ДЕРЖАВНА СЛУЖБА УКРАЇНИ З НАДЗВИЧАЙНИХ СИТУАЦІЙ

ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ БЕЗПЕКИ ЖИТТЄДІЯЛЬНОСТІ

Навчально-науковий інститут цивільного захисту

Кафедра інформаційних технологій та систем електронних комунікацій

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ (СИЛАБУС)

«Об’єктно-орієнтоване програмування**»**

(назва навчальної дисципліни)

Рівень вищої освіти: перший (бакалаврський), заочна форма навчання

Галузь знань: F Інформаційні технології

Спеціальність (*спеціалізація*) : F3 Комп’ютерні науки

Спеціалізація *(за наявності):*

Освітня програма: Комп’ютерні науки

Львів 2026

РОЗРОБНИК
Доцент кафедри інформаційних технологій та систем електронних комунікацій
кандидат технічних наук
_____ Роман ГОЛОВАТИЙ
(підпис) (ім’я, прізвище)
«___» _____ 20__ р.

ЗАТВЕРДЖЕНО
Начальник кафедри інформаційних технологій та систем електронних комунікацій
кандидат технічних наук, доцент
_____ Назарій БУРАК
(підпис) (ім’я, прізвище)
«___» _____ 20__ р.

Заступник начальника кафедри інформаційних технологій та систем електронних комунікацій
доктор філософії
_____ Юлія НАЗАР
(підпис) (ім’я, прізвище)
«___» _____ 20__ р.

ПОГОДЖЕНО
Гарант освітньої програми «Комп’ютерні науки» першого (бакалаврського) рівня вищої освіти
кандидат технічних наук, доцент

_____ Ольга СМОТР
(підпис) (ім’я, прізвище)
«___» _____ 20__ р.

ПОГОДЖЕНО
Тимчасово виконуючий обов'язки заступника начальника навчально-наукового інституту цивільного захисту
кандидат технічних наук, доцент

_____ Роман ВЕСЕЛІВСЬКИЙ
(підпис) (ім’я, прізвище)
«___» _____ 20__ р.

Розглянуто та затверджено на засіданні кафедри інформаційних технологій та систем електронних комунікацій, протокол від «28» серпня 2026 № 1

Розглянуто Вченою радою навчально-наукового інституту цивільного захисту, протокол від «28» серпня 2026 № 1

Актуалізовано:

Дата перегляду / внесення змін					
Підпис					
Ім’я, прізвище викладача					

1. Загальна інформація

Назва дисципліни	Об’єктно-орієнтоване програмування
Статус дисципліни	Нормативна
Рік навчання, семестр	2-й рік (3-й та 4-й семестри), заочна форма навчання
Мова викладання	українська
Викладачі	Головатий Роман Русланович, доцент кафедри інформаційних технологій та систем електронних комунікацій, кандидат технічних наук Назар Юлія Сергіївна, заступник начальника кафедри інформаційних технологій та систем електронних комунікацій, доктор філософії
Контактний телефон	
Електронна пошта	r.golovatyi@ldubgd.edu.ua y.kordunova@ldubgd.edu.ua
Сторінка курсу в ВУ	III семестр http://virt.ldubgd.edu.ua/course/view.php?id=1771 IV семестр http://virt.ldubgd.edu.ua/course/view.php?id=2192
Консультації	Згідно з розкладом консультацій кафедри інформаційних технологій та систем електронних комунікацій

2. Анотація до курсу

Освітня програма підготовки бакалавра з спеціальності «Комп’ютерні науки» передбачає оволодіння здобувачами вищої освіти низки фахових компетенцій в області програмування, досягнення яких організовано шляхом вивчення низки курсів, а саме «Основи програмування», «Основи Web-розробки», «Дискретна математика», «Чисельні методи», «Операційні системи та системне програмування», «Клієнт-серверне програмування», «Алгоритми та структури даних», «Web-технології та web-дизайн», «Об’єктне моделювання прогармних систем», «Програмування для мобільних платформ», «Людино-машинна взаємодія» тощо. Проте одним із основних курсів, який являється фундаментальним для оволодіння результатами навчання щодо проєктування та розробки програмних систем, є курс «Об’єктно-орієнтоване програмування». Сам курс опирається на результати навчання, що були отримані в рамках 2-ї частини курсу «Основи програмування».

Предметом вивчення навчального курсу є інструментальні засоби та технології створення програмних систем за об’єктно-орієнтованою парадигмою програмування. В ході курсу здобувачі вивчають тонкощі застосування об’єктно-орієнтованого підходу до побудови програмних систем використовуючи як інструмент мову програмування Java та середовища розробки Eclipse і IntelliJ IDEA. За результатами вивчення курсу здобувачі готують командні проекти (в окремих випадках індивідуальні) у вигляді реалізації програмної системи за об’єктно-орієнтованою парадигмою.

3. Мета і завдання курсу

Мета навчальної дисципліни “Об’єктно-орієнтоване програмування” – поглиблення знань здобувачів освіти з технології об’єктно-орієнтованого програмування, принципами модульної побудови програмних систем, детальне вивчення етапів розробки, збору та розгортання програмного забезпечення, а також розроблення програмного застосунку в межах виконання курсової роботи.

Завдання:

- формування поглиблених знань з об’єктно-орієнтованого проєктування програмних систем, їх прототипування використовуючи як засіб реалізації мову програмування Java, середовища розробки Eclipse та IntelliJ IDEA, а також фреймворк Spring.
- засвоєння поняття про архітектуру програмних систем, основних етапів їх розробки, збирання, розгортання та підтримки;
- розвиток навиків командної роботи під час реалізації командних проєктів (колективне виконання курсової роботи).

Процес вивчення дисципліни спрямований на формування елементів таких **компетентностей**:

загальних:

- ЗК08 Здатність генерувати нові ідеї (креативність);
- ЗК09 Здатність працювати в команді.

