPy [Електронний ресурс]. – Режим доступу : <http://www.numpy.org>.

15. <http://progbook.net/ss/> Книжки по методикам програмування на C++.