

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

**ЗАТВЕРДЖЕНО**  
на засіданні кафедри  
інформаційних систем  
Протокол № 1 від 22.08.2023 р.

**ПОГОДЖЕНО**  
Проректор з навчально-методичної роботи



Каріна НЕМАШКАЛО

**ПРОГРАМУВАННЯ**

робоча програма навчальної дисципліни (РПНД)

Галузь знань 12 "Інформаційні технології"  
Спеціальність 121 "Інженерія програмного забезпечення"  
Освітній рівень перший (бакалаврський)  
Освітня програма "Інженерія програмного забезпечення"

Статус дисципліни обов'язкова  
Мова викладання, навчання та оцінювання українська

Розробник:  
к.т.н., доцент підписано КЕП Володимир ФЕДОРЧЕНКО

Завідувач кафедри інформаційних систем Дмитро БОНДАРЕНКО

Гарант програми Олег ФРОЛОВ

Харків  
2024

## ВСТУП

На даний час умови промислового виробництва та ведення бізнесу вимагають від фахівців з економічного управління всебічного використання новітніх інформаційних технологій та сучасного програмного забезпечення (ПЗ). Широкі можливості комп'ютерної техніки в питаннях збирання, обробки та видачі необхідної інформації здатні значно підвищити якість розрахунків, зробити більш ефективним процес обґрунтування економічних рішень.

Навчальна дисципліна “Програмування” є інструментальною основою для виконання аналітичної частини подальших спецкурсів, а також курсових і дипломних проектів. Вона забезпечує вивчення наступних дисциплін: “Об'єктно-орієнтоване програмування”, “Алгоритми та структури даних”, “Операційні системи”, “Бази даних”, “Розподілені та паралельні обчислення”, “WEB-технології та WEB-дизайн”, “Технології розробки та тестування програмного забезпечення”, “Кросплатформене програмування” та “Програмування для мобільних пристроїв”.

Метою викладання навчальної дисципліни “Програмування” є: засвоєння здобувачами вищої освіти необхідних знань щодо основних понять алгоритмізації і техніки застосування у програмуванні базових алгоритмічних структур (організація програм) і базових типів даних, вивчення основних етапів процесу проектування програмного забезпечення і визначення принципів процедурного програмування щодо розроблення програм мовами C/C++, Python.

Завданнями навчальної дисципліни є:

- отримати вміння аналізувати існуючий стан предметної області, аналізувати та розробляти вимоги до ПЗ, що створюється;
- визначення концепцій і вивчення основних принципів організації програм;
- поглиблення умінь роботи з ПЗ персонального комп'ютера;
- отримати вміння використовувати сучасне інструментальне програмне забезпечення; користуватися раніше складеними програмами і здійснювати супровід програм, вносити зміни в програму, виконувати впровадження програм за допомогою вбудованих інструментальних засобів;
- оволодіння практичними вміннями використовувати сучасні методи та засоби для проектування програмного забезпечення та компонентів архітектури програмного продукту;
- отримати вміння використовувати ефективні алгоритми та структури даних для розробки програмних продуктів;
- оволодіння вміннями застосовувати сучасні інформаційні ресурси та сервіси у процесі розв'язання професійних задач, вміти конструювати програмне забезпечення для рішень в інфокомунікаціях;

- формування навичок застосування типових підходів до розроблення і аналізу найбільш розповсюджених алгоритмів вирішення економіко-математичних задач;

- отримати вміння застосовувати та створювати програми на мовах програмування C/C++, Python з використанням сучасних інструментальних середовищ;

- формування навичок роботи з технічною документацією до ПЗ;

- використовувати свої знання для побудови математичної моделі та оцінювання коректності постановок задач, а також самостійно передбачати наслідки отриманих результатів.

Предметом вивчення навчальної дисципліни є теорія і практика застосування у програмуванні базових алгоритмічних структур і базових структур даних на базі сучасних технологій розроблення програмного забезпечення.

Об'єктом навчальної дисципліни є програмне забезпечення інформаційних систем.

Навчальна дисципліна “Програмування” вивчається здобувачами спеціальності «Інженерія програмного забезпечення» усіх форм навчання на першому курсі протягом першого та другого семестру.

Після успішного закінчення курсу, здобувач повинен бути в змозі продемонструвати здатність вивчати нові технології, методи та прийоми щодо розроблення ПЗ, а також критично аналізувати їх для професійної роботи і подальшого використання.

