

**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. Від 25 лютого 2019 р.)

Ректор

А. А. Мазаракі



**АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ/
ALGORITHMIZATION AND PROGRAMMING**

**ПРОГРАМА/
CURRICULUM**

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>

Київ 2019

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

ПРОГРАМА/
CURRICULUM

CURRICULUM

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	Кількість годин			Форми контролю	
	Всього годин/кредитів	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
Разом за I семестр	180/6	56	56	68	x
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 Pandal	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					
Разом за II семестр	180/6	34	68	78	х
Разом за 2 семестри	360/12	90	124	146	х

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
I семестр			
Тема 1. Введення в алгоритмічні мови програмування			
Знати: Основні поняття алгоритмізації	Лекція №1. Введення в алгоритмічні мови програмування 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	
	Завдання для самостійної роботи: 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: • Підготовка персонального домашнього комп'ютера до роботи з IDE (PyCharm, Notebook) для Python. • Вивчення основних її можливостей та вбудованих робочих інструментів. • Вивчення стандартів синтаксису мови програмування Python, зарезервованих слів.	4	
Тема 2. Об'єкти і структури даних.			
Знати: Основні модулі, пакети та бібліотеки; основні змінні у	Лекція №2. Базові модулі та бібліотеки Python План 1. Модулі і бібліотеки. 2. Система управління пакетами.	2	

Python	<p>3. <i>Локальний простір імен.</i></p> <p>4. <i>Типи змінних у алгоритмічних мовах програмування.</i></p> <p>Список рекомендованих джерел: <i>Основний: 1-4.</i> <i>Додатковий: 5, 6, 8.</i> <i>Інтернет-ресурси: 10-12, 16.</i></p>		
	<p>Завдання для самостійної роботи:</p> <p>1. Вивчення матеріалу лекцій.</p> <p>2. Опрацювання та підготовка доповідей з питань:</p> <ul style="list-style-type: none"> • Вивчити основних модулів та бібліотек, що використовуються у Python. • Встановити необхідні додаткові інструменти (пакети, модулі, бібліотеки) у робоче середовища для виконання самостійної роботи. • Ознайомитись з поняттям локального простору імен. 	2	
<p>Вміти: Додавати до проекту необхідні модулі, пакети та окремі бібліотеки; використовувати різні змінні</p>	<p>Лабораторне заняття №1. Типи змінних у Python, принципи їх перетворення.</p> <p>Мета: ознайомитись з основними типами змінних та операціями над ними.</p> <p>Завдання: за допомогою програмного забезпечення PyCharm або Python Notebook виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> 1. Типи числових змінних. 2. Текстові, бінарні та інші типи змінних. 3. Об'єкти. 4. Правила взаємодії та переводу значень одних змінних в інші. 	3	5
<p>Знати: Основні операції та функції.</p>	<p>Лекція №3 <i>План лекції:</i></p> <ol style="list-style-type: none"> 1. <i>Основні оператори та функції. (презентація, відео для самостійної роботи)</i> 2. Створення консольних настільних додатків. 	2	

	<p>3. Поняття динамічного простору імен. <i>Список рекомендованих джерел:</i> Основний: 1-3 Додатковий: 5, 6, 8 Інтернет-ресурси: 9-16</p>		
	<p>Завдання для самостійної роботи: 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань:</p> <ul style="list-style-type: none"> • Вивчити лекційний матеріал. • Вивчити основні оператори та функції, навчитись їх застосовувати у PythonNotebook. • Практикуватись у створенні консольних настільних додатків у PyCharm різної тематики. 	2	
<p>Вміти: Створювати консольні додатки</p>	<p style="text-align: center;">Лабораторне заняття №2. <i>План заняття:</i></p> <p>Створення перших консольних додатків Мета: навчитись створювати самостійно власні консольні додатки Завдання: за допомогою програмного забезпечення PyCharm або PythonNotebook виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> 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 	3	5
Тема 3. Логічні конструкції.			
<p>Знати: Алгебри логіки та засобів її практичної</p>	<p>Лекція № 4. Логічні оператори в Python План</p> <ol style="list-style-type: none"> 1. Алгебра логіки. 2. Блок-схеми. 	8	

