

ДЕРЖАВНА СЛУЖБА УКРАЇНИ З НАДЗВИЧАЙНИХ СИТУАЦІЙ
ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ БЕЗПЕКИ ЖИТТЄДІЯЛЬНОСТІ
Факультет 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)

Галузь знань: 12 Інформаційні технології

Спеціальність (спеціалізація) : 122 Комп'ютерні науки

Спеціалізація (за наявності): _____

Освітня програма: Комп'ютерні науки

Львів 2024

РОЗРОБНИК

Старший викладач кафедри
інформаційних технологій та систем
електронних комунікацій
доктор філософії

(підпис) Юлія НАЗАР
(ім'я, прізвище)
«__» _____ 20__ р.

ЗАТВЕРДЖЕНО

Начальник кафедри інформаційних
технологій та систем електронних
комунікацій
кандидат технічних наук, доцент

(підпис) Олександр ПРИДАТКО
(ім'я, прізвище)
«__» _____ 20__ р.

ПОГОДЖЕНО

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

(підпис) Ігор МАЛЕЦЬ
(ім'я, прізвище)
«__» _____ 20__ р.

ПОГОДЖЕНО

Заступник начальника факультету
цивільного захисту
кандидат фізико-математичних наук,
доцент

(підпис) Ольга МЕНЬШИКОВА
(ім'я, прізвище)
«__» _____ 20__ р.

Розглянуто та затверджено на засіданні кафедри _____,
протокол від «__» _____ 20__ № _____

Розглянуто Вченою радою факультету _____,
протокол від «__» _____ 20__ № _____

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 та 4 семестри)
Мова викладання	українська
Викладачі	Назар Юлія Сергіївна, старший викладач кафедри інформаційних технологій та систем електронних комунікацій Головатий Роман Русланович, доцент кафедри інформаційних технологій та систем електронних комунікацій
Контактний телефон	
Електронна пошта	kordunovayulia@gmail.com
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 телекомунікаційних систем

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 програмних 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 проєкти (в окремих випадках індивідуальні) у вигляді реалізації програмної системи за об'єктно-орієнтованою парадигмою.

До розробки курсу долучено зовнішніх стейкхолдерів: Кокотко Богдан Ярославович, Software engineer, компанія Scanmarket.

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 написання першого застосунку в рамках виконання курсової роботи.

Завдання:

- формування поглиблених знань з об’єктно-орієнтованого проектування 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, збору 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;

- зв'язки між класами, асоціація, агрегація, метакласи;
- поняття «енамів» та «інтерфейсів» в програмуванні мовою Java;
- особливості опрацювання символьних стрічок;
- 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, читання з файлу тощо;
- основні принципи функціонального програмування та застосування лямбда-виразів;
- класи, методи бібліотек java.util , java.lang та особливості застосування;
- основні поняття роботи в мережі;
- мережеві класи та інтерфейси.

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 парадигми програмування.

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	Навчальний матеріал дисципліни структурований за двома змістовими модулями і складається з шістнадцяти тем, які є логічно завершеними, відносно самостійними, цілісними частинами. Засвоєння теоретичного матеріалу курсу передбачає відвідування 32 годин лекційних занять та здачу тестових завдань на базі електронного освітнього середовища (до кожної теми). Засвоєння практичної частини курсу зосереджено в рамках 32 годин практичних занять та 32 годин лабораторних занять. В рамках практичних занять здобувачі освіти закріплюють теоретичний матеріал шляхом виконання практичних робіт у відповідності до методичних рекомендацій, що розміщені на платформі електронного освітнього середовища «Віртуальний університет». В рамках проведення лабораторних занять здобувачі освіти виконують та захищають лабораторні роботи у відповідності до індивідуальних або групових завдань, що розміщені на платформі електронного освітнього середовища «Віртуальний університет».
---------------------	--

	Підсумковим етапом вивчення курсу «Об’єктно-орієнтоване програмування» є написання курсової 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 лабораторних робіт за умови проходження он-лайн/очних курсів “Java Core” або “Java Advanced” (до прикладу від SoftServe IT Academy, Java BaseCamp від GlobalLogic та споріднені курси інших компаній) та представлення сертифікату про проходження курсів із перліком вивчених тем (опрацьованого контенту, реалізованого проєкту тощо). Перезарахування відбувається за індивідуальним узгодженням із викладачем та висвітлюється в електронному журналі курсу.
Обсяг дисципліни:	8 кредитів / 240 академічних годин, з яких: лекції - 48 год., практичні - 48 год., лабораторні – 16 год., самостійна робота – 128 год.
Форми навчання	лекції, практичні заняття, лабораторні заняття, консультації, самостійна робота (в т. ч. здача тестових завдань та виконання лабораторних робіт з їх подальшим захистом), робота в командах, виконання курсової роботи.

