


**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»
Навчально-науковий інститут денної освіти
Кафедра комп'ютерних наук та інформаційних технологій**

ЗАТВЕРДЖУЮ

Завідувач кафедри


Олена ОЛЬХОВСЬКА

«25» січня 2023 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 25 січня 2023 року, №8


Полтава 2023

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

«25» січня 2023 року

Зміст
робочої програми початкової дисципліни

Розділ 1. Загальна характеристика навчальної дисципліни	4
Розділ 2. Перелік компетентностей та програмні результати навчання з навчальної дисципліни	4
Розділ 3. Програма навчальної дисципліни	6
Розділ 4. Тематичний план навчальної дисципліни	6
Розділ 5. Оцінювання результатів навчання	10
Розділ 6. Інформаційні джерела	11
Розділ 7. Програмне забезпечення навчальної дисципліни	12

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> – Дипломне проектування, Атестація
Мова викладання	Українська
Статус дисципліни	Обов'язкова
Курс/семестр вивчення	4/1
Кількість кредитів ECTS/кількість модулів	4/1
Денна форма навчання:	
Кількість годин: 120 год – загальна кількість: 1 семестр – 120 год	
- Лекції: 16 год.	
- Практичні (семінарські, лабораторні) заняття: 32 год.	
- Самостійна робота: 72 год.	
- Вид підсумкового контролю (ПМК, екзамен): ПМК	
Заочна форма навчання:	
Кількість годин: 120 год – загальна кількість: 1 семестр – 120 год	
- Лекції: 4 год.	
- Практичні (семінарські, лабораторні) заняття: 2 год.	
- Самостійна робота: 114 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, які забезпечує навчальна дисципліна «Сучасні парадигми програмування»

Компетентності, якими повинен оволодіти	Програмні результати навчання
---	-------------------------------

здобувач	
<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 математичного формулювання та досліджування неперервних та дискретних математичних моделей, обґрунтування вибору методів і підходів для розв'язування теоретичних і прикладних задач у галузі комп'ютерних наук, аналізу та інтерпретування (СК1).</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, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем (СК3).</p> <p>Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними</p>	<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>моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління (СК8).</p>
--

Розділ 3. Програма навчальної дисципліни

Модуль 1 «Основи програмування алгоритмічною мовою»

Тема 1. Парадигми програмування. Імперативне програмування. Функційне програмування. Логічне програмування. Структурне програмування. Об'єктно-орієнтоване програмування.

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

Типи даних. Поєднання функційного та об'єктно-орієнтованого підходів. Інкапсуляція. Успадкування. Поліморфізм. Інтерфейси. Абстрактні класи. Перевантаження.

Розділ 4. Тематичний план навчальної дисципліни

Таблиця 3. 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. Основні парадигми програмування					
Тема 1. Парадигми програмування. <u>Лекція 1.</u> Вступ до предмету 1. Вступ. 2. Основні поняття. 3. Предметна область інформатики. 4. Функції і особливості мислення програмістів. <u>Лекція 2.</u> Парадигми програмування. 1. Методології програмування та їх класифікація 2. Методологія імперативного програмування. 3. Методологія об'єктно-орієнтованого	2	<u>Лабораторна робота 1-2.</u> Структурне та об'єктно-орієнтоване програмування. <u>Лабораторна робота 3-5.</u> Логічне програмування. <u>Лабораторна робота 6-7.</u> Лінійні обчислення. <u>Лабораторна робота 8.</u> Робота з Stream в Java. <u>Лабораторна робота 9.</u> КР №1. Парадигми програмування.	4 6 4 2 2	опрацювати лекційний матеріал до теми 1, готуватись до практичних занять, пройти тест до теми в ДК Опрацювати теми: 1. Мови функціонального програмування. 2. Паралельне програмування. 3. Функційне програмування і мова Java	36

1	2	3	4	5	6
операторів Лекція 8. Успадкування 1. Конструктори і успадкування 2. Перевизначення (заміщення) методів 3. Абстрактні класи					
Всього, годин	16		32		144

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. Основні парадигми програмування					
Тема <u>1</u> . Парадигми програмування. Лекція 1. Вступ до предмету 1. Вступ. 2. Основні поняття. 3. Предметна область інформатики. 4. Функції і особливості мислення програмістів. Лекція 2. Парадигми програмування. 1. Методології програмування та їх класифікація 2. Методологія імперативного програмування. 3. Методологія об'єктно-орієнтованого програмування 4. Функціональне програмування. 5. Методологія логічного програмування. 6. Методологія програмування в обмеженнях 7. Структурне програмування. 8. Інші методології та	2	Лабораторна робота 1-2. Структурне та об'єктно-орієнтоване програмування. Лабораторна робота 3-5. Логічне програмування. Лабораторна робота 6-7. Лінійні обчислення. Лабораторна робота 8. Робота з Stream в Java. Лабораторна робота 9. КР №1. Парадигми програмування.	2	опрацювати лекційний матеріал до теми 1, готуватись до практичних занять, пройти тест до теми в ДК Опрацювати теми: 1. Мови функціонального програмування. 2. Паралельне програмування. 3. Функціональне програмування і мова Java	36

