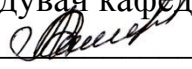


ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ
Навчально-науковий інститут денної освіти
Кафедра комп'ютерних наук та інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри КНІТ

 **Олена ОЛЬХОВСЬКА**

« 30 » _____ 06 _____ 2025 р.

РОБОЧА ПРОГРАМА

навчальної дисципліни
освітня програма
спеціальність
галузь знань
ступінь вищої освіти

«Сучасні парадигми програмування»
Комп'ютерні науки
122 Комп'ютерні науки
12 Інформаційні технології
бакалавр

Робоча програма навчальної дисципліни «Сучасні парадигми програмування»
рекомендована до використання в освітньому процесі на засіданні кафедри комп'ютерних
наук та інформаційних технологій
Протокол від _30_. _06_.2025 року, № _15_

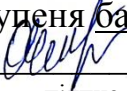
Полтава 2025

Укладач: Олексійчук Юрій Федорович, доцент кафедри комп'ютерних наук та інформаційних технологій, к. ф.-м. н.

ПОГОДЖЕНО:

Гарант освітньої програми «Комп'ютерні науки»
спеціальності "Комп'ютерні науки"

ступеня бакалавр

 Оксана ЧЕРНЕНКО
підпис ініціали, прізвище

«_30_» _____ 06 _____ 2025 року

Розділ 1. Опис навчальної дисципліни

Таблиця 1. Опис навчальної дисципліни "Сучасні парадигми програмування"

Місце у структурно-логічній схемі підготовки	<i>Пререквізити:</i> – Програмування <i>Постреквізити:</i> – Бакалаврська робота
Мова викладання	Українська
Статус дисципліни	Обов'язкова
Курс/семестр вивчення	3/1, 4/1
Кількість кредитів ECTS/кількість модулів	4/2
Денна форма навчання:	
Кількість годин: 120 год	
- Лекції: 16 год	
- Практичні заняття: 32 год	
- Самостійна робота: 72 год.	
- Вид підсумкового контролю (ПМК, екзамен): ПМК	
Заочна форма навчання:	
Кількість годин: 120 год	
- Лекції: 4 год	
- Практичні (семінарські, лабораторні) заняття: 4 год	
- Самостійна робота: 112 год	
- Вид підсумкового контролю (ПМК, екзамен): ПМК	

Розділ 2. Перелік коментентностей, які забезпечує дана навчальна дисципліна, програмні результати навчання

Основною метою вивчення дисципліни є формування у студентів системного мислення, узагальнення та систематизація знань та навичок з програмування, розуміння різних парадигм програмування, поглиблення знань з об'єктно-орієнтованого програмування.

Таблиця 2. Перелік компетентностей та програмні результати навчання, які забезпечує навчальна дисципліна «СПП»

Програмні результати навчання	Компетентності, якими повинен оволодіти здобувач
ПР1. 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. ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач,	Здатність до абстрактного мислення, аналізу та синтезу (ЗК1). Здатність застосовувати знання у практичних ситуаціях (ЗК2). Знання та розуміння предметної області та розуміння професійної діяльності (ЗК3). Здатність спілкуватися державною мовою як усно, так і письмово (ЗК4). Здатність вчитися й оволодівати сучасними

оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій. ПР9. 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.	знаннями (ЗК6). Здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК7). Здатність генерувати нові ідеї (креативність) (ЗК8). Здатність працювати в команді (ЗК9). Здатність бути критичним і самокритичним (ЗК10). Здатність приймати обґрунтовані рішення (ЗК11) Здатність оцінювати та забезпечувати якість виконуваних робіт (ЗК12). Здатність діяти на основі етичних міркувань (ЗК13). Здатність до математичного формулювання та досліджування неперервних та дискретних математичних моделей, обґрунтовування вибору методів і підходів для розв'язування теоретичних і прикладних задач у галузі комп'ютерних наук, аналізу та інтерпретування (СК1). Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем (СК3). 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 (СК8).
---	--

Розділ 3. Програма навчальної дисципліни

Модуль 1. Парадигми програмування

Тема 1. Основні парадигми програмування.

Імперативне програмування. Функційне програмування. Логічне програмування. Структурне програмування. Об'єктно-орієнтоване програмування.

Тема 2. Логічне та функційне програмування

Лінійні обчислення. Способи передачі параметрів в методи. Prolog. Stream API в Java.