спеціальних/фахових (СК/ФК):

- СК08 Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об’єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

програмні результати навчання:

- РН1 Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук;
- РН9 Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв’язання задач в галузі комп’ютерних наук;
- РН14 Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

У результаті вивчення навчальної дисципліни здобувач вищої освіти повинен:

знати:

- поняття класів та екземплярів класів в ООП;
- особливості застосування методів в ОПП;
- поняття змінних екземплярів класів та конструкторів;
- поняття констант, статичних змінних і методів в ООП;
- поняття внутрішніх та вкладених класів;
- різновиди модифікаторів доступу та особливості їх застосування;
- поняття інкапсуляції в програмуванні;
- поняття поліморфізму в програмуванні;
- поняття наслідування в програмуванні;

- поняття «енамів» та «інтерфейсів» в програмуванні мовою Java;
- основні поняття про колекції та бібліотеки в програмуванні;
- основи об’єктно-орієнтованого моделювання та мови UML;
- особливості побудови та використання діаграм класів;
- винятки в програмуванні, перехоплення та опрацювання винятків;
- особливості реалізації багатопотокового програмування;
- принципи застосування потоків введення-виведення, збереження даних, запису даних у файл, чтитування з файлу тощо;
- основні принципи функціонального програмування та застосування лямбда-виразів;
- принципи побудови REST API;
- основи використання Spring Boot;
- основи мови SQL та принципи взаємодії Java-застосунків із базами даних засобами JDBC;
- особливості управління залежностями за допомогою Maven.

вміти:

- застосовувати методи абстрактного мислення, аналізу та синтезу при побудові алгоритмів та їх реалізації;
- застосовувати отримані знання з основ процедурного та об’єктно-орієнтованого програмування в практичних ситуаціях;
- самостійно вчитись та оволодівати сучасними підходами, механізмами та технологіями створення програмних продуктів;
- генерувати нові ідеї та бути визначеним і наполегливим в процесі вирішення прикладних завдань;
- розробляти програмне забезпечення із застосуванням об’єктно-орієнтованої парадигм програмування;
- проєктувати та реалізовувати REST API засобами Spring Boot;
- виконувати SQL-запити та організовувати доступ до баз даних засобами JDBC;
- застосовувати інструменти автоматизації збірки Maven.

4. Передумови для вивчення дисципліни

Передумовами для вивчення курсу є оволодіння здобувачами вищої освіти низки фахових компетенцій в області програмування, досягнення яких організовано шляхом вивчення курсів “Основи програмування”, «Основи Web-розробки», «Дискретна математика».

5. Формат і обсяг дисципліни

Формат курсу	Навчання здійснюється за заочною формою. Навчальний матеріал дисципліни структурований за двома змістовими модулями і складається з 22 тем, які є логічно завершеними, відносно самостійними, цілісними частинами. Під час настановних та екзаменаційних сесій проводяться лекційні (8 год), практичні (4 год) та лабораторні (4 год) заняття, на яких розглядаються ключові теми курсу та демонструються інструменти розробки. У міжсесійний період здобувачі освіти самостійно опрацьовують теоретичний матеріал, проходять тестові завдання на базі електронного освітнього середовища та виконують практичні й лабораторні роботи відповідно до методичних рекомендацій, що розміщені на платформі електронного освітнього середовища «Віртуальний університет»; результати робіт подаються на перевірку через

	електронний курс. Підсумковим етапом вивчення курсу «Об’єктно-орієнтоване програмування» є написання курсові роботи, ключовим завданням якої є реалізація програмної системи за об’єктно-орієнтованою моделлю. Курсова робота може виконуватись за індивідуальним завданням згідно методичних вказівок, або в складі команди R&D проєкту за попереднім погодженням із викладачем. Методичні вказівки для виконання курсової роботи розміщені на платформі електронного освітнього середовища «Віртуальний університет». В процесі вивчення курсу здобувачі освіти можуть реалізовувати власні студентські R&D проєкти в рамках виконання курсової роботи. Здобувачам освіти можливе перезарахування окремих тем та індивідуальних робіт за умови проходження он-лайн/очних курсів “Java Core” або “Java Advanced” (до прикладу від SoftServe IT Academy, Java BaseCamp від GlobalLogic, Epam Intership та споріднені курси інших компаній) та представлення сертифікату про проходження курсів із переліком вивчених тем (опрацьованого контенту, реалізованого проєкту тощо). Перезарахування відбувається за індивідуальним узгодженням із викладачем та висвітлюється в електронному журналі курсу.
Обсяг дисципліни:	6,5 кредитів / 195 академічних годин, з яких: лекції – 8 год., практичні – 4 год., лабораторні – 4 год., самостійна робота – 179 год.
Форми навчання	лекції, практичні та лабораторні заняття під час сесій, консультації, самостійна робота (в т. ч. здача тестових завдань та виконання лабораторних робіт та подання їх результатів на перевірку), робота в командах, виконання курсової роботи.

6. Тематика та зміст курсу

Назви змістових модулів і тем	Усього	Кількість годин			
			у тому числі		
		Лекції	Практичні	Лабораторні	Самостійна робота
3-ий семестр					
Тема №1. Базові поняття об'єктно-орієнтованого програмування. Класи, методи, конструктори	8	-	-	-	8
Тема №2. Сумісне застосування наслідування та поліморфізму	8	-	-	-	8
Тема №3. Абстракція	8	-	-	-	8
Тема №4. Сумісне застосування парадигм ООП при створенні програмних систем	12	2	2	-	8
Тема №5. Принципи SOLID та якість об'єктно-орієнтованого проєктування	8	-	-	-	8
Тема №6. Колекції. Реалізація різних структур даних (пакет java.util)	12	2	2	-	8
Тема №7. Узагальнення (Generics). Сортування даних. Comparable та Comparator	8	-	-	-	8
Тема №8. Оброблення винятків	8	-	-	-	8
Тема №9. Потоки вводу-виводу	8	-	-	-	8
Тема №10. Робота з рядками та регулярними виразами	8	-	-	-	8
Тема №11. Багатопотокове програмування	8	-	-	-	8
Тема №12. Робота з датою та часом (java.time)	8	-	-	-	8
Тема №13. Архітектура чат-боту. Розгортання проєкту Java	8	-	-	-	8
Тема №14. Повторення вивченого за семестр	8	-	-	-	8
Усього годин за семестр	**120**	**4**	**4**	**0**	**112**
4-ий семестр					
Тема №15. Вступ до Stream API. Лямбда-вирази	8	-	-	-	8
Тема №16. Stream API	12	2	-	2	8
Тема №17. Інструмент збірки проєктів Maven. UML-діаграми класів	8	-	-	-	8
Тема №18. Анотації в Java. Reflection API	8	-	-	-	8
Тема №19. JDBC+SQL	9	-	-	-	9
Тема №20. Основи фреймворку Spring Boot	9	-	-	-	9
Тема №21. Розроблення REST API засобами Spring Boot	12	2	-	2	8
Тема №22. Повторення та підсумкове проєктування програмної системи	9	-	-	-	9
Усього годин за семестр	**75**	**4**	**0**	**4**	**67**
Усього годин	**195**	**8**	**4**	**4**	**179**