реалізації у Python	<p>3. <i>Оператори керування логікою if/elif/else. (презентація, відео для самостійної роботи)*</i></p> <p>4. <i>Оператори керування логікою switch-case. (презентація, відео для самостійної роботи)*</i></p> <p>Список рекомендованих джерел: <i>Основний:</i> 1-4. <i>Додатковий:</i> 5, 6, 8. <i>Інтернет-ресурси:</i> 10-12, 16.</p>		
	<p>Завдання для самостійної роботи:</p> <p>1. Вивчення матеріалу лекцій.</p> <p>2. Опрацювання та підготовка доповідей з питань:</p> <ul style="list-style-type: none"> • Освоїти теоретичний лекційний матеріал. • Використовуючи завдання з лабораторної 3, побудувати блок-схеми для вирішення цих завдань. • Придумати та вирішити 5 типових задач різної тематики для використання конструкції if/elif/else. • Підготувати доповідь на тему: «Специфіка використання логічних операторів у програмуванні». 	10	
<p>Вміти: Використовувати логічні оператори if/elif/else</p>	<p style="text-align: center;">Лабораторна робота № 3 Логічні оператори</p> <p>Мета: Навчитись використовувати логічні конструкції у власних програмах.</p> <p>Завдання: за допомогою програмного забезпечення PyCharm або Python Notebook виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> 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 if/elif/else. 4. Вирішення завдань з декількома логічними умовами за допомогою 	6	11

	інструкцій switch-case		
Тема 4. Вступ до теорії алгоритмів.			
Знати: Конструкції циклі в у Python; принципи створення користувацьких функцій	<p style="text-align: center;">Лекція № 5. Цикли у Python</p> <p style="text-align: center;">План</p> <ol style="list-style-type: none"> 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. <i>Функція range().</i> 4. <i>Користувальницькі функції. (презентація, відео для самостійної роботи)*</i> <p>Список рекомендованих джерел: <i>Основний: 1-4.</i> <i>Додатковий: 5, 6, 8.</i> <i>Інтернет-ресурси: 10-12, 16</i></p>	8	
	<p>Завдання для самостійної роботи:</p> <ol style="list-style-type: none"> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> • Освоїти теоретичний лекційний матеріал. • Опрацювати завдання з інших варіантів лабораторної роботи №4, будуючи тільки умову циклу та його готіло. • Навчитись створювати власні користувацькі функції. • Переробити рішення власного варіанту лабораторної роботи №3 використовуючи користувацькі функції 	12	
Вміти: Користуватись циклічними алгоритмами; створювати користувацькі функції	<p style="text-align: center;">Лабораторна робота №4</p> <p style="text-align: center;">Операції з циклами</p> <p>Мета: навчитись використовувати циклічні конструкції у власних програмах, а також створювати та використовувати власні функції</p> <p>Завдання: за допомогою програмного забезпечення PyCharm або Python Notebook виконати ситуаційне завдання</p>	10	18

	<p>ння по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> 1. Формування блок-схеми типової задачі з циклами 2. Вирішення циклічних задач за допомогою інструкції for 3. Вирішення циклічних задач за допомогою інструкції while 4. Розширення типових задач з циклами за допомогою умовних операторів 5. Зациклення <p>Вирішення задач з необхідністю написання власних користувальницьких функцій.</p>		
Тема 5. Структури даних.			
<p>Знати: Принципи взаємодії об'єктів</p>	<p style="text-align: center;">Лекція №6. <i>Лекція № 6. Об'єкти та масиви у Python</i> План</p> <ol style="list-style-type: none"> 1. Поняття об'єкта у алгоритмічних об'єктно-орієнтованих мовах програмування. 2. Види об'єктів, їх властивості та операції маніпулювання ними. 3. <i>Масиви та основні операції над ними у Python. (презентація, відео для самостійної роботи)*</i> 4. <i>Багатовимірні масиви. (презентація, відео для самостійної роботи)*</i> 5. Призначення та структура бібліотеки NumPy. <p>Список рекомендованих джерел: Основний: 1-4. Додатковий: 5, 6, 8. Інтернет-ресурси: 10-12, 16</p>	5	
<p>Вміти: Складати складні алгоритми для вирішення задач</p>	<p>Завдання для самостійної роботи:</p> <ol style="list-style-type: none"> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> • Вивчити теоретичний лекційний матеріал. • Зрозуміти поняття «об'єкта», підготувати доповідь на тему: «Сутність 	5	

