

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ:

В. о. декана факультету
комп'ютерних інформаційних
технологій

Ігор ЯКИМЕНКО

“ ” 2023 р.

ЗАТВЕРДЖУЮ:

Директор навчально-наукового
інституту новітніх освітніх
технологій

Святослав ПИТЕЛЬ

“ ” 2023 р.

ЗАТВЕРДЖУЮ:

В. о. проректора з науково-педагогічної
роботи

Віктор ОСТРОВЕРХОВ

“ ” 2023 р.

РОБОЧА ПРОГРАМА

з дисципліни «Сучасні парадигми програмування»

ступінь вищої освіти – бакалавр

галузь знань – 12 “Інформаційні технології”

спеціальність – 122 „Комп’ютерні науки”

освітньо-професійна програма – „Комп’ютерні науки”

Кафедра інформаційно-обчислювальних систем і управління

Форма навчання	Курс	Семестр	Лекції (год.)	Лабораторні заняття (год.)	ІРС (год.)	Тренінг, КПІЗ (год.)	Самост. робота студ. (год.)	Разом (год.)	Залік (сем.)
Денна	1	2	30	30	3	8	79	150	2
Заочна	1	2	8	4	–	–	138	150	2

Тернопіль – ЗУНУ
2023

31.08.2023

Робоча програма складена на основі освітньо-професійної програми «Комп'ютерні науки» підготовки бакалавра галузі знань 12 «Інформаційні технології» спеціальності 122 «Комп'ютерні науки», затвердженої Вченою радою ЗУНУ (протокол №10 від 23.06.2023 р.).

Робочу програму склав доцент кафедри ІОСУ, к.т.н. Павло БИКОВИЙ

Робоча програма затверджена на засіданні кафедри інформаційно-обчислювальних систем і управління, протокол № 1 від 29 серпня 2023 р.

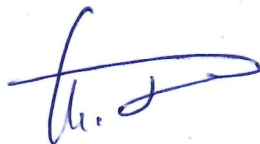
Завідувач кафедри



Мирослав КОМАР

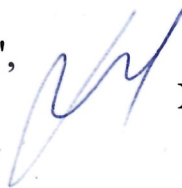
Розглянуто та схвалено групою забезпечення спеціальності „Комп'ютерні науки”, протокол № 1 від 29 серпня 2023 р.

Голова групи
забезпечення спеціальності,
д-р техн. наук, професор



Мирослав КОМАР

Гарант освітньо-професійної
програми "Комп'ютерні науки",
канд. техн. наук, доцент



Христина ЛІП'ЯНІНА-ГОНЧАРЕНКО

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ " СУЧАСНІ ПАРАДИГМИ ПРОГРАМУВАННЯ "

1. Опис дисципліни " СУЧАСНІ ПАРАДИГМИ ПРОГРАМУВАННЯ "

Дисципліна «Сучасні парадигми програмування»	Галузь знань, спеціальність, СВО	Характеристика навчальної дисципліни
Кількість кредитів – 5	Галузь знань – 12 “Інформаційні технології”	Статус дисципліни: обов’язкова дисципліна циклу професійної підготовки Мова навчання: українська
Кількість залікових модулів – 3	Спеціальність – 122 «Комп’ютерні науки»	Рік підготовки: 1 Семестр: <i>Денна – 2</i> <i>Заочна – 2</i>
Кількість змістових модулів – 3	Освітньо- професійна програма «Комп’ютерні науки»	Лекції: <i>Денна – 30 год.</i> <i>Заочна – 8 год.</i> Лабораторні заняття: <i>Денна – 30 год.</i> <i>Заочна – 4 год.</i>
Загальна кількість годин – 150	Ступінь вищої освіти – бакалавр	Самостійна робота: <i>Денна – 79 год.,</i> <i>Заочна – 138 год.</i> Тренінг, КПЗ: <i>Денна – 8 год.</i> Індивідуальна робота: <i>Денна – 3 год.</i>
Тижневих годин – 10, з них аудиторних – 4 год.		Вид підсумкового контролю – залік

2. Мета і завдання дисципліни "СУЧАСНІ ПАРАДИГМИ ПРОГРАМУВАННЯ"

2.1. Мета вивчення дисципліни

Мета вивчення дисципліни «Сучасні парадигми програмування» – дати студентам систематизовані відомості про основні принципи розробки програмного забезпечення, методи та особливості реалізації програмного коду та провести огляд основних технологій розробки програмного забезпечення.