7. Інформаційний обсяг навчальної дисципліни

7.1. Теми лекційних занять

№	Назва теми	Кількість годин
1.	Об’єктно-орієнтована парадигма програмування: класи, наслідування, поліморфізм, абстракція, принципи SOLID (оглядова лекція за темами №1–5)	2
2.	Колекції та узагальнення. Реалізація структур даних засобами пакета java.util (оглядова лекція за темами №6–7)	2
3.	Функціональне програмування в Java: лямбда-вирази та Stream API (оглядова лекція за темами №15–16)	2
4.	Робота з базами даних та розроблення REST API засобами Spring Boot (оглядова лекція за темами №19–21)	2
	Разом	8

Тема №1. Базові поняття об'єктно-орієнтованого програмування. Класи, методи, конструктори.

Введення в класи. Структура класу. Введення в методи. Типізовані методи, void-методи, методи, що приймають параметри. Конструктори. Пусті конструктори, генерування конструкторів. Створення власних класів з унікальною логікою. Застосування конструкторів. Створення власних методів з унікальною логікою.

Тема №2. Сумісне застосування наслідування та поліморфізму

Введення в концепції наслідування та поліморфізму. Опис, як створювати класи-нащадки. Застосування методу super для доступу до батьківського класу. Поліморфізм: динамічне визначення методів. Перевизначення методів батьківських класів. Створення програмних систем з використанням наслідування та поліморфізму.

Тема №3. Абстракція

Вивчення принципів абстракції в Java. Створення та використання абстрактних класів і методів. Огляд інтерфейсів та їх ролі у проєктуванні програмних систем. Реалізація множинної поведінки за допомогою інтерфейсів. Порівняння абстрактних класів та інтерфейсів, особливості їх застосування у розробці програмного забезпечення.

Тема №4. Сумісне застосування парадигм ООП при створенні програмних систем

Огляд ключових принципів об'єктно-орієнтованого програмування: інкапсуляції, наслідування, поліморфізму та абстракції. Проєктування програмних систем із використанням об'єктно-орієнтованого підходу. Реалізація взаємодії між класами та об'єктами. Використання композиції та агрегації. Застосування шаблонів проєктування для побудови гнучких та масштабованих програмних систем.

Тема №5. Принципи SOLID та якість об'єктно-орієнтованого проєктування

Огляд принципів SOLID як основи якісного проєктування програмного забезпечення. Принцип єдиної відповідальності (SRP), відкритості-закритості (OCP), підстановки Лісков (LSP), розділення інтерфейсів (ISP) та інверсії залежностей (DIP). Виявлення типових проблем архітектури програмних систем та їх усунення шляхом рефакторингу. Аналіз прикладів якісного та неякісного проєктування класів.

Тема №6. Колекції. Реалізація різних структур даних (пакет java.util)

Огляд ієрархії колекцій пакета java.util. Вивчення інтерфейсу List та його реалізацій ArrayList і LinkedList: додавання, видалення, пошук і сортування елементів. Вивчення інтерфейсу Set та його реалізацій HashSet, LinkedHashSet і TreeSet: особливості зберігання унікальних елементів, принципи хешування. Вивчення інтерфейсу Map та його реалізацій HashMap, LinkedHashMap і TreeMap: робота з парами «ключ-значення». Використання ітераторів для обходу колекцій. Практичне застосування різних структур даних для зберігання та опрацювання інформації.

Тема №7. Узагальнення (Generics). Сортування даних. Comparable та Comparator

Вивчення механізму узагальнень (Generics) у Java. Створення узагальнених класів, методів та колекцій. Переваги типобезпечності під час роботи з даними. Сортування об'єктів за допомогою інтерфейсів Comparable та Comparator. Реалізація власних критеріїв сортування та впорядкування колекцій.

Тема №8. Оброблення винятків

Основи обробки винятків в Java. Створення блоків try-catch для перехоплення винятків. Ключові слова throw та throws. Створення власних класів винятків. Використання множинних блоків catch і нарощування обробки винятків у програмах.

Тема №9. Потоки вводу-виводу

Введення в Java I/O. Робота з файлами за допомогою класів File, FileReader, FileWriter, BufferedReader та BufferedWriter. Читання і запис текстових та бінарних даних. Робота з потоками байтів і символів. Використання буферизації для підвищення ефективності операцій введення-виведення.

Тема №10. Робота з рядками та регулярними виразами

Вивчення класу String та основних операцій над рядками. Методи обробки текстової інформації. Використання класів StringBuilder та StringBuffer для оптимізації роботи з рядками. Основи регулярних виразів. Пошук, перевірка та заміна текстових фрагментів за допомогою шаблонів.

Тема №11. Багатопотокове програмування

Основи багатопотокового програмування в Java. Створення потоків виконання засобами класу Thread та інтерфейсу Runnable. Життєвий цикл потоку. Синхронізація доступу до спільних ресурсів: ключове слово synchronized, монітори. Проблеми конкурентного виконання (стан гонитви, взаємне блокування) та шляхи їх уникнення. Використання пулів потоків (ExecutorService) для ефективного управління паралельним виконанням завдань.

Тема №12. Робота з датою та часом (java.time)

Вивчення сучасного API для роботи з датою та часом у Java. Класи LocalDate, LocalTime, LocalDateTime, Duration та Period. Форматування та перетворення дат. Виконання арифметичних операцій над датами й часом. Порівняння застарілих і сучасних засобів роботи з датою та часом.

Тема №13. Архітектура чат-боту. Розгортання проєкту Java

Огляд архітектурних підходів до побудови чат-ботів засобами Java: взаємодія з API месенджера, обробка вхідних повідомлень та формування відповідей, організація логіки діалогу. Основи пакування Java-застосунку (побудова виконуваного jar-файлу), налаштування середовища виконання та конфігурації застосунку. Огляд підходів до розгортання (deployment) Java-застосунків на сервері.