6. Тематика та зміст курсу

Назви змістових модулів і тем			Кількість годин			
	Усього			у тому числі		
		Лекції	Практичні	Лабораторні	Самостійна робота	
3-й семестр						
Тема №1. Базові поняття об'єктно-орієнтованого програмування. Класи, методи, конструктори	7	2	2	-	3	
Тема №2. Сумісне застосування наслідування та поліморфізму	7	2	2	-	3	
Тема №3. Абстракція	7	2	2	-	3	
Тема №4. Сумісне застосування парадигм ООП при створенні програмних систем (частина 1)	7	2	-	2	3	
Тема №5. Сумісне застосування парадигм ООП при створенні програмних систем (частина 2)	15	4	2	2	7	
Тема №6. Колекції. Реалізація різних структур даних (пакет java.util) (частина 1)	7	2	2	-	3	
Тема №7. Колекції. Реалізація різних структур даних (пакет java.util) (частина 2)	15	4	-	4	7	
Тема №8. Опрацювання винятків	8	2	2	-	4	
Тема №9. Потоки вводу-виводу	8	2	2	-	4	
Тема №10. Робота з рядками	8	2	2	-	4	
Тема №11. Багатопотокове програмування	8	2	2	-	4	
Тема №12. Робота з часом та датою	8	2	2	-	4	
Тема №13. Архітектура чат-боту. Розгортання проєкту Java (частина 1)	8	2	2	-	4	
Тема №14. Архітектура чат-боту. Розгортання проєкту Java (частина 2)	8	2	2	-	4	
Усього годин за семестр	121	32	24	8	57	
4-ий семестр						
Тема №15. Вступ до Stream API. Лямбда-вирази.	15	2	2	2	9	
Тема №16. Stream API	15	2	2	2	9	
Тема №17. Інструмент збірки проєктів Maven. UML-діаграми класів.	15	2	2	2	9	
Тема №18. Анотації в Java. Reflection API	15	2	2	2	9	
Тема №19. JDBC+SQL	15	2	4	-	9	
Тема №20. Архітектура web-сервісу на java (Spring Boot) (частина 1)	14	2	4	-	8	
Тема №21. Архітектура web-сервісу на java (Spring Boot) (частина 2)	14	2	4	-	8	
Тема №22. Повторення вивченого за курс	16	2	4	-	10	
Усього годин за семестр	119	16	24	8	71	
Усього годин	240	48	48	16	128	

7. Інформаційний обсяг навчальної дисципліни

7.1. Теми лекційних занять

№	Назва теми	Кількість годин
1.	Базові поняття об'єктно-орієнтованого програмування. Класи, методи, конструктори	2
2.	Сумісне застосування наслідування та поліморфізму	2
3.	Абстракція	2
4.	Сумісне застосування парадигм ООП при створенні програмних систем (частина 1)	2
5.	Сумісне застосування парадигм ООП при створенні програмних систем (частина 2)	4
6.	Колекції. Реалізація різних структур даних (пакет java.util) (частина 1)	2
7.	Колекції. Реалізація різних структур даних (пакет java.util) (частина 2)	4
8.	Опрацювання винятків	2
9.	Потоки вводу-виводу	2
10.	Робота з рядками	2
11.	Багатопотокове програмування	2
12.	Робота з часом та датою	2
13.	Архітектура чат-боту. Розгортання проєкту Java (частина 1)	2
14.	Архітектура чат-боту. Розгортання проєкту Java (частина 2)	2
15.	Вступ до Stream API. Лямбда-вирази.	2
16.	Stream API	2
17.	Інструмент збірки проєктів Maven. UML-діаграми класів.	2
18.	Анотації в Java. Reflection API	2
19.	JDBC+SQL	4
20.	Архітектура web-сервісу на java (Spring Boot) (частина 1)	2
21.	Архітектура web-сервісу на java (Spring Boot) (частина 2)	
22.	Повторення вивченого за курс	
	Разом	48

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 основ абстракції в Java. Створення абстрактних класів та методів. Огляд інтерфейсів та їх роль в абстракції. Застосування абстрактних класів та інтерфейсів у розробці. Порівняння абстрактних класів та інтерфейсів.