1	2	3	4	5	6
<p>перспективи розвитку.</p> <p>9. Методології паралельного програмування</p> <p><u>Лекція 3.</u> Логічне програмування</p> <p>1. Основи мови Prolog</p> <p>2. Приклади</p> <p><u>Лекція 4.</u> Особливості функціонального програмування</p> <p>1. Лінійні обчислення</p> <p>2. Способи передачі аргументів</p> <p>3. Лінійні обчислення в ООП</p>					
<p><u>Тема 2.</u> Особливості реалізації об'єктно-орієнтованого програмування в різних мовах програмування.</p> <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> <p>1. Класи і об'єкти</p> <p>2. Складові елементи класів</p> <p>3. Модифікатори доступу</p> <p>4. Відношення класів</p> <p><u>Лекція 7.</u> Перевантаження</p> <p>1. Перевантаження методів та конструкторів</p> <p>2. Перевантаження операторів</p> <p><u>Лекція 8.</u> Успадкування</p> <p>1. Конструктори і успадкування</p> <p>2. Перевизначення (заміщення) методів</p> <p>3. Абстрактні 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> <p><u>Лабораторна робота 14-15.</u> Успадкування.</p> <p><u>Лабораторна робота 16.</u> КР №2. Особливості ООП</p>		<p>опрацювати лекційний матеріал до теми 2, готуватись до практичних занять, пройти тести до теми в ДК</p> <p>Опрацювати теми:</p> <p>1. Інтерфейси</p> <p>2. Анотації в Java</p> <p>3. Функціональне програмування в об'єктно-орієнтованих мовах.</p> <p>4. Багато-парадигменні мови програмування.</p>	36
Всього, годин	4		2		112

Розділ 5. Оцінювання результатів навчання

Таблиця 5. Розподіл балів за результатами вивчення навчальної дисципліни

Форма навчальної роботи	Вид навчальної роботи	Бали
1. Аудиторна 1.1. Лекція	▪ Відвідування всіх лекцій лекції та лабораторних	20
1.2. Практичні заняття	▪ Підготовка до лабораторної роботи та її виконання (4x14=56)	56
2. Підсумковий контроль.	КР (2*12=24)	24
Усього за семестр		100

Додаткові бали:

Форма роботи	Вид роботи	Бали
1. Навчальна	1. Участь в предметних олімпіадах: університетських, міжвузівських, всеукраїнських, міжнародних	5
	2. Участь в конкурсах на кращого знавця дисципліни: університетських, міжвузівських, всеукраїнських, міжнародних	3
2. Науково-дослідна	1. Участь в наукових гуртках	2
	2. Участь в наукових студентських клубах	2
	3. Участь в наукових магістерських семінарах	2
	4. Участь в конкурсах студентських робіт: університетських, міжвузівських, всеукраїнських, міжнародних	5
	5. Участь в наукових студентських конференціях: університетських, міжвузівських, всеукраїнських, міжнародних	5

Шкала оцінювання здобувачів вищої освіти за результатами вивчення навчальної дисципліни

<i>Сума балів за всі види навчальної діяльності</i>	<i>Оцінка за шкалою ЄКТС</i>	<i>Оцінка за національною шкалою</i>
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

Основна

1. Васильєв О. М. Програмування мовою Java / О. М. Васильєв - Bohdan Books - 2022.
2. SWI-Prolog: comprehensive free Prolog environment / Download, Documentation, Tutorials, Community. URL: <http://www.swi-prolog.org/>
3. Schildt, Herbert. Java: a beginner's guide. McGraw-Hill Education, 2022.
4. Downey, Allen B., and Chris Mayfield. Think Java: How to think like a computer scientist. O'Reilly Media, 2019.
5. Oaks, Scott. Java performance: in-depth advice for tuning and programming Java 8, 11, and beyond. " O'Reilly Media, Inc.", 2020.
6. Farrell, Joyce. Java programming. Cengage Learning, 2022.
7. 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. Технічні науки. - 2021. - № 3. - С. 34-38.

Додаткова

8. Xiao, Perry. Practical Java Programming for IoT, AI, and Blockchain. John Wiley & Sons, 2019.
9. Buonanno, Enrico. Functional Programming in C. Simon and Schuster, 2022.
10. Beßler, Daniel, Sascha Jongebloed, and Michael Beetz. "Prolog as a Querying Language for MongoDB."arXiv preprint arXiv:2110.01284 (2021).
11. Llanes, Jose E. Zalacain. "Java Prolog Interface."arXiv preprint arXiv:2203.17134 (2022).
12. Geleßus, David, and Michael Leuschel. "Making ProB compatible with SWI-Prolog."arXiv preprint arXiv:2205.04373 (2022).
13. 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.
14. Baraik, D. K. (2022). Modern JAVA: Functional Programming. Dhiraj Baraik.
15. 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.

Розділ 7. Програмне забезпечення навчальної дисципліни

Використовується наступне програмне забезпечення:

1. Середовище розробки для мови програмування Java NetBeans (або Eclipse, або IntelliJ IDEA).
2. SWI-Prolog.