	<p>об'єкта у сучасних об'єктно-орієнтованих мовах програмування».</p> <ul style="list-style-type: none"> • Ознайомитись з поняттям масиву та багатомірного масиву. • Використовуючи засоби бібліотеки NumPy, практикуватись у створенні простих та багатовимірних масивів (не менше 10 штук кожного виду). 		
<p>Знати: Всі види колекцій та методи роботи з ними</p>	<p>Лекція № 7. Колекції та робота з ними у Python План</p> <ol style="list-style-type: none"> 1. Колекції та їх види: списки. (презентація, відео для самостійної роботи)* 2. Колекції та їх види: кортежі. (презентація, відео для самостійної роботи)* 3. Колекції та їх види: множини. 4. Колекції та їх види: словники. 5. Методи роботи з колекціями. <p>Список рекомендованих джерел: Основний: 1-4. Додатковий: 5, 6, 8. Інтернет-ресурси: 10-12, 16</p>	5	
	<p>Завдання для самостійної роботи:</p> <ol style="list-style-type: none"> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> • Вивчити теоретичний лекційний матеріал. • Ознайомитись з особливостями використання різних видів колекцій у інших мовах програмування. • Практикуватись у створенні всіх типів колекцій у Python. • Придумати по 3 ситуації для кожної колекції, коли б вони повинні були використовуватися 	5	
<p>Вміти: Використовувати всі види колекцій на практиці</p>	<p>Лабораторна робота № 6 Колекції</p> <p>Мета: розглянути особливості використання колекцій та методи роботи з ними.</p>	12	18

	<p>Завдання: за допомогою програмного забезпечення PyCharm або Python Notebook виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> 1. Сутність колекцій та методи їх використання. 2. Типи колекцій, які використовуються у Python. 3. Методи для роботи з колекціями у Python. 		
Тема 6. Основи об'єктно-орієнтованого програмування.			
<p>Знати: Принципи ООП</p>	<p style="text-align: center;">Лекція №14. План лекції № 8</p> <ol style="list-style-type: none"> 1. Основні принципи об'єктно-орієнтованого програмування. 2. <i>Наслідування. Класи, їх структура. Екземпляри. (презентація, відео для самостійної роботи)*</i> 3. <i>Інкапсуляція. (презентація, відео для самостійної роботи)*</i> 4. <i>Поліморфізм. (презентація, відео для самостійної роботи)*</i> 5. Методи у об'єктно-орієнтованому програмуванні. <p>Список рекомендованих джерел: Основний: 1-4. Додатковий: 5, 6, 8. Інтернет-ресурси: 10-12, 18</p>	6	
	<p>Завдання для самостійної роботи:</p> <ol style="list-style-type: none"> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> • Вивчити теоретичний лекційний матеріал. • Використовуючи літературні джерела, написати реферат на тему: «Принципи об'єктно-орієнтованого програмування у сучасних мовах програмування». • Знайти по декілька прикладів у реальному житті на кожний з вивчених принципів побудови алгоритмів 	8	