Тема №4. Сумісне застосування парадигм ООП при створенні програмних систем (частина 1)

Огляд ключових парадигм ООП (інкапсуляція, наслідування, поліморфізм, абстракція). Введення до проектування програмних систем на основі об'єктно-орієнтованих принципів. Реалізація проекту з використанням базових концепцій ООП.

Тема №5. Сумісне застосування парадигм ООП при створенні програмних систем (частина 2)

Продовження проектування складних програмних систем. Оптимізація та рефакторинг коду на основі ООП. Використання шаблонів проектування.

Тема №6. Колекції. Реалізація різних структур даних (пакет java.util) (частина 1)

Вивчення основ колекцій в Java. Огляд інтерфейсів колекцій: List, Set, Queue. Реалізація різних типів списків (ArrayList, LinkedList). Використання ітераторів для перебору елементів колекцій.

Тема №7. Колекції. Реалізація різних структур даних (пакет java.util) (частина 2)

Продовження вивчення колекцій. Огляд інтерфейсів Map та їх реалізацій (HashMap, TreeMap). Особливості сортування та пошуку в колекціях. Використання компараторів та компарабельних елементів.

Тема №8. Опрацювання 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 (пакет java.io). Читання та запис файлів, використання класів File, FileReader, FileWriter. Робота з потоками байтів та символів. Використання буферизованих потоків для оптимізації вводу/виводу. Читання даних з консолі та запис даних у файл.

Тема №10. Робота з рядками.

Вивчення класу String в Java. Робота з рядками: створення, маніпуляція, порівняння. Методи String: substring(), indexOf(), replace(), тощо. Використання класів StringBuilder та StringBuffer для оптимальної роботи з рядками. Застосування регулярних виразів для пошуку і обробки тексту.

Тема №11. Багатопотокове програмування

Основи багатопотоковості в Java. Створення потоків: клас Thread та інтерфейс Runnable. Синхронізація потоків, ключове слово synchronized. Використання wait() та notify() для управління потоками. Управління станами потоків: створення, зупинка, завершення.

Тема №12. Робота з часом та датою

Вивчення нових API для роботи з датою та часом (java.time пакет). Класи LocalDate, LocalTime, LocalDateTime. Операції з датами та часом: додавання, віднімання, форматування.

Вимірювання тривалості виконання операцій за допомогою класу Duration. Порівняння старого (java.util.Date) та нового API.

Тема №13. Архітектура чат-боту. Розгортання проєкту Java (частина 1)

Введення в архітектуру чат-боту. Розробка базової архітектури чат-боту на Java. З'єднання з API месенджерів або платформ для обробки повідомлень. Основні компоненти: обробник повідомлень, менеджер сесій, бізнес-логіка. Тестування базових функцій чат-боту.

Тема №14. Архітектура чат-боту. Розгортання проєкту Java (частина 2)

Продовження розробки чат-боту. Реалізація складних сценаріїв взаємодії та логіки. Масштабування чат-боту для обробки великої кількості користувачів. Розгортання проєкту на сервері або хмарному сервісі. Використання інструментів для моніторингу та обслуговування чат-боту. Повторення вивченого за 1 семестр.

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

Основи лямбда-виразів в Java. Функціональні інтерфейси та їх застосування. Використання лямбд для скорочення коду. Метод референс (::) та його застосування. Комбінування лямбда-виразів із Stream API для обробки даних.

Тема №16. Stream API

Введення в 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.

Вивчення основ UML (Unified Modeling Language). Створення діаграм класів для візуалізації структури проєкту. Визначення класів, їх атрибутів, методів та зв'язків. Використання UML для планування великих проєктів. Огляд інших типів діаграм UML (діаграми послідовностей, діаграми станів). Видача завдань на курсові роботи, обговорення вимог до проєктів.

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 + MySQL

Основи роботи з базами даних через JDBC (Java Database Connectivity). Підключення до бази даних MySQL. Створення запитів: SELECT, INSERT, UPDATE, DELETE. Використання класів Connection, Statement, PreparedStatement, ResultSet. Управління транзакціями. Обробка результатів SQL-запитів та інтеграція з Java-додатками.

