

**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 кібернетики та системного аналізу**

**ЗАТВЕРДЖЕНО**

вченою радою

(пост. п. 5 від 25 лютого 2019 р.)

Ректор

А.А. Мазаракі



**АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ/  
ALGORITHMIZATION AND PROGRAMMING**

**РОБОЧА ПРОГРАМА /  
SYLLABUS**

освітній ступінь	бакалавр	bachelor
галузь знань	<u>12 Інформаційні технології</u>	<u>Information Technology</u>
спеціальність	<u>124 Системний аналіз</u>	<u>System analysis</u>

Розповсюдження і тиражування без офіційного дозволу КНТЕУ  
заборонено

Автор: В. В. Кулаженко, канд. екон. наук, доц. кафедри кібернетики та системного аналізу КНТЕУ

Розглянуто і схвалено на засіданні кафедри кібернетики та системного аналізу 11.03.2019, протокол № 8.

Рецензенти: Роскладка А.А., д.е.н., професор, завідувач кафедри кібернетики та системного аналізу КНТЕУ

# АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ/ ALGORITHMIZATION AND PROGRAMMING

## РОБОЧА ПРОГРАМА / SYLLABUS

освітній ступінь	бакалавр	bachelor
галузь знань	<u>12 Інформаційні технології</u>	<u>Information Technology</u>
спеціальність	<u>124 Системний аналіз</u>	<u>System analysis</u>

# 1. МЕТА, ЗАВДАННЯ ТА РЕЗУЛЬТАТИ ВИВЧЕННЯ ДИСЦИПЛІНИ (КОМПЕТЕНТНОСТІ), ЇЇ МІСЦЕ В ОСВІТНЬОМУ ПРОЦЕСІ

**Метою дисципліни** «Алгоритмізація та програмування» є набуття майбутніми фахівцями в галузі інформаційних технологій необхідного рівня знань та практичних навичок для алгоритмізації методів системного аналізу та їх практичній реалізації у вигляді програмних додатків.

**Завданням** вивчення дисципліни «Алгоритмізація та програмування» є теоретична та практична підготовка майбутніх фахівців з питань конфігурації, налагодження та використання інструментальних засобів алгоритмічних мов програмування і створення програмних додатків з дружнім до користувача інтерфейсом.

**Предметом** вивчення дисципліни є алгоритми програмування, реалізовані за допомогою технологій, методик та програмного забезпечення на основі мови програмування Python.

У результаті вивчення дисципліни студенти повинні:

## **Знати**

- концепції, покладені в основу алгоритмічних мов;
- синтаксис, семантику та оператори мови програмування;
- призначення та особливості використання об'єктів мови Python;
- структуру та функціональний склад бібліотек, які використовуються в Python;
- основні принципи створення користувацького інтерфейсу програмних додатків;

## **Вміти:**

- розробляти та оптимізувати програмні додатки мовою програмування Python;
- долучати до проекту програмного додатку зовнішні бібліотеки, а також створювати їх самостійно;
- створювати код програми;
- створювати ергономічний користувацький інтерфейс;
- створювати графіки, діаграми та звіти на основі отриманих при аналізі даних;
- створювати програмні додатки для автономної роботи у операційній системі Windows;
- реалізовувати концепції об'єктно-орієнтованого програмування.

## **Практичні навички:**

- Ініціювати, розробляти та обґрунтовувати проекти в галузі систем прийняття рішень, а саме: формулювати концепцію та

ідею проекту; проводити кінцеве тестування;

- формувати інформаційне забезпечення процесу розробки додатків;

- використовувати у своїх проектах методику математичного аналізу, імітаційного моделювання, тощо;

- застосовувати складні алгоритмічні структури та принципи об'єктно-орієнтованого програмування.

Дисципліна «Алгоритмізація та програмування» є професійно-орієнтованою дисципліною і базується на знаннях, отриманих студентами під час вивчення навчальної дисципліни «Дискретна математика» та шкільного курсу з інформатики.

Також дана дисципліна надає студентам необхідні знання для успішного освоєння наступних навчальних дисциплін:

- «Методи оптимізації та дослідження операцій»;
- «Математична логіка і теорія алгоритмів»;
- «Технологія Java»;
- «Технології аналізу даних»;
- «WEB-дизайн та WEB-програмування»;
- «Технологія проектування та адміністрування баз та сховищ даних».

## 2. СТРУКТУРА ДИСЦИПЛІНИ ТА РОЗПОДІЛ ГОДИН ЗА ТЕМАМИ

Для вивчення дисципліни відводиться 360 год. / 12 кредитів ECTS (лекцій – 90 год., практичних занять – 124 год., самостійної роботи – 146 год.).