	чних програм.		
Вміти: Використовувати всі можливості ООП під час створення власних програм	Лабораторна робота № 7 Об'єктно-орієнтоване програмування Мета: Освоїти на практиці принципи об'єктно-орієнтованого програмування. Завдання: за допомогою програмного забезпечення PyCharm або Python Notebook виконати ситуаційне завдання по власному варіанту за наступними темами: 1. Конструкція об'єктів, які використовуються у об'єктно-орієнтованому програмуванні 2. Створення програм на основі принципу наслідування 3. Використання інкапсуляції та поліморфізму на практиці	7	13
Знати: Основні шаблони ООП	Лекція № 9. Шаблони у об'єктно-орієнтованому програмуванні План 1. Поняття шаблону. 2. Види шаблонів: структурні, поведінкові, створюючі. 3. <i>Основні шаблони в алгоритмічних мовах програмування. (презентація, відео для самостійної роботи)*</i> Список використаних джерел: <i>Основний:</i> 1-4. <i>Додатковий:</i> 5, 6, 8. <i>Інтернет-ресурси:</i> 10-12, 18	6	
	Завдання для самостійної роботи: 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: • Вивчити теоретичний лекційний матеріал. • Знайти основні типи шаблонів у мовах програмування. • Розібратись, в яких випадках краще використовувати ті, чи інші шаблони. • Знайти у «реальному» житті	8	

	ситуації, в яких той чи інший шаблон застосовується.		
Вміти: Створювати основні шаблони програмування	Лабораторне заняття №8. Шаблони об'єктно-орієнтованого програмування Мета: Ознайомитись з базовими шаблонами програмування та навчитись їх приміняти на практиці. Завдання: за допомогою програмного забезпечення PyCharm або Python Notebook виконати ситуаційне завдання по власному варіанту за наступними темами: 1. Ознайомлення з основними шаблонами програмування 2. Використання шаблонів «Стратегія», «Адаптер», «Фабрика», «Декоратор», «Сінглтон».	7	12
Тема 7. Робота з базами даних.			
Знати: Технологію роботи з базами даних у Python	Лекція №10. Робота з БД у SQLite <i>План лекції:</i> 1. Робота з даними за допомогою SQL. (презентація, відео для самостійної роботи)* 2. DBBrowserforSQLite. (презентація, відео для самостійної роботи)* 3. Бібліотека sqlite3. 4. Особливості роботи з даними. Список рекомендованих джерел: Основний: 1-3 Додатковий: 4,5,7,8 Інтернет-ресурси: 9, 10, 11, 12	10	
	Завдання для самостійної роботи: 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: • Виконання додаткових завдань для редагування даних (sqlbolt.com). • Створення прототипу автоматизованої системи обліку товарів підприємства	10	
Вміти:	Лабораторне заняття №9.	12	18

Зберігати, зчитувати та редагувати бази даних в середовищі Python	<p align="center">Робота з БД</p> <p>Мета: Ознайомитись з базовими шаблонами роботи з базами даних.</p> <p>Завдання: за допомогою програмного забезпечення PyCharm та BDBrowserforSQLite виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> 1. Редагування даних у таблицях баз даних 2. Обмін даними між БД та користувацьким додатком 3. Умовні стандарти побудови роботи з БД 		
Підсумковий контроль – письмовий екзамен			
Разом за семестр		180	100
II семестр			
Тема 8. Обробка текстової інформації			
Знати: Механізми обробки текстових даних	<p align="center">Лекція № 1. Функції роботи з текстовими змінними</p> <ol style="list-style-type: none"> 1. Робота з текстовими рядками. 2. <i>Основні методи роботи з текстовими рядками, їх розширення. (презентація, відео для самостійної роботи)*</i> 3. <i>Форматування текстових змінних (презентація, відео для самостійної роботи)*</i> 4. <i>Алгоритм роботи програм антиплагіату</i> <p>Список рекомендованих джерел: <i>Основний:</i> 1, 3, 4. <i>Додатковий:</i> 6, 7. <i>Інтернет-ресурси:</i> 10-12, 15, 18</p>	6	
	<p>Завдання для самостійної роботи:</p> <ol style="list-style-type: none"> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> • Вивчитеоретичнийлекційнийматеріал. • Познайомитись з основними модулями для антиплагіату у Python • Визначити основні напрямки 	8	

