

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний авіаційний університет

Аерокосмічний факультет
Кафедра комп'ютеризованих електротехнічних систем та технологій



Система менеджменту якості

НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС
навчальної дисципліни

«Обчислювальна техніка та алгоритмічні мови»

Освітній ступінь Бакалавр

Освітньо-професійна програма: «Електротехнічні системи електроспоживання»

Галузь знань:14 «Електрична інженерія»

Спеціальність:141 «Електроенергетика, електротехніка та електромеханіка»

СМЯ НАУ НМК 07.01.07–01–2022

КИЇВ



Система менеджменту якості.
НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС
навчальної дисципліни
«Обчислювальна техніка та алгоритмічні
мови»

Шифр
документа

СМЯ НАУ
НМК 07.01.07-01-2022

Стор. 2 із 153

Навчально-методичний комплекс розробив:
доцент кафедри комп'ютеризованих електротехнічних
систем та технологій, к.т.н., доц. _____ Сергій ЄГОРОВ

Навчально-методичний комплекс обговорено та схвалено на засіданні
кафедри комп'ютеризованих електротехнічних систем та технологій протокол
№ 13 від «22» серпня 2022 р.

Гарант освітньо-професійної програми _____ Сергій ЄГОРОВ

Завідувач кафедри _____ Володимир КВАСНИКОВ

Навчально-методичний комплекс обговорено та схвалено на засіданні
НМРР Аерокосмічного факультету, протокол № 1 від «08» вересня 2022 р.

Голова НМРР _____ Катерина БАЛАЛАСВА

Рівень документа – 3б

Плановий термін між ревізіями – 1 рік

Контрольний примірник



ЗМІСТ НАВЧАЛЬНО-МЕТОДИЧНОГО КОМПЛЕКСУ

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

Освітньо-професійна програма: «Електротехнічні системи електроспоживання»

Галузь знань: 14 «Електрична інженерія»

Спеціальність: 141 «Електроенергетика, електротехніка та електромеханіка»

№	Складова комплексу	Позначення електронного файлу ¹⁾	Наявність	
			друкований вигляд ²⁾	електронний вигляд ³⁾
1	Робоча програма дисципліни та/або силабус	01_ОТАМ_РП	18.11.2022	18.11.2022
2	Курс лекцій або розширений план лекцій	02_ОТАМ_Л		
3	Методичні рекомендації з підготовки студентів до практичних (семінарських) занять	03_XXX_МР_ПЗ		
4	Методичні рекомендації з підготовки студентів до лабораторних занять	04_ОТАМ_МР_ЛЗ		
5	Методичні рекомендації з виконання домашнього завдання (розрахунково-графічної роботи)	05_ОТАМ_МР_ДЗ (РГР)		
6	Методичні рекомендації з виконання контрольних (домашніх) робіт для студентів заочної форми навчання	06_ОТАМ_МР_КРз		
7	Методичні рекомендації до виконання курсової роботи (проекту)	07_XXX_МР_КР (КП)		
8	Завдання з дисципліни для поточного контролю (тести або практичні ситуаційні задачі)	08_XXX_З		
9	Модульні контрольні роботи ⁴⁾	09_ОТАМ_МКР_1 10_ОТАМ_МКР_2		
10	Затверджені екзаменаційні білети	11_ОТАМ_ЕБ		х

1) XXX – скорочена назва дисципліни (перші літери кожного слова з назви дисципліни)

2) Вказується дата затвердження до друку та номер справи у Номенклатурі справ кафедри

3) Вказується дата розміщення у інституційному репозитарії АБО дата та місце розміщення на кафедрі

4)



Система менеджменту якості.
НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС
навчальної дисципліни
«Обчислювальна техніка та алгоритмічні
мови»

Шифр
документа

СМЯ НАУ
НМК 07.01.07–01–2022

Стор. 4 із 153

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Аерокосмічний факультет
Кафедра комп'ютеризованих електротехнічних систем та технологій

КОНСПЕКТ ЛЕКЦІЙ

з дисципліни **«Обчислювальна техніка та алгоритмічні мови»**
за напрямом 141 «Електроенергетика, електротехніка та електромеханіка»

Укладач: к.т.н., доц. С.В. Єгоров

Конспект лекцій розглянутий та схвалений на
засіданні кафедри комп'ютеризованих
електротехнічних систем та технологій
Протокол № _____ від « _____ » 20 ____ р.

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 5 із 153	

МОДУЛЬ № 1

«ВВЕДЕННЯ В РОЗРОБКУ І КОДУВАННЯ АЛГОРИТМІВ»

Лекція №1

Тема лекції: «Установча лекція»


План лекції:

1. Місце дисципліни в системі підготовки фахівця з електроенергетики, електротехніки та електромеханіки.
2. Порядок проходження дисципліни.
3. Розклад годин, лабораторні завдання, контрольні роботи, форми звітності.

Зміст лекції:

Список джерел

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. - Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. - Тернопіль : Навчальна книга – Богдан, 2019. - 504 с.
5. Трофименко О.Г. С++. 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.
6. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 6 із 153	

7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами : лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).

Лекція №2

Тема лекції: «Вступ»

План лекції:

1. Алгоритм як основне поняття програмування.
2. Вступ до дисципліни.
3. Етапи розробки та виконання програм на ЕОМ.
4. Поняття алгоритму, властивості алгоритму.
5. Типові алгоритмічні конструкції.
6. Розробка алгоритму методом покрокового уточнення.

Зміст лекції:


Програмою ми будемо називати втілення алгоритмів, що змушує комп'ютер виконувати необхідні дії.

Алгоритм – це кінцева послідовність дій, яка призводить до потрібного результату.

Операційна система (ОС) – це програма або множина програм, яка керує ресурсами комп'ютера та іншими програмами. Зазвичай ОС складається з модулів – множини системних програм та бібліотек.

Прикладна програма – це програма, яка працює під керуванням операційної системи, використовує її сервіс.

Програмне забезпечення (ПЗ) (англ. *software*) – набір програм, які можуть бути встановлені в комп'ютерній системі користувачем цієї системи. Можна виділити такі типи ПЗ: системне, прикладне, інструментальне.

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 7 із 153	

Мікропрограмне забезпечення (англ. *firmware*) – постійне або напівпостійне програмне забезпечення, для встановлення та зміни якого потрібні спеціальні знання про систему, значні зусилля та майстерність.

Апаратне забезпечення (англ. *hardware*) – фізичні, апаратні компоненти системи. Зазвичай вони є найбільш постійними і незмінними, проте деякі комп'ютерні системи дозволяють відносно легко змінювати апаратні компоненти.