2.2. Завдання вивчення дисципліни

В результаті вивчення дисципліни студент повинен:

- розуміти процеси та технології створення програмного забезпечення, сервісів, та інтерфейсів з інтеграцією зовнішніх даних і програмних продуктів;
- аналізувати та аргументувати рішення щодо методів розробки веб-застосувань, моделей їх архітектур, засобів інтеграції програмного забезпечення із зовнішніми даними та програмними продуктами, методів забезпечення безпеки коду та даних тощо;
- здійснювати синтез архітектури і ефективних алгоритмів функціонування програмного забезпечення в процесі, моделювання, проектування та програмування програмного забезпечення складних об'єктів і систем професійної діяльності.
- проводити емпіричні дослідження, оцінювати ефективність рішень та якість програмного забезпечення, визначати критерії вибору платформи для розробки програмного забезпечення.

2.3. Найменування та опис компетентностей, формування яких забезпечує вивчення дисципліни:

СК3. 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.

СК8. 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.

СК13. Здатність до розробки мережевого програмного забезпечення, що функціонує на основі різних топологій структурованих кабельних систем, використовує комп'ютерні системи і мережі передачі даних та аналізує якість роботи комп'ютерних мереж.

2.4. Передумови для вивчення дисципліни

Вивчення курсу „Сучасні парадигми програмування” передбачає наявність систематичних та ґрунтовних знань із дисциплін: «Основи програмування»,

«Алгоритми та структури даних».

2.5. Результати навчання

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.

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.

ПР13. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.

3. Програма навчальної дисципліни «СУЧАСНІ ПАРАДИГМИ ПРОГРАМУВАННЯ» (Частина 1)

Змістовий модуль 1. Процедурно-орієнтоване програмування

Тема 1. Парадигми програмування

Парадигми і мови програмування. Узагальнене, об'єктно-орієнтоване, функціональне і логічне програмування. Програмні середовища.

Тема 2. Базові поняття програмування. Алгоритми розв'язання обчислювальних та логічних задач

Фундаментальні типи даних. Арифметичні та логічні операції. Оператори розгалуження, переходу та циклу. Масиви. Пошук і сортування. Структури. Бітові поля. Об'єднання.

Тема 3. Оцінка ефективності та складності алгоритмів

Часова складність алгоритмів. Поліноміальні та експоненціальні алгоритми.

Тема 4. Функції

Визначення й оголошення функцій. Механізми передавання параметрів значенням. Параметри-відсилки. Передавання параметрів-масивів. Особливості передавання сталих параметрів. Шаблони функцій. Техніка рекурсивного програмування. Ефективна рекурсія. Вказівники на функції.

Змістовий модуль 2. Об'єктне програмування

Тема 5. Об'єктно - орієнтоване програмування та його головні принципи.

Абстрагування. Обмеження доступу. Модульність. Ієрархія

Тема 6. Класи

Протокол опису класу. Створення об'єктів. Доступ до полів та методів. Використання специфікаторів доступу класу. Правила визначення конструкторів. Деструктори. Порядок виклику конструкторів та деструкторів. Статичні члени класу.

Тема 7. Уніфікована мова моделювання UML

Загальна характеристика. Архітектурний базис. Відношення. Діаграми. Зображення класу. Асоціації між класами. Агрегація та композиція. Узагальнення та залежності.

Тема 8. Успадкування

Механізм успадкування. Керування доступом при успадкуванні. Друзі-класи та друзі-функції.