Тема №14. Повторення вивченого за семестр

Узагальнення та систематизація теоретичного і практичного матеріалу, опрацьованого протягом семестру: базові поняття ООП, наслідування, поліморфізм, абстракція, принципи SOLID, колекції, узагальнення, обробка винятків, потоки введення-виведення, багатопотокове програмування. Розбір типових помилок і складних питань. Підготовка здобувачів до проходження підсумкового тестування за темами семестру.

Тема №15. Вступ до Stream API. Лямбда-вирази

Основи функціонального програмування в Java. Створення та використання лямбда-виразів. Функціональні інтерфейси та їх застосування. Методичні посилання (method references) та їх використання. Скорочення та спрощення програмного коду за допомогою функціонального підходу. Вступ до Stream API як інструменту функціональної обробки колекцій даних.

Тема №16. Stream API

Поглиблене вивчення Stream API для роботи з потоками даних. Створення потоків із колекцій та масивів. Методи обробки потоків: filter(), map(), reduce(). Паралельні потоки: прискорення обробки великих обсягів даних. Використання потоків для трансформації та агрегації даних.

Тема №17. Інструмент збірки проєктів Maven. UML-діаграми класів

Введення в систему автоматизації збірки Maven. Структура Maven-проєкту. Файл конфігурації pom.xml. Підключення та управління зовнішніми бібліотеками й залежностями. Життєвий цикл проєкту Maven. Використання Maven для компіляції, тестування та пакування Java-застосунків. Вивчення основ UML (Unified Modeling Language). Створення діаграм класів для візуалізації структури програмних систем. Визначення атрибутів, методів та зв'язків між класами. Використання UML під час проєктування програмного забезпечення.

Тема №18. Анотації в Java. Reflection API

Основи створення та використання анотацій у Java. Принципи роботи з Reflection API для динамічного аналізу класів, методів та полів під час виконання програми. Застосування анотацій та рефлексії у розробці гнучких і масштабованих Java-додатків.

Тема №19. JDBC+SQL

Основи мови SQL: створення таблиць, вибірка, додавання, оновлення та видалення даних. Основи роботи з базами даних через JDBC (Java Database Connectivity). Підключення до систем управління базами даних. Використання класів Connection, Statement, PreparedStatement та ResultSet. Виконання SQL-запитів засобами Java. Управління транзакціями та інтеграція баз даних із Java-застосунками.

Тема №20. Основи фреймворку Spring Boot

Введення у фреймворк Spring Boot та його призначення. Створення та налаштування Spring Boot-проєкту. Основні компоненти Spring Boot: контролери, сервіси та репозиторії. Інверсія керування (IoC) та впровадження залежностей (Dependency Injection). Використання Spring Boot для створення серверної частини застосунків.

Тема №21. Розроблення REST API засобами Spring Boot

Основи концепції REST та RESTful веб-сервісів. Використання HTTP-методів GET, POST, PUT та DELETE. Створення контролерів для обробки запитів клієнтів. Передача даних у форматі JSON. Тестування REST API та інтеграція веб-сервісів із клієнтськими застосунками.

Тема №22. Повторення та підсумкове проєктування програмної системи

Узагальнення та систематизація знань, отриманих протягом курсу. Аналіз ключових технологій та підходів до розробки Java-застосунків. Проєктування структури програмної системи з використанням принципів ООП, UML, баз даних та веб-технологій. Підготовка до підсумкового оцінювання, захисту курсових робіт та закріплення практичних навичок розробки програмного забезпечення.

8. Теми практичних, семінарських та лабораторних занять

8.1. Теми практичних занять

№	Назва теми	Кількість годин
1.	Розроблення цілісної об’єктно-орієнтованої програми (класи, наслідування, поліморфізм, інтерфейси)	2
2.	Реалізація структур даних засобами колекцій List, Set, Map	2
	Разом	4

8.2. Теми семінарських занять

«Семінарські заняття не передбачені»

8.3. Теми лабораторних занять

Лабораторні заняття з дисципліни проводяться лише в 4-му семестрі.

№	Назва теми	Кількість годин
1.	Опрацювання даних засобами лямбда-виразів та Stream API	2
2.	Розроблення веб-сервісу (REST API) засобами Spring Boot з доступом до бази даних	2
	Разом	4

9. Завдання для самостійного опрацювання

№	Назва теми	Кількість годин
1.	Опрацювати поняття класів, об'єктів, методів і конструкторів. Створити клас для опису об'єкта обраної предметної області.	8
2.	Дослідити особливості наслідування та поліморфізму в Java. Підготувати приклади використання перевизначення методів.	8
3.	Порівняти абстрактні класи та інтерфейси. Навести приклади їх застосування.	8
4.	Проаналізувати приклади використання композиції та агрегації в програмних системах.	8
5.	Дослідити принципи SOLID та підготувати приклади порушення кожного з них.	8
6.	Опрацювати реалізації колекцій List, Set, Map пакета java.util. Порівняти ArrayList/LinkedList, HashSet/TreeSet, HashMap/TreeMap.	8
7.	Дослідити можливості Generics та механізми сортування колекцій за допомогою Comparable і Comparator.	8
8.	Опрацювати ієрархію винятків у Java. Навести приклади перевірюваних та неперевірюваних винятків.	8
9.	Ознайомитися з класами пакета java.io та java.nio для роботи з файлами та потоками введення-виведення.	8
10.	Дослідити можливості регулярних виразів для пошуку та перевірки текстових даних.	8
11.	Опрацювати засоби пакета java.util.concurrent. Підготувати приклади синхронізації потоків та роботи з ExecutorService.	8
12.	Опрацювати класи пакета java.time та підготувати приклади роботи з датами й часом.	8
13.	Дослідити підходи до побудови чат-ботів засобами Java та способи пакування й розгортання Java-застосунків.	8
14.	Підготувати конспект повторення матеріалу семестру та відповіді на контрольні питання за вивченими темами.	8
15.	Опрацювати стандартні функціональні інтерфейси пакета java.util.function та основи лямбда-виразів.	8
16.	Дослідити додаткові операції Stream API та особливості паралельних потоків даних.	8
17.	Ознайомитися зі структурою Maven-проєкту та життєвим циклом збірки. Створити UML-діаграму класів для обраної предметної області.	8
18.	Дослідити застосування анотацій та Reflection API у сучасних Java-фреймворках.	8
19.	Опрацювати основи мови SQL та проєктування реляційних баз даних.	9
20.	Ознайомитися з екосистемою Spring Framework та основними модулями Spring Boot.	9
21.	Дослідити принципи побудови REST API та особливості використання HTTP-протоколу.	8
22.	Підготувати структурну схему програмної системи з використанням технологій, вивчених протягом курсу.	9
	Разом	179

10. Методи навчання

Основні форми організації навчання за заочною формою: лекції, практичні та лабораторні заняття під час сесій; самостійна робота в міжсесійний період із використанням електронного курсу «Віртуального університету»; виконання курсової роботи; командна робота; консультації, у тому числі онлайн у MS Teams.