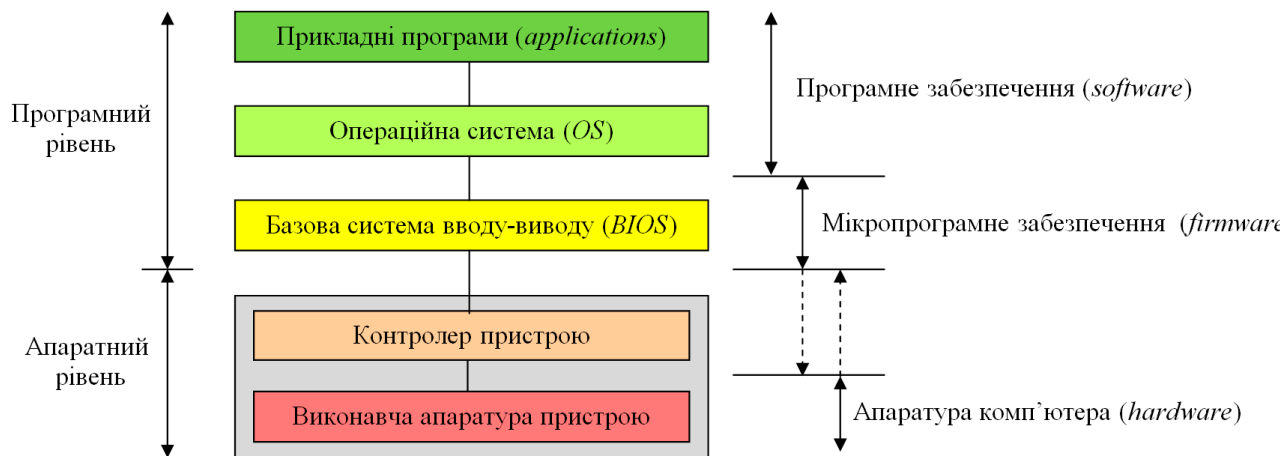



Рис. 1.1. Ієрархія взаємодії програмного забезпечення та апаратури

Основними функціями операційної системи будемо вважати керування ресурсами комп'ютера і надання можливостей користувачу та прикладним програмам використовувати ці ресурси.

В даний час найбільше поширення одержали дискові операційні системи. Для них характерним є те, що більшість необхідних для роботи комп'ютера програм і даних зберігається на диску у вигляді **файлів**. Операційна система являє собою набір програмних модулів, що читаються з файлів у міру необхідності. До таких модулів відносяться бібліотеки

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 8 із 153	


підтримки, драйвери пристроїв і інші. Та частина операційної системи, що повинна постійно знаходитися в оперативній пам'яті (ОП), називається її **ядром**.

Операційна система також дає користувачу комп'ютера свій інтерфейс для налаштування системи, роботи з файлами, для запуску прикладних програм і інших операцій. Цей інтерфейс в деяких ОС реалізований на основі **командної мови**, а в інших ОС - на основі **графічного інтерфейсу користувача GUI** (*Graphical User Interface*). Графічний інтерфейс є сучаснішим та зручнішим.

Прикладні програми зазвичай зберігаються у вигляді файлів. У комп'ютерах типу IBM PC під керуванням ОС Windows прикладні програми зберігаються у вигляді файлів із розширенням EXE. При запуску прикладної програми ОС забезпечує завантаження програми з диска в оперативну пам'ять. Після завантаження всієї програми або необхідної її частини в ОП, ОС передає процесору адресу точки входу. До цього часу програма повинна бути розгорнута в ОП – як інструкції так і дані у вигляді машинних кодів. Під час виконання програми процесор зчитує чергову інструкцію, виконує її і переходить до чергової інструкції.

Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 9 із 153	


3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. Трофименко О.Г. С++. 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.
6. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.
7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).

Лекція №3.

Тема лекції: «Введення в С++»

План лекції:

1. Лексичні елементи мови С++.
2. Операнд, змінна.
3. Оператор привласнення.
4. Оголошення та ініціалізація змінних.
5. Поняття типу даних. Класифікація і представлення даних.
6. Базові типи даних.
7. Перетворення типів.
8. Константні величини. Операції.
9. Пріоритети операцій.
10. Структура програми С++.


	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 10 із 153	

11. Класи пам'яті.

Зміст лекції:

Всі змінні у мові програмування C++ мають бути оголошені до їх використання. Це необхідно для компілятора, якому потрібно мати інформацію про типи даних, які містяться в змінних. Тільки у цьому випадку компілятор зможе належним чином скомпілювати настанови, у яких використовуються змінні. У мові програмування C++ визначено сім основних типів даних: символьний, символьний двобайтовий, цілочисельний, з плинною крапкою, з плинною крапкою подвійної точності, логічний (або булевий) і такий, що "не має значення". Для оголошення змінних цих типів використовують ключові слова `char`, `wchar_t`, `int`, `float`, `double`, `bool` і `void` відповідно. Типи і розміри значень в бітах і діапазони представлення для кожного з цих семи типів наведено в табл. 3.1. Необхідно пам'ятати, що розміри і діапазони, які використовуються Вашим компілятором, можуть відрізнятися від наведених у цьому посібнику. Найбільша відмінність існує між 16- і 32-розрядними середовищами: для представлення цілочисельного значення в 16-розрядному середовищі використовується, як правило, 16 біт, а в 32-розрядному – 32.

Змінні типу `char` використовуються для зберігання 8-розрядних ASCII-символів (наприклад букв Л, Б або В) або будь-яких інших 8-розрядних значень. Щоб задати символ, необхідно помістити його в одинарні лапки. Тип `wchar_t` призначений для зберігання символів, що входять до складу великих символьних наборів. Ймовірно, Вам відомо, що в деяких природних мовах (наприклад китайській) визначено дуже велику кількість символів, для яких 8-розрядного представлення (забезпечуване типом `char`) зовсім недостатньо. Для вирішення проблем такого роду у мові програмування C++ і був доданий тип `wchar_t`, який Вам стане у пригоді, якщо Ви плануєте виходити з своїми програмами на міжнародний ринок.

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 11 із 153	

Змінні типу `int` дають змогу зберігати цілочисельні значення (що не містять дробових компонентів). Змінні цього типу часто використовують для керування циклами і в логічних настановах. До змінних типу `float` і `double` звертаються або для оброблення чисел з дробовою частиною, або у разі потреби виконання операцій над дуже великими чи дуже малими числами. Типи `float` і `double` відрізняються значенням найбільшого (і найменшого) числа, які можна зберігати за допомогою змінних цих типів. Як це показано в табл. 3.1, тип `double` у мові програмування C++ дає змогу зберігати число, що приблизно вдсятеро перевищує значення типу `float`.

Тип `bool` призначений для зберігання булевих (тобто ІСТИНА/ФАЛЬШ) значень. У мові програмування C++ визначені дві булеві константи: `true` і `false`, що є єдиними значеннями, які можуть мати змінні типу `bool`. Як уже розглядалося вище, тип `void` використовують для оголошення функції, яка не повертає значення.

Оголошення змінних

Загальний формат настанови оголошення змінних має такий вигляд:

тип перелік_змінних x;

У цьому записі елемент тип означає допустимий у мові програмування C++ ип даних, а елемент перелік_змінних може складатися з одного або декількох імен (ідентифікаторів), розділених між собою комами. Ось декілька прикладів оголошень змінних:


`int c, d, f;`

`char ch, chr;`

`float f, balance;`

`double d;`

Згідно зі стандартом мови програмування C++, перші 1024 символи будь-якого імені (у тому числі і імені змінної) є значущими. Це означає, що коли двоє імен

	<p style="text-align: center;">Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 12 із 153	


відрізняються між собою хоча б одним символом з перших 1024, то компілятор розглядатиме їх як різні імена.

У мові програмування C++ ім'я будь-якої змінної ніяк не пов'язане з її типом даних, які вона може зберігати.

Змінні можуть бути оголошені всередині функцій, у визначенні параметрів функцій і поза всіма функціями. Залежно від місця оголошення вони називаються локальними змінними, формальними параметрами і глобальними змінними відповідно. Про важливість цих трьох типів змінних ми поговоримо далі, а поки що стисло розглянемо кожен з них окремо.

Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. 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.
6. Васильєв, О. Програмування на C++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 13 із 153	

7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К.: НАУ, 2023. – 48 с. (електронна версія).

Лекція №4

Тема лекції: «Програмування обчислювальних процесів»

План:

1. Загальні відомості про систему вводу-виводу даних.
2. Оператор привласнення.
3. Стандартні функції.
4. Умовний оператор.

Зміст лекції:

Система введення-виведення мови C вимагає містити у програмі заголовний файл `stdio.h` (йому відповідає заголовок `<stdio>`, що відповідає новому стилю). Кожна C-програма повинна використовувати заголовний файл `stdio.h`, оскільки мова C не підтримує C++-стиль внесення заголовків. C++-програма може виконувати дії з використанням будь-якого з цих двох варіантів. Заголовок `<stdio>` поміщає свій зміст у простір імен `std`, а заголовний файл `stdio.h` – у глобальний простір імен, що відповідає C-орієнтації. У цьому додатку як приклади наведені C-програми, тому вони використовують C-стиль внесення заголовного файлу `stdio.h` і не вимагають встановлення простору імен.

Двома найбільш популярними C-функціями введення-виведення даних є `printf()` і `scanf()`. Функція `printf()` записує дані у стандартний пристрій виведення (консоль), а функція `scanf()`, її доповнення, зчитує дані з клавіатури. Оскільки мова C не підтримує перевизначення операторів і не використовує операторів "`<<`" і "`>>`" як оператори введення-виведення, то для консольного введення-виведення використовуються саме функції `printf()` і `scanf()`. Обидві вони можуть обробляти



дані будь-яких вбудованих типів, в т.ч. символи, рядки і числа. Але оскільки ці функції не є об'єктно-орієнтованими, їх не можна використовувати безпосередньо для введення-виведення об'єктів класів, що створюються програмістом.

Присвоєнням значень змінним

Можливо, найважливішою конструкцією в будь-якій мові програмування є присвоєння змінній деякого значення. Змінна – іменована область пам'яті, у якій можуть зберігатися різні значення. При цьому значення змінної у процесі виконання програми можна змінити один або кілька разів. Іншими словами, вміст змінної є змінним, а не фіксованим.

У наведеному нижче коді програми створюється цілочисельна змінна з іменем *x*, якій присвоюється значення 1023, а потім на екрані монітора з'являється таке повідомлення:

Ця програма виводить значення змінної *x*: 1023.

Код програми 2.2. Демонстрація механізму розроблення другої програми

// Програма №2 – використання змінної.

```
#include <iostream> // Потокowe введення-виведення
```

```
using namespace std; // Використання стандартного простору імен
```

```
int main()
```

```
{
```

```
int x; // Тут визначається тип змінної
```

```
x = 1023; // Тут змінній x присвоюється число 1023.
```

```
cout << "Ця програма виводить значення змінної x: ";
```

```
cout << x; // Відображення числа 1023.
```

```
getch(); return 0;
```

```
}
```



Що ж нового у цьому коді програми? По-перше, настанова:

```
int x; // Тут визначається тип змінної.
```

оголошує змінну з іменем `x` цілого типу. У мові програмування C++ всі змінні мають бути оголошені до їх використання. У визначенні змінної, окрім її імені, треба вказати, значення якого типу вона може зберігати. Тим самим

оголошується тип змінної. У цьому випадку змінна `x` може зберігати цілочисельні значення, тобто цілі числа, що знаходяться в діапазоні `-32 768 , +32 767`. У мові програмування C++ для оголошення змінної цілого типу достатньо поставити перед її іменем ключове слово `int`. Таким чином, настанова `int x;` оголошує змінну `x` типу `int`. Нижче дізнаємося, що мова програмування C++ підтримує широкий діапазон вбудованих типів змінних¹.

По-друге, у процесі виконання такої настанови змінній присвоюється конкретне значення:

```
x = 1023; // Тут змінній x присвоюється число 1023.
```


У мові програмування C++ оператор присвоєння позначають одиночним знаком рівності (`=`). Його дія полягає в копіюванні значення, розташованого праворуч від оператора, у змінну, вказану зліва від нього. Після виконання цієї настанови присвоєння змінна `x` міститиме число `1023`.

Результати, що були видані цією програмою, відображаються на екрані за допомогою двох настанов `cout`. Зверніть увагу на використання такої настанови для виведення значення змінної `x`:

```
cout << x; // Відображення числа 1023.
```

У загальному випадку для відображення значення змінної достатньо в настанові `cout` помістити її ім'я праворуч від оператора `"<<"`:

```
cout << "Ця програма виводить значення змінної x: " << x;
```

	<p align="center">Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 16 із 153	

Оскільки у цьому конкретному випадку змінна x містить число 1023, то його і буде відображено на екрані.

Перед останнім оператором `return` в секції `main()` знаходиться функція `getch()`, за допомогою якої Ви зможете побачити результати роботи програми на екрані.

Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. Трофименко О.Г. С++. 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.
6. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.
7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).

Лекція №5



Тема лекції: «Програмування обчислювальних процесів (продовження)»

План:

1. Оператор вибору варіанту.
2. Оператор while.
3. Оператор do-while.

Зміст лекції:


Ітераційна настанова while

Загальна форма організації циклу while має такий вигляд:

while(вираз) настанова;

У цьому записі під елементом настанова розуміють або одиночну настанову, або блок настанов. Роботою циклу керує елемент вираз, який є будь-яким допустимим C++-виразом. Елемент настанова здійснюється доти, доки умовний вираз повертає значення ІСТИНА. Як тільки цей вираз стає помилковим, то керування передається настанові, яка знаходиться за цим циклом. Настанова while — один із способів організації ітераційних циклів у мові програмування C++.

Використання циклу while можна продемонструвати на прикладі такої невеликої програми. Практично всі компілятори підтримують розширений набір символів, який не обмежується символами ASCII. У розширеному наборі часто містяться спеціальні символи і деякі букви з алфавітів іноземних мов. ASCII-символи використовують значення, що не перевищують число 127, а розширений набір символів – значення з діапазону 128-255. У процесі виконання цієї програми виводяться всі символи, значення яких лежать в діапазоні 32-255 (32 – код пропуску). Виконавши цю програму, програміст повинен побачити ряд дуже цікавих символів.


	Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 18 із 153	

Код програми 4.12. Демонстрація механізму виведення усіх друкованих символів, в т.ч. розширений набір символів, якщо такі існують

```
#include <iostream> // Потокowe введення-виведення
#include <conio> // Консольний режим роботи
using namespace std; // Використання стандартного простору імен
int main()
{
    unsigned char ch;
    clrscr(); // Очищення екрану
    ch = 32;
    while(ch) {
        cout << ch; ch++;
    }
    getch(); return 0;
}
```

Розглянемо while-вираз з попередньої програми. Можливо, Вас здивувало, що він складається всього тільки з однієї змінної ch. Але "скринька" тут відкривається просто. Оскільки змінна ch має тут тип unsigned char, то вона може містити значення від 0 до 255. Якщо її значення дорівнює 255, то після інкрементування воно "скидається" в нуль. Отже, факт рівності значення змінної ch нулю слугує зручним способом завершити while-цикл.