Тема 9. Поліморфізм

Віртуальні функції. Чисті віртуальні функції та абстрактні базові класи. Розміщення VPTR та таблиці VMT у пам'яті. Віртуальні деструктори.

Тема 10. Перевантаження операцій

Перевизначення з використанням функцій членів та не членів класу. Особливості реалізації оператора присвоєння.

Тема 11. Шаблони

Параметризовані функції. Параметризовані класи

Тема 12. Стандартна бібліотека шаблонів

Огляд стандартної бібліотеки шаблонів. Поняття про контейнерні класи. Механізми роботи з векторами. Механізми роботи зі списками. Поняття про відображення – асоціативний контейнер. Алгоритми оброблення контейнерних даних. Особливості використання об'єктів класу string.

Змістовий модуль 3. Взаємодія між програмними компонентами комп'ютерних систем

Тема 13. Розробка Windows-базованого програмного забезпечення.

Основні характеристики Windows. Поняття API. Компонування бібліотек у Windows. Об'єкти ядра.

Тема 14. Процеси в ОС Windows

Поняття процесу. Створення та завершення процесів

Тема 15. Потіки в ОС Windows

Поняття потоку. Планування потоків і пріоритет. Синхронізація потоків

Тема 16. Робота з мережевими протоколами

Використання стеку протоколів TCP/IP. Стандартні і нестандартні протоколи прикладного рівня, приклади їх використання. Приклад використання Telnet.

Тема 17. Програмний інтерфейс протоколів. Сокети API.

Специфікація прикладного програмного інтерфейсу (API). Дескриптори сокетів. Використання сокетів. Встановлення адреси кінцевої точки. Загальна структура адреси. Функції в сокетах API.

4. Структура залікового кредиту з дисципліни «СУЧАСНІ ПАРАДИГМИ ПРОГРАМУВАННЯ» (Частина 1)

Денна форма навчання

Тема	Кількість годин					
	Лекції	Лабораторні заняття	Індивідуальна робота	Тренінг, КПЗ	Самостійна робота	Контрольні заходи
Змістовий модуль 1. Процедурно-орієнтоване програмування						
Тема 1. Парадигми програмування	2	-	1	2	5	Опитування під час заняття
Тема 2. Базові поняття програмування. Алгоритми розв'язання обчислювальних та логічних задач	2	3	-		5	Опитування під час заняття
Тема 3. Оцінка ефективності та складності алгоритмів	1	-	-		5	Опитування під час заняття
Тема 4. Функції	1	2	-		5	Опитування під час заняття
Змістовий модуль 2. Об'єктне програмування						
Тема 5. Об'єктно-орієнтоване програмування та його головні принципи.	2	-	1	3	5	Опитування під час заняття
Тема 6. Класи	2	3	-		5	Опитування під час заняття

Тема 7. Уніфікована мова моделювання UML	2	2	-		5	Опитування під час заняття
Тема 8. Успадкування	2	3	-		5	Опитування під час заняття
Тема 9. Поліморфізм	1	3	-		5	Опитування під час заняття
Тема 10. Перевантаження операцій	2	-	-		4	Опитування під час заняття
Тема 11. Шаблони	1	2	-		4	Опитування під час заняття
Тема 12. Стандартна бібліотека шаблонів	2	-	-		4	Опитування під час заняття
Змістовий модуль 3. Взаємодія між програмними компонентами комп'ютерних систем						
Тема 13. Розробка Windows-базованого програмного забезпечення.	2	-	1	3	5	Опитування під час заняття
Тема 14. Процеси в ОС Windows	2	2	-		4	Опитування під час заняття
Тема 15. Потoki в ОС Windows	2	2	-		4	Опитування під час заняття
Тема 16. Робота з мережевими протоколами	2	4	-		5	Опитування під час заняття
Тема 17. Програмний інтерфейс протоколів. Сокети API.	2	4	-		4	Опитування під час заняття
Разом	30	30	3	8	79	