Модуль 2. Особливості реалізації об'єктно-орієнтованого програмування в різних мовах програмування

Тема 3. Особливості об'єктно-орієнтованого програмування.

Типізація даних в різних мовах програмування. Поєднання функційного та об'єктно-орієнтованого підходів. Інкапсуляція. Успадкування. Поліморфізм. Інтерфейси. Абстрактні класи. Перевантаження.

Тема 4. Spring як мультипарадигмовий фреймворк

Spring Framework. POJO, Entity та DTO-об'єкти. Основи Domain-Driven Design (DDD). Сервіси у Spring Framework. Репозиторії в Spring Data JPA. Аспектно-орієнтоване програмування. Реактивне програмування.

Розділ 4. Тематичний план навчальної дисципліни

Таблиця 4. Тематичний план навчальної дисципліни для студентів денної/ форми навчання

1	2	3	4	5	6
Назва теми (лекції) та питання теми (лекції)	кількість годин	Назва теми та питання семінарського, практичного або лабораторного заняття	кількість годин	Завдання самостійної роботи в розрізі тем	кількість годин
Модуль 1 Парадигми програмування					
Тема 1. Основні парадигми програмування					
Лекція 1. Вступ до предмету 1. Основні поняття 2. Методології розробки ПЗ 3. Функції і особливості мислення програмістів Лекція 2. Парадигми програмування 1. Парадигми (методології) програмування та їх класифікація 2. Основні парадигми програмування 3. Інші парадигми та перспективи	2 2	Практична робота №1. Структурне та об'єктно-орієнтоване програмування Практична робота №2. Структурне та об'єктно-орієнтоване програмування	2 2	опрацювати лекційний матеріал до теми 1, готуватись до практичних занять, опрацювати питання: 1. Альтернативні класифікації парадигм програмування. 2. Паралельне програмування.	18
Тема 2. Логічне та функційне програмування					

1	2	3	4	5	6
Лекція 3. Мова логічного програмування Prolog 1. Основи мови Prolog 2. Приклади Лекція 4. Особливості функційного програмування 1. Основні особливості функційного програмування 2. Лінійні обчислення 3. Способи передачі аргументів 4. Stream API в Java	2 2	Практична робота №3-4. Логічне програмування Практична робота №5-6. Лінійні обчислення Практична робота №7. Функційний підхід в реалізації циклів Практична робота №8. Робота з Stream.API в Java Практична робота №9. МКР№1. Парадигми програмування	4 4 2 2 2	опрацювати лекційний матеріал до теми 2, готуватись до практичних занять, пройти тести в ДК Опрацювати теми: 1. Списки в функційних та логічних мовах програмування. 2. Мови функційного програмування	18
Модуль 2. Особливості реалізації об'єктно-орієнтованого програмування в різних мовах програмування					
<u>Тема 3. Особливості об'єктно-орієнтованого програмування.</u>					
Лекція 5. Типізація даних 1. Система типізації 2. Статична та динамічна типізація 3. Строга (сильна) та нестрога (слабка) типізація 4. Явна та неявна типізація 5. Тип dynamic в C# Лекція 6. Особливості ООП 1. Класи і об'єкти 2. Складові елементи класів 3. Модифікатори доступу 4. Відношення класів Лекція 7. Перевантаження в об'єктно-орієнтованих мовах 1. Перевантаження	2 2	Практична робота №10. Типізація даних Практична робота №11. Відношення між класами Практична робота №12. Перевантаження в ООП	2 2 2	опрацювати лекційний матеріал до теми 3, готуватись до практичних занять Опрацювати теми: 1. Функційне програмування в об'єктно-орієнтованих мовах. 2. Багато-парадигменні мови програмування.	18

1	2	3	4	5	6
методів та конструкторів 2. Перевантаження операторів					
Тема 4. Spring як мультипарадигмовий фреймворк					
Лекція 8. Парадигми програмування в Spring Framework 1. Spring Framework 2. Відмінності між POJO, Entity та DTO-об'єктами 3. Основи Domain-Driven Design (DDD) 4. Сервіси у Spring Framework 5. Репозиторії в Spring Data JPA 6. Аспектно-орієнтоване програмування у Spring 7. Реактивне програмування у Spring	2 2	Практичне заняття №13-14. Реактивне програмування Практичне заняття №15. Аспектно-орієнтоване програмування Практична робота №16. МКР №2. Особливості реалізації ООП в різних мовах програмування	4 2 2	опрацювати лекційний матеріал до теми 4, готуватись до практичних занять, пройти тести в ДК Опрацювати теми: 1. Патерни у DDD. 2. Конфігурація Spring Boot.	18
Всього, годин	16		32		72