Подібно до циклу for, умовний вираз перевіряється під час входу в цикл while, а це означає, що тіло циклу (при помилковому результаті обчислення умовного виразу) може не виконатися жодного разу. Ця властивість циклу усуває необхідність окремого тестування до початку виконання циклу. Наступна

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 19 із 153	

програма виводить рядок, що складається з крапок. Кількість виведених крапок дорівнює значенню, яке вводить користувач. Програма не дає змоги виводити рядки, якщо їх довжина перевищує 80 символів. Перевірка на допустимість кількості крапок, що виводяться, здійснюється усередині умовного виразу циклу, а не зовні.

Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. Трофименко О.Г. С++. 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.
6. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.
7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).

Лекція №6



Тема лекції: «Програмування обчислювальних процесів (продовження)»

План:

1. Оператор вибору варіанту.
2. Циклічні оператори for.
3. Оператори переходу (виходу).

Зміст лекції:

Язык Си++ унаследовал от языка Си соглашение, согласно которому целое значение 0 считается логическим "false" (ложное значение), а ненулевое целое – логическим "true" (истинным значением). Но выражения вроде

Мова С++ успадкувала від мови С конвенцію, згідно з якою ціле значення 0 вважається логічним «false» (хибне значення), а ненульове ціле число — логічним «true» (істинне значення). Але такі вирази, як

Умова1 == 1

або

Умова2 == 0

не дуже зручні при читанні людиною тесту програм. Логічні вирази краще буде записати інтуїтивно зрозумілим способом:

Умова1 == true

і

Умова2 == false



Ось чому до C++ було додано спеціальний логічний тип "bool". Змінні типу bool можуть приймати значення "true" і "false", які при необхідності автоматично перетворюються у виразах в значення 1 і 0.

Тип даних "bool" може використовуватися в програмах так само, як і типи "int", "char" і т.д. (наприклад, для опису змінних або для створення функцій, що повертають значення типу bool).

Вкладені оператори «if», призначені для виконання «множинного розгалуження», вже зустрічалися в Лекції 1. Спрощена версія цього прикладу показана нижче:

...

...

```
if ( total_test_score < 50 )
cout << "Ви не пройшли тест. Вивчіть матеріал як слід.\n";
else if ( total_test_score < 60 )
cout << "Ви пройшли тест із середнім балом.\n";
else if ( total_test_score < 80 )
cout << "Ви добре впоралися з тестом.\n";
else if ( total_test_score < 100 )
cout << "Ви показали відмінний результат.\n";
else
{
cout << "Ви виконали тест нечесно";
cout << "(Оцінка повинна бути менше 100 балів)!\n";
}
...
```

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 22 із 153	

...

Вкладені оператори "if" виглядають занадто громіздкими, тому в C++ реалізований інший метод множинного розгалуження - оператор "switch". Він дозволяє вибрати один з декількох операторів для виконання, в залежності від поточного значення конкретної змінної або виразу.

У наведеному вище прикладі повідомлення для друку вибирається на основі значення змінної "total_test_score". Це може бути довільне ціле число в діапазоні від 0 до 100. Діапазон значень, що перевіряються можна звужити, якщо врахувати, що оцінка тесту перевіряється з точністю до 10 балів. Введемо додаткову цілочисельну змінну «score_out_of_ten» і присвоїмо їй значення:

```
score_out_of_ten = total_test_score/10;
```

Тепер перевірку в програмі можна сформулювати так: (1) якщо змінна "score_out_of_ten" дорівнює 0, 1, 2, 3 або 4, то виведіть повідомлення "Ви не пройшли тест. Вивчіть матеріал як слід.», (2) якщо «score_out_of_ten» дорівнює 5, то виведіть повідомлення «Ви пройшли тест із середнім результатом.» і т. д. У загальному випадку оператор «switch» буде виглядати так:

...

...

```
score_out_of_ten = total_test_score / 10;
switch ( score_out_of_ten )
{
```



case 0 :

case 1 :

case 2 :

case 3 :

case 4 :

```
cout << "Ви не пройшли тест. Вивчіть матеріал як слід.\n";
```

```
break;
```

case 5 :

```
cout << "Ви пройшли тест із середнім балом.\n";
```

```
break;
```

case 6 :

case 7 :

```
cout << "Ви добре впоралися з тестом.\n";
```

```
break;
```

case 8 :

case 9 :

case 10 :

```
cout << "Ви показали відмінний результат.\n";
```

```
break;
```

default :

```
cout << "Ви провели тест нечесно\n";
```

```
cout << "(Оцінка не повинна перевищувати 100 балів)!\n";
```

```
}
```

...

...

Фрагмент програми 3.1.



Оператор "switch" має такий синтаксис:

switch (селектор)

{

case Мітка1 :

<Оператори1>

break;

...

...

...

case міткаN :

<операторыN>

break;

default :

<Оператори>

}

Зробимо кілька важливих зауважень про оператор "switch":

- У середині «switch» виконуються оператори, що містяться між міткою, яка відповідає поточному значенню селектора, і першим оператором «break», що зустрічається після цієї мітки.
- Оператори «break» необов'язкові, але вони покращують читабельність програм. З ними відразу видно, де закінчується кожен варіант множинного розгалуження. Як тільки під час виконання операторів всередині «перемикача» зустрічається «break», то тут же виконується перехід до першого оператора програми,

	<p align="center">Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 25 із 153	

розташованого після оператора «switch». В іншому випадку триває послідовне виконання операторів всередині «switch».

Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. Трофименко О.Г. С++. 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.
6. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.
7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).

Лекція №7

Тема лекції: «Функції»

План:

1. Загальні відомості про функції.
2. Заголовні файли.




3. Оголошення функції.
4. Структура функції.
5. Параметри функції.
6. Значення, що повертаються.
7. Локальні і гло-бальні змінні.
8. Стек і функції. Функції, що підставляються.
9. Перевантаження функцій. Рекурсія.

Зміст лекції:

Функція – у програмуванні – один з видів підпрограми. Особливість, що відрізняє її від іншого виду підпрограм – процедури, полягає в тому, що функція повертає значення, а її виклик може використатися в програмі як вираз. Підпрограма – частина програми, яка реалізує певний алгоритм і дає змогу звернення до неї з різних частин загальної (головної) програми. Підпрограма часто використовується для скорочення розмірів програм у тих завданнях, у процесі розв'язання яких необхідно виконати декілька разів однаковий алгоритм при різних значеннях параметрів. Оператори (команди), які реалізують відповідну підпрограму, записують один раз, а в необхідних місцях розміщують оператори передачі управління на цю підпрограму.

З погляду теорії систем, функція в програмуванні – окрема система (підсистема, підпрограма), на вхід якої надходять керуючі впливи у вигляді значень аргументів. На виході системи одержуємо результат виконання програми, що може бути як скалярною величиною, так і векторним значенням. За ходом виконання функції можуть виконуватися також деякі зміни в керованій системі, причому як оборотні, так і необоротні.

У деяких мовах програмування (наприклад, у мові Pascal) функції існують поряд із процедурами (підпрограмами, що не повертають значення), в інші,

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 27 із 153	

наприклад, у мові C++, є єдиним реалізованим видом підпрограми (тобто всі підпрограми є функціями й можуть повертати значення).