Заочна форма навчання

Тема	Кількість годин			
	Лекції	Лабораторні заняття	Тренінг, КПІЗ	Самостійна робота
Змістовий модуль 1. Процедурно-орієнтоване програмування				
Тема 1. Парадигми програмування	2	2	-	6
Тема 2. Базові поняття програмування. Алгоритми розв'язання обчислювальних та логічних задач				6
Тема 3. Оцінка ефективності та складності алгоритмів				6
Тема 4. Функції				6
Змістовий модуль 2. Об'єктне програмування				
Тема 5. Об'єктно-орієнтоване програмування та його головні принципи.	2	2	-	10
Тема 6. Класи				6
Тема 7. Уніфікована мова моделювання UML				6
Тема 8. Успадкування	2	-	-	6
Тема 9. Поліморфізм				6
Тема 10. Перевантаження операцій				8
Тема 11. Шаблони				10
Тема 12. Стандартна бібліотека шаблонів				10
Змістовий модуль 3. Взаємодія між програмними компонентами комп'ютерних систем				
Тема 13. Розробка Windows-базованого програмного забезпечення.	2	-	-	12
Тема 14. Процеси в ОС Windows				10
Тема 15. Потoki в ОС Windows				10
Тема 16. Робота з мережевими протоколами				10
Тема 17. Програмний інтерфейс протоколів. Сокети API.				10
Разом	8	4	-	138

5. Тематика лабораторних занять

1. Керуючі структури в C++.
2. Функції. Шаблони функцій.
3. Робота з класами. реалізація принципу інкапсуляції. Конструктори та деструктори.
4. Процес аналізу та проектування програмних систем на основі мови UML.
5. Робота з класами. Реалізація принципу успадкування.
6. Реалізація принципу поліморфізму в ООП. Перевантаження операцій.

7. Віртуальні функції. Шаблони класів.
8. Створення процесів і потоків в ОС Windows
9. Розробка мережевих застосунків в ОС Windows

6. Комплексне практичне індивідуальне завдання

Комплексне практичне індивідуальне завдання (КПІЗ) виконується кожним студентом одноосібно. Студенти повинні вибрати одну із запропонованих тем або запропонувати свою, відповідно до власних вподобань та в рамках даного предмету, обговорити та затвердити у викладача конкретне завдання у вибраній області.

Індивідуальне науково-дослідне завдання повинно містити:

- теоретичний опис обраної області;
- опис поставленого завдання;
- шляхи розв'язання поставленого завдання;
- представлення результатів.

Теми комплексних практичних індивідуальних завдань:

1. Створити вектор, що містить об'єкти класу "учасник": прізвище, телефон, адреса. Продемонструвати основні операції з контейнером.
2. Створити список, що містить об'єкти класу "фігура": координати на шахівниці, колір. Продемонструвати основні операції з контейнером.
3. Створити чергу, що містить об'єкти класу "прямокутник": координати верхнього лівого і нижнього правого кутів, порядковий номер. Продемонструвати основні операції з контейнером.
4. Створити стек, що містить об'єкти класу "нота": назва, октава, тривалість звучання. Продемонструвати основні операції з контейнером.
5. Створити вектор, що містить об'єкти класу "іграшка": ціна, назва, кількість на складі. Продемонструвати основні операції з контейнером.
6. Створити список, що містить об'єкти класу "годинник": стиль відображення (24 чи 12), години, хвилини, секунди. Продемонструвати основні операції з контейнером.
7. Створити чергу, що містить об'єкти класу "Товар": назва, порядковий номер, постачальник, ціна, кількість одиниць. Продемонструвати основні операції з контейнером.
8. Створити список, що містить об'єкти класу "Точка на площині": координати. Продемонструвати основні операції з контейнером.
9. Створити список, що містить об'єкти класу "учасник змагань": країна, вид спорту, назва учасника. Продемонструвати основні операції з контейнером.
10. Створити стек, що містить об'єкти класу "довге число": кількість знаків, основа системи числення. Продемонструвати основні операції з контейнером.
11. Створити вектор, що містить об'єкти класу "РадіоСтанція": координати_частота, назва. Продемонструвати основні операції з контейнером.
12. Створити вектор, що містить об'єкти класу " істота "": координати, вік, назва. Продемонструвати основні операції з контейнером.

