

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ЄРЕВАНСЬКИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ**



ЗАТВЕРДЖУЮ
Директор Єреванського навчально-
наукового інституту ЗУНУ

_____ аак ГУДРАТЯН
_____ 2022 р.

ЗАТВЕРДЖУЮ

Перший проректор

Микола ЦИНКАРИК

_____ 2022 р.



РОБОЧА ПРОГРАМА

з дисципліни

«Прикладне програмне забезпечення для комп'ютерних систем та мереж»

ступінь вищої освіти – **бакалавр**

галузь знань - 12 Інформаційні технології

спеціальність - 123 Комп'ютерна інженерія

освітньо-професійна програма – «Комп'ютерна інженерія»

Кафедра фундаментальних дисциплін

Форма навчання	Курс	Семестр	Лекції (год.)	Лабораторні (год.)	ІРС (год.)	Тренінг (год.)	Самост. робота студ.(год.)	Разом (год.)	Екз. (сем.)
Заочна	1	2	8	4	-	-	108	120	2

Робоча програма складена на основі освітньо-професійної програми «Комп'ютерна інженерія» підготовки здобувачів вищої освіти на першому (бакалаврському) рівні за спеціальністю 123 «Комп'ютерна інженерія» галузі знань 12 Інформаційні технології, затвердженої Вченою радою ЗУНУ (протокол № 9 від 15.06.2022р.).

Робочу програму розробила к.т.н., доцент, доцент кафедри фундаментальних дисциплін ЄННІ ЗУНУ Анна ОВАКІМЯН.

Робоча програма затверджена на засіданні кафедри фундаментальних дисциплін, протокол № 1 від 26.08.2022 року.

Завідувач кафедри
фундаментальних дисциплін,
доктор технічних наук, професор



Гамлет АРУТЮНЯН

Розглянуто та схвалено групою забезпечення спеціальності 123 «Комп'ютерна інженерія», протокол № 1 від 30.08.2022 р.

Голова ГЗС,
д.т.н., професор



Гамлет АРУТЮНЯН

Гарант
д.т.н., професор



Гамлет АРУТЮНЯН

**СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
" ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ
КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ"**

**1. Опис дисципліни " Прикладне програмне забезпечення для
комп'ютерних систем та мереж "**

Дисципліна – «Прикладне програмне забезпечення для комп'ютерних систем та мереж»	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни
Кількість кредитів – 4	галузь знань – 12 „Інформаційні технології”	Статус дисципліни – обов'язкова Мова навчання – українська, вірменська
	Спеціальність – 123 „Комп'ютерна інженерія”	Рік підготовки: 1 Семестр: 2
Кількість змістових модулів – 4	Ступінь вищої освіти - бакалавр	Лекції: 8 год., Лабораторні заняття: 4 год.
Загальна кількість годин – 120 год.,		Самостійна робота: 108 год.
		Вид підсумкового контролю–екзамен

2 Мета й завдання дисципліни

" Прикладне програмне забезпечення для комп'ютерних систем та мереж "

2.1 Мета вивчення дисципліни

Програма та тематичний план дисципліни „Прикладне програмне забезпечення для комп'ютерних систем та мереж” орієнтовані на глибоке та ґрунтовне засвоєння студентами систематичних знань та практичних навичок для створення програмних продуктів з ефективним використанням сучасних технологій.

Метою дисципліни є вивчення науково-практичного інструментарію технологій структурного та об'єктно-орієнтованого підходів при проектуванні комп'ютерних програм та його використанні при реалізації програмних додатків.

2.2 Завдання вивчення дисципліни

Завданням вивчення дисципліни є вивчення науково-практичного інструментарію проектування прикладних програмних додатків та їх складових за допомогою сучасних засобів автоматизованого проектування.

В результаті вивчення дисципліни студенти мають вміти визначати алгоритми вирішення поставлених задач, проектувати загальну структуру програмних додатків, реалізовувати поставлені задачі за допомогою основних бібліотек та команд мови C++.

2.3 Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни:

СК2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

СК3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

2.4 Передумови для вивчення дисципліни

Зазначена дисципліна включена до циклу дисциплін професійної підготовки за переліком програми. У структурно-логічній схемі навчання зазначена дисципліна розміщена на I-му курсі. Вивчення курсу „Прикладне програмне забезпечення для комп'ютерних систем та мереж” передбачає наявність систематичних та ґрунтовних знань із суміжних курсів «Основи програмування», а також цілеспрямованої роботи на лекційних та лабораторних заняттях, самостійної роботи студентів.

2.5 Результати навчання

В результаті вивчення дисципліни студенти повинні:

ПРН1. Знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж.

ПРН3. Знати новітні технології в галузі комп'ютерної інженерії.

ПРН8. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.

ПРН10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.

ПРН14. Вміти поєднувати теорію і практику, а також приймати рішення та виробляти стратегію діяльності для вирішення завдань спеціальності з урахуванням загальнолюдських цінностей, суспільних, державних та виробничих інтересів.

ПРН16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

ПРН20. Усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

ПРН21. Якісно виконувати роботу та досягати поставленої мети з дотриманням вимог професійної етики.

3. Зміст дисципліни

«Прикладне програмне забезпечення для комп'ютерних систем та мереж»

Змістовий модуль 1. Об'єктно-орієнтоване програмування.

Тема 1. Об'єктно-орієнтоване програмування.

1. Об'єктно-орієнтовані технології. 2. Переваги та недоліки ООП. 3. Парадигми ООП: інкапсуляція, поліморфізм, наслідування. 4. Ієрархія.

Тема 2. Поняття класу.

1. Поняття класу. 2. Поняття екземпляру класу. 3. Конструктор та Деструктор класу. Конструктор копіювання. 4. Поля та методи класів.

Тема 3. Поняття інкапсуляції.

1. Поняття інкапсуляції. Переваги та недоліки використання інкапсуляції. 2. Модифікатори доступу. 3. Використання інкапсуляції при проектуванні класів. 4. Сеттери та геттери при реалізації класу.

Тема 4. Принцип наслідування в об'єктно-орієнтованому підході.

1. Ієрархія класів. 2. Базовий клас. 3. Абстрактний клас. 4. Модифікатори доступу при наслідуванні. 5. Переваги та недоліки використання наслідування.

Тема 5. Поліморфізм.

1. Поняття поліморфізму. 2. Пізніє та раннє зв'язування. 3. Віртуальний метод. 4. Чисто віртуальні методи. 5. Абстрактні класи. 6. Дружні функції. 7. Дружні методи. 8. Таблиця віртуальних методів.

Змістовий модуль 2. Особливості обробки та проектування класів.

Тема 6. Перезавантаження унарних операторів.

1. Принципи перезавантаження операторів. 2. Перезавантаження оператора присвоєння (=). 3. Перезавантаження оператора інкременту (префіксна та постфіксна форми), 4. Перезавантаження оператора декретенту (префіксна та постфіксна форми).

Тема 7. Перезавантаження бінарних операторів.

1. Перезавантаження математичних операторів (+, -, *, /, %). 2. Перезавантаження математичних операторів з присвоєнням (+=, -=, *=, /=, %=). 3. Перезавантаження логічних операторів (>, <, !=, ==, >=, <=). 4. Перезавантаження операторів вводу/виводу (<</>>). 5. Перезавантаження операторів індексації ([], ()).

Тема 8. Шаблони класів.

1. Поняття шаблону класу. 2. Контейнерні класи. 3. Шаблони класів в стандартній бібліотеці класів. 4. Шаблони класів та заголовочні файли.

Змістовий модуль 3. Алгоритми і структури даних.

Тема 9. Standard Template Library принципи, вимоги, типи

1. Поняття колекції (vector, list, map, set, multiset, multimap) 2. STL Ітератори. 3. Поняття ітераторів. 4. Типи ітераторів. 5. Вимоги та реалізація.

Тема 10. Алгоритми бібліотеки STL

1. Методи перебору та обробки колекцій. 2. Методи сортування елементів колекцій. 3. Методи виконання математичних операцій над елементами колекцій. 4. Поняття ітераторів.

Тема 11. Списки.

1. Поняття списку в програмуванні. 2. Принципи організації списків. 3. Переваги та недоліки використання списків. 4. Види списків.

Тема 12. Дерева та стеки.

Поняття дерева. Двійкове дерево. Дерево пошуку. Обхід двійкового дерева. Обчислення арифметичних виразів. Використання пов'язаних структур. Зберігання двійкового дерева в масиві. Стек як структура даних. 5. Сегмент стеку викликів.

Змістовий модуль 4. Стандартні та нестандартні виключення.

Тема 13. Обробка виключень.

1. Поняття виняткової ситуації. 2. Оператор try. 3. Оператор Throw. 4. Оператор catch.

Тема 14. Лямбда вирази.

1. Введення в лямбда-вирази. 2. Тип лямбда-виразів. 3. Загальні/Узагальнені лямбди. 4. Загальні лямбди та статичні змінні. 5. Виведення типу, що повертається, і типи, що повертаються trailing.

Тема 15. Бібліотека роботи з графічними елементами SFML.

1. Поняття технології SFML. 2. Принципи підключення функцій бібліотеки SFML в проект. 3. Створення візуального вікна для відображення графічних елементів. 4. Відображення графічних примітивів за допомогою функцій бібліотеки SFML.

3. Структура залікового кредиту дисципліни «Прикладне програмне забезпечення для комп'ютерних систем та мереж»

(заочна форма навчання)

	Кількість годин		
	Лекції	Лабораторні	Самостійна
		заняття	робота
Змістовий модуль 1. Об'єктно-орієнтоване програмування			
Тема 1. Об'єктно-орієнтоване програмування	2	-	6
Тема 2. Поняття класу	-	-	6
Тема 3. Поняття інкапсуляції	2	-	6
Тема 4. Принцип наслідування в об'єктно-орієнтовано підході	-	2	6
Тема 5. Поліморфізм			6
Змістовий модуль 2. Створення програмних додатків на мові Java.			
Тема 6. Перезавантаження унарних операторів.	-	-	6
Тема 7. Перезавантаження бінарних операторів.	-	-	8
Тема 8. Шаблони класів.	-	-	8
Змістовий модуль 3. Алгоритми і структури даних			
Тема 9. Standard Template Library принципи, вимоги, типи	2	2	8
Тема 10. Алгоритми бібліотеки STL	-	-	8
Тема 11. Списки	-		8
Тема 12. Деревя та стеки			8
Змістовий модуль 4. Стандартні та нестандартні виключення			
Тема 13. Обробка виключень	-	-	8
Тема 14. Лямбда вирази	2	-	8
Тема 15. Бібліотека роботи з графічними елементами SFML.			8
Разом	8	4	108

4. Тематика лабораторних занять

Лабораторна робота №1.

Тема: Парадигми ООП та проектування класів.

Мета: Отримати практичні навички по створенні та роботі з класами на мові C++.

Питання для обговорення:

1. Парадигми ООП.
2. Поняття класу.
3. Модифікатори доступу до полів та методів класу.

Лабораторна робота №2.

Тема: Проектування ієрархії класів на мові програмування C++.

Мета: Отримати практичні навички по проектуванні ієрархії класів на мові програмування C++.

Питання для обговорення:

1. Поняття наслідування
2. Одинарне та множинне наслідування.
3. Модифікатори доступу при наслідуванні.

Лабораторна робота №3.

Тема: Перезавантаження операторів на мові програмування C++.

Мета: Отримати практичні навички по перезавантаженню операторів на мові програмування C++

Питання для обговорення:

1. Поняття перезавантаження операторів
2. Перезавантаження унарних операторів.
3. Перезавантаження бінарних операторів.

Лабораторна робота №4.

Тема: Алгоритми сортування та пошуку

Мета: Вивчення сучасних алгоритмів пошуку та сортування засобами мови C++

Питання для обговорення:

1. Основні операції над масивами
2. Алгоритми сортування.
3. Алгоритми пошуку.

Лабораторна робота №5.

Тема: Робота з бібліотекою STL

Мета: Вивчення базових функцій бібліотеки STL

Питання для обговорення:

1. Основні колекції STL
2. Обробка векторів.
3. Алгоритми STL.

Лабораторна робота №6.

Тема: Графічна бібліотека SFML та C++

Мета: Створення програмних додатків з графічними елементами з використанням бібліотеки SFML

Питання для обговорення:

1. Основні структура даних SFML
2. Робота з геометричними примітивами.
3. Принципи створення анімації.

5. Самостійна робота студентів

(заочна форма навчання)

№ п/п	Тематика
1	Змістовне поняття алгоритму та його основні властивості.
2	Абстракція даних.
3	Оператори управління МП.
4	Оператори вибору: типи, синтаксис і семантика. Техніка програмування розгалужень.
5	Оператори повторення: типи, синтаксис і семантика. Техніка програмування циклів.
6	Передумови, постумови і інваріанти циклів.
7	Методологія процедурного програмування.
8	Форми рекурсії. Лінійна рекурсія. Повторна рекурсія. Каскадна рекурсія.
9	Рекурсивні та ітеративні описи алгоритмів.
10	Одномірні масиви. Багатомірні масиви і загальні типи індексів. Динамічні і гнучкі масиви.
11	Задачі обробки масивів. Лінійний пошук у масиві. Бінарний пошук у масиві. Алгоритми сортування масивів.
12	Прості алгоритми сортування: сортування обмінами (бульбашкове), сортування вибором.
13	Робота з файлами. Текстові та двійкові файли.
14	Прямий доступ. Посилальний тип даних. Операції над вказівниками.
15	Розподіл пам'яті.
16	Динамічні інформаційні структури та їх реалізація.
17	Послідовності як абстрактні типи даних.
18	Обчислювальні структури послідовностей. Списки. Лінійні списки.
19	Структура програми на C++ та етапи її обробки.
20	Поняття компілятора, інтерпретатора.
21	Заголовні файли.
22	Правила «гарного стилю» в програмуванні.
23	Система числення. Подання чисел в різних системах числення. Двійкова, шістнадцяткова система.
24	Константи, змінні, типи даних. Класифікація типів. Перетворення типів.
25	Машинне подання дійсних типів.

6. Засоби оцінювання та методи демонстрування результатів навчання

У навчальному процесі застосовуються: лекції, в тому числі з використанням мультимедіапроектора та інших ТЗН; практичні заняття; самостійна робота студента, робота в Інтернет.

У процесі вивчення дисципліни «Прикладне програмне забезпечення для комп'ютерних систем та мереж» використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- поточні опитування;

- презентації результатів виконання завдань та досліджень;
- оцінювання результатів самостійної роботи;
- екзамен.

7. Критерії, форми поточного та підсумкового контролю

В процесі вивчення дисципліни «Прикладне програмне забезпечення для комп'ютерних систем та мереж» використовуються наступні методи оцінювання навчальної роботи студента:

- поточні опитування;
- презентації результатів виконання завдань та досліджень;
- оцінювання результатів самостійної роботи;
- екзамен.

Оцінювання здійснюється за 100 бальною шкалою.

Критерії оцінювання знань:

90–100 балів – у повному обсязі володіє навчальним матеріалом, вільно самостійно та аргументовано його викладає під час відповідей, глибоко та всебічно розкриває зміст питань;

75–89 балів – достатньо повно володіє навчальним матеріалом, але при викладанні деяких питань не вистачає достатньої глибини та аргументації, допускаються при цьому окремі несуттєві неточності та незначні помилки;

65–74 бали – в цілому володіє навчальним матеріалом та викладає його основний зміст, але без глибокого всебічного аналізу, обґрунтування та аргументації, допускаючи при цьому окремі суттєві неточності та помилки;

60–64 бали – не в повному обсязі володіє навчальним матеріалом, фрагментарно (без

аргументації та обґрунтування) його викладає, недостатньо розкриває зміст теоретичних питань, допускаючи при цьому суттєві неточності;

1–59 балів – не володіє навчальним матеріалом, не розкриває зміст теоретичних питань.

Шкала оцінювання

За шкалою університету	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.	DEV C++	1-15
2.	MS Visual Studio Community Edition	1-15

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Alexander Stepanov and Paul McJones Elements of Programming Semigroup Press, 2019, 218
2. Kenneth Leroy Busbee, Dave Braunschweig. Programming Fundamentals - A Modular Structured Approach, 2nd Edition Houston Community College; eBook (Creative Commons Licensed)
3. Changkun Ou Modern C++ Tutorial: C++11/14/17/20 On the Fly - The Fastest Guide towards Modern C++ GitHub; eBook (Creative Commons Licensed), 92.
4. Johan Sannemo Principles of Algorithmic Problem Solving KTH Royal Institute of Technology, 2018, 351.
5. Divakar Viswanath Scientific Programming and Computer Architecture. The MIT Press, 2017, 434.
6. Frank B. Brokken C++ Annotations: An Extensive Tutorial about the C++ Programming Language. University of Groningen, 2021.
7. Bjarne Stroustrup, Herb Sutter C++ Core Guidelines Standard C++ Foundation and GitHub. 2022.
8. C++ Notes for Professionals. Stack Overflow Contributors. Goalkicker. 2018, 708.
9. Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi How to Design Programs, second edition: An Introduction to Programming and Computing (The MIT Press) second edition 2018, 792.
10. Roldan Pozo, Ronald Kriemann, et al. C++ Programming for Scientists NIST, Ohio Supercomputer Center
11. Dr. Rian Quinn. Hands-On System Programming with C++: Build Performant and Concurrent Unix and Linux Systems with C++17. Packt Publishing 2018, 522.
12. Florian Jatton The Constitution of Algorithms: Ground-Truthing, Programming, Formulating. The MIT Press, 2021, 154.
13. Xinyu Liu. Elementary Algorithms. GitHub.com, 2021, 503.
14. Granville Barnett, Luca Del Tongo, John Bullinaria. Data Structures and Algorithms: Annotated Reference with Examples. University of Birmingham, 2019, 112.
15. Об'єктно-орієнтоване програмування: конспект лекцій / Порєв В.М.; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 271 с.
16. Зеленський О.С., Лисенко В.С. – Кривий Ріг: Державний університет економіки і технологій, 2023.-269 с.