Побічним ефектом функції називається будь-яка зміна функцією стану програмного середовища, крім повернення результату (зміна значень глобальних змінних, виділення й звільнення пам'яті, введення-виведення і так далі). Теоретично найбільш правильним є використання функцій, що не мають побічного ефекту (тобто таких, внаслідок виклику яких повертається обчислене значення і тільки), хоча на практиці доводиться використати функції з побічним ефектом, хоча б для забезпечення введення-виведення й відображення результатів роботи програми. Існує специфічна парадигма програмування – функціональне програмування, у якій будь-яка програма є набір вкладених викликів функцій, що не викликають побічних ефектів. Найбільш відома мова програмування, що реалізує цю парадигму – Лісп1. У ньому будь-яка операція, будь-яка конструкція мови, будь-який вираз, окрім константи, є викликами функцій.

Основні поняття про функції

Будь-яка C++-програма складається з "будівельних блоків", що називаються функціями. Функція – підпрограма, яка містить одну або декілька C++-настанов і здійснює одну або декілька задач. Хороший стиль програмування мовою C++ передбачає, що кожна функція виконує тільки одну задачу, наприклад, окремо введення та виведення даних.

Кожна функція має ім'я, яке використовують для її виклику. Своїм функціям програміст може давати будь-які імена за винятком імені `main()`, зарезервованого для функції, з якої починається виконання програми.

У мові програмування C++ жодна функція не може бути вбудована в іншу. На відміну від таких мов програмування, як Pascal, Modula-2 і деяких інших, які



дають змогу використовувати вкладені функції, у мові програмування C++ всі функції розглядаються як окремі компоненти.

Під час позначення функцій у цьому посібнику використовують домовленість (зазвичай її дотримується в літературі, присвяченій мові програмування C++), згідно з якою ім'я функції завершується парою круглих дужок. Наприклад, якщо функція має ім'я `getVal`, то її згадка в тексті позначиться як `Put()`. Дотримання цієї домовленості дасть змогу легко відрізнити імена змінних від імен функцій.

У вже розглянутих прикладах програм функція `main()` була єдиною. Як було зазначено вище, функція `main()` – перша функція, яка виконується під час запуску програми. Її повинна містити кожна C++-програма. Взагалі, функції, які Вам належить використовувати, бувають двох типів. До першого типу належать функції, написані програмістом (`main()` – приклад функції такого типу). Функції іншого типу знаходяться в стандартній бібліотеці C++-компілятора. Як правило, C++-програми містять як функції, написані програмістом, так і функції, які надає компілятор.

Оскільки функції утворюють фундамент мови програмування C++, зайнемося ними ґрунтовніше. Наведена вище програма містить дві функції: `main()` і `FunC()`. Ще до виконання цієї програми (або читання подальшого опису) уважно вивчіть її текст і спробуйте передбачити, що вона повинна відобразити на екрані.

Код програми 2.6. Демонстрація механізму реалізації структури програми,
що містить дві функції: `main()` і `FunC()`

```
#include <iostream> // Потокowe введення-виведення  
using namespace std; // Використання стандартного простору імен  
void FunC(); // Попереднє оголошення прототипу функції FunC()
```




```
int main()
{
cout << "У функції main().";
FunC(); // Викликаємо функцію FunC().
cout << "Знову у функції main()."
getch(); return 0;
}
void FunC() // Визначення функції FunC()
{
cout << " У функції FunC(). ";
}
```

Програма працює таким чином. Спочатку викликається функція `main()` і здійснюється її перша `cout`-настанова. Потім з функції `main()` викликається функція `FunC()`. Зверніть увагу на те, як цей виклик реалізується у програмі: вказується ім'я функції `FunC`, за яким стоїть пара круглих дужок і крапка з комою. Виклик будь-якої функції є C++-настановою і тому повинен завершуватися крапкою з комою. Потім функція `FunC()` здійснює свою єдину `cout`-настанову і передає керування назад функції `main()`, причому тому рядку коду програми, який розташований безпосередньо за викликом функції. Нарешті, функція `main()` здійснює свою другу `cout`-настанову, яка завершує всю програму. Отже, на екрані ми повинні побачити такі результати:

У функції `main()`. У функції `FunC()`. Знову у функції `main()`.

У цьому коді програми необхідно розглянути таку настанову: `void FunC()`;
// Попереднє оголошення прототипу функції `FunC()`

	<p style="text-align: center;">Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 30 із 153	

Як зазначено в коментарі, це – прототип оголошення функції `FunC()`. Хоча детальніше прототипи буде розглянуто нижче, все ж таки без коротких пояснень тут не обійтися. Прототип функції оголошує функцію до її визначення. Прототип дає змогу компіляторові дізнатися тип значення, що повертається цією функцією, а також кількість і тип параметрів, які вона може мати. Компіляторові потрібно знати цю інформацію до першого виклику функції. Тому прототип розташовується ще до функції `main()`. Єдиною функцією, яка не вимагає прототипу, є `main()`, оскільки вона є вбудованою у мові програмування C++.

Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. 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.
6. Васильєв, О. Програмування на C++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.
7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).



МОДУЛЬ № 2

«Похідні типи даних та введення в система вводу-виводу C++»

Лекція №1

Тема лекції: «Похідні типи даних»

План:

1. Масиви.
2. Алгоритми обробки масивів.
3. Оголошення масивів.
4. Ініціалізація масивів.
5. Обробка одновимірних масивів даних економічного характеру.
6. Алгоритми сортування масивів.

Зміст лекції:

Масив (array) – перелік змінних однакового типу, звернення до яких відбувається із застосуванням імені, загального для всіх його елементів. У мові програмування C++ масиви можуть бути одно-, дво- та багатовимірними, хоча в основному використовуються одновимірні масиви. Масиви є зручними засобами для групування взаємопов'язаних між собою змінних.

Рядок (string) – тип даних, значеннями якого є довільна послідовність символів алфавіту. Кожна змінна такого типу може бути представлена фіксованою кількістю байтів або мати довільну довжину. Деякі мови програмування накладають обмеження на максимальну довжину рядка, але в більшості мов подібні обмеження відсутні.



Основні проблеми в машинному представленні рядкового типу: рядки можуть мати достатньо істотний розмір (до декількох десятків мегабайтів); з часом розмір може змінюватися, тобто виникають труднощі з додаванням і видаленням символів.

Поряд з числовими найчастіше використовуються символні масиви, у яких зберігаються рядки. Як було зазначено вище, у мові програмування C++ не визначено вбудованого типу даних для зберігання рядків. Тому рядки реалізуються як масиви символів. Такий підхід до реалізації рядків дає C++-програмісту більше "важелів" керування порівняно з тими мовами, у яких використовується окремий рядковий тип даних.

5.1. Одновимірні масиви

Одновимірний масив – перелік взаємопов'язаних між собою змінних. Для оголошення одновимірного масиву використовують така форма запису:

тип ім'я_масиву[розмір];

У цьому записі за допомогою елемента запису тип оголошується базовий тип масиву. Базовий тип визначає тип даних кожного елемента, з яких складається масив. Кількість елементів, які зберігатимуться в масиві, визначається елементом розмір. Наприклад, у процесі виконання наведеної нижче настанови оголошується int-масив (що складається з 10 елементів) з іменем Array:

```
int Array[10];
```

Індекс у прямокутних дужках після імені масиву вказує на конкретний елемент масиву.