Назва теми	Всього годин/ кредитів	Кількість годин			Форми контролю
		Лекції	Лабораторні заняття/ МК	Самостійна робота	
1	2	3	4	5	6
Тема 1. Введення в алгоритмічні мови програмування	8	4	-	4	ЛР, Т, П
Тема 2. Об'єкти і структури даних.	14	4	6	4	ЛР, Т, П
Тема 3. Логічні конструкції	24	8	6	10	ЛР, Т, П
Тема 4. Вступ до теорії алгоритмів	30	8	10	12	ЛР, Т, П, М
Тема 5. Структури даних.	32	10	10	12	ЛР, Т, П
Тема 6. Основи об'єктно-орієнтованого програмування.	42	12	14	16	ЛР, Т, П
Тема 7. Робота з базами даних	32	10	10	12	ЛР, Т, П, М
<b>Разом за I семестр</b>	<b>180/6</b>	<b>56</b>	<b>56</b>	<b>68</b>	<b>x</b>
Тема 8. Обробка текстової інформації	20	6	6	8	ЛР, Т, П
Тема 9. Парсинг текстових даних. BeautifulSoup	28	4	12	12	ЛР, Т, П
Тема 10. Jupyter Notebook	14	2	-	8	ЛР, Т, П

Тема 11. Модуль Pandalas	52	10	22	24	ЛР, Т, П
Тема 12. Основи аналізу статистичної інформації за допомогою Python	66	12	28	26	ЛР, Т, П, М
Курсова робота					
Підсумковий контроль – письмовий екзамен					
<b>Разом за II семестр</b>	<b>180/6</b>	<b>34</b>	<b>68</b>	<b>78</b>	<b>х</b>
<b>Разом за 2 семестри</b>	<b>360/12</b>	<b>90</b>	<b>124</b>	<b>146</b>	<b>х</b>

**Примітка:** **Т** – тестування; **ЛР** – виконання лабораторних робіт; **П** – розроблення проекту, **М** – модульний контроль.

### 3. ТЕМАТИКА ТА ЗМІСТ ЛЕКЦІЙНИХ, ПРАКТИЧНИХ ЗАНЯТЬ, САМОСТІЙНОЇ РОБОТИ СТУДЕНТА

Результати навчання	Навчальна діяльність	Робочий час студента	Оцінювання в балах
<b>I семестр</b>			
<b>Тема 1. Введення в алгоритмічні мови програмування</b>			
<b>Знати:</b> Основні поняття алгоритмізації	<b>Лекція №1. Введення в алгоритмічні мови програмування</b> План лекції: 1. Вступ. Мета та завдання дисципліни, її місце у освітньому процесі . 2. Парадигма та основні ідеї, покладені у сучасні алгоритмічні мови програмування. 3. <i>Основні ресурси, спільноти користувачів і розробників (презентація, відео для самостійної роботи).*</i> 4. Мови програмування Python: призначення, основні особливості. <i>Список рекомендованих джерел:</i> Основний: 1-3 Додатковий: 5, 6, 8 Інтернет-ресурси: 9-16	4	
	<b>Завдання для самостійної роботи:</b> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> <li>• Підготовка персонального домашнього комп'ютера до роботи з IDE (PyCharm, Notebook) для Python.</li> <li>• Вивчення основних її можливостей та вбудованих робочих інструментів.</li> <li>• Вивчення стандартів синтаксису мови програмування Python, зарезервованих слів.</li> </ul>	4	
<b>Тема 2. Об'єкти і структури даних.</b>			
<b>Знати:</b> Основні модулі, пакети та бібліотеки; основні змінні у Python	<b>Лекція №2.</b> <b>Базові модулі та бібліотеки Python</b> План 1. Модулі і бібліотеки. 2. Система управління пакетами. 3. Локальний простір імен. 4. Типи змінних у алгоритмічних мовах програмування. <i>Список рекомендованих джерел:</i> Основний: 1-4. Додатковий: 5, 6, 8. Інтернет-ресурси: 10-12, 16.	2	
	<b>Завдання для самостійної роботи:</b>	2	