Таблиця 4. Тематичний план навчальної дисципліни для студентів заочної форми навчання

1	2	3	4	5	6
Назва теми (лекції) та питання теми (лекції)	кількість годин	Назва теми та питання семінарського, практичного або лабораторного заняття	кількість годин	Завдання самостійної роботи в розрізі тем	кількість годин
Модуль 1 Парадигми програмування					
Тема 1. Основні парадигми програмування					
Лекція 1. Вступ до предмету 1. Основні поняття 2. Методології розробки ПЗ 3. Функції і особливості мислення програмістів Лекція 2. Парадигми програмування 1. Парадигми (методології)	 2	Практична робота №1. Структурне та об'єктно-орієнтоване програмування Практична робота №2. Структурне та об'єктно-орієнтоване програмування		опрацювати лекційний матеріал до теми 1, готуватись до практичних занять, опрацювати питання: 1. Альтернативні класифікації парадигм програмування. 2. Паралельне програмування.	28

1	2	3	4	5	6
програмування та їх класифікація 2. Основні парадигми програмування 3. Інші парадигми та перспективи					
Тема 2. Логічне та функційне програмування					
Лекція 3. Мова логічного програмування Prolog 1. Основи мови Prolog 2. Приклади Лекція 4. Особливості функційного програмування 1. Основні особливості функційного програмування 2. Лінійні обчислення 3. Способи передачі аргументів 4. Stream API в Java		Практична робота №3-4. Логічне програмування Практична робота №5-6. Лінійні обчислення Практична робота №7. Функційний підхід в реалізації циклів Практична робота №8. Робота з Stream.API в Java Практична робота №9. МКР№1. Парадигми програмування	2	опрацювати лекційний матеріал до теми 2, готуватись до практичних занять, пройти тести в ДК Опрацювати теми: 1. Списки в функційних та логічних мовах програмуваннях. 2. Мови функційного програмування	28
Модуль 2. Особливості реалізації об'єктно-орієнтованого програмування в різних мовах програмування					
Тема 3. Особливості об'єктно-орієнтованого програмування.					
Лекція 5. Типізація даних 1. Система типізації 2. Статична та динамічна типізація 3. Строга (сильна) та нестрога (слабка) типізація 4. Явна та неявна типізація 5. Тип dynamic в C# Лекція 6. Особливості ООП 1. Класи і об'єкти 2. Складові елементи класів 3. Модифікатори доступу		Практична робота №10. Типізація даних Практична робота №11. Відношення між класами Практична робота №12. Перевантаження в ООП		опрацювати лекційний матеріал до теми 3, готуватись до практичних занять Опрацювати теми: 1. Функційне програмування в об'єктно-орієнтованих мовах. 2. Багато-парадигменні мови програмування.	28

1	2	3	4	5	6
4. Відношення класів Лекція 7. Перевантаження в об'єктно-орієнтованих мовах 1. Перевантаження методів та конструкторів 2. Перевантаження операторів					
Тема 4. Spring як мультипарадигмовий фреймворк					
Лекція 8. Парадигми програмування в Spring Framework 1. Spring Framework 2. Відмінності між POJO, Entity та DTO- об'єктами 3. Основи Domain- Driven Design (DDD) 4. Сервіси у Spring Framework 5. Репозиторії в Spring Data JPA 6. Аспектно- орієнтоване програмування у Spring 7. Реактивне програмування у Spring	2	Практичне заняття №13-14. Реактивне програмування Практичне заняття №15. Аспектно- орієнтоване програмування Практична робота №16. МКР №2. Особливості реалізації ООП в різних мовах програмування	2	опрацювати лекційний матеріал до теми 4, готуватись до практичних занять, пройти тести в ДК Опрацювати теми: 1. Патерни у DDD. 2. Конфігурація Spring Boot.	28
Всього, годин	4		4		112

Розділ 5. Система оцінювання знань студентів

Таблиця 5.1 - Розподіл балів за результатами вивчення навчальної дисципліни

Вид робіт	Максимальна кількість балів
Модуль 1 Парадигми програмування	
<u>Практичні роботи 1-2 (тема 1)</u>	2 практичних роботи x 5 балів = 10
<u>Практичні роботи 3-8 (тема 2)</u>	6 практичних робіт x 5 балів = 30
<u>Практична робота 9</u> (Контрольна робота №1)	15
Всього за модулем 1	55
Модуль 2. Особливості реалізації об'єктно-орієнтованого програмування в різних	

