

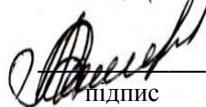
ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»

Навчально-науковий інститут денної освіти

Кафедра комп'ютерних наук та інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри



Олена ОЛЬХОВСЬКА
ініціали, прізвище

«_28_» _____06_____ 2024 р.

РОБОЧА ПРОГРАМА

навчальної дисципліни	<u>"Сучасні парадигми програмування"</u>
освітня програма/спеціалізація	<u>«Комп'ютерні науки»</u>
спеціальність	<u>122 "Комп'ютерні науки"</u>
галузь знань	<u>12"Інформаційні технології"</u>
ступінь вищої освіти	<u>бакалавр</u>

Робоча програма навчальної дисципліни "Сучасні парадигми програмування" схвалена та рекомендована до використання в освітньому процесі на засіданні кафедри комп'ютерних наук та інформаційних технологій

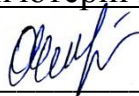
Протокол № _13_ від «_28_» ___06___ 2024 року

Полтава 2024

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

ПОГОДЖЕНО:

Гарант освітньої програми «Комп'ютерні науки»
спеціальності 122 "Комп'ютерні науки"
ступеня бакалавр



Оксана ЧЕРНЕНКО

підпис

ініціали, прізвище

«_28_» червня 2024 року

Розділ 1. Загальна характеристика навчальної дисципліни

Таблиця 1. Загальна характеристика навчальної дисципліни «Сучасні парадигми програмування»

Місце у структурно-логічній схемі підготовки	<i>Пререквізити:</i> Програмування <i>Постреквізити:</i> Кваліфікаційна робота	
Мова викладання	Українська	
Статус дисципліни	Обов'язкова	
Курс/семестр вивчення	4/8	
Кількість кредитів ECTS/кількість модулів	4/1	
Денна форма навчання:		
Кількість годин: 120 год – загальна кількість		
- Лекції: 16 год		
- Практичні заняття: 32 год		
- Самостійна робота: 72 год		
- Вид підсумкового контролю (ПМК, екзамен): ПМК		
Заочна форма навчання:		
Кількість годин: 120 год – загальна кількість		
- Лекції: 6 год		
- <u>Практичні</u> (семінарські, лабораторні) заняття: 4 год		
- Самостійна робота: 110 год		
- Вид підсумкового контролю (ПМК, екзамен): ПМК		

Розділ 2. Перелік компетентностей та програмні результати навчання з навчальної дисципліни

Предметом дисципліни «Сучасні парадигми програмування» є процес створення програмного продукту, використання при цьому сучасних парадигм програмування.

Метою вивчення дисципліни «Сучасні парадигми програмування» є формування у студентів системного мислення, узагальнення та систематизація знань та навичок з програмування, поглиблення знань з об'єктно-орієнтованого програмування.

Основним завданням дисципліни є набуття студентами знань та вмінь використання сучасних парадигм програмування.

Таблиця 2. Перелік компетентностей та програмні результати навчання

Програмні результати навчання	Компетентності, якими повинен оволодіти здобувач
<p>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.</p> <p>ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.</p> <p>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.</p>	<p>Здатність до абстрактного мислення, аналізу та синтезу (ЗК1).</p> <p>Здатність застосовувати знання у практичних ситуаціях (ЗК2).</p> <p>Знання та розуміння предметної області та розуміння професійної діяльності (ЗК3).</p> <p>Здатність спілкуватися державною мовою як усно, так і письмово (ЗК4).</p> <p>Здатність вчитися й оволодівати сучасними знаннями (ЗК6).</p> <p>Здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК7).</p> <p>Здатність генерувати нові ідеї (креативність) (ЗК8).</p> <p>Здатність працювати в команді (ЗК9).</p> <p>Здатність бути критичним і самокритичним (ЗК10).</p> <p>Здатність приймати обґрунтовані рішення (ЗК11)</p> <p>Здатність оцінювати та забезпечувати якість виконуваних робіт (ЗК12).</p> <p>Здатність діяти на основі етичних міркувань (ЗК13).</p> <p>Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності,</p>

розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем (СКЗ).
 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

Тема 1. Парадигми програмування.