Результати навчання та компетентності, які формує навчальна дисципліна визначено в табл. 1.

Таблиця 1

### Результати навчання та компетентності, які формує навчальна дисципліна

Результати навчання	Компетентності, якими повинен оволодіти здобувач вищої освіти
PH07	СК7, СК10
PH13	СК3, СК7
PH15	ЗК2, СК10, СК11, СК13

де, PH07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

PH13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

PH15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

ЗК02. Здатність застосовувати знання у практичних ситуаціях.

ЗК03. Здатність спілкуватися державною мовою як усно, так і письмово.

СК7. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.

СК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

СК11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

СК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

## **ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

### **Зміст навчальної дисципліни**

#### **Змістовий модуль 1. Програмування на мовах C/C++. Лексичні основи мов C/C++**

##### **Тема 1. Етапи розроблення та впровадження програм**

Початкові відомості про технологію програмування: процедурне, структурне і об'єктно-орієнтоване програмування. Мови програмування: процедурні, аплікативні, системи правил, об'єктно-орієнтовані. Стандартизація мов та середовища проектування. Транслятори. Редактори. Компонувальники. Відладчики. Керівні структури: оператори, вирази та підпрограми. Огляд сучасних інтегрованих систем програмування. Інтегроване середовище системи програмування Visual Studio C++.NET. Платформа DOT.NET. Етапи розроблення та впровадження програм. Вимоги до програмного коду.

Поняття алгоритму. Властивості алгоритму. Типові алгоритмічні конструкції. Розроблення алгоритму методом покрокового уточнення.

##### **Тема 2. Архітектура комп'ютерів, принципи Джона фон Неймана**

Електронний числовий інтегратор ENIAC. Логічна організація універсального обчислювального пристрою. Відкрита архітектура системних блоків. Принципи Джона фон Неймана.

##### **Тема 3. Позиційні системи числення**

Класифікація систем числення. Особливості позиційних систем числення. Основа, алфавіт та базис позиційних систем числення. Подання чисел в позиційних системах числення. Переведення чисел з однієї системи числення в іншу.

##### **Тема 4. Елементи алгоритмічних мов C/C++: концепція типів даних, імена, значення, покажчики, змінні, константи, операції, вирази**

Стандарти мов програмування C/C++. Структура C++ програми.

Лексичні елементи мови C++: алфавіт, коментарі, ідентифікатори, службові слова, дані, вираз, операнд, змінна, операція. Домовленості про імена.

Поняття типу даних. Класифікація і подання даних. Базові типи даних: логічний, символний, цілий, речовинний. Перетворення типів: неявні перетворення, явні перетворення типу.

Пріоритети операцій. Таблиця пріоритетності й асоціативності операцій.

Операції. Унарні операції: унарний мінус, унарний плюс, порозрядне інвертування, логічне заперечення, інкремент, декремент, операція обчислення розміру (sizeof). Бінарні операції: адитивні, мультиплікативні, операції зсувів, порозрядні, операції відносин, логічні, привласнення, операція “кома”.

Стандартні математичні функції.

Константні величини: цілі, дійсні, перелічувальні, символічні (літерні), рядкові (рядки або літерні рядки). Правила визначення компілятором констант. Визначення констант за допомогою ключового слова const.

## **Тема 5. Структурне програмування: послідовність, розгалуження та цикли**

Загальні відомості про систему вводу-виводу даних.

Вирази, символи пропусків, блоки і комплексні вирази. Операнд, змінна. Оператор привласнення. Оголошення й ініціалізація змінних.

Типи операторів. Найпростіший оператор, оператор-оголошення, оператор-визначення, оператор-вираз.

Керівні оператори: оператори проходження; оператори вибору (єдиний вибір – if, подвійний вибір – if/else, множинний вибір – switch, умовна операція); оператори повторення (оператор while, оператор do-while, оператор for). Вкладені цикли. Керівні оператори в циклах: оператор break, оператор continue, оператор goto. Рекомендації щодо вибору циклів.

## **Тема 6. Передпроцесорна обробка**

Директиви препроцесора. Основи апарату макросів.

Директива препроцесора #include і файли, що включаються. Директива препроцесора #define: оголошення констант і макросів. Умовна компіляція. Використовування ключового слова typedef. Відмінність директиви #define від оператора typedef. Оператор typeid.

## **Тема 7. Процедурно-орієнтоване програмування. Рекурсія**

Загальні відомості про функції. Структура функції.

Значення, параметри і аргументи, що повертаються. Оголошення функції. Прототипи функцій. Визначення функції. Виконання функції.

Локальні та глобальні змінні. Правило видимості змінних. Перетворення типів аргументів функцій. Правила автоматичного (неявного) перетворення типів. Явні перетворення типів. Правила роботи з функціями. Класи пам'яті.

Список параметрів функції. Параметри за умовчанням. Способи передачі параметрів. Способи повернення значення.

Функції, що підставляються.

Створення власних заголовних файлів.

Перевантаження функцій. Рекурсія. Типи рекурсії.

Робота функцій. Розбиття пам'яті. Стек і функції.

Модифікатори функцій.

## **Тема 8. Бібліотеки динамічного компонування (DLL)**

Статичні й динамічні бібліотеки. Загальні відомості про DLL. Структура DLL. Оголошення функції в DLL.

Способи завантаження DLL. Явне (статичне) завантаження DLL. Неявне (динамічне) завантаження DLL.

## **Тема 9. Методології розроблення програм: низхідне та висхідне проектування, модульне програмування**

Методологія розроблення ПЗ: каскадне розроблення, інтерактивне розроблення.

Проектування програм, принцип модульності.

Методи проектування програм: низхідне та висхідне проектування, метод розширення ядра.

## **Змістовий модуль 2. Програмування на мовах C/C++. Основи програмування на мовах C/C++**

### **Тема 10. Масиви**

Оголошення масивів. Ініціалізація масивів. Обробка одновимірних масивів даних економічного характеру.

Алгоритми сортування масивів.

Багатовимірні масиви. Ініціалізація багатовимірного масиву. Типові приклади обробки матриць.

Масиви і графи. Масиви як параметри функцій.

### **Тема 11. Похідні типи даних. Рядки в стилі C**

Рядки як масиви символів. Операції з рядками. Введення-виведення рядків.

Поняття покажчика, посилання. Покажчики і масиви. Адресна арифметика. Посилання. Приклади використання покажчиків і посилань. Покажчики на функції.

Параметри функцій як посилання.

Організація пам'яті в сучасних процесорах і покажчики мови C++. Моделі пам'яті. Статичні і динамічні змінні. Оператори new і delete. Динамічні масиви. Динамічні масиви як параметри функцій.

### **Тема 12. Структури та об'єднання. Динамічні структури даних**

Структури, оголошення та визначення об'єктів типу структура. Відмінності структур мов C та C++. Методи в структурах. Статичні члени структури. Структури з бітовими полями. Вкладені структури. Доступ до елементів структур. Операції з структурами. Структури як параметри функцій. Структури як значення функції, що повертається.

Масиви структур. Покажчики на структури. Передача по посиланню масивів структур.

Об'єднання. Операції з об'єднаннями. Переліки. Змінні структури.

Визначення, класифікація динамічних структур даних, способи оголошення, ініціалізація динамічних структур, методи доступу до даних динамічних структур, розміщення їх у пам'яті, переваги і недоліки використання динамічних структур в програмах.

Зв'язні списки, однозв'язні і двонаправлені списки, циклічні списки. Створення однозв'язного списку. Прохід однозв'язного списку. Включення нового елемента в існуючий список. Виключення елемента із списку.

Створення двонаправленого списку. Прохід двонаправленого списку. Включення нового елемента в існуючий список. Виключення елемента із списку.

### **Тема 13. Введення в систему вводу-виводу C/C++. Файлові структури даних**

Базові положення системи вводу-виводу C/C++. Потоки і буфери. Стандартні об'єкти вводу-виводу. Ієрархія класів вводу-виводу C++.

Ввід даних за допомогою глобального об'єкту `cin`.

Вивід рядків. Введення одного символу. Використовування функції `get()`: без параметрів, з параметрами. Ввід рядків із стандартного пристрою введення. Використовування функції `getline()`.

Вивід даних за допомогою глобального об'єкту `cout`. Очищення буфера виведу – `flush()`. Використовування функцій `put()` і `write()`. Ввід – вивід даних, що форматується. Маніпулятори вводу-виводу. Функції `width()`, `precision()`, `fill()`. Маніпулятори, визначувані користувачем.

Використовування файлів для вводу-виводу даних. Створення файлу. Створення потоку. Відкриття потоку. «Приєднання» файлу до потоку.

Режими відкриття та доступу до файлів. Обміни з файлом з допомогою потоку. «Від'єднання» потоку від файлу. Закриття файлу. Знищення файлу. Визначення стану потоку. Функції управління вводу-виводу.

### **Тема 14. Шаблони. Стандартна бібліотека шаблонів**

Основи апарату шаблонів. Шаблони функцій. Шаблони структур. Перевантаження шаблонів функцій. Шаблони функцій сортування.

Стандартна бібліотека шаблонів (STL). Призначення та склад STL. Контейнери. Алгоритми. Функціональні об'єкти. Ітератори, ітератори потоку. Робота з векторами, списками, стеками, чергами.

### **Тема 15. Стандартний клас string**

Тип даних `string`. Зв'язок класу `string` з STL. Оголошення та визначення об'єктів типу `string`. Методи та властивості класу `string`. Операції з рядками. Функції роботи з рядками. Ввід-вивід рядків. Перетворення між `string` та рядками в стилі C.

### **Тема 16. Обробка виключень. Особливості стандартів C17, C++17, C++20**

Обробка виключень в мові C, функції `abort()`, `exit()`, `atexit()`.

Принципи обробки виключень в мові C++. Генерація виключень. Послідовність обробки виключень. Особливості обробки виключень при динамічному виділенні пам'яті.

Цикл по колекції. Списки ініціалізації, універсальна ініціалізація. Переліки зі строгою типізацією. Додаткові контейнери та зміни в контейнерах. Зовнішні шаблони. Шаблони зі змінною кількістю аргументів. Клас `std::tuple`. Вивід типів. Лямда-функції, альтернативний синтаксис функцій. Регулярні вирази. Багатопоточне програмування, бібліотека `Thread`.

## **Змістовий модуль 3. Програмування на мові Python. Лексичні основи мови Python**

**Тема 17. Синтаксис та семантика мови Python концепція типів даних, імена, значення, покажчики, змінні, константи, операції, вирази**

Скриптові мови програмування. Стандарти, сфери застосування Python. Парадигми програмування, що підтримує Python. Огляд сучасних інтегрованих систем програмування для Python. Інтегроване середовище системи програмування PyCharm.

Концепція типів даних Python. Імена, значення, покажчики, змінні, операції, вирази. Коментарі, Shebang. Пріоритети операцій. Таблиця пріоритетності й асоціативності операцій. Операції зі строками. Вираз if/else. Основні функції.

**Тема 18. Оператори мови Python**

Оператор привласнення. Оголошення й ініціалізація змінних.

Типи операторів.

Умовна конструкція if. Цикли. Слово else. Керівні оператори в циклах: оператор break, оператор continue. Функція range. Вкладені цикли.

**Тема 19. Функції в мові Python**

Іменні функції, інструкція def. Аргументи функції. Функції з довільним числом аргументів, функції з позиційними і іменованими аргументами, обов'язковими і необов'язковими. Вкладені функції. Анонімні функції, інструкція lambda. Оператор return. Функція main. Декоратори. Область видимості змінних, ключове слово global.

**Тема 20. Робота з модулями**

Інструкції import і from. Підключення модуля зі стандартної бібліотеки. Використання псевдонімів. Налаштування простору імен. Змінні `__name__`, `__main__`. Области видимості.

**Тема 21. Правила написання і документування коду на мові Python**

Зовнішній вигляд коду. Прогалини в виразах і інструкціях. Коментарі. Контроль версій. Угоди по іменуванню.

Рядки документації. Атрибут `__doc__`. Однорядкові та багаторядкові рядки документації.

**Змістовий модуль 4. Програмування на мові Python. Основи програмування на мові Python**

**Тема 22. Списки, кортежі і словники**

Індекси і зрізи. Перебір елементів, порівняння списків, методи і функції по роботі зі списками. Генератори і ітератори. Корутіни. Додавання і видалення елементів. Перевірка наявності елемента. Сортування, підрахунок входжень. Копіювання списків, з'єднання списків. Списки списків. Складні кортежі. Функція tuple(). Перетворення зі списку в словник. Отримання і зміна елементів словника. Функція dict(). Метод get. Оператор del, метод pop(). Копіювання і об'єднання словників. Методи items(), keys(), values(). Комплексні словники.

Множина (set, frozenset). Масиви, модуль NumPy. Функціональне програмування.



### **Тема 23. Обробка виключень в мові Python**

Runtime error. Інструкція try...except. Блок finally і інструкція else. Ієрархія виключень. Отримання інформації про виключення, оператор as.

Генерація виключень, оператор raise.

Менеджери контексту, конструкція with ... as.

### **Тема 24. Робота з файлами**

Відкриття та закриття файлу, функція open, метод close(). Режими відкриття файлу. Текстові файли. Файли CSV. Бінарні файли, модуль pickle. Читання з файлу. Запис в файл.

Модуль shelve. Модуль OS і робота з файловою системою.

### **Тема 25. Робота з рядками**

Літерали рядків. Отримання підрядка, функції ord і len, пошук в рядку, перебір рядка. Функції і методи рядків. Форматування рядків, метод format. Байтові рядки.

### **Тема 26. Класи і об'єкти**

Визначення класу, методи та атрибути, властивості, анотації властивостей. Посилання на об'єкт класу, конструктори, метод \_\_init(). Визначення класів в модулях та підключення. Клас object. Строкове представлення об'єкту.

### **Тема 27. Основні вбудовані модулі**

Модулі random, math, locale, decimal. Робота з датами і часом, модуль datetime. Процеси та потоки. Графічний інтерфейс.

Перелік лабораторних занять за навчальною дисципліною наведено в табл. 2.

Таблица 2

### **Перелік лабораторних занять**

Назва теми	Зміст
Тема 1-5.	Знайомство з інтегрованим середовищем розробки програм MS Visual C++.NET. Підготовка і розв'язання на ПК задач лінійного та розгалуженого характеру на мові C/C++.
Тема 5.	Підготовка і розв'язання на ПК задач з використанням циклів на мові C/C++.
Тема 6-9.	Підготовка і розв'язання на ПК задач з використанням функцій і макросів на мові C/C++.
Тема 10.	Підготовка і розв'язання на ПК задач обробки масивів на мові C/C++.
Тема 11.	Підготовка і розв'язання на ПК задач оброблення масивів і рядків в стилі C з використанням покажчиків.
Тема 12-16.	Підготовка і рішення на ПК задач обробки масивів структур на мові C/C++, та динамічних структур даних з використанням STL.
Тема 17, 18.	Знайомство з інтегрованим середовищем розроблення програм PyCharm. Підготовка і розв'язання на ПК задач з використанням розгалужень і циклів на мові Python.
Тема 19.	Підготовка і розв'язання на ПК задач з використанням функцій на мові Python.

Тема 20-22.	Підготовка і розв'язання на ПК задач оброблення масивів на мові Python.
Тема 22, 23.	Підготовка і розв'язання на ПК задач оброблення колекцій з використанням засобів обробки виключень на мові Python.
Тема 24, 25.	Підготовка і розв'язання на ПК задач оброблення рядків з використанням файлів на мові Python.
Тема 26, 27.	Підготовка і розв'язання на ПК задач оброблення структурованих даних на мові Python.

Перелік самостійної роботи за навчальною дисципліною наведено в табл. 3

Таблиця 3

### Перелік самостійної роботи

Назва теми	Зміст
Тема 1 – 27	Вивчення лекційного матеріалу
Тема 1 – 27	Підготовка до лабораторних занять
Тема 1 – 27	Підготовка до екзамену

Кількість годин лекційних та лабораторних занять, а також годин самостійної роботи наведено в робочому плані (технологічній карті) з навчальної дисципліни.

## МЕТОДИ НАВЧАННЯ

У процесі викладання навчальної дисципліни для набуття визначених результатів навчання, активізації освітнього процесу передбачено застосування таких методів навчання, як:

Словесні (лекція (Тема 1, 4-7, 10-15, 17-20, 22-26), проблемна лекція (Тема 8, 9, 15), семінар-дискусія (Тема 2, 16, 27)).

Наочні (демонстрація (Тема 1, 4-7, 10-15, 17-20, 22-26)).

Лабораторна робота (Тема 1 – 27), кейс-метод (Тема 2, 3, 8, 9, 16, 21, 27).

## ФОРМИ ТА МЕТОДИ ОЦІНЮВАННЯ

Університет використовує 100 бальну накопичувальну систему оцінювання результатів навчання здобувачів вищої освіти.

**Поточний контроль** здійснюється під час проведення лекційних та лабораторних занять і має на меті перевірку рівня підготовленості здобувача вищої освіти до виконання конкретної роботи і оцінюється сумою набраних балів:

– для дисциплін з формою семестрового контролю екзамен (іспит): максимальна сума – 60 балів; мінімальна сума, що дозволяє здобувачу вищої освіти скласти екзамен (іспит) – 35 балів.

**Підсумковий контроль** включає семестровий контроль та атестацію здобувача вищої освіти.

**Семестровий контроль** проводиться у формі семестрового екзамену (іспиту). Складання семестрового екзамену (іспиту) здійснюється під час екзаменаційної сесії.

Максимальна сума балів, яку може отримати здобувач вищої освіти під час екзамену (іспиту) – 40 балів. Мінімальна сума, за якою екзамен (іспит) вважається складеним – 25 балів.

**Підсумкова оцінка за навчальною дисципліною** визначається сумуванням балів за поточний та підсумковий контроль.

Під час викладання навчальної дисципліни використовуються наступні контрольні заходи:

Поточний контроль: захист лабораторних робіт (42 бала), письмові контрольні роботи (тестування) (18 балів).

Семестровий контроль: екзамен (40 балів).

Більш детальну інформацію щодо системи оцінювання наведено в робочому плані (технологічній карті) з навчальної дисципліни.

Приклад екзаменаційного білета та критерії оцінювання для навчальної дисципліни.

### **Приклад екзаменаційного білета**

Харківський національний економічний університет імені Семена Кузнеця  
Перший (бакалаврський) рівень вищої освіти  
Спеціальність «Інженерія програмного забезпечення»  
Освітньо-професійна програма «Інженерія програмного забезпечення»  
Семестр I  
Навчальна дисципліна «Програмування»

### **ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 1**

**ЗАГАЛЬНІ ВИМОГИ:** дві задачі у одному проекті; багатомодульний проект; наявність меню; обробка виключних ситуацій; процедурна парадигма.

**Завдання 1** (евристичне, 20 балів).

Дана прямокутна матриця цілих чисел. Визначити кількість від'ємних елементів у тих рядках, які містять хоча б один нульовий елемент та суму додатних діагональних елементів. Вихідні дані записати до текстового файлу.

**Завдання 2** (евристичне, 20 балів).

Ввести з файлу текст. Вивести слово максимальної довжини, тобто яке має максимальну кількість символів. При розв'язанні задачі використати макрос з параметрами. Для обробки тексту використати функції бібліотеки `cstring`.

Затверджено на засіданні кафедри інформаційних систем. Протокол № \_\_\_\_ від «\_\_» \_\_\_\_\_ 20\_\_ р.

Екзаменатор

к.т.н., доц. Федорченко В.М.

Зав. кафедрою

к.т.н., доц. Бондаренко Д.О.

### Критерії оцінювання

Екзаменаційний білет складається із двох евристичних завдань у вигляді задач. Для розв'язання кожної задачі потрібно розробити програму на визначеній викладачем алгоритмічній мові (C(C++) або Python). Підсумкова оцінка за екзамен є сумою оцінок за кожне завдання. Кожне завдання оцінюється від 0 до 20 балів у відповідності з наступною шкалою (табл. 4):