	<p>1. Вивчення матеріалу лекцій.</p> <p>2. Опрацювання та підготовка доповідей з питань:</p> <ul style="list-style-type: none"> <li>• Вивчити основних модулів та бібліотек, що використовуються у Python.</li> <li>• Встановити необхідні додаткові інструменти (пакети, модулі, бібліотеки) у робочесередовища для виконання самостійної роботи.</li> <li>• Ознайомитись з поняттям локального простору імен.</li> </ul>		
<p><b>Вміти:</b>          Додавати до проекту необхідні модулі, пакети та окремі бібліотеки; використовувати різні змінні</p>	<p align="center"><b>Лабораторне заняття №1.</b>  <b>Типи змінних у Python, принципи їх перетворення.</b></p> <p><b>Мета:</b> ознайомитись з основними типами змінних та операціями над ними.</p> <p><b>Завдання:</b> за допомогою програмного забезпечення PyCharm або Python Notebook виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> <li>1. Типи числових змінних.</li> <li>2. Текстові, бінарні та інші типи змінних.</li> <li>3. Об'єкти.</li> <li>4. Правила взаємодії та переводу значень одних змінних в інші.</li> </ol>	3	5
<p><b>Знати:</b>          Основні операції та функції.</p>	<p align="center"><b>Лекція №3</b>  <i>План лекції:</i></p> <ol style="list-style-type: none"> <li>1. Основні оператори та функції. (презентація, відео для самостійної роботи)</li> <li>2. Створення консольних настільних додатків.</li> <li>3. Поняття динамічного простору імен.</li> </ol> <p><i>Список рекомендованих джерел:</i>          Основний: 1-3          Додатковий: 5, 6, 8          Інтернет-ресурси: 9-16</p>	2	
	<p><b>Завдання для самостійної роботи:</b></p> <ol style="list-style-type: none"> <li>1. Вивчення матеріалу лекцій.</li> <li>2. Опрацювання та підготовка доповідей з питань:             <ul style="list-style-type: none"> <li>• Вивчити лекційний матеріал.</li> <li>• Вивчити основні оператори та функції, навчитись їх застосовувати у Python Notebook.</li> <li>• Практикуватись у створенні консольних настільних додатків у PyCharm різної тематики.</li> </ul> </li> </ol>	2	
<p><b>Вміти:</b>          Створювати консольні додатки</p>	<p align="center"><b>Лабораторне заняття №2.</b>  <i>План заняття:</i></p> <p><b>Створення перших консольних додатків</b></p> <p><b>Мета:</b> навчитись створювати самостійно власні консольні додатки</p> <p><b>Завдання:</b> за допомогою програмного забезпечення PyCharm або Python Notebook виконати ситуаційне завдання по</p>	3	5



	власному варіанту за наступними темами: 1. Створення консольних додатків. Можливості IDE при створенні консольних додатків 2. Написання перших програм, вивід текстової інформації на екран 3. Створення додатків, що контактують з користувачем 4. Створення калькулятора як консольного додатку		
<b>Тема 3. Логічні конструкції.</b>			
<b>Знати:</b> Алгебри логіки та засобів її практичної реалізації у Python	<b>Лекція № 4. Логічні оператори в Python</b> План 1. Алгебра логіки. 2. Блок-схеми. 3. <i>Оператори керування логікою if/elif/else. (презентація, відео для самостійної роботи)*</i> 4. <i>Оператори керування логікою switch-case. (презентація, відео для самостійної роботи)*</i> <b>Список рекомендованих джерел:</b> <i>Основний:</i> 1-4. <i>Додатковий:</i> 5, 6, 8. <i>Інтернет-ресурси:</i> 10-12, 16.	8	
	<b>Завдання для самостійної роботи:</b> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> <li>• Освоїти теоретичний лекційний матеріал.</li> <li>• Використовуючи завдання з лабораторної 3, побудувати блок-схеми для вирішення цих завдань.</li> <li>• Придумати та вирішити 5 типових задач різної тематики для використання конструкції if/elif/else.</li> <li>• Підготувати доповідь на тему: «Специфіка використання логічних операторів у програмуванні».</li> </ul>	10	
<b>Вміти:</b> Використовувати логічні оператори if/elif/else	<b>Лабораторна робота № 3</b> <b>Логічні оператори</b> <b>Мета:</b> Навчитись використовувати логічні конструкції у власних програмах. <b>Завдання:</b> за допомогою програмного забезпечення PyCharm або Python Notebook виконати ситуаційне завдання по власному варіанту за наступними темами: 1. Побудова блок-схем задач з логічними складовими 2. Вирішення завдання з одною логічною умовою (якщо ..., то ...) 3. Вирішення завдань з декількома логічними умовами за допомогою інструкцій	6	11