Доступ до окремого елемента масиву здійснюється за допомогою індексу, який описує позицію елемента усередині масиву. У мові програмування C++ перший елемент масиву має нульовий індекс. Оскільки масив Array містить 10 елементів, то його індекси змінюються від 0 до 9. Щоб отримати доступ до елемента масиву за



індексом, достатньо вказати потрібний номер елемента в квадратних дужках. Так, наприклад, першим елементом масиву `Array` є `Array[0]`, а останнім – `Array[9]`. Розглянемо наведену нижче програму, яка ініціалізує масив `Array` квадратами чисел від 1 до 10.

```
Код програми 5.1. Демонстрація механізму ініціалізації елементів  
одновимірного масиву  
#include <iostream> // Потокowe введення-виведення  
#include <conio> // Консольний режим роботи  
using namespace std; // Використання стандартного простору імен  
int main()  
{  
int Array[10]; // Ця настанова резервує область пам'яті для 10 елементів типу int.  
clrscr(); // Очищення екрану  
// Поміщаємо в масив значення.  
for(int t=0; t<10; ++t) Array[t] = (t+1)*(t+1);  
// Відображається масив.  
for(int t=0; t<10; ++t) cout << Array[t] << " ";  
getch(); return 0;  
}
```

У мові програмування C++ всі масиви займають суміжні елементи пам'яті. Іншими словами, елементи масиву в пам'яті розташовані послідовно один за одним. Клітина з найменшою адресою належить до першого елемента масиву, а з найбільшою – до останнього. Наприклад, після виконання цього фрагмента коду програми

```
int Array[7];
```

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 34 із 153	

```
for(int j=0; j<7; j++) Array[j] = j*j;
```

Для одновимірних масивів загальний розмір масиву в байтах обчислюється так:
всього байтів = розмір типу елемента в байтах * кількість елементів.

Масиви часто використовують під час програмування, оскільки дають змогу легко обробляти велику кількість взаємопов'язаних між собою змінних. Наприклад, у наведеному нижче коді програми створюється масив з десяти елементів, кожному елементу присвоюється випадкове число, а потім на екрані відображаються мінімальне і максимальне його значення.

Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. Трофименко О.Г. С++. 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.
6. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.
7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).



Лекція №2

Тема лекції: «Похідні типи даних (продовження)»

План:

1. Багатовимірні масиви.
2. Ініціалізація багатовимірного масиву.
3. Типові приклади обробки матриць.
4. Функції і масиви.

Зміст лекції:

Дво- та багатовимірні масиви

У програмуванні масив (англ. array) – одна з найпростіших структур даних, сукупність елементів одного типу даних, впорядкованих за індексами, які зазвичай репрезентовані натуральними числами, що визначають положення елемента в масиві. Масив може бути двовимірним (матрицею), та багатовимірним, тобто таким, де індексом є не одне число, а кортеж (сукупність) з декількох чисел, кількість яких збігається з розмірністю масиву. У переважній більшості мов програмування масив є стандартною вбудованою структурою даних.

Організація двовимірних масивів

У мові програмування C++ можна використовувати двовимірні масиви. Двовимірний масив, по суті, є списком одновимірних масивів. Щоб оголосити двовимірний масив цілочисельних значень розміром 10*20 з іменем num, достатньо записати таке:

```
int num[10][20];
```

Зверніть особливу увагу на це оголошення. На відміну від багатьох інших мов програмування, у яких під час оголошення масиву значення розмірностей




відокремлюються комами, у мові програмування C++ кожна розмірність полягає у власну пару квадратних дужок.

Щоб отримати доступ до елемента масиву `num` з координатами 3'5, необхідно використовувати запис `num[3][5]`. У наведеному нижче прикладі в двовимірний масив поміщаються послідовні числа від 1 до 12.

Код програми 5.14. Демонстрація механізму роботи з елементами
двовимірного масиву

```
#include <iostream> // Потокове введення-виведення
using namespace std; // Використання стандартного простору імен
int main()
{
    int t, i, num[3][4];
    for(t=0; t<3; ++t) {
        for(i=0; i<4; ++i) {
            num[t][i] = (t*4)+i+1;
            cout << num[t][i] << " ";
        }
        cout << endl;
    }
    getch(); return 0;
}
```

У наведеному прикладі елемент масиву `num[0][0]` набуде значення 1, елемент `num[0][1]` – значення 2, елемент `num[0][2]` – значення 3 і т.д. Значення елемента `num[2][3]` буде дорівнювати числу 12. Схематично цей масив можна представити так, як це показано на рис. 5.1.

	<p style="text-align: center;">Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 37 із 153	

У двовимірному масиві позиція будь-якого елемента визначається двома індексами. Якщо представити двовимірний масив у вигляді таблиці даних, то один індекс означає рядок, а другий – стовпець. З цього виходить, якщо доступ до елементів масиву надати в порядку, у якому вони реально зберігаються в пам'яті, то правий індекс змінюватиметься швидше, ніж лівий.

Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. Трофименко О.Г. С++. 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.
6. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.
7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).

Лекція №3

Тема лекції: «Похідні типи даних. Показчики.»



План:

1. Поняття покажчика, посилання.
2. Адресна арифметика. Організація пам'яті.
3. Статичні і динамічні змінні.
4. Оператори new і delete.
5. Покажчики і масиви.

Зміст лекції:

Покажчики, поза сумнівом, – один з найважливіших і складних аспектів мови програмування C++. Значною мірою потужність багатьох засобів мови програмування C++ визначається використанням покажчиків. Наприклад, завдяки ним забезпечується підтримка зв'язних списків і динамічного виділення області пам'яті, саме вони дають змогу функціям змінювати значення своїх аргументів. Однак ці питання будуть розглядатися в подальших розділах, а поки що (тобто у цьому розділі) розглянемо основні особливості застосування покажчиків і продемонструємо, як можна уникнути деяких потенційних проблем, пов'язаних з їх використанням.

Під час розгляду основних понять про покажчики нам доведеться використовувати таке поняття як розмір базових C++-типів даних. Для цього запам'ятаємо тільки те, що символи займають в пам'яті один байт, цілочисельні значення – чотири, значення з плинною крапкою типу float – чотири, а значення з плинною крапкою типу double – вісім (ці розміри характерні для типового 32-розрядного середовища).

6.1. Основні поняття про покажчики

Покажчики – змінні, які зберігають адреси іншої змінної. Найчастіше ці адреси позначають місцезнаходження в пам'яті інших змінних. Наприклад, якщо

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯНАУ НМК 07.01.07–01–2022
		Стор. 39 із 153	

змінна x містить адресу змінної y , то про змінну x говорять, що вона "вказує" на y .

Змінні-показчики (або змінні типу показчика) мають бути відповідно оголошені. Формат оголошення змінної-показчика є таким:

тип `*ім'я_змінної`;

У цьому записі елемент тип означає базовий тип показчика, причому він повинен бути допустимим C++-типом. Елемент `ім'я_змінної` є іменем змінної показчика.

Для розуміння сказаного розглянемо такий приклад. Щоб оголосити змінну p показчиком на `int`-значення, використаємо таку настанову:

`int *p;`

Для оголошення показчика на `float`-значення використаємо таку настанову:

`float *p;`

У загальному випадку використання символу "зірочка" (*) перед іменем змінної в настанові оголошення перетворює цю змінну на показчик.

Тип даних, на які посилатиметься показчик, визначається його базовим типом. Розглянемо ще один приклад:

`int *ip; // Показчик на цілочисельне значення`

`double *dp; // Показчик на значення типу double`

Як зазначено в коментарях до цієї програми, змінна `ip` – показчик на `int`-значення, оскільки його базовим типом є тип `int`, а змінна `dp` – показчик на `double`-значення, оскільки його базовим типом є тип `double`. Отже, в попередніх прикладах змінну `ip` можна використовувати для вказівки на `int`-значення, а змінну `dp` – на `double`-значення. Проте запам'ятайте: не існує реального засобу, який би зміг перешкодити показчику посилатися на "казна-що". Ось через це показчики потенційно небезпечні.



Механізм використання покажчиків у поєднанні з операторами присвоєння


Оператори роботи з покажчиками

З покажчиками використовуються два оператори: "*" і "&". Оператор "&" – унарний. Він повертає адресу пам'яті, за якою розташований його операнд 1. Наприклад, у процесі виконання такого фрагмента коду програми

```
ptr = &balance;
```

у змінну ptr поміщається адреса змінної balance. Ця адреса відповідає області у внутрішній пам'яті комп'ютера, яка належить змінній balance. Виконання цієї настанови ніяк не впливає на значення змінної balance. Призначення оператора "&" можна "перекласти" українською мовою як "адреса змінної", перед якою він знаходиться. Отже, наведену вище настанову присвоєння можна виразити так: "змінна ptr отримує адресу змінної balance". Щоб краще зрозуміти суть цього присвоєння, припустимо, що змінна balance розташована у області пам'яті з адресою 100. Отже, після виконання цієї настанови змінна ptr набуде значення 100.

Другий оператор роботи з покажчиками (*) слугує доповненням до першого (&). Це також унарний оператор, але він звертається до значення змінної, розташованої за адресою, заданою його операндом. Іншими словами, він посилається на значення змінної, яка адресується заданим покажчиком. Якщо (продовжуючи роботу з попередньою настановою присвоєння) змінна ptr містить адресу змінної balance, то у процесі виконання настанови value = *ptr; змінній value буде присвоєне значення змінної balance, на яку вказує змінна ptr. Наприклад, якщо змінна balance містить значення 3200, то після виконання останньої настанови змінна value міститиме значення 3200, оскільки це якраз те

	<p style="text-align: center;">Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 41 із 153	

значення, яке зберігається за адресою 100. Призначення оператора "*" можна виразити словосполученням "за адресою".

Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. Трофименко О.Г. С++. 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.
6. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.
7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).

Лекція №4

Тема лекції: «Похідні типи даних. Показчики (продовження)»

План:

1. Показчики і функції.
2. Динамічні масиви як параметри функцій.



3. Зв'язні списки.

Зміст лекції:

Показчики можуть передаватися як параметри функції. У програмі 1.5 показчик перевіряється на коректність в окремій функції, яка створює динамічну цілочисельну змінну:

```
void assign_new_int( IntPtrType& ptr )  
{  
    ptr = new int;  
    if ( ptr == NULL )  
    {  
        cout << "На жаль, недостатньо оперативної пам'яті";  
        exit(1);  
    }  
}
```

Фрагмент програми 1.5.

Правила створення і знищення динамічних змінних типів "int", "char", "double" і т. д. (див. п. 1.3) Застосовуються і до динамічних масивів. Що стосується масивів, динамічний розподіл пам'яті особливо корисний, оскільки іноді є великі масиви.

Динамічний масив з 10 цілих чисел можна створити наступним чином:

```
int* number_ptr;  
number_ptr = new int[10];
```



Змінні масиви в C++ є покажчиками, тому можна отримати доступ до всіх 10-ти елементів динамічного масиву:

```
number_ptr[0] number_ptr[1] ... number_ptr[9]
```

або:

```
number_ptr *(number_ptr + 1) ... *(number_ptr + 9)
```

Для руйнування динамічного масиву використовуйте оператор "delete" з квадратними дужками "[]":

```
delete[] number_ptr;
```

Дужки "[]" відіграють важливу роль. Вони кажуть оператору знищити всі елементи масиву, а не тільки перший.

Робота з динамічними масивами показана в програмі 3.1. У наведеному фрагменті програми користувачеві пропонується список цілих чисел, потім обчислюється їх середнє значення і відображається на екрані.

...

...

```
int no_of_integers, *number_ptr;  
cout << "Введіть кількість цілих чисел у списку: ";  
cin >> no_of_integers;  
number_ptr = new int[no_of_integers];
```




```
if ( number_ptr == NULL )
{
    cout << " Вибачте, недостатньо пам'яті.\n";
    exit(1);
}
cout << "Наберіть " << no_of_integers;
cout << " цілих чисел, розділяючи їх пробілами:\n";
for ( int count = 0; count < no_of_integers; count++ )
    cin >> number_ptr[count];
cout << " Середнє значення: ";
cout << average( number_ptr, no_of_integers );
delete[] number_ptr;
...
...
```

Фрагмент програми 3.1.

Динамічні масиви, як і звичайні масиви, можуть передаватися як параметри функції. Тому в програмі 3.1 функція "average()" з попередньої лекції (лекція 6, п. 2) працюватиме без змін.

Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 45 із 153	

3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. Трофименко О.Г. С++. 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.
6. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.
7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).

Лекція №5


Тема лекції: «Похідні типи даних. Рядки (продовження)»

План:

1. Рядки і масиви символів.
2. Операції з рядками.

Зміст лекції:

Найчастіше одновимірні масиви використовуються для побудови символічних рядків. У мові програмування С++ рядок визначається як символічний масив, який завершується нульовим символом ('\0'). Під час визначення довжини символічного масиву необхідно враховувати ознаку його завершення, тобто задавати його довжину на одиницю більше довжини найбільшого рядка, які передбачають зберігати у цьому масиві.

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 46 із 153	

Рядок – символний масив, який завершується нульовим символом. 5.2.1. Оголошення рядкового літерала Оголошуючи масив `str`, призначеного для зберігання 10-символьного рядка, потрібно використовувати таку настанову:

```
char str[11];
```

Заданий тут розмір (11) дає змогу зарезервувати місце для нульового символу в кінці рядка.

Як було зазначено вище, мова програмування C++ дає змогу визначати рядкові літерали. Пригадаємо, що рядковий літерал – перелік символів, поміщений в подвійні лапки. Ось декілька прикладів:

```
"Привіт"
```

```
"Мені подобається мова програмування C++"
```

```
"#$%@@#$"
```

```
""
```

Рядок, наведений останнім (""), називається нульовим. Він складається тільки з одного нульового символу (ознаки завершення рядка). Нульові рядки використовуються для представлення порожніх рядків.

Програмісту не потрібно вручну добавляти в кінець рядкових констант нульові символи. C++-компілятор робить це автоматично.

Зчитування рядків з клавіатури

Найпростіше зчитувати рядок з клавіатури, створивши масив, який прийме цей рядок за допомогою настанови `cin`. Зчитування рядка, введеного користувачем з клавіатури, відображено у наведеному нижче коді програми.

Код програми 5.4. Демонстрація механізму використання `cin`-настанови для зчитування рядка з клавіатури

```
#include <iostream> // Потокowe введення-виведення
```



```
#include <conio> // Консольний режим роботи
using namespace std; // Використання стандартного простору імен
int main()
{
char str[80];
cout << "Введіть рядок: "; cin >> str; // Зчитуємо рядок з клавіатури.
cout << "Ось Ваш рядок: ";
cout << str;
getch(); return 0;
}
```

Хоча ця програма формально коректна, однак вона не позбавлена недоліків.

Розглянемо такий результат її виконання:

Введіть рядок: Це перевірка

Ось Ваш рядок: Це

Як бачимо, під час виведення рядка, введеного з клавіатури, програма відображає тільки слово "Це", а не весь рядок. Йдеться про те, що оператор "<<" припиняє зчитування рядка, як тільки трапляється символ пропуску, табуляції або нового рядка (називатимемо ці символи пропусковими). Для вирішення цього питання можна використовувати ще одну бібліотечну функцію `gets()`. Загальний формат її виклику є таким:

```
gets(ім'я_масиву);
```

Якщо у програмі необхідно зчитувати рядок з клавіатури, то викличте функцію `gets()`, а як аргумент передайте ім'я масиву, не вказуючи індексу. Після виконання цієї функції заданий масив міститиме текст, введений з клавіатури. Функція `gets()` зчитує символи, які вводяться користувачем, доти, доки він не натисне на клавішу



<Enter>. Для виклику функції `gets()` у програму необхідно включити заголовок `<cstdio>`.

У наведеній нижче версії попередньої програми продемонстровано механізм використання функції `gets()`, яка дає змогу ввести в масив рядок символів, що містить пропуски.

Код програми 5.5. Демонстрація механізму використання функції `gets()`

для зчитування рядка з клавіатури

```
#include <iostream> // Потокowe введення-виведення
#include <cstdio> // Підтримка системи введення-виведення
#include <conio> // Консольний режим роботи
using namespace std; // Використання стандартного простору імен
int main()
{
char str[80];
cout << "Введіть рядок: "; gets(str); // Зчитуємо рядок з клавіатури.
cout << "Ось Ваш рядок: ";
cout << str;
getch(); return 0;
}
```

Цього разу після запуску нової версії програми на виконання і введення з клавіатури тексту "Це простий тест" рядок зчитується повністю, а потім так само повністю і відображається:

Введіть рядок: Це простий тест

Ось Ваш рядок: Це простий тест

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 49 із 153	


У цьому кодї програми зверніть увагу на таку настанову:

```
cout << str;
```

У цьому записі (замість звичного літерала) використовують ім'я рядкового масиву. І хоча причина такого використання настанови `cout` нам стане зрозумілою після вивчення ще декількох розділів цього навчального посібника, поки що стисло зазначимо, що ім'я символічного масиву, який містить рядок, можна використовувати скрізь, де допустимо застосування рядкового літерала. При цьому майте на увазі, що ні оператор "`<<`", ні функція `gets()` не виконують граничної перевірки (на відсутність порушення меж масиву). Тому, якщо користувач введе рядок, довжина якого перевищує розмір масиву, то можливі неприємності, про які згадувалося вище. Із сказаного виходить, що обидва описаних тут варіанти зчитування рядків з клавіатури є потенційно небезпечними.

Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. Трофименко О.Г. С++. 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.

	<p align="center">Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 50 із 153	

6. Васильєв, О. Програмування на C++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.

7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).

Лекція №6

Тема лекції: «Похідні типи даних. Рядки (закінчення)»

План:

1. Тип даних string.
2. Обробка об'єктів типа string.
3. Ввод-вивід рядків.

Зміст лекції:


Мова програмування C++ підтримує багато функцій для оброблення рядків. Найпоширенішими з них є такі: `strcpy()`, `strncpy()`, `strlen()`, `strcmp()`. Для виклику всіх цих функцій у програму необхідно включити заголовок `<cstring>`. Тепер познайомимося з кожною функцією окремо.

Механізм використання функції `strcpy()`

Загальний формат виклику функції `strcpy()` є таким:

`strcpy(to, from);`

Функція `strcpy()` копіює вміст рядка `from` в рядок `to`. Необхідно пам'ятати, масив, який використовують для зберігання рядка `to`, повинен бути достатньо великим, щоб в нього можна було помістити рядок з масиву `from`. Інакше масив `to` переповниться, тобто відбудеться вихід за його межі, що може призвести до руйнування програми.

	Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 51 із 153	

Використання функції `strcpy()` продемонстровано у наведеному нижче коді програми, яка копіює рядок "Привіт" в рядок `str`:

Код програми 5.6. Демонстрація механізму використання функції `strcpy()`

```
#include <iostream> // Потокowe введення-виведення
#include <cstring> // Робота з рядковими типами даних
using namespace std; // Використання стандартного простору імен
int main()
{
char str[80];
strcpy(str, "Привіт");
cout << str << endl << "";
getch(); return 0;
}
```

Механізм використання функції `strcat()`

Звернення до функції `strcat()` має такий формат:

```
strcat(s1, s2);
```

Функція `strcat()` приєднує рядок `s2` до кінця рядка `s1`; при цьому рядок `s2` не змінюється. Обидва рядки повинні завершуватися нульовим символом.

Результат виклику цієї функції, тобто остаточний рядок `s1` також завершуватиметься нульовим символом. Використання функції `strcat()` продемонстровано у наведеному нижче коді програми, яка має вивести на екран рядок

```
"Привіт усім!".
```

Код програми 5.7. Демонстрація механізму використання функції `strcat()`



```
#include <iostream> // Потокowe введення-виведення
#include <cstring> // Робота з рядковими типами даних
#include <conio> // Консольний режим роботи
using namespace std; // Використання стандартного простору імен
int main()
{
char s1[20], s2[10];
strcat(s1, "Привіт");
strcat(s2, " усім!");
strcat(s1, s2);
cout << s1;
getch(); return 0;
}
```

Звернення до функції `strcmp()` має такий формат:

```
strcmp(s1, s2);
```

Функція `strcmp()` порівнює рядок `s2` з рядком `s1` і повертає значення 0, якщо вони однакові. Якщо рядок `s1` лексикографічно (тобто відповідно до алфавітного порядку) більший від рядка `s2`, то повертається позитивне число. Якщо рядок `s1` лексикографічно менший від рядка `s2`, то повертається негативне число. Використання функції `strcmp()` продемонстровано у наведеному нижче кодї програми, яка слугує для перевірки правильності пароля, введеного користувачем (для введення пароля з клавіатури і його верифікації слугує функція `password()`).

Код програми 5.8. Демонстрація механізму використання функції `strcmp()`

```
#include <iostream> // Потокowe введення-виведення
```



```
#include <cstring> // Робота з рядковими типами даних
#include <cstdio> // Підтримка системи введення-виведення
using namespace std; // Використання стандартного простору імен
bool password();
int main()
{
if(password()) cout << "Вхід дозволено" << endl;
else cout << "У доступі відмовлено" << endl;
getch(); return 0;
}
// Функція повертає значення true, якщо пароль прийнятий, і значення false
інакше.
bool password()
{
char str[80];
cout << "Введіть пароль: "; gets(str);
if(strcmp(str, "пароль")) { // Рядки різні.
cout << "Пароль недійсний" << endl;
return false;
}
// Порівнювані рядки збігаються
return true;
}
```

Під час застосування функції `strcmp()` важливо пам'ятати, що вона повертає число Про (тобто значення `false`), якщо порівнювані рядки однакові. Отже, якщо Вам

	<p align="center">Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 54 із 153	

необхідно виконати певні дії за умови збігу рядків, то програміст повинен використовувати оператор НЕ (!). Наприклад, у процесі виконання наведеної нижче програми запит вхідних даних продовжується доти, доки користувач не введе слово "Вихід".


Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. Трофименко О.Г. С++. 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.
6. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.
7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).

Лекція №7

Тема лекