

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ФАХОВИЙ КОЛЕДЖ ПРОМИСЛОВОЇ АВТОМАТИКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ОДЕСЬКОГО НАЦІОНАЛЬНОГО ТЕХНОЛОГІЧНОГО УНІВЕРСИТЕТУ»**

Циклова комісія комп'ютерних наук та інженерії програмного забезпечення

ЗАТВЕРДЖУЮ
директор ФКПАІТ ОНТУ
підписано Ольга ЄПУР
30.08.2023 року

**ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ОСНОВИ ПРОГРАМУВАННЯ ТА АЛГОРИТМІЧНІ МОВИ**

обов'язкова

Освітньо-професійна програма	<u>Інженерія програмного забезпечення</u>
Код та найменування спеціальності	<u>121 «Інженерія програмного забезпечення»</u>
Шифр та найменування галузі знань	<u>12 «Інформаційні технології»</u>
Мова навчання	<u>українська</u>

Розроблено та забезпечується: цикловою комісією Комп'ютерних наук та інженерії програмного забезпечення ВСП «Фаховий коледж промислової автоматики та інформаційних технологій Одеського національного технологічного університету»

Розробники:

- Тетяна КОСТИРЕНКО, викладач вищої кваліфікаційної категорії ФКПАІТ ОНТУ
- Наталія СЛУШНА, викладач I кваліфікаційної категорії ФКПАІТ ОНТУ

Розглянуто та схвалено на засіданні циклової комісії Комп'ютерних наук та інженерії програмного забезпечення

Протокол №1 від 28.08.2023 р.

Голова циклової комісії

підписано
(підпис)

Тетяна КОСТИРЕНКО
(Власне ім'я, ПРІЗВИЩЕ)

Гарант освітньо-професійної програми

підписано
(підпис)

Тетяна КОСТИРЕНКО
(Власне ім'я, ПРІЗВИЩЕ)

Розглянуто та схвалено Методичною радою ФКПАІТ ОНТУ

Протокол №1 від 29.08.2023 р.

Голова Методичної ради ФКПАІТ ОНТУ підписано Вікторія ОКСАНІЧЕНКО
(підпис)

1. Пояснювальна записка

Вступ

Програма вивчення навчальної дисципліни «Основи програмування та алгоритмічні мови» складена відповідно до Стандарту фахової передвищої освіти із спеціальності 121 Інженерія програмного забезпечення та освітньо-професійної програми «Інженерія програмного забезпечення» підготовки фахових молодших бакалаврів.

Предмет вивчення

Предмет вивчення дисципліни "Основи програмування та алгоритмічні мови" - це комплекс понять, принципів, інструментів та практичних навичок, пов'язаних з процесом програмування та створенням алгоритмів.

Міждисциплінарні зв'язки

Попередні – операційні системи, послідовні – об'єктно-орієнтоване програмування, бази даних, Інженерна і комп'ютерна графіка

Мета та завдання навчальної дисципліни

Метою викладання дисципліни «*Основи програмування та алгоритмічні мови*» є формування у здобувачів освіти комплексу знань та навичок, необхідних для розуміння та використання базових принципів програмування а алгоритмізації. Основними цілями є: розуміння основ програмування, оволодіння алгоритмічними принципами, використання мов програмування, розробка програмних продуктів, стимулювання критичного мислення, готовність до подальшого навчання. Цілі спрямовані на те, щоб забезпечити здобувачам освіти необхідні знання та навички для успішної роботи у сфері програмування та розробки програмного забезпечення.

Основними завданнями вивчення дисципліни «*Основи програмування та алгоритмічні мови*» є:

1. Вивчення основ програмування:

- Ознайомлення здобувачів освіти із базовими поняттями, такими як змінні, типи даних, операції.
- Розуміння роботи умов та циклів у програмуванні.
- Вивчення базових структур даних та їх використання в програмуванні.

2. Навчання алгоритмізації:

- Розвиток навичок розробки та аналізу алгоритмів.
- Вивчення методів сортування, пошуку та інших базових алгоритмів.
- Практика вирішення завдань, що вимагають використання алгоритмів.

3. Ознайомлення із синтаксисом мови програмування:

- Вивчення синтаксису мови програмування C/C++.
- Практика написання програм, використовуючи основи синтаксису.

4. Робота з програмним забезпеченням:

- Розробка невеликих програмних продуктів.
- Використання різних інструментів для написання, тестування та налагодження коду.

5. Використання засобів розв'язання задач:

- Застосування отриманих знань для розв'язання різноманітних завдань та проблем.
- Самостійне створення програмних рішень для конкретних сценаріїв.