Методи організації та здійснення навчально-пізнавальної діяльності:

- лекції – словесні та наочні методи навчання із елементами мозкового штурму;
- практичні та лабораторні роботи – репродуктивний метод, метод проблемного викладу матеріалу, частково-пошуковий метод навчання;
- курсова робота – дослідницький метод, частково-пошуковий метод навчання, робота в групах (командна робота);
- консультації – словесний та дискусійний методи.

11. Технічне й програмне забезпечення /обладнання

Комп’ютери на базі процесорів Intel Pentium Gold G5400 або Intel Core i5 (i7, i9), компоненти програмного забезпечення MS Office 365 (Teams, PowerPoint, Word,), інтегровані середовища розробки: Eclipse, IntelliJ IDEA, комплект розробника java (JDK), система контролю версій Git, інтерактивна система управління комп'ютерами Veyon, електронне освітнє середовище “Віртуальний університет” (на базі платформи Moodle) та іншіі програмні засоби навчання.

12. Критерії оцінювання

Оцінювання результатів навчання здобувачів вищої освіти здійснюється відповідно до «Положення про організацію освітнього процесу у ЛДУ БЖД» https://drive.google.com/file/d/1JIjfqHQMV83poyjEENbqM6pYE9lxxhNs/view та «Положення про порядок та критерії оцінювання результатів навчання здобувачів вищої освіти у ЛДУ БЖД» https://drive.google.com/file/d/1lCGnfmr0hNmvE_YeDA96cxCsMLQsu3cE/view

Поточний контроль	
Поточний контроль проводитися у формі виконання тестових завдань на базі платформи електронного освітнього середовища «Віртуальний університет», усного опитування, виконання практичних робіт та їх захисту на оцінку. Оцінювання результатів поточного контролю здійснюється за національною (4 бальною) шкалою із подальшим переведенням у 100-бальну шкалу. Результати поточного контролю (поточна успішність) враховуються викладачем при визначені допуску до підсумкового контролю.	
3-й семестр	
Вид робіт	**Формат проведення та критерії оцінювання**
Тестові завдання	У першій частині курсу передбачено проходження 7 тестових завдань за відповідними темами дисципліни. За успішне виконання кожного тестового завдання здобувач освіти може отримати 5 балів. Максимальна кількість балів за тестові завдання становить **35 балів.**
Практичні та	Першою частиною курсу передбачено виконання та обов’язковий

лабораторні роботи	захист 10 практичних робіт (частина – під час аудиторних занять сесії, решта – у міжсесійний період із поданням результатів через «Віртуальний університет»), за результатами яких здобувачі освіти можуть отримати сумарно 50 балів. Ще **15 балів** здобувачі освіти можуть отримати за роботу протягом семестру: активність на лекційних, практичних та лабораторних заняттях, участь в олімпіадах, наукових конференціях, конкурсах студентських наукових робіт та інших формах наукової й освітньої діяльності. **Максимальна кількість балів за поточний контроль становить 100 балів.**
4-й семестр	
Вид робіт	**Формат проведення та критерії оцінювання**
Тестові завдання	У другій частині курсу передбачено проходження 6 тестових завдань за відповідними темами дисципліни та одного підсумкового тестування. За успішне виконання кожного тестового завдання здобувач освіти може отримати 5 балів. Максимальна кількість балів за виконання тестових завдань становить 30 балів. Підсумкове тестування оцінюється максимально у 20 балів. Загальна кількість балів за тестовий контроль становить **50 балів.**
Практичні та лабораторні роботи	У межах другої частини курсу передбачено виконання 6 практичних (лабораторних) робіт (частина – під час аудиторних занять сесії, решта – у міжсесійний період із поданням результатів через «Віртуальний університет»). За результатами виконання та захисту обраних робіт здобувачі освіти можуть отримати сумарно до 30 балів. Ще **20 балів** здобувачі освіти можуть отримати за роботу протягом семестру: активність на лекційних, практичних та лабораторних заняттях, участь в олімпіадах, наукових конференціях, конкурсах студентських наукових робіт та інших формах наукової й освітньої діяльності. **Максимальна кількість балів за поточний контроль становить 100 балів.** Другою частиною курсу також передбачено виконання та захист курсової роботи. Методичні рекомендації та критерії оцінювання курсової роботи наведені в методичних вказівках та розміщені на платформі електронного освітнього середовища «Віртуальний університет»

Курсова робота
Під час оцінювання курсової роботи до уваги приймаються три критерії, що в сумі формують загальну оцінку: якість реалізації програмного продукту ($K_{код}$); якість оформлення пояснювальної записки ($K_{записка}$); рівень презентації та захисту результатів власної роботи ($K_{захист}$). Оцінювання проводиться за 100-бальною шкалою за формулою $K = K_{код} + K_{записка} + K_{захист}$. **Оцінювання програмного продукту** $K_{код}$ (діапазон балів 0-50): - 40 – 50 балів – програмний продукт повністю відповідає усім вимогам методичних вказівок, відтворює функціональність за отриманим тематичним напрямком, характеризується стійкістю (захистом від некоректного введення даних, стійкістю роботи тощо); - 30 – 40 балів – програмний продукт містить незначні неточності у реалізації

принципів об’єктно-орієнтованого підходу, алгоритм або діаграма класів не в повній мірі відповідають поданому коду, або структура програмного продукту частково не відповідає вимогам методичних вказівок. Загалом продукт є працездатним та відповідає тематичному напрямку;

- 20 – 30 балів – програмний продукт працездатний проте з суттєвими зауваженнями щодо стійкості його роботи. Архітектура та реалізація програмної системи не відповідає поставленим у методичних вказівках вимогам (не більше ніж на 40%) або програмна реалізація частково не відповідає тематичному напрямку;

- 0 – 20 балів – програмний код не відповідає поставленим у методичних вказівках вимогам більше ніж на 40%, проте орієнтований на реалізацію ідеї згідно одержаного завдання. Програмна система не відтворює задекларованої функціональності та/або є частково не працездатно. Програмний продукт видає некоректні результати роботи та/або є не стійким при введенні різних вхідних даних;

- 0 балів: програмний продукт не представлено в повній мірі на захисті, він не є працездатним, або запозичений з відкритих ресурсів.

Оформлення пояснювальної записки $K_{записка}$ (діапазон балів 0-30):

- 25 – 30 балів: пояснювальна записка в повній мірі відповідає вимогам розділів 2 «Обов’язковий зміст основних розділів пояснювальної записки» та 6 «Вимоги до оформлення пояснювальної записки» методичних вказівок, містить увесь необхідний ілюстративний та графічний матеріал, подано усі лістинги програмних модулів, коментарі змістовно та лаконічно відтворюють функціонал окремих методів;

- 20 – 25 балів – пояснювальна записка відповідає вимогам розділів 2 «Обов’язковий зміст основних розділів пояснювальної записки» та 6 «Вимоги до оформлення пояснювальної записки» методичних вказівок не менше ніж на 80%, має незначні зауваження щодо повноти забезпечення ілюстративним та графічним матеріалом, або до якості його виконання, коментарі не в повній мірі відтворюють функціонал окремих модулів програм або мають неточності чи невідповідність поданим лістингам;

- 10 – 20 балів – робота має значні зауваження щодо якості оформлення та відповідності вимогам розділів 2 «Обов’язковий зміст основних розділів пояснювальної записки» і 6 «Вимоги до оформлення пояснювальної записки» методичних вказівок, проте всі обов’язкові складові (розділи) подано у пояснювальній записці. В пояснювальній записці наявні не всі лістинги програмного коду, коментарі сформульовано не зрозуміло та/або у невідповідності поданим лістингам;

- 0 – 10 балів – пояснювальна записка не містить усіх необхідних компонентів передбачених вимогам розділу 2 «Обов’язковий зміст основних розділів пояснювальної записки» методичних вказівок. Програмний код наведено не в повній мірі у результаті чого не можливо перевірити його працездатність. Графічний та ілюстративний матеріал відсутній або не відповідає програмному коду чи завданню на курсову роботу;

- 0 балів – пояснювальну записку не подано на рецензування, або вона не відповідає поставленому завданню та ключовим вимогам методичних вказівок.

Захист курсової роботи $K_{захист}$ (діапазон балів 0-20):

- 15 – 20 балів – здобувач освіти демонструє вільне володіння матеріалом курсової роботи, оперує термінами та демонструє розуміння основних понять об’єктно-орієнтованого програмування, презентована ідея та шляхи її реалізації є унікальними та креативними, вірно відповідає на усі додаткові запитання;

- 10 – 15 балів – здобувач відповідає основним критеріям на оцінку (15-20) проте під час відповідей на додаткові або уточнюючі питання допускає певні неточності;

- 0 – 10 балів – під час презентації допускаються певні неточності або невідповідності поданого в пояснювальній записці матеріалу, при наданні додаткових або уточнювальних питань відповіді є не вірними і не точними, що може свідчити про виконання роботи з використанням готових алгоритмів без їх творчого опрацювання;

- 0 балів – здобувач не може пояснити поданий на захист матеріал та коректно його прокоментувати, що може свідчити про несамостійність виконаної роботи.

Здобувач, що отримав 0 балів з першого або третього критерію, вважається таким що не виконав завдання на курсову роботу та обирає нову тему з її подальшим перезахистом.

Загальна оцінка за курсову роботу з 100-бальної шкали переводиться у національну шкалу у відповідності до наведеної шкали оцінювання результатів навчання (подано далі)

Підсумковий контроль

3-ий семестр

Семестровий контроль проводиться у формі екзамену. До підсумкового контролю допускаються здобувачі освіти, які мають належний рівень поточних знань та одержали понад **60** зі **100** можливих балів за результатами проходження курсу (поточного оцінювання) на базі електронного освітнього середовища «Віртуальний університет».

Одержання необхідної кількості балів поточного контролю можливе за рахунок проходження тестових завдань (теоретична частина) та виконання і захисту практичних та/або лабораторних робіт (практична частина). Розподіл між запропонованими складовими поточного контролю знань за навчальний семестр наступний: тестові завдання – до **35 балів**, виконання та захист практичних та лабораторних робіт – **до 50 балів**. Ще **15 балів** студенти можуть отримати за роботу під час семестру (активність на лекціях та практичних/лабораторних заняттях, участь у конференціях та наукових роботах).

Під час виставлення підсумкової екзаменаційної оцінки результати поточного контролю перераховуються у 30-бальну шкалу шляхом **множення суми балів, отриманих здобувачем освіти протягом семестру, на коефіцієнт 0,3.** Таким чином, до підсумкової оцінки може бути зараховано не більше **30 балів** за результатами поточного контролю.

Екзамен (максимально 70 балів) складається із теоретичної та практичної частини (50 балів теоретична частина та 20 – практична) і проводиться на базі електронного освітнього середовища «Віртуальний університет». На теоретичну частину виноситься **50 випадково згенерованих тестових питань** закритої форми з вибором однієї правильної відповіді за матеріалами семестрової частини курсу. Вага кожного питання тесту = 1 балу. Практична частина екзамену передбачає **розв'язання практичного завдання шляхом написання програмного коду відповідно до отриманого завдання.** Завдання практичної частини екзамену автоматично генерується засобами електронного освітнього середовища «Віртуальний університет» із банку завдань, сформованого відповідно до програмних результатів навчання та змісту дисципліни. Практичне завдання є індивідуальним для кожного здобувача освіти та спрямоване на перевірку рівня сформованості практичних навичок застосування об'єктно-орієнтованого підходу, використання основних засобів мови програмування Java та вміння розробляти програмний код для розв'язання прикладних задач.

Критерії оцінювання виконаних завдань:

- 20 балів – написаний код є працездатним, а структура та результат його виконання в повній мірі відповідають поставленому завданню;
- 15-20 балів – написаний код є працездатним, але його структура або результат виконання не в повній мірі відповідають поставленому завданню;
- 10-15 балів – написаний код є працездатним, але його структура або результат виконання лише частково відтворюють умову поставленого завдання;
- 5-10 балів – написаний код не працездатний, проте в його структурі спостерігається вірно обраний напрям вирішення завдання;
- 0-5 балів – завдання не виконане або не відповідає поставленому завданню.

4-ий семестр

Семестровий контроль проводиться у формі екзамену. До підсумкового контролю допускаються здобувачі освіти, які мають належний рівень поточних знань та одержали понад **60** зі **100** можливих балів за результатами проходження курсу (поточного оцінювання) на базі електронного освітнього середовища «Віртуальний університет».

Одержання необхідної кількості балів поточного контролю можливе за рахунок проходження тестових завдань (теоретична частина) та виконання і захисту практичних та/або лабораторних робіт (практична частина). Розподіл між запропонованими складовими поточного контролю знань за навчальний семестр наступний: тестові завдання – до **50 балів**, виконання та захист практичних та лабораторних робіт – **до 30 балів**. Ще **20 балів** студенти можуть отримати за роботу під час семестру (активність на лекціях та практичних/лабораторних заняттях, участь у конференціях та наукових роботах).

Під час виставлення підсумкової екзаменаційної оцінки результати поточного контролю перераховуються у 30-бальну шкалу шляхом **множення суми балів, отриманих здобувачем освіти протягом семестру, на коефіцієнт 0,3.** Таким чином, до підсумкової оцінки може бути зараховано не більше **30 балів** за результатами поточного контролю.

Екзамен (максимально 70 балів) складається із теоретичної та практичної чатини (50 балів теоретична частина та 20 – практична) і проводиться на базі електронного освітнього середовища «Віртуальний університет». На теоретичну частину виноситься **50 випадково згенерованих тестових питань** закритої форми з вибором однієї правильної відповіді за матеріалами семестрової частини курсу. Вага кожного питання тесту = 1 балу. Практична частина екзамену передбачає **розв'язання практичного завдання шляхом написання програмного коду відповідно до отриманого завдання.** Завдання практичної частини екзамену автоматично генерується засобами електронного освітнього середовища «Віртуальний університет» із банку завдань, сформованого відповідно до програмних результатів навчання та змісту дисципліни. Практичне завдання є індивідуальним для кожного здобувача освіти та спрямоване на перевірку рівня сформованості практичних навичок застосування об'єктно-орієнтованого підходу, використання основних засобів мови програмування Java та вміння розробляти програмний код для розв'язання прикладних задач.

Критерії оцінювання виконаних завдань:

- 20 балів – написаний код є працездатним, а структура та результат його виконання в повній мірі відповідають поставленому завданню;
- 15-20 балів – написаний код є працездатним, але його структура або результат виконання не в повній мірі відповідають поставленому завданню;
- 10-15 балів – написаний код є працездатним, але його структура або результат виконання лише частково відтворюють умову поставленого завдання;
- 5-10 балів – написаний код не працездатний, проте в його структурі спостерігається вірно обраний напрям вирішення завдання;
- 0-5 балів – завдання не виконане або не відповідає поставленому завданню.

Підсумкову семестрову оцінку розраховують як суму балів теоретичної та практичної частини за 100-бальною шкалою і переводяться в національну (чотирибальну) шкалу (“відмінно”, “добре”, “задовільно”, “незадовільно”, для заліків – “зараховано”, “не зараховано”).

Підсумкові оцінки заносяться до екзаменаційної відомості в національній, 100-бальній шкалі та шкалі ЄКТС відповідно до співвідношень, поданих у наступній таблиці.

Шкала оцінювання результатів навчання здобувачів вищої освіти

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, диференційованого заліку, курсового проекту (роботи), практики	для заліку
91–100	A	відмінно	зараховано
81–90	B	добре	
71–80	C		
61–70	D	задовільно	
51–60	E		
36–50	FX	незадовільно	не зараховано
0–35	F		
Невиконання умов допуску до поточного контролю			не допущений

13. Політика курсу

Виконання навчальних завдань і робота на курсі має відповідати вимогам «Кодексу академічної доброчесності та корпоративної культури ЛДУ БЖД» (https://drive.google.com/file/d/1ya2if6epebCiJKxeKDw3Y2xbFbAwH9xL/view), «Положення про порядок та критерії оцінювання результатів навчання здобувачів вищої освіти у ЛДУ БЖД» (https://drive.google.com/file/d/1lCGnfmr0hNmvE_YeDA96cxCsMLQsu3cE/view) та «Положення про організацію освітнього процесу у ЛДУ БЖД» (https://drive.google.com/file/d/1JIjfqHQMV83poyjEENbqM6pYE9lxxhNs/view).

Академічні очікування від здобувачів: своєчасне виконання тестових завдань, передбачених силабусом дисципліни; обов'язкове відвідування аудиторних занять під час настановних та екзаменаційних сесій; виконання усіх індивідуальних завдань, передбачених силабусом курсу (практичні роботи та курсова робота); дотримання принципів академічної доброчесності під час усіх видів контролю.

Відвідування занять та відпрацювання пропусків: відвідування аудиторних занять під час сесій є обов'язковим. Пропуски занять поділяються на пропуски з поважної причини та без поважної причини. До поважних причин належать:

- хвороба, підтверджена медичною довідкою;
- участь у конкурсах, конференціях, олімпіадах, грантових програмах, спортивних змаганнях (за погодженням із деканатом).

За відсутності підтвердних документів пропуск заняття вважається таким, що стався без поважної причини. Пропущені практичні та лабораторні заняття відпрацьовуються в обсязі пропущених годин згідно з графіком консультацій кафедри інформаційних технологій та систем електронних комунікацій; пропущені лекційні теми опрацьовуються самостійно на платформі «Віртуальний університет». Відпрацювання пропущених занять має бути завершене до початку екзаменаційної сесії; здобувач, який не відпрацював пропуски у встановлений термін, не допускається до семестрового контролю з дисципліни.

Політика щодо термінів виконання завдань та ліквідації академічної заборгованості: терміни виконання завдань вказуються в електронному курсі «Віртуального університету». Після завершення терміну прийому завдань система може блокувати можливість їх завантаження для подальшої оцінки викладачем, окрім випадків, пов'язаних із поважними причинами, про що здобувач особисто повідомляє викладача. Відпрацювання академічної заборгованості з дисципліни можливе до дня проведення підсумкового контролю (залік, екзамен) згідно з графіком консультацій кафедри або за індивідуальними домовленостями з викладачем.

Недопущені до підсумкового контролю здобувачі освіти здійснюють перездачу в терміни, відведені для усунення академічної заборгованості, у два етапи:

- заборгованість із поточного контролю;
- заборгованість із підсумкового контролю.