	практичної роботи з текстовими даними		
Вміти: Обробляти та аналізувати текстові дані засобами Python	<p align="center">Лабораторне заняття №1.</p> <p align="center">Форматування текстових даних у Python</p> <p>Мета: навчитись працювати з текстовими даними у Python. Завдання: використовуючи PyCharm виконати ситуаційне завдання по власному варіанту 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</p>	6	9
Тема 9. Парсинг текстових даних. BeautifulSoup			
Знати: Принципи парсингу веб-сайтів	<p>Лекція № 2. Парсинг текстових даних</p> <ol style="list-style-type: none"> Основні принципи парсингу веб-сторінок. Модуль <i>BeautifulSoup</i>. (презентація, відео для самостійної роботи)* Використання допоміжних засобів для роботи з <i>BeautifulSoup</i> (презентація, відео для самостійної роботи)* Побудова прототипу десктопного додатку парсингу веб-сайтів <p>Список рекомендованих джерел: Основний: 1, 3, 4. Додатковий: 6, 7. Інтернет-ресурси: 10-12, 15, 18</p>	4	
	<p>Завдання для самостійної роботи:</p> <ol style="list-style-type: none"> Вивчення матеріалу лекцій. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> Вивчитеоретичнийлекційнийматеріал. Проаналізувати структуру побудови сторінок у html підвищеної складності Ознайомитись з аналогічними засобами парсингу текстових даних у Python 	12	
Вміти:	Лабораторне заняття №2.	12	18

Використовувати зовнішній модуль BeautifulSoup для парсингу веб-сайтів	<p>Розробка додатків для видобутку та первинного аналізу текстових даних</p> <p>Мета: навчитись створювати десктопні додатки для видобутку даних.</p> <p>Завдання: використовуючи PyCharm виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> 1. Завантаження текстової частини веб-сторінок на локальний диск 2. Аналіз веб-сторінок за допомогою модуля BeautifulSoup 3. Принципи постобробки отриманих текстових даних 		
Тема 10. Jupyter Notebook			
Знати: функціонал роботи Jupyter Notebook	<p>Лекція № 3. Jupyter Notebook</p> <ol style="list-style-type: none"> 1. Альтернативне середовище розробки для Python. 2. Встановлення та налаштування середовища Jupyter Notebook. (презентація, відео для самостійної роботи)* 3. Приклади вирішення практичних задач на базі Jupyter Notebook (презентація, відео для самостійної роботи)* <p>Список рекомендованих джерел: Основний: 1, 3, 4. Додатковий: 6, 7. Інтернет-ресурси: 10-12, 15, 18</p>	2	
Вміти: використовувати Jupyter Notebook у якості середовища розробки	<p>Завдання для самостійної роботи:</p> <ol style="list-style-type: none"> 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> • Вивчитеоретичний лекційний матеріал. • Познайомитись з розширеним функціоналом середовища Jupyter Notebook • Реалізація алгоритмів роботи з текстовими даними у Jupyter Notebook 	8	
Тема 11. Модуль Pandas			
Знати: Принципи	<p>Лекція № 4. Використання модулю Pandas у системному аналізі</p>	20	