Таблиця 4

#### Критерії оцінювання за діагностичним завданням

Бали	Критерій
20 балів	Завдання виконане в повному об'ємі. Алгоритм розв'язання задачі є оптимальним. Програма працює правильно на всіх тестах. Інтерфейс програми задовольняє встановленим вимогам. Текст програми супроводжується коментарями.
18-19 балів	Завдання виконане в повному об'ємі. Алгоритм розв'язання задачі є оптимальним. Програма працює правильно. Є невеликі зауваження до організації інтерфейсу користувача. У тексті програми відсутні коментарі.
16-17 балів	Завдання в основному виконано. Алгоритм є правильним, але не оптимальним. Програма працює правильно, але є невеликі зауваження до організації інтерфейсу користувача. У тексті програми відсутні коментарі.
14-15 балів	Завдання в основному виконано. Алгоритм є правильним, але не оптимальним. Програма працює правильно, але не реалізована одна з вимог або можливостей, зазначених у завданні.
12-13 балів	Завдання виконане, але не в повному об'ємі. Алгоритм містить незначні помилки. Програма працює, але не реалізовані дві вимоги або можливості, зазначені в завданні.
10-11 балів	Завдання виконане, але не в повному об'ємі. Алгоритм містить помилки. Програма працює, але не реалізовані три вимоги або можливості, зазначені в завданні.
8-9 балів	Алгоритм розроблений з помилками, реалізований у вигляді програми. Програма запускається і як мінімум дозволяє виконати одну дію (можливість).
6-7 балів	Алгоритм розроблений з значними помилками, реалізований у вигляді програми. Програма запускається, але містить грубі помилки, у тому числі й при організації уведення/виведення вхідних/вихідних даних. Під час роботи програма або зависає або аварійно перериває роботу.
4-5 балів	Алгоритм розроблений з значними помилками, реалізований у вигляді програми, але програма не запускається.
3 бала	Алгоритм розроблений з помилками і не реалізований у вигляді програми.
2 бала	Програма не відповідає постановці завдання.
1 бал	Програма не містить програмного коду, розробленого здобувачом. Програма має явні ознаки несамостійності її розробки.
0 балів	Алгоритм розв'язання задачі відсутній. Програма відсутня.

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

### Основна

1. Лосєв М. Ю. Програмування мовою Python: навчальний посібник / М. Ю. Лосєв, В. М. Федорченко – Харків, – Львів: Видавництво ПП «Новий Світ – 2000», 2023. – 178 с. (Укр. мов.)
2. Lospinoso J. C++ Crash Course: A Fast-Paced Introduction. No Starch Press, 2019, 792 p.
3. Bancila M. Modern C++ Programming Cookbook - Third Edition: Master modern C++ including the latest features of C++23 with 140+ practical recipes: 3rd ed. Edition. Packt Publishing, 2024, 816 p.
4. Stroustrup B. Tour of C++, A (C++ In-Depth Series): 3rd Edition. Addison-Wesley Professional, 2022, 320 p.
5. Ivor Horton, Peter Van Weert. Beginning C++23: From Beginner to Pro, 7th Edition. Apress, 2023, 948 p.

#### **Додаткова**

6. Трофименко О.Г. C++. Алгоритмізація та програмування: підручник / О.Г. Трофименко, Ю.В. Прокоп, Н.І. Логінова, О.В. Задерейко. 2-ге вид. перероб. і доповн. Одеса : Фенікс, 2019. – 477 с.
7. Bjarne Stroustrup. Tour of C++, A (C++ In-Depth Series), 3rd Edition. Addison-Wesley Professional, 2022. – 320 p.
8. Васильєв, Олексій Миколайович. Програмування мовою Python / О.М. Васильєв. Тернопіль, Видавництво "Навчальна книга-Богдан", 2021. – 503 с.
9. Висоцька, Вікторія Анатоліївна. PYTHON: Алгоритмізація та програмування: навчальний посібник / В.А. Висоцька, О.В. Оборська; Міністерство освіти і науки України, Національний університет "Львівська політехніка". Львів, Видавництво "Новий Світ-2000", 2021. – 514 с.
10. C. Matthews, Robert. Coding in Python: A Comprehensive Beginners Guide to Learn the Realms of Coding in Python. 2020. – 211 p.
11. Маттес, Ерік. Пришвидшений курс Python: практичний, проєктно-орієнтований вступ до програмування / Ерік Маттес ; з англійської переклала Ольга Белова. Львів, Видавництво Старого Лева, 2021. – 556 с.

#### **Інформаційні ресурси**

12. International Standard ISO/IEC 14882:2014(E) – Programming Language C++ [Електронний ресурс]. – Режим доступу: <https://isocpp.org/std/the-standard>.
13. C++ reference [Електронний ресурс]. – Режим доступу: <https://en.cppreference.com/w/>.
14. Уроки програмування на C++ [Електронний ресурс]. – Режим доступу: <https://acode.com.ua/uroki-po-cpp/>
15. C++ language documentation [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-gb/cpp/cpp/?view=msvc-170/>.
16. Python documentation [Електронний ресурс]. – Режим доступу: <https://www.python.org/doc/>.
17. Learn Python Programming [Електронний ресурс]. – Режим доступу: <https://www.programiz.com/python-programming>.

18.3 Python Підручник [Електронний ресурс]. – Режим доступу:  
<https://w3schoolsua.github.io/python/index.html#gsc.tab=0>.