Ліквідація заборгованості поточного контролю відбувається шляхом проходження тестових завдань та виконання практичних робіт згідно з РНП. Ліквідація заборгованості з підсумкового контролю організовується в форматі перездачі екзамену та/або повторного виконання і захисту курсової роботи у два етапи: перша спроба приймається провідним викладачем дисципліни одноособово, друга – комісією у складі не менше трьох викладачів кафедри, яка призначається начальником навчально-наукового інституту цивільного захисту; оцінка комісії є остаточною.

Дотримання принципів академічної доброчесності: усі результати контролю (тестові завдання, практичні та лабораторні роботи, курсова робота) зараховуються виключно за умови дотримання здобувачем норм академічної доброчесності. Презумпція академічної доброчесності означає, що результати виконаних завдань вважаються отриманими самостійно й чесно, доки протилежне не буде аргументовано доведено. Роботи (завдання) виконуються здобувачами самостійно; ідеї та напрацювання інших авторів використовуються лише за умови належного цитування. Групове вирішення завдань дозволяється за попереднього узгодження ідеї з викладачем та в межах реалізації студентських R&D проєктів.

Використання засобів штучного інтелекту (ШІ): у процесі вивчення дисципліни допускається та заохочується використання інструментів штучного інтелекту (ШІ) як сучасних інформаційних технологій, що є невід'ємною складовою професійної діяльності сучасного ІТ спеціаліста. ШІ може використовуватися для пошуку та структурування інформації, генерування ідей за умови критичного оцінювання отриманих результатів. Використання ШІ не повинно підміняти самостійну навчальну діяльність здобувача. Здобувач несе відповідальність за зміст, достовірність і коректність результатів, отриманих із використанням ШІ, та має бути здатним пояснити застосовані підходи, алгоритми й отримані результати. У разі суттєвого використання ШІ під час виконання завдання здобувач має зазначити відповідний інструмент та мету його застосування.

Поведінка в аудиторії: неприпустиме запізнення на заняття; користування телефоном або іншими гаджетами на занятті, за винятком виконання громіздких обчислень та використання додаткових програм в освітніх цілях; повага до думки інших колег; дотримання норм культури мовлення; соціальна солідарність тощо.

14. Рекомендована література

Основна:

1. *Основи програмування (мовою Java)* : курс лекцій / Придатко О., Хлєвной О., Бурак Н. – Львів : ЛДУ БЖД, 2019. – 180 с.
2. *Java 8. The Complete Reference*. 9 edition / Herbert Schildt. – New York : Oracle Press, 2015. – 1274 с.
3. *Java: керівництво для початківців* / Шилдт Г. – К. : «Діалектик», 2020. – 816 с.
4. *Java. Бібліотека професіонала*. Т 1. Основи. 11-е видання / Кей С. Хорстманн. – Наука і техінка, 2020. – 864 с.
5. *Spring 5 для професіоналів* / Кларенс Хо, Кріс Шефер, Роб Харроп, Юліана Козміна. – Наука і техінка, 2020. – 1120 с.
6. *Head First Java* / Kathy Sierra, Bert Bates. – «O'REILLY», 2012. – 678 с.
7. *Дизайн-патерни – просто, як двері* : підручник / Андрій Будай : «Developer's SUCCESS», 2012. – 90 с.
8. *Об'єктно-орієнтоване моделювання програмних систем* : навчальний посібник. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2007. – 108 с.

Додаткова:

1. Придатко О. В. Інтеграція новаційного методу мобільного навчання в освітні проекти підготовки розробників програмного забезпечення / О. В. Придатко, В. В. Придатко, Ю. О. Борзов, В. Є. Дзень // Вісник ЛДУБЖД: Зб. наук. праць. Львів: ЛДУ БЖД, 2018. – №18. – С.70-80.
2. *Придатко О. В.* Адаптивна інформаційно-довідкова система "UniBell" як складова частина проєкту "Smart-університет" / О. В. Придатко, Н. Є. Бурак, В. Є. Дзень, М. С. Кунинець // Науковий вісник НЛТУ України : Зб. наук. пр. Львів: НЛТУ, 2020. - т. 30, № 5 – С. 113–121.
3. *Андрушко О. А.* Аналіз процесів використання Docker для побудови мікросервісів / О. А. Андрушко, Ю. О. Борзов, І. О. Малець, О. В. Придатко // Науковий вісник НЛТУ України: Зб. нак. праць. Львів: НЛТУ, 2017. - №9(27) – С.95-98.
4. *Prydatko O*. Informational System of Project Management in the Areas of Regional Security Systems' Development / O. Prydatko, O. Smotr, Yu. Borzov, I. Solotvinskyi, O. Didyk // 2018 IEEE Second Conference on Data Stream Mining & Processing. Lviv, 2018. – №2 – 187-192.
5. *Concurrent Programming in Java*™: Design Principles and Patterns, Second Edition : book / Doug Lea : «Addison Wesley», 1999. – 432 p.
6. *Основи програмування на Java* : методичні вказівки до лабораторного практикуму та самостійної роботи, частина перша / Бивойно П. Г., Бивойно Т. П. – Чернігів: ЧНТУ, 2014. – 79 с.
7. *Кордунова Ю. С.* Аналіз та розроблення концептуальної моделі управління життєвим циклом спеціалізованого програмного забезпечення безпеко-орієнтованого спрямування. Український журнал інформаційних технологій. 2023. Т. 5, № 2. С. 72–78. https://doi.org/10.23939/ujit2023.02.072
8. Kordunova Y., Prydatko O., Smotr O., Golovatyi R. Expert Decision Support System Modeling in Lifecycle Management of Specialized Software. Lecture Notes on Data Engineering and Communications Technologies, Springer, Switzerland. Vol. 149, 2022, pp. 367-383, https://doi.org/10.1007/978-3-031-16203-9_22
9. Шопський, О. М., Лінник, М. Б., Лінник, М. Б., & Назар, Ю. С. (2025). Розроблення та впровадження платформи для автоматизованого тестування систем оповіщення та моніторингу надзвичайних ситуацій. Scientific Bulletin of UNFU, 35(3), 193-199. https://doi.org/10.36930/40350321

Інформаційні ресурси:

1. *Віртуальний університет ЛДУ БЖД* [Електронний ресурс]. — Доступний з http://virt.ldubgd.edu.ua/