Тема №20. Архітектура web-сервісу на java (Spring Boot) (частина 1)

Введення в архітектуру веб-додатків. Основні компоненти веб-додатків: клієнт, сервер, база даних. Створення простого веб-додатку на Java з використанням сервлетів та JSP. Підключення веб-додатку до бази даних. Налаштування серверу (Tomcat, Jetty). Розгортання проєкту на хостингу або хмарному сервісі.

Тема №21. Архітектура web-сервісу на java (Spring Boot) (частина 2)

Введення в архітектуру сучасних веб-сервісів на Java з використанням Spring Boot. Основи концепції REST (Representational State Transfer) та реалізація RESTful API.

Використання HTTP-методів (GET, POST, PUT, DELETE) для обміну даними. Розробка серверної частини, обробка запитів, налаштування маршрутизації та інтеграція веб-сервісів із зовнішніми клієнтами. Основи тестування та забезпечення безпеки web-сервісів.

Тема №22. Повторення вивченого за курс

Узагальнення та систематизація знань, отриманих протягом курсу: від основ об'єктно-орієнтованого програмування до розробки веб-сервісів і роботи з базами даних. Підготовка до підсумкового оцінювання та закріплення практичних навичок створення Java-додатків.

8. Теми практичних, семінарських та лабораторних занять

8.1. Теми практичних занять

№	Назва теми	Кількість годин
1.	Створення класів та методів з унікальною логікою	2
2.	Наслідування та поліморфізм у дії	2
3.	Створення абстрактних класів та інтерфейсів	2
4.	Розроблення цілісної об'єктно-орієнтованої програми	2
5.	Реалізація різних структур даних (пакет java.util)	2
6.	Робота з рядками	2
7.	Опрацювання винятків	2
8.	Робота із файлами, потоки вводу-виводу	2
9.	Робота із багатопотоковим програмуванням	2
10.	Робота з часом та датою	2
11.	Робота із Stream API	2
12.	Робота із лямбда-виразами, функціональними інтерфейсами	2
13.	Розгортання чат-бот проєкту Java	2
14.	Робота із UML діаграмами. Видача завдань на курсову роботу	2
15.	Робота в мережі	2
16.	Створення першої клієнт-сервісної програми	2
	Разом	24

8.2. Теми семінарських занять

«Семінарські заняття не передбачені»

8.3. Теми лабораторних занять

№	Назва теми	Кількість годин
1.	Розроблення цілісної об'єктно-орієнтованої програми	4
2.	Реалізація різних структур даних (пакет java.util)	2
3.	Розгортання чат-бот проєкту Java	2
4.	Робота з базами даних в Java	2
5.	Розробка Unit тестів	2
6.	Integrating Testing	2
7.	Розгортання web-проєкту на Java	2
	Разом	16

9. Завдання для самостійного опрацювання

№	Назва теми	Кількість годин
1.	Базові поняття об'єктно-орієнтованого програмування. Класи, методи, конструктори	3

2.	Сумісне застосування наслідування та поліморфізму	3
3.	Абстракція	3
4.	Сумісне застосування парадигм ООП при створенні програмних систем (частина 1)	3
5.	Сумісне застосування парадигм ООП при створенні програмних систем (частина 2)	3
6.	Колекції. Реалізація різних структур даних (пакет java.util) (частина 1)	3
7.	Колекції. Реалізація різних структур даних (пакет java.util) (частина 2)	3
8.	Робота з рядками	4
9.	Опрацювання винятків	4
10.	Потоки вводу-виводу	4
11.	Багатопотокове програмування	4
12.	Робота з часом та датою	4
13.	Stream API	4
14.	Лямбда-вирази. Функціональні інтерфейси. API потоку	4
15.	Архітектура чат-боту. Розгортання проєкту Java (частина 1)	4
16.	Архітектура чат-боту. Розгортання проєкту Java (частина 2)	4
17.	JDBC та роботи з базами даних. СУБД MySQL	9
18.	Unit Testing	9
19.	Integrating Testing	9
20.	Архітектура web-додатків. Розгортання проєкту Java	9
21.	Базові поняття UML. Діаграми класів. Видача завдань на курсову роботу	9
22.	Робота в мережі	8
23.	Сервіси REST. Створення першої клієнт-сервісної програми	8
24.	Методологія CI/CD. Збір програмного продукту. Налаштування проєкту з вихідним кодом на GitHub.	10
	Разом	128

10. Методи навчання

Основні форми організації навчання: лекції; практичні заняття з проведенням зрізів знань; лабораторні заняття та захистів звітів лабораторних робіт; виконання курсової роботи; командна робота; консультації.