	if\elif\else. 4. Вирішення завдань з декількома логічними умовами за допомогою інструкцій switch-case		
<b>Тема 4. Вступ до теорії алгоритмів.</b>			
<b>Знати:</b> Конструкції циклів у Python; принципів створення користувальницьких функцій	<b>Лекція № 5. Цикли у Python</b> <b>План</b> 1. <i>Поняття циклу. Інструкції for, while. (презентація, відео для самостійної роботи)*</i> 2. <i>Інструкції break, continue. (презентація, відео для самостійної роботи)*</i> 3. <b>Функція range().</b> 4. <i>Користувальницькі функції. (презентація, відео для самостійної роботи)*</i> <b>Список рекомендованих джерел:</b> <i>Основний: 1-4.</i> <i>Додатковий: 5, 6, 8.</i> <i>Інтернет-ресурси: 10-12, 16</i>	8	
	<b>Завдання для самостійної роботи:</b> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> <li>• Освоїти теоретичний лекційний матеріал.</li> <li>• Опрацювати завдання з інших варіантів лабораторної роботи №4, будуючи тільки умову циклу та його тіло.</li> <li>• Навчитись створювати власні користувальницькі функції.</li> <li>• Переробити рішення власного варіанту лабораторної роботи №3 використовуючи користувальницькі функції</li> </ul>	12	
<b>Вміти:</b> Користуватись циклічними алгоритмами; створювати користувальницькі функції	<b>Лабораторна робота №4</b> <b>Операції з циклами</b> <b>Мета:</b> навчитись використовувати циклічні конструкції у власних програмах, а також створювати та використовувати власні функції <b>Завдання:</b> за допомогою програмного забезпечення PyCharm або Python Notebook виконати ситуаційне завдання по власному варіанту за наступними темами: <ol style="list-style-type: none"> <li>1. Формування блок-схеми типової задачі з циклами</li> <li>2. Вирішення циклічних задач за допомогою інструкції for</li> <li>3. Вирішення циклічних задач за допомогою інструкції while</li> <li>4. Розширення типових задач з циклами за допомогою умовних операторів</li> <li>5. Зациклення</li> </ol> Вирішення задач з необхідністю написання власних користувальницьких функцій.	10	18

<b>Тема 5. Структури даних.</b>			
<b>Знати:</b> Принципи взаємодії об'єктів	<p style="text-align: center;"><b>Лекція №6.</b> <i>Лекція № 6. Об'єкти та масиви у Python</i> План</p> <ol style="list-style-type: none"> <li>1. Поняття об'єкта у алгоритмічних об'єктно-орієнтованих мовах програмування.</li> <li>2. Види об'єктів, їх властивості та операції маніпулювання ними.</li> <li>3. <i>Масиви та основні операції над ними у Python. (презентація, відео для самостійної роботи)*</i></li> <li>4. <i>Багатовимірні масиви. (презентація, відео для самостійної роботи)*</i></li> <li>5. Призначення та структура бібліотеки NumPy.</li> </ol> <p><b>Список рекомендованих джерел:</b> Основний: 1-4. Додатковий: 5, 6, 8. Інтернет-ресурси: 10-12, 16</p>	5	
<b>Вміти:</b> Складати складні алгоритми для вирішення задач	<b>Завдання для самостійної роботи:</b> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> <li>• Вивчити теоретичний лекційний матеріал.</li> <li>• Зрозуміти поняття «об'єкта», підготувати доповідь на тему: «Сутність об'єкта у сучасних об'єктно-орієнтованих мовах програмування».</li> <li>• Ознайомитись з поняттям масиву та багатомірного масиву.</li> <li>• Використовуючи засоби бібліотеки NumPy, практикуватись у створенні простих та багатовимірних масивів (не менше 10 штук кожного виду).</li> </ul>	5	
<b>Знати:</b> Всі види колекцій та методи роботи з ними	<p style="text-align: center;"><b>Лекція № 7. Колекції та робота з ними у Python</b> План</p> <ol style="list-style-type: none"> <li>1. <i>Колекції та їх види: списки. (презентація, відео для самостійної роботи)*</i></li> <li>2. <i>Колекції та їх види: кортежі. (презентація, відео для самостійної роботи)*</i></li> <li>3. Колекції та їх види: множини.</li> <li>4. Колекції та їх види: словники.</li> <li>5. Методи роботи з колекціями.</li> </ol> <p><b>Список рекомендованих джерел:</b> Основний: 1-4. Додатковий: 5, 6, 8. Інтернет-ресурси: 10-12, 16</p>	5	
	<b>Завдання для самостійної роботи:</b> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань:	5	