6. Акцент на критичному мисленні та оптимізації:

- Розвиток навичок аналізу та оцінки ефективності програмних рішень.
- Здатність вибирати оптимальні підходи та алгоритми для вирішення конкретних задач.

7. Підготовка до роботи в команді:

- Використання проєктів та завдань, що сприяють розвитку навичок командної роботи.
- Здатність спільно працювати над програмними проєктами.

Ці завдання спрямовані на те, щоб здобувачі освіти отримали практичні навички у програмуванні, зрозуміли основи алгоритмізації та могли успішно застосовувати їх у різних сферах програмування та розробки ПЗ.

Компетентності та результати навчання

У результаті вивчення навчальної дисципліни «*Основи програмування та алгоритмічні мови*» здобувач освіти отримує наступні програмні компетентності та програмні результати навчання, які визначені в Стандарті фахової передвищої освіти із спеціальності 121 Інженерія програмного забезпечення (<https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/12/21/121-inzhener.programn.zabezp.bakalavr-1.pdf>) та освітньо-професійній програмі «Інженерія програмного забезпечення» (<https://dev-kpa.fakel.com.ua/storage/uploads/4t0YvRV8MBZ1IXWV9i190ZBGF5H7rglXYy sWLzuH.pdf>) підготовки фахових молодших бакалаврів.

Загальні компетентності:

ЗК05. Знання та розуміння предметної області та розуміння професійної діяльності

ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел

ЗК07. Здатність застосовувати знання у практичних ситуаціях

Спеціальні (фахові, предметні) компетентності:

СК01. Здатність алгоритмічно та логічно мислити

СК02. Здатність накопичувати знання в галузі інформаційних технологій та усвідомлювати важливість навчання протягом усього життя.

СК03. Здатність застосовувати теоретичні та емпіричні знання для розроблення, тестування, впровадження та супроводу програмного забезпечення.

СК04. Здатність дотримуватися стандартів при розробці програмного забезпечення.

СК05. Здатність брати участь у визначенні та формулюванні вимог до програмного продукту

СК07. Здатність розробляти модулі і компоненти програмного забезпечення за допомогою типових алгоритмів та інструментів.

СК09. Здатність вибирати та використовувати ефективні інструментальні засоби розробки програмного продукту.

СК10. Здатність реалізовувати всі етапи життєвого циклу програмного забезпечення.

Програмні результати навчання:

РН02. Систематизувати та узагальнювати інформацію про підходи, методи та засоби розробки супроводу програмного забезпечення.

РН03. Застосовувати спеціалізовані емпіричні та теоретичні знання у сфері інженерії програмного забезпечення.

РН05. Розробляти та супроводжувати програмне забезпечення.

РН06. Використовувати основні методології та підходи до організації життєвого циклу програмного забезпечення.

РН07. Застосовувати стандарти, специфікації в процесах життєвого циклу програмного забезпечення.

РН08. Аналізувати вимоги до програмного забезпечення.

РН09. Розуміти основні принципи командної роботи при розробці програмного забезпечення.

РН10. Обирати та застосовувати ефективні методи оптимізації алгоритмів.

РН11. Обирати інструментальні засоби, ефективні методи та здійснювати тестування програмних систем.

РН15. Аналізувати та узагальнювати необхідну інформацію з різних джерел та ресурсів для розв'язання професійних задач з урахуванням сучасних досягнень інформаційних технологій.

2. Інформаційний обсяг навчальної дисципліни

2.1 Тематичний план

Таблиця 2.1 – Тематичний план навчальної дисципліни

№	Назва змістових модулів і тем
Змістовий модуль 1. Організація програм. Основні поняття програмування та основи алгоритмізації	
Тема 1	Основні поняття та означення програмування.
Тема 2	Створення програм. Середовище програмування Visual Studio
Тема 3	Системи числення
Тема 4	Алгоритм
Змістовий модуль 2: Елементи мови програмування	
Тема 5	Елементи мови C. Типи даних
Тема 6	Вирази та операції
Тема 7	Структура програми мовою C. Об'єкти програми.
Тема 8	Механіка програмування. Анатомія програми на мові C. Відладка програми.
Тема 9	Функції вводу-виводу.
Тема 10	Стандартні бібліотеки C.
Змістовий модуль 3: Програмування базових алгоритмів	
Тема 11	Оператори розгалуження.
Тема 12	Оператор безумовного переходу goto
Тема 13	Оператори циклів.
Змістовий модуль 4: Масиви. Показчики. Динамічні структури даних. Робота з файлами	
Тема 14	Поняття масиву.
Тема 15	Показчики
Тема 16	Сортування масивів.
Тема 17	Символи і рядки
Тема 18	Структури в мові програмування C.
Тема 19	Файли. Бінарні файли
Змістовий модуль 5: Процедурно-орієнтоване програмування. Функції користувача	
Тема 20	Функції в мові програмування C/C++. Параметри функції. Шаблони функцій
Тема 21	Компіляція програм з двох чи більше файлів.
Тема 22	Списки, стеки, черги
Змістовий модуль 6: Особливості програмування в середовищі Windows	
Тема 23	Графічні елементи вікна Windows
Тема 34	Архітектура Windows-додатка
Разом: 180 годин	

2.2 Зміст дисципліни

Змістовий модуль 1. Організація програм. Основні поняття програмування та основи алгоритмізації

Тема 1. Основні поняття та означення програмування.

Поняття програмного забезпечення, види програмного забезпечення, засоби та технології створення програм, поняття середовища розробки додатків.

Тема 2. Створення програм

Засоби створення програм. Класифікація мов програмування. Технологія створення програм. Етапи розробки та виконання програм на ПК.

Середовище програмування Visual Studio

Технологія створювання інтерфейсних програм. Засоби середовища Visual Studio. Послідовність створювання програмного проекту в Visual Studio

Тема 3. Системи числення

Різновиди систем числення. Одиниці інформації. Основні позиції систем числення. Переведення чисел з однієї СЧ в іншу.

Тема 4. Алгоритм

Поняття алгоритму.

Властивості алгоритму, типові алгоритмічні конструкції.

Способи представлення алгоритмів. Конструкції розгалуження. Циклічні конструкції. Графічні елементи, що використовуються на блок-схемах.

Змістовий модуль 2: Елементи мови програмування

Тема 5. Елементи мови С. Типи даних

Особливості мови С. Алфавіт мови. Поняття ідентифікатора. Ключові слова мови. Використання коментарів в програмі. Константи.

Поняття типу даних, види типів даних в С, діапазон значення для кожного типу.

Визначення цілей, проектування програми, кодування, компіляція, запуск, тестування, супровід.

Тема 6. Вирази та операції

Поняття виразу. Унарні та бінарні операції. Визначення логічних та арифметичних операцій. Пріоритет операцій. Поняття інкременту та декременту, префіксна та постфіксна форми.

Тема 7. Структура програми мовою С. Об'єкти програми.

Як організована програма мовою С. Час життя та область видимості об'єктів. Оголошення та ідентифікація змінних. Глобальні та локальні змінні.

Тема 8. Механіка програмування. Анатомія програми на мові С. Відладка програми.

Види файлів, їх призначення. Бібліотеки в С. Призначення бібліотек. Особливості програмування на мові С в різних ОС. Структура програми. Основна функція програми. Особливості відладки програм мовою С.

Тема 9. Функції вводу-виводу.

Формат написання команд вводу-виводу. Формат вводу. Бібліотека `stdio.h`. Функції `printf()`, `scanf()`

Тема 10. Стандартні бібліотеки C.

Бібліотека математичних функцій: `math.h`.

Функції загального призначення: `stdlib.h`. Функції перевірки та обробки літер: `ctype.h`. Функції вводу-виводу, робота з файлами: `stdio.h`

Змістовий модуль 3: Програмування базових алгоритмів

Тема 11. Оператори розгалуження.

Оператори розгалужень. Повна та неповна форми оператора `if ... else`. Особливості використання оператора `switch`.

Тема 12. Оператор безумовного переходу goto

Основи використання оператора безумовного переходу `goto`.

Тема 13. Оператори циклів.

Поняття циклу. Види циклів. Формат запису різних циклів. Особливості використання.

Опрацювання прикладів використання вкладених циклів.

Приклад циклу, що використовує значення функцій що повертається.

Змістовий модуль 4: Масиви. Показчики. Динамічні структури даних. Робота з файлами

Тема 14. Поняття масиву.

Основні поняття масиву. Оголошення одновимірних масивів. Введення-виведення одновимірних масивів.

Організація двовимірних масивів. Введення-виведення двовимірних масивів.

Алгоритми пошуку мінімуму та максимуму в масиві. Пошук елементів масиву. Робота з елементами масиву. Блок-схеми основних алгоритмів для роботи з масивами.

Опрацювання прикладів на використання основних алгоритмів роботи з масивами.

Тема 15. Показчики

Показчики в мові програмування C/C++. Ініціалізація показчиків. Операції з показчиками. Посилання.

Динамічні одновимірні масиви, динамічні багатовимірні масиви (матриці).