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, SharePoint, PowerApps), 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 інші програмні засоби навчання.

12. Критерії оцінювання

Оцінювання результатів навчання здобувачів вищої освіти здійснюється відповідно до «Положення про організацію освітнього процесу у ЛДУ БЖД» <https://cutt.ly/0WRAkEh> та «Положення про порядок та критерії оцінювання результатів навчання здобувачів вищої освіти у ЛДУ БЖД» <https://cutt.ly/BIMOvwn>.

Поточний контроль	
Поточний контроль проводиться у формі виконання тестових завдань на базі платформи електронного освітнього середовища «Віртуальний університет», усного опитування, виконання лабораторних 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 визначенні допуску до підсумкового контролю.	
1-й семестр	
Вид робіт	Формат проведення та критерії оцінювання
Тестові завдання	В першій частині курсу (змістовий модуль №1) передбачено проходження 6 тестових завдань та 1 підсумкового тестування. За результатами позитвої здачі усіх тестових завдань виводиться середня оцінка за національною шкалою та шляхом пропорційного співвідношення переводиться у 100-бальну шкалу. Здобувачі освіти мають можливість одержати за проходження усіх тестових завдань сумарно до 55 балів (5,0 за тест за нац.шкалою = 30 балам та підсумкове тестування за нац.шкалою = 25 балам) . Критерії оцінювання наведені в електронному курсі «Віртуального університету».
Практичні та лабораторні роботи	Першою частиною курсу (змістовий модуль №1) передбачено виконання 8 практичних та 1 лабораторної робіт. За успішне виконання та захист практичних та лабораторної роботи виводиться середня оцінка за національною шкалою та шляхом пропорційного співвідношення переводиться у 100-бальну шкалу. Здобувачі освіти мають можливість одержати за виконання та захист усіх лабораторних робіт сумарно до 45 балів (5,0 за нац.шкалою = 45 балам) . Критерії оцінювання наведені в електронному курсі «Віртуального університету».
2-й семестр	
Вид робіт	Формат проведення та критерії оцінювання
Тестові завдання	В другій частині курсу (змістовий модуль №2) передбачено проходження 7 тестових завдань (за кожною темою) та 1 підсумкове тестування. За результатами позитвої здачі усіх тестових завдань

	виводиться середня оцінка за національною шкалою та шляхом пропорційного співвідношення переводиться у 100-бальну шкалу. Здобувачі освіти мають можливість одержати за проходження усіх тестових завдань сумарно до 45 балів (5,0 за нац.школою = 45 балам). Критерії оцінювання наведені в електронному курсі «Віртуального університету».
Лабораторні роботи	Другою частиною курсу (змістовий модуль №2) передбачено виконання 7 лабораторних робіт. За успішне виконання та захист лабораторних робіт виводиться середня оцінка за національною шкалою та шляхом пропорційного співвідношення переводиться у 100-бальну шкалу. Здобувачі освіти мають можливість одержати за виконання та захист усіх лабораторних робіт сумарно до 35 балів (5,0 за нац.школою = 35 балам). Критерії оцінювання наведені в електронному курсі «Віртуального університету». Ще 20 балів студенти можуть отримати за роботу під час семестру (активність на лекціях та практичних/лабораторних заняттях, участь у конференціях та наукових роботах). 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 несамотійність 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

1-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 лабораторних робіт (практична частина). Розподіл між запропонованими складовими поточного

контролю знань за навчальний семестр наступний: тестові завдання – до 30 балів, виконання та захист практичних та лабораторних робіт – до 45 балів та підсумкове тестування – до 25 балів.

Екзамен (максимально 100 балів) складається із теоретичної та практичної частини (по 50 балів кожна) 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 передбачає розв'язок 2-х практичних завдань, а саме написання програмного коду згідно отриманого завдання. Кожне питання у білеті оцінюється у 25 балів (максимально).