<p>роботи модулю Pandas</p>	<p>4. Сутність та основні засоби Pandas. 5. Структури даних – серії та датафрейми. (презентація, відео для самотійної роботи)* 6. Завантаження даних у датафрейми(презентація, відео для самотійної роботи)* 7. Отримання даних з датафреймів 8. Редагування датафреймів Список рекомендованих джерел: <i>Основний:</i> 1, 3, 4. <i>Додатковий:</i> 6, 7. <i>Інтернет-ресурси:</i> 10-12, 15, 18</p>		
	<p>Завдання для самотійної роботи: 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань:</p> <ul style="list-style-type: none"> • Вивчитеоретичнийлекційнийматеріал. • Опрацювати базові методи розрахункових метрик даних • Підготовка даних для машинного навчання 	24	
<p>Вміти: Працювати з датафреймами у середовищі Python</p>	<p>Лабораторне заняття №4. Робота з даними на базі Pandas</p> <p>Мета: Отримати практичні навички для роботи з датафреймами. Завдання: використовуючи Jupyter Notebook виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> 1. Створення датафреймів та наповнення їх даними 2. Дослідження завантажених даних 3. Методи об'єднання датафреймів 4. Вирішення практичних задач у сфері роботи з даними 	22	32
<p>Тема 12. Основи аналізу статистичної інформації за допомогою Python</p>			
<p>Знати: інструменти передобробки даних у Python</p>	<p>Лекція № 5. Преобробка великих даних <i>План лекції:</i></p> <ol style="list-style-type: none"> 1. Великі дані (Big Data). 2. Засоби предобробки великих даних. (презентація, відео для самотійної 	6	

	<p>роботи)*</p> <p>3. Особливості роботи з великими даними(презентація, відео для самостійної роботи)*</p> <p>4. Описова статистика великих даних</p> <p>5. Використання графіків у аналізі великих даних</p> <p>Список рекомендованих джерел: <i>Основний:</i> 1, 3, 4. <i>Додатковий:</i> 6, 7. <i>Інтернет-ресурси:</i> 10-12, 15, 18</p>		
	<p>Завдання для самостійної роботи:</p> <p>1. Вивчення матеріалу лекцій.</p> <p>2. Опрацювання та підготовка доповідей з питань:</p> <ul style="list-style-type: none"> • Вивчититеоретичнийлекційнийматеріал. • Познайомитись з основними напрямками використання великих даних 	13	
<p>Вміти: здійснювати початковий аналіз та роботи преобробку великих даних</p>	<p>Лабораторне заняття №5. Обробка великих даних у Python</p> <p>Мета: навчитись виконувати предобробку великих даних у Python.</p> <p>Завдання: використовуючи Jupyter Notebook виконати ситуаційне завдання по власному варіанту за наступними темами:</p> <ol style="list-style-type: none"> 1. Робота з великими даними 2. Проведення предобробки даних 3. Проведення описового статистичного аналізу 4. Графічний аналіз даних 	14	21
<p>Знати: Основні засоби статистичного аналізу у Python</p>	<p>Лекція № 6. Статистичний аналіз у сфері великих даних</p> <ol style="list-style-type: none"> 1. Аналіз номінативних даних. 2. <i>Кореляція та лінійна регресія (презентація, відео для самостійної роботи)*</i> 3. <i>Діагностика моделі(презентація, відео для самостійної роботи)*</i> 4. Логістична регресія <p>Список рекомендованих джерел: <i>Основний:</i> 1, 3, 4.</p>	6	

	<i>Додатковий: 6, 7. Інтернет-ресурси: 10-12, 15, 18</i>		
	Завдання для самостійної роботи: 1. Вивчення матеріалу лекцій. 2. Опрацювання та підготовка доповідей з питань: <ul style="list-style-type: none"> • Вивчитиматематичний апарат статистичного аналізу; • Ознайомитись з іншими сучасними популярними методами аналізу великих даних. 	13	
Вміти: проводити статистичний аналіз великих даних	Лабораторне заняття №6. Статистичний аналіз даних у Python Мета: навчитись виконувати статистичний аналіз даних у Python на прикладі великих даних. Завдання: використовуючи Jupyter Notebook виконати ситуаційне завдання по власному варіанту за наступними темами: <ol style="list-style-type: none"> 1. Номінативний аналіз даних 2. Регресійний аналіз даних 3. Кореляційний аналіз 	14	20
Підсумковий контроль – письмовий екзамен			
Разом за семестр		180	100
Разом		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. Основи програмування. Python. Частина 1 [Електронний ресурс]: підручник для студ. спеціальності 122 "Комп'ютерні науки", спеціалізації "Інформаційні технології в біології та медицині" / А. В. Яковенко ; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. 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/>