	<ul style="list-style-type: none"> <li>• Вивчити теоретичний лекційний матеріал.</li> <li>• Ознайомитись з особливостями використання різних видів колекцій у інших мовах програмування.</li> <li>• Практикуватись у створенні всіх типів колекцій у Python.</li> <li>• Придумати по 3 ситуації для кожної колекції, коли б вони повинні були використовуватися</li> </ul>		
<b>Вміти:</b> Використовувати всі види колекцій на практиці	<p style="text-align: center;"><b>Лабораторна робота № 6</b> <b>Колекції</b></p> <p><b>Мета:</b> розглянути особливості використання колекцій та методи роботи з ними.</p> <p><b>Завдання:</b> за допомогою програмного забезпечення PyCharm або Python Notebook виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> <li>1. Сутність колекцій та методи їх використання.</li> <li>2. Типи колекцій, які використовуються у Python.</li> <li>3. Методи для роботи з колекціями у Python.</li> </ol>	12	18
<b>Тема 6. Основи об'єктно-орієнтованого програмування.</b>			
<b>Знати:</b> Принципи ООП	<p style="text-align: center;"><b>Лекція №14.</b> <b>План лекції № 8</b></p> <ol style="list-style-type: none"> <li>1. Основні принципи об'єктно-орієнтованого програмування.</li> <li>2. <i>Наслідування. Класи, їх структура. Екземпляри. (презентація, відео для самостійної роботи)*</i></li> <li>3. <i>Інкапсуляція. (презентація, відео для самостійної роботи)*</i></li> <li>4. <i>Поліморфізм. (презентація, відео для самостійної роботи)*</i></li> <li>5. Методи у об'єктно-орієнтованому програмуванні.</li> </ol> <p><b>Список рекомендованих джерел:</b>  Основний: 1-4.  Додатковий: 5, 6, 8.  Інтернет-ресурси: 10-12, 18</p>	6	
	<p><b>Завдання для самостійної роботи:</b></p> <ol style="list-style-type: none"> <li>1. Вивчення матеріалу лекцій.</li> <li>2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> <li>• Вивчите теоретичний лекційний матеріал.</li> <li>• Використовуючи літературні джерела, написати реферат на тему: «Принципи об'єктно-орієнтованого програмування у сучасних мовах програмування».</li> </ul> </li> </ol>	8	

	<ul style="list-style-type: none"> <li>Знайти по декілька прикладів у реальному житті на кожний з вивчених принципів побудови алгоритмічних програм.</li> </ul>		
<b>Вміти:</b> Використовувати всі можливості ООП під час створення власних програм	<p style="text-align: center;"><b>Лабораторна робота № 7</b>  <b>Об'єктно-орієнтоване програмування</b></p> <p><b>Мета:</b> Освоїти на практиці принципи об'єктно-орієнтованого програмування.</p> <p><b>Завдання:</b> за допомогою програмного забезпечення PyCharm або Python Notebook виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> <li>Конструкція об'єктів, які використовуються у об'єктно-орієнтованому програмуванні</li> <li>Створення програм на основі принципу наслідування</li> <li>Використання інкапсуляції та поліморфізму на практиці</li> </ol>	7	13
<b>Знати:</b> Основні шаблони ООП	<p style="text-align: center;"><b>Лекція № 9. Шаблони у об'єктно-орієнтованому програмуванні</b>  <b>План</b></p> <ol style="list-style-type: none"> <li>Поняття шаблону.</li> <li>Види шаблонів: структурні, поведінкові, створюючі.</li> <li><i>Основні шаблони в алгоритмічних мовах програмування. (презентація, відео для самостійної роботи)*</i></li> </ol> <p><b>Список використаних джерел:</b>  <i>Основний:</i> 1-4.  <i>Додатковий:</i> 5, 6, 8.  <i>Інтернет-ресурси:</i> 10-12, 18</p>	6	
	<p><b>Завдання для самостійної роботи:</b></p> <ol style="list-style-type: none"> <li>Вивчення матеріалу лекцій.</li> <li>Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> <li>Вивчити теоретичний лекційний матеріал.</li> <li>Знайти основні типи шаблонів у мовах програмування.</li> <li>Розібратись, в яких випадках краще використовувати ті, чи інші шаблони.</li> <li>Знайти у «реальному» житті ситуації, в яких той чи інший шаблон застосовується.</li> </ul> </li> </ol>	8	
<b>Вміти:</b> Створювати основні шаблони програмування	<p style="text-align: center;"><b>Лабораторне заняття №8.</b>  <b>Шаблони об'єктно-орієнтованого програмування</b></p> <p><b>Мета:</b> Ознайомитись з базовими шаблонами програмування та навчитись їх примінити на практиці.</p> <p><b>Завдання:</b> за</p>	7	12