Критерії оцінювання виконаних завдань:

- 25 балів – написаний код є працездатним, а структура та результат його виконання в повній мірі відповідають поставленому завданню;
- 20-24 балів – написаний код є працездатним, але його структура або результат виконання не в повній мірі відповідають поставленому завданню;
- 0-19 балів – написаний код є працездатним, але його структура або результат виконання лише частково відтворюють умову поставленого завдання;
- 1-9 балів – написаний код не працездатний, проте в його структурі спостерігається вірно обраний напрям вирішення завдання;
- 0 балів – завдання не виконане або не відповідає поставленому завданню.

2-ий семестр

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

Одержання необхідної кількості балів поточного контролю можливе за рахунок проходження тестових завдань (теоретична частина) та виконання і захисту лабораторних робіт (практична частина). Розподіл між запропонованими складовими поточного контролю знань за навчальний семестр наступний: тестові завдання – до 45 балів, виконання та захист лабораторних робіт – до 35 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).

На екзамен здобувачам перераховуються бали еквівалентні: сума балів за семестр*0,3, тобто не більше 30% семестрових балів.

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 передбачає розв'язок практичного завдання, а саме написання програмного коду згідно отриманого завдання. Максимальна оцінка за практичну частину екзамену – 20 балів.

Критерії оцінювання виконаних завдань:

- 25 балів – написаний код є працездатним, а структура та результат його виконання в повній мірі відповідають поставленому завданню;
- 20-24 балів – написаний код є працездатним, але його структура або результат виконання не в повній мірі відповідають поставленому завданню;
- 0-19 балів – написаний код є працездатним, але його структура або результат виконання лише частково відтворюють умову поставленого завдання;
- 1-9 балів – написаний код не працездатний, проте в його структурі спостерігається вірно обраний напрям вирішення завдання;
- 0 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		

13. Політика курсу

Виконання навчальних завдань і робота на курсі має відповідати вимогам «Кодексу академічної доброчесності та корпоративної культури ЛДУ БЖД» <https://cutt.ly/5WTlc6y>.

Академічні очікування від здобувачів: своєчасне виконання тестових завдань, передбачених силабусом дисципліни; обов’язкове відвідування практичних і лабораторних занять, а також виконання усіх індивідуальних завдань передбачених силабусом курсу (лабораторні роботи та курсова робота).

Політика щодо термінів виконання завдань та ліквідації академічної заборгованості: терміни виконання завдань вказуються у 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 можливо до дня проведення підсумкового контролю (залік, екзамен) згідно графіку консультацій випускової 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 лабораторних робіт згідно із тематичним планом курсу. Ліквідація заборгованості з підсумкового контролю організовується в форматі перездачі екзамену та/або повторного виконання і захисту курсової роботи.

Дотримання принципів академічної доброчесності: роботи (завдання) виконуються здобувачами самостійно, ідеї та ініціативи інших авторів використовуються лише при належно оформленню цитуванні. Групове вирішення завдань дозволяється за попереднього декларування ідеї та в рамках реалізації студентських R&D проєктів.

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., Хлевной 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 Будаї : «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.
9. Придатко О. В., Бурак Н. С. *Об'єктно-орієнтоване програмування* : методичні рекомендації для написання курсової роботи. Львів: ЛДУ БЖД. – 2019. – 62 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. Martyn Ye. Software for Shelter's Fire Safety and Comfort Levels Evaluation / Martyn Ye., Smotr O., Burak N., Prydatko O., Malets I. // Communications in Computer and Information Science, Springer, Cham. – Vol. 1158, 2020. pp. 457-469 https://doi.org/10.1007/978-3-030-61656-4_31
3. Придатко О. В. Адаптивна інформаційно-довідкова система "UniBell" як складова частина проекту "Smart-університет" / О. В. Придатко, Н. С. Бурак, В. Є. Дзень, М. С. Кунинець // Науковий вісник НЛТУ України : Зб. наук. пр. Львів: НЛТУ, 2020. - т. 30, № 5 – С. 113–121.
4. 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. наук. праць. Львів: НЛТУ, 2017. - №9(27) – С.95-98.
5. 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.
6. *Concurrent Programming in Java™: Design Principles and Patterns*, Second Edition : book / Doug Lea : «Addison Wesley», 1999. – 432 p.
7. *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.
8. 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>
9. Kordunova Y., Prydatko O., Smotr O., Golovaty 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

Інформаційні ресурси:

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