13. Створити чергу, що містить об'єкти класу "кліматичні умови": температура, освітленість, вологість, кислотність гранта . Продемонструвати основні операції з контейнером.
14. Створити чергу, що містить об'єкти класу "давач": поточне значення, максимально і мінімально допустимі, тип (t, p, i, pH), сигнал тривоги. Продемонструвати основні операції з контейнером.
15. Створити список, що містить об'єкти класу "товар на складі": назва, кількість, місце розташування. Продемонструвати основні операції з контейнером.
16. Створити дек, що містить об'єкти класу "фраза": (кількість слів, кількість символів, кількість різних символів. Продемонструвати основні операції з контейнером.
17. Створити вектор, що містить об'єкти класу "коло": x, y, R. Продемонструвати основні операції з контейнером.
18. Створити список, що містить об'єкти класу "книга": код УДК, назва, автор, рік видання, кількість сторінок. Продемонструвати основні операції з контейнером.
19. Створити вектор, що містить об'єкти класу "обладнання": назва, вартість, дата виготовлення, група (5, 15, 25 – амортизація) і клас "модернізація обладнання" (дата ремонту, вартість ремонту). Продемонструвати основні операції з контейнером.
20. Створити дек, що містить об'єкти класу "товар": назва, виробник. Продемонструвати основні операції з контейнером.
21. Створити вектор, що містить об'єкти класу "банківський рахунок": назва банку, номер рахунку, МФО. Продемонструвати основні операції з контейнером.
22. Створити чергу з пріоритетом, що містить об'єкти класу "підключення": назва мережі, тариф, наявність пільги. Продемонструвати основні операції з контейнером.
23. Створити вектор, що містить об'єкти класу "іграшка": ціна, назва, кількість на складі. Продемонструвати основні операції з контейнером.
24. Створити дек, що містить об'єкти класу "учасник змагань": країна, вид спорту, назва учасника. Продемонструвати основні операції з контейнером.
25. Створити дек, що містить об'єкти класу "товар": назва, виробник. Продемонструвати основні операції з контейнером.

7. Самостійна робота

№ п/п	Тематика	Кількість годин	
		ДФН	ЗФН
1.	Об'єктний підхід в об'єктно-орієнтованому програмуванні.	3	5
2.	Класифікація в об'єктно-орієнтованому аналізі.	3	5
3.	Об'єктно-орієнтоване проектування. Система позначень.	3	5
4.	Ідентифікація зв'язків між класами та об'єктами.	3	5
5.	Об'єктно-орієнтоване програмування в життєвому циклі розробки програми.	3	5
6.	Переваги та недоліки об'єктно-орієнтованого програмування.	3	5
7.	Відношення між об'єктами.	3	5
8.	Відношення між класами.	3	5
9.	Взаємозв'язки класів та об'єктів.	3	5
10.	Перевантаження операторів.	3	5
11.	Наслідування.	3	5
12.	Віртуальні функції.	3	5
13.	Абстрактні класи.	3	5
14.	Контроль доступу до членів класу.	3	5
15.	Шаблони класів.	3	5
16.	Параметризовані функції.	3	5
17.	Обробка виняткових ситуацій.	3	5
18.	Множинне наслідування. Керування доступом.	4	5
19.	Інформація про тип на етапі виконання.	3	5
20.	Стандартні контейнери.	3	5
21.	Визначення нового контейнера.	4	10
22.	Алгоритми стандартної бібліотеки.	4	9
23.	Ітератори і розподіл пам'яті.	4	9
24.	Потоки вводу/виводу.	3	5
25.	Стандартні бібліотеки.	3	5
ВСЬОГО		79	138