	<p>допомогою програмного забезпечення PyCharm або Python Notebook виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> <li>1. Ознайомлення з основними шаблонами програмування</li> <li>2. Використання шаблонів «Стратегія», «Адаптер», «Фабрика», «Декоратор», «Сінглтон».</li> </ol>		
<b>Тема 7. Робота з базами даних.</b>			
<p><b>Знати:</b> Технологію роботи з базами даних у Python</p>	<p style="text-align: center;"><b>Лекція №10. Робота з БД у SQLite</b> <i>План лекції:</i></p> <ol style="list-style-type: none"> <li>1. Робота з даними за допомогою SQL. <i>(презентація, відео для самостійної роботи)*</i></li> <li>2. DB Browser for SQLite. <i>(презентація, відео для самостійної роботи)*</i></li> <li>3. Бібліотека sqlite3.</li> <li>4. Особливості роботи з даними.</li> </ol> <p><b>Список рекомендованих джерел:</b> Основний: 1-3 Додатковий: 4,5,7,8 Інтернет-ресурси: 9, 10, 11, 12</p>	10	
	<p><b>Завдання для самостійної роботи:</b></p> <ol style="list-style-type: none"> <li>1. Вивчення матеріалу лекцій.</li> <li>2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> <li>• Виконання додаткових завдань для редагування даних (sqlbolt.com).</li> <li>• Створення прототипу автоматизованої системи обліку товарів підприємства</li> </ul> </li> </ol>	10	
<p><b>Вміти:</b> Зберігати, зчитувати та редагувати бази даних в середовищі Python</p>	<p style="text-align: center;"><b>Лабораторне заняття №9. Робота з БД</b></p> <p><b>Мета:</b> Ознайомитись з базовими шаблонами роботи з базами даних.</p> <p><b>Завдання:</b> за допомогою програмного забезпечення PyCharm та DB Browser for SQLite виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> <li>1. Редагування даних у таблицях баз даних</li> <li>2. Обмін даними між БД та користувацьким додатком</li> <li>3. Умовні стандарти побудови роботи з БД</li> </ol>	12	18
<b>Підсумковий контроль – письмовий екзамен</b>			
<b>Разом за семестр</b>		180	<b>100</b>
<b>II семестр</b>			
<b>Тема 8. Обробка текстової інформації</b>			
<p><b>Знати:</b> Механізми обробки текстових даних</p>	<p style="text-align: center;"><b>Лекція № 1. Функції роботи з текстовими змінними</b></p> <ol style="list-style-type: none"> <li>1. Робота з текстовими рядками.</li> <li>2. <i>Основні методи роботи з текстовими рядками, їх розширення. (презентація, відео для самостійної роботи)*</i></li> <li>3. <i>Форматування текстових</i></li> </ol>	6	

	змінних(презентація, відео для самостійної роботи)* 4. Алгоритм роботи програм антиплагіату <b>Список рекомендованих джерел:</b> Основний: 1, 3, 4. Додатковий: 6, 7. Інтернет-ресурси: 10-12, 15, 18		
	<b>Завдання для самостійної роботи:</b> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> <li>• Вивчитеоретичнийлекційнийматеріал.</li> <li>• Познайомитись з основними модулями для антиплагіату у Python</li> <li>• Визначити основні напрямки практичної роботи з текстовими даними</li> </ul>	8	
<b>Вміти:</b> Обробляти та аналізувати текстові дані засобами Python	<b>Лабораторне заняття №1.</b> <b>Форматування текстових даних у Python</b> <b>Мета:</b> навчитись працювати з текстовими даними у Python. <b>Завдання:</b> використовуючи PyChartвиконатиситуаційнезавдання по власномуваріанту за наступними темами: 1. Основи роботи з текстовими даними у Python 2. Методи роботи з даними, представлені у стандартному наборі Python 3. Методи форматування текстових даних	6	9
<b>Тема 9. Парсинг текстових даних. BeautifulSoup</b>			
<b>Знати:</b> Принципи парсингу веб-сайтів	<b>Лекція № 2. Парсинг текстових даних</b> 1. Основні принципи парсингу веб-сторінок. 2. Модуль BeautifulSoup. (презентація, відео для самостійної роботи)* 3. Використання допоміжних засобів для роботи з BeautifulSoup(презентація, відео для самостійної роботи)* 4. Побудова прототипу десктопного додатку парсингу веб-сайтів <b>Список рекомендованих джерел:</b> Основний: 1, 3, 4. Додатковий: 6, 7. Інтернет-ресурси: 10-12, 15, 18	4	
	<b>Завдання для самостійної роботи:</b> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> <li>• Вивчитеоретичнийлекційнийматеріал.</li> <li>• Проаналізувати структуру побудови сторінок у html підвищеної складності</li> <li>• Ознайомитись з аналогічними засобами</li> </ul>	12	