Вид робіт	Максимальна кількість балів
мовах програмування	
<i>Практичні роботи 10-12 (тема 3)</i>	3 практичних роботи x 5 балів = 15
<i>Практичні роботи 13-15 (тема 4)</i>	3 практичних роботи x 5 балів = 15
<i>Практична робота 20</i> (Контрольна робота №2)	15
Всього за модулем 2	45
Всього по курсу	<u>100</u>

Система нарахування додаткових балів за видами робіт з вивчення навчальної дисципліни

Форма роботи	Вид роботи	Бали
Навчальна робота	Опрацювання додаткових навчальних курсів, написання та наукових публікацій на тему, що відповідає дисципліні	10*

За додаткові види навчальних робіт студент може отримати не більше 10 балів. Додаткові бали додаються до загальної підсумкової оцінки за вивчення навчальної дисципліни, але загальна підсумкова оцінка не може перевищувати 100 балів.

Таблиця 5.2 – Шкала оцінювання знань здобувачів вищої освіти за результатами вивчення навчальної дисципліни

Сума балів за всі види навчальної діяльності	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	Відмінно
82-89	B	Дуже добре
74-81	C	Добре
64-73	D	Задовільно
60-63	E	Задовільно достатньо
35-59	FX	Незадовільно з можливістю повторного складання
0-34	F	Незадовільно з обов'язковим повторним вивченням навчальної дисципліни

Розділ 6. Інформаційні джерела

1. Buch, G., R. A. Maksimchuk, and M. U. Engle. "Object-oriented analysis and design with examples of applications."

2. Eckel B. Thinking in Java, 4th Edition. Prentice-Hall PTR, 2006.
3. Васильєв О. М. Програмування мовою Java / О. М. Васильєв – Bohdan Books, 2022.
4. Новожилова М. В. Використання мови логічного програмування Visual Prolog для розробки експертних систем : навч. посібник / М. В. Новожилова, О. О. Петрова – Харків : ХНУМГ ім. О. М. Бекетова, 2019. – 89 с.
5. Коваль А. І. Порівняння об'єктно-орієнтованої та функційної парадигм програмування у проектуванні програмного забезпечення / А. І. Коваль, О. М. Яшина, Г. І. Радельчук, Ю. В. Форкун // Вісник Хмельницького національного університету, №3, 2021 - С. 34-38.
6. Schildt H. "The complete reference Java.", 2020.
7. SWI-Prolog: comprehensive free Prolog environment / Download, Documentation, Tutorials, Community. URL: <http://www.swi-prolog.org/>
8. Beßler, Daniel, Sascha Jongebloed, and Michael Beetz. "Prolog as a Querying Language for MongoDB." arXiv preprint arXiv:2110.01284 (2021).
9. Llanes, Jose E. Zalacain. "Java Prolog Interface." arXiv preprint arXiv:2203.17134 (2022).
10. Geleßus, David, and Michael Leuschel. "Making ProB compatible with SWI-Prolog." arXiv preprint arXiv:2205.04373 (2022).
11. Nostas, J., Alcocer, J. P. S., Costa, D. E., & Bergel, A. (2021, September). How do developers use the Java Stream API? In International Conference on Computational Science and Its Applications (pp. 323-335). Springer, Cham.
12. Baraik, D. K. (2022). Modern JAVA: Functional Programming. Dhiraj Baraik.
13. Zheng, M., Yang, J., Wen, M., Zhu, H., Liu, Y., & Jin, H. (2021, November). Why Do Developers Remove Lambda Expressions in Java?. In 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE) (pp. 67-78). IEEE
14. Олексійчук Ю. Ф., Ольховська О. В., Ольховський Д. М., Орлова Д. І. Проектування та розробка web-сервісу для генерування та розсилки pdf-документів. / Системи та технології, 65(1), 2023 – С. 39-45. <https://doi.org/10.32782/2521-6643-2023.1-65.5>

Розділ 7. Програмне забезпечення навчальної дисципліни

Для вивчення навчальної дисципліни використовується наступне програмне забезпечення:

1. NetBeans (або інше IDE для мови програмування Java)
2. SWI-Prolog