8. Тренінг з дисципліни

№ з/п	Вид роботи	Порядок проведення тренінгу
1.	Вступна частина	Ознайомлення студентів з темою тренінгового заняття «Використання С++ для роботи з базами даних» і видача завдання
2.	Практична частина	<ol style="list-style-type: none"> 1. Здійснити аналіз керування базою даних за допомогою С++. 2. На основі аналізу навчитись визначати підхід до побудови бази даних. 3. Розглянути проектування систем управління базами даних з використанням С++. 4. Представити результати у вигляді короткого звіту, що повинен містити основні теоретичні відомості, копії вихідного коду програми та результати його виконання. Описати хід виконання кожного завдання.
3.	Підведення підсумків	Обговорення результатів виконаних завдань

9. Засоби оцінювання та методи демонстрування результатів навчання

У процесі вивчення дисципліни «Сучасні парадигми програмування» використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- поточне тестування та опитування;
- оцінювання результатів лабораторних робіт;
- ректорська контрольна робота;
- оцінювання виконання завдань тренінгу;
- оцінювання результатів КППЗ.

10. Критерії, форми поточного та підсумкового контролю

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Сучасні парадигми програмування» визначається як середньозважена величина, залежно від питомої ваги кожної складової залікового кредиту:

Заліковий модуль 1	Заліковий модуль 2	Заліковий модуль 3
30 %	40 %	30 %
<ol style="list-style-type: none"> 1. Виконання та захист лабораторних робіт (5 робіт по 10 балів) – 50 балів 2. Модульна контрольна робота – 50 балів 	<ol style="list-style-type: none"> 1. Виконання та захист лабораторних робіт (4 роботи по 10 балів) – 40 балів 2. Ректорська контрольна робота – 60 балів 	<ol style="list-style-type: none"> 1. Виконання завдань під час тренінгу – 20 балів 2. Написання та захист КППЗ – 80 балів

Шкала оцінювання:

За шкалою ЗУНУ	За національною шкалою	За шкалою ECTS
90-100	Відмінно	A (відмінно)
85-89	Добре	B (дуже добре)
75-84		C (добре)
65-74	Задовільно	D (задовільно)
60-64		E (достатньо)
35-59	Незадовільно	FX (незадовільно, з можливістю повторного складання)
1-34		F (незадовільно, з обов'язковим повторним курсом)

11. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

№	Найменування	Номер теми
1.	Редактор SublimeText	1, 2
2.	Редактор Visual Studio Code	1, 2
3.	Компілятор MinGW	1, 2
4.	Середовище Code::Blocks	3-17
5.	Середовище MS Visual Studio	3-17

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Tony Gaddis. Starting Out with C++ from Control Structures to Objects. 9th edition, Pearson, 2022, 1311 p.

2. Bjarne Stroustrup. Tour of C++, A (C++ In-Depth Series), 3rd Edition. Addison-Wesley Professional, 2022, 320 p.

3. Електронний курс з дисципліни «Сучасні парадигми програмування» для студентів галузі знань 12 «Інформаційні технології» спеціальності 122 «Комп'ютерні науки» на платформі Moodle ЗУНУ /Биковий П.Є., Кіт І.Р. - Тернопіль, 2021. <https://moodle.wunu.edu.ua>.

4. Биковий П.Є. Опорний конспект лекцій з дисципліни «Сучасні парадигми програмування» для студентів спеціальності 122 «Комп'ютерні науки». - Тернопіль: ЗУНУ, 2021. - 80 с.

5. Биковий П.Є., Кіт І.Р. Методичні вказівки до виконання комплексного практичного індивідуального завдання з дисципліни «Сучасні парадигми програмування» для студентів спеціальності «Комп'ютерні науки». - Тернопіль: ЗУНУ, 2021. - 12 с.