	парсингу текстових даних у Python		
<b>Вміти:</b> Використовувати зовнішній модуль BeautifulSoup для парсингу веб-сайтів	<b>Лабораторне заняття №2. Розробка додатків для видобутку та первинного аналізу текстових даних</b> <b>Мета:</b> навчитись створювати десктопні додатки для видобутку даних. <b>Завдання:</b> використовуючи PyCharm виконати ситуаційне завдання по власному варіанту за наступними темами: 1. Завантаження текстової частини веб-сторінок на локальний диск 2. Аналіз веб-сторінок за допомогою модуля BeautifulSoup 3. Принципи постобробки отриманих текстових даних	12	18
<b>Тема 10. Jupyter Notebook</b>			
<b>Знати:</b> функціонал роботи Jupyter Notebook	<b>Лекція № 3. Jupyter Notebook</b> 1. Альтернативне середовище розробки для Python. 2. <i>Встановлення та налаштування середовища Jupyter Notebook. (презентація, відео для самостійної роботи)*</i> 3. <i>Приклади вирішення практичних задач на базі Jupyter Notebook (презентація, відео для самостійної роботи)*</i> <b>Список рекомендованих джерел:</b> <i>Основний:</i> 1, 3, 4. <i>Додатковий:</i> 6, 7. <i>Інтернет-ресурси:</i> 10-12, 15, 18	2	
<b>Вміти:</b> використовувати Jupyter Notebook якості середовища розробки	<b>Завдання для самостійної роботи:</b> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: • Вивчитеоретичний лекційний матеріал. • Познайомитись з розширеним функціоналом середовища Jupyter Notebook • Реалізація алгоритмів роботи з текстовими даними у Jupyter Notebook	8	
<b>Тема 11. Модуль Pandas</b>			
<b>Знати:</b> Принципи роботи модулю Pandas	<b>Лекція № 4. Використання модулю Pandas у системному аналізі</b> 4. Сутність та основні засоби Pandas. 5. <i>Структури даних – серії та датафрейми. (презентація, відео для самостійної роботи)*</i> 6. <i>Завантаження даних у датафрейми (презентація, відео для самостійної роботи)*</i> 7. Отримання даних з датафреймів 8. Редагування датафреймів <b>Список рекомендованих джерел:</b> <i>Основний:</i> 1, 3, 4.	20	



	<p><i>Додатковий: 6, 7.</i> <i>Інтернет-ресурси: 10-12, 15, 18</i></p>		
	<p><b>Завдання для самостійної роботи:</b> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань:</p> <ul style="list-style-type: none"> <li>• Вивчититеоретичнийлекційнийматеріал.</li> <li>• Опрацювати базові методи розрахункових метрик даних</li> <li>• Підготовка даних для машинного навчання</li> </ul>	24	
<p><b>Вміти:</b> Працювати з датафреймами у середовищі Python</p>	<p style="text-align: center;"><b>Лабораторне заняття №4. Робота з даними на базі Pandas</b></p> <p><b>Мета:</b> Отримати практичні навички для роботи з датафреймами. <b>Завдання:</b> використовуючи Jupyter Notebook виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> <li>1. Створення датафреймів та наповнення їх даними</li> <li>2. Дослідження завантажених даних</li> <li>3. Методи об'єднання датафреймів</li> <li>4. Вирішення практичних задач у сфері роботи з даними</li> </ol>	22	32
<b>Тема 12. Основи аналізу статистичної інформації за допомогою Python</b>			
<p><b>Знати:</b> інструменти передобробки даних у Python</p>	<p style="text-align: center;"><b>Лекція № 5. Преобробка великих даних</b> <i>План лекції:</i></p> <ol style="list-style-type: none"> <li>1. Великі дані (Big Data).</li> <li>2. Засоби предобробки великих даних. (презентація, відео для самостійної роботи)*</li> <li>3. Особливості роботи з великими даними (презентація, відео для самостійної роботи)*</li> <li>4. Описова статистика великих даних</li> <li>5. Використання графіків у аналізі великих даних</li> </ol> <p><b>Список рекомендованих джерел:</b> <i>Основний: 1, 3, 4.</i> <i>Додатковий: 6, 7.</i> <i>Інтернет-ресурси: 10-12, 15, 18</i></p>	6	
	<p><b>Завдання для самостійної роботи:</b> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань:</p> <ul style="list-style-type: none"> <li>• Вивчититеоретичнийлекційнийматеріал.</li> <li>• Познайомитись з основними напрямками використання великих даних</li> </ul>	13	
<p><b>Вміти:</b> здійснювати початковий аналіз та роботи</p>	<p style="text-align: center;"><b>Лабораторне заняття №5. Обробка великих даних у Python</b></p>	14	21

преобробку великих даних	<p><b>Мета:</b> навчитись виконувати преобробку великих даних у Python.</p> <p><b>Завдання:</b> використовуючи Jupyter Notebook виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> <li>1. Робота з великими даними</li> <li>2. Проведення преобробки даних</li> <li>3. Проведення описового статистичного аналізу</li> <li>4. Графічний аналіз даних</li> </ol>		
<b>Знати:</b> Основні засоби статистичного аналізу у Python	<p><b>Лекція № 6. Статистичний аналіз у сфері великих даних</b></p> <ol style="list-style-type: none"> <li>1. Аналіз номінативних даних.</li> <li>2. <i>Кореляція та лінійна регресія (презентація, відео для самостійної роботи)*</i></li> <li>3. <i>Діагностика моделі (презентація, відео для самостійної роботи)*</i></li> <li>4. Логістична регресія</li> </ol> <p><b>Список рекомендованих джерел:</b>  <i>Основний:</i> 1, 3, 4.  <i>Додатковий:</i> 6, 7.  <i>Інтернет-ресурси:</i> 10-12, 15, 18</p>	6	
	<p><b>Завдання для самостійної роботи:</b></p> <ol style="list-style-type: none"> <li>1. Вивчення матеріалу лекцій.</li> <li>2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> <li>• Вивчитиматематичний апарат статистичного аналізу;</li> <li>• Ознайомитись з іншими сучасними популярними методами аналізу великих даних.</li> </ul> </li> </ol>	13	
<b>Вміти:</b> проводити статистичний аналіз великих даних	<p><b>Лабораторне заняття №6. Статистичний аналіз даних у Python</b></p> <p><b>Мета:</b> навчитись виконувати статистичний аналіз даних у Python на прикладі великих даних.</p> <p><b>Завдання:</b> використовуючи Jupyter Notebook виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> <li>1. Номінативний аналіз даних</li> <li>2. Регресійний аналіз даних</li> <li>3. Кореляційний аналіз</li> </ol>	14	20
<b>Підсумковий контроль – письмовий екзамен</b>			
<b>Разом за семестр</b>		180	100
<b>Разом</b>		360	

\*курсивом виділено інтерактивні методики навчання

#### 4. СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

##### *Основний*

1. Introduction To Python 3: (Python Documentation Manual Part 1) / by Guido Van Rossum, Fred L. Drake. - CreateSpace Independent Publishing Platform, March 20, 2009. – 264 p.
2. Michael Dawson Python Programming for the Absolute Beginner / Michael Dawson. - Cengage Learning PTR. – 480 pages.
3. Mark L. Learning Python, 5th Edition / L. Mark – Sebastopol: O'Reilly Media, 2013. – 648 p.
4. Яковенко А. В. Основипрограмування. Python. Частина 1 [Електроннийресурс]: підручникдлястуд. спеціальності 122 "Комп'ютернінауки", спеціалізації "Інформаційнітехнологіївбіологіїта медицині" / А. В. Яковенко ; КПІім. Ігоря Сікорського.– Київ : КПІім. Ігоря Сікорського, 2018. – 195 с.
5. Kenneth A. Lambert Fundamentals of Python: First Programs 002 Edition / by Kenneth A. Lambert. – Boston: Cengage Learning, January 1, 2018. – 379 p.

##### *Додатковий*

6. Raschka S. Python Machine Learning, 1st Edition / S. Raschka. – Birmingham: Packt Publishing, 2015. – 456 p.
7. Richert W. Building Machine Learning Systems with Python / W. Richert, L. P. Coelho – Birmingham: Packt Publishing, 2013. – 290 с.
8. 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.

##### *Internet-ресурси*

9. Сайт розробника Python [Електронний ресурс]. — Режим доступу: <https://www.python.org>.
10. Довідник мови програмування C++ [Електронний ресурс]. – Режим доступу: <https://msdn.microsoft.com/uk-ua/library/3bstk3k5.aspx>.
11. Сайт про мову програмування C++ [Електронний ресурс]. – Режим доступу: <http://www.cplusplus.com/>.
12. Довідник по IDE VisualStudio [Електронний ресурс]. – Режим доступу: .
13. Сайт підтримки бібліотеки NumPy [Електронний ресурс]. – Режим доступу : <http://www.numpy.org>.
14. Сайт підтримки бібліотеки Pandas [Електронний ресурс]. –

Режим доступу : <http://www.pandas.pydata.org/>.

15. Безкоштовні онлайн курси [Електронний ресурс]. – Режим доступу : <https://prometheus.org.ua/>.

16. Суспільство програмістів [Електронний ресурс]. – Режим доступу: <https://www.dou.ua/>