Парадигми програмування. Імперативне програмування. Функційне програмування. Логічне програмування. Структурне програмування. Об'єктно-орієнтоване програмування. Мова програмування Prolog. Лінійні обчислення. Лінійні обчислення в ООП. Способи передачі аргументів функцій. Функційне програмування в Java. Stream API.

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

Типи даних. Системи типізації. Статична та динамічна типізація. Строга (сильна) та нестрога (слабка) типізація. Явна та неявна типізація. Поєднання функційного та об'єктно-орієнтованого підходів. Інкапсуляція. Успадкування. Поліморфізм. Інтерфейси. Абстрактні класи. Перевантаження.

Розділ 4. Тематичний план навчальної дисципліни

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. Сучасні Java-фреймворки для роботи з реляційними базами даних					
Тема 1. Основні парадигми програмування					
<u>Лекція 1.</u> Парадигми програмування. 1. Методології програмування та їх класифікація	2	<u>Лабораторна робота 1-2.</u> Структурне та об'єктно-орієнтоване програмування. <u>Лабораторна робота</u>	4 6	опрацювати лекційний матеріал до теми 1, готуватись до практичних занять	36

1	2	3	4	5	6
<p>2. Методологія імперативного програмування.</p> <p>3. Методологія об'єктно-орієнтованого програмування</p> <p>4. Функціональне програмування.</p> <p>5. Методологія логічного програмування.</p> <p>6. Методологія програмування в обмеженнях</p> <p>7. Структурне програмування.</p> <p>8. Інші методології та перспективи розвитку.</p> <p>9. Методології паралельного програмування</p> <p><u>Лекція 2.</u> Логічне програмування</p> <p>1. Основи мови Prolog</p> <p>2. Приклади</p> <p><u>Лекція 3.</u> Особливості функційного програмування</p> <p>1. Лінійні обчислення</p> <p>2. Способи передачі аргументів</p> <p>3. Лінійні обчислення в ООП</p> <p><u>Лекція 4.</u> Особливості функційного програмування в Java</p> <p>1. Функційне програмування в Java</p> <p>2. Stream API</p>		<p><u>3-5.</u> Логічне програмування.</p> <p><u>Лабораторна робота 6-7.</u> Лінійні обчислення.</p> <p><u>Лабораторна робота 8.</u> Робота з Stream в Java.</p> <p><u>Лабораторна робота 9.</u> КР №1. Парадигми програмування.</p>	<p>4</p> <p>2</p> <p>2</p>		
<u>Тема 2.</u> Особливості реалізації об'єктно-орієнтованого програмування в різних мовах програмування					
<p><u>Лекція 5.</u> Типізація даних</p> <p>1. Система типізації</p> <p>2. Статична та динамічна типізація</p> <p>3. Строга (сильна) та нестрога (слабка) типізація</p> <p>4. Явна та неявна типізація</p> <p>5. Тип dynamic в C#</p> <p><u>Лекція 6.</u> Класи</p>	2	<p><u>Лабораторна робота 10-11.</u> Поєднання об'єктно-орієнтованого та функційного підходів.</p> <p><u>Лабораторна робота 12-13.</u> Перевантаження.</p>		опрацювати лекційний матеріал до теми 2	36

1	2	3	4	5	6
1. Класи і об'єкти 2. Складові елементи класів 3. Модифікатори доступу 4. Відношення класів <u>Лекція 7.</u> Перевантаження 1. Перевантаження методів та конструкторів 2. Перевантаження операторів <u>Лекція 8.</u> Успадкування 1. Конструктори і успадкування 2. Перевизначення (заміщення) методів 3. Абстрактні класи		<u>Лабораторна робота 14-15.</u> Успадкування. <u>Лабораторна робота 16.</u> КР №2. Особливості ООП			
Всього, годин	16		32		72

Розділ 5. Система поточного та підсумкового контролю знань студентів

Форма навчальної роботи	Вид навчальної роботи	Бали
1. Аудиторна	▪	
1. Практичні заняття	▪ Підготовка до лабораторної роботи та її виконання (5x14=70)	70
2. Підсумковий контроль.	КР№1	15
	КР№2	15
Усього за семестр		100

Розділ 6. Програмне забезпечення комп'ютерної підтримки освітнього процесу

Використовується наступне програмне забезпечення:

1. Середовище розробки для мови програмування Java NetBeans (або Eclipse, або IntelliJIDEA).

2. SWI-Prolog.

Розділ 7. Рекомендовані джерела інформації

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