Тема 16. Сортування масивів.

Поняття сортування. Методи сортування масивів. Приклади програм. Алгоритми сортування. Методи прямого включення і прямого вибору, метод «бульбашки» та метод Шелла.

Тема 17. Символи і рядки

Визначення символів та рядків як основних типів даних у програмуванні. Роль символів і рядків у представленні текстової інформації. Опис поняття символа та його представлення в різних кодуваннях (ASCII, Unicode). Використання символів для зберігання та обробки окремих знаків. Визначення рядка та його основні характеристики. Розгляд операцій, що виконуються над рядками: конкатенація, порівняння, копіювання тощо.

Тема 18. Структури в мові програмування С.

Поняття структури. Ініціалізація структур. Доступ до полів структури. Вкладені структури.

Динамічні структури.

Тема 19. Файли. Бінарні файли

Загальні відомості про файл. Зчитування і запис у файл. Послідовне записування і зчитування. Довільне записування і зчитування.

Робота з бінарними файлами у стилі С. Робота з бінарними файловими потоками у стилі С++. Опрацювання бінарних файлів за допомогою дескрипторів.

Змістовий модуль 5: Процедурно-орієнтоване програмування. Функції користувача

Тема 20. Функції в мові програмування С/С++. Параметри функції. Шаблони функцій

Загальні відомості про функції в мові програмування С/С++. Стандартні функції. Оголошення функції. Значення, що повертають функції.

Створення та використання простої функції.

Визначення функції з аргументами: формальні параметри.

Прототипування функцій в стандарті ANSI С.

Передача масивів у якості параметрів, передача імен функції у якості параметрів. Параметри зі значенням за замовчуванням. Пряма і непряма рекурсія. Глибина рекурсії. Рекурсивні функції. Приклади програмування задач з використанням рекурсивних функцій.

Розглянути додаткові відомості про функції, масиви та покажчики

Розглянути створення функцій, що працюють з багатовимірними масивами.

Порядок завдання шаблону. Використання шаблонів.

Тема 21. Компіляція програм з двох чи більше файлів.

Огляд процесу компіляції та його важливість для створення складних програм. Розгляд ситуацій, коли програма розбивається на декілька файлів. Переваги використання розділення програми на модулі чи файли. Розуміння модульності та підтримки коду відокремлено. Використання заголовкових файлів для опису інтерфейсів та визначень функцій. Роль директиви препроцесора `#include` та її вплив на компіляцію. Розрізнення між зовнішніми та внутрішніми змінними.

Тема 22. Списки, стеки, черги

Визначення основних структур даних: списки, стеки, черги. Роль цих структур у програмуванні та їх важливість для оптимізації алгоритмів. Розгляд різних видів списків: однозв'язні, двозв'язні, циклічні. Визначення стеку та його принцип роботи (Last In, First Out - LIFO). Визначення черги та її принцип роботи (First In, First Out - FIFO).

Змістовий модуль 6: Особливості програмування в середовищі Windows

Тема 23. Графічні елементи вікна Windows

Інтерфейс користувача. Розглянути графічні елементи вікна Windows. Створення дружнього інтерфейсу користувача. Робота з буферизованим вводом. Перевірка допустимості вводу

Тема 24. Архітектура Windows-дodatка

Загальний огляд архітектурних принципів та особливостей Windows-дodatків. Роль архітектури у створенні ефективних, масштабованих та стабільних додатків. Визначення ключових компонентів операційної системи Windows, таких як Windows API, реєстр, служби, процеси та потоки. Розподіл функціональності додатка на рівні (presentation layer, business logic layer, data access layer). Визначення інтерфейсів між рівнями та їх взаємодія.

2.3 Перелік практичних робіт з дисципліни

Таблиця 2.2 – Тематичний план практичних робіт

№	Тема практичної роботи
3.	Робота в інтегрованому середовищі.
4.	Програмування лінійного обчислювального процесу.
5.	Розробка додатку з візуальними компонентами.
6.	Програмування розгалуженого алгоритму.
7.	Робота з перемикачами CheckBox, RadioButton
8.	Програмування циклічного алгоритму.
9.	Робота з масивами
10.	Робота з багатовимірними масивами
11.	Сортування масивів
12.	Обробка масивів за допомогою StringGrid.
13.	Робота з рядками символів.
14.	Робота зі структурами
15.	Функції в C
16.	Головна та підпорядкована форми.
17.	Невізуальні компоненти
18.	Мультиплікація. Аудіо-файли, відео
19.	Зображення
20.	Графіки
	Всього: 50 годин

2.4 Кількість годин разом за програмою дисципліни

Розподіл дисципліни у годинах									
Курс	I		II		III		IV		Всього
Семестр	1	2	3	4	5	6	7	8	
Повний обсяг часу			88	92					180
Аудиторні заняття, годин			45	62					107
із них (кількість годин):									
лекції			25	32					57
практичні			20	30					50
Самостійна робота			43	30					83

3. Інструменти, обладнання та програмне забезпечення, використання яких передбачає дисципліна

1. Інтегроване середовище розробки Visual Studio.// Microsoft: [сайт]/ <https://visualstudio.microsoft.com/ru/downloads/>

4. Інформаційні ресурси

Базові (основні):

1. Татарчук Д. Д., Діденко Ю. В. Програмування мовами С та С++: навч. посіб. / Д.Д. Татарчук, Ю.В. Діденко. – К.: , 2012. – 112 с.

2. С++. Основи програмування. Теорія та практика : підручник / [О.Г. Трофименко, Ю.В. Прокоп, І.Г. Швайко, Л.М. Буката та ін.]; за ред. О.Г.Трофименко. – Одеса: Фенікс, 2010. – 544 с.

3. Вінник В.Ю. Алгоритмічні мови та основи програмування: мова С. – Житомир. Видавництво ЖДТУ. 2007. – 328с. [Електронне навчальне видання] <http://programming.in.ua/programming/c-language.html>

4. Бандоріна Л.М., Климкович Т.О., Удачина К.О. Основи алгоритмізації та програмування : навч. посібник. УДУНТ, 2022. 158 с.

Додаткові:

1. Бібліотека MSDN [Електронний ресурс]//Microsoft: [сайт]/ MSDN Library Numerical, 2012. – Режим доступу: <http://msdn.microsoft.com/library/default.aspx>

2. Лабораторний практикум з алгоритмізації та програмування. Мова С++: Електронна книга [Олена Булига. Видавництво ФОП Панов А.М.. 2022. – 112 с.

5. Форма підсумкового контролю

Залік (III семестр, IV семестр), поточний контроль.

6. Засоби діагностики результатів навчання

Перевірка та оцінювання знань здобувачів освіти може проводитись кількома методами:

1. Оцінювання знань здобувача освіти під час практичних занять.
2. Виконання індивідуальних навчально-дослідних завдань.
3. Захист практичних робіт.
4. Тестування.
5. Проведення поточно-модульного контролю.
6. Проведення заліку.

6.1 Питання для самоконтролю

1. Основи програмування:
 - a. Які є базові елементи програмування у мові C/C++?
 - b. Що таке змінна та як вона оголошується?
 - c. Які типи даних підтримуються у мові C/C++?
2. Оператори та Вирази:
 - a. Які основні арифметичні та логічні оператори використовуються у мові C/C++?
 - b. Які є умовні конструкції (if, else) та як вони використовуються?
 - c. Як працює цикл for у мові C/C++?
3. Вказівники та Робота з Пам'яттю:
 - a. Що таке вказівник та як він використовується для роботи з пам'яттю?
 - b. Як використовувати оператори & та * у зв'язці з вказівниками?
 - c. Як уникнути проблем пам'яті, таких як витік чи дублювання?
4. Функції та Модульність:
 - a. Як оголосити та визначити функцію в мові C/C++?
 - b. Як передавати аргументи в функцію та отримувати результат?
 - c. Як розділити програму на модулі та чому це важливо?
5. Рядки та Масиви:
 - a. Як створити та обробляти рядки у мові C/C++?
 - b. Як ініціалізувати та працювати з масивами?
 - c. Як обходити елементи масиву за допомогою циклів?
6. Структури та Об'єктно-Орієнтоване Програмування:
 - a. Як визначити та використовувати структури в C/C++?
 - b. Що таке класи та як вони використовуються в об'єктно-орієнтованому програмуванні?
 - c. Які основні принципи ООП реалізовані в C++?
7. Робота з Файлами:
 - a. Як відкривати, читати та записувати файли у мові C/C++?
 - b. Як обробляти помилки при роботі з файлами?
8. Динамічна Пам'ять:
 - a. Як виділяти та звільняти динамічну пам'ять у мові C/C++?
 - b. Як уникнути витоку пам'яті при використанні динамічних об'єктів?
9. Робота з Вказівниками та Строками:
 - a. Як працювати з вказівниками на функції?
 - b. Як використовувати функції для операцій над рядками?
10. Оптимізація та Ефективність:
 - a. Як використовувати оптимізаційні методи для покращення ефективності коду?
 - b. Як уникнути зайвого використання ресурсів та забезпечити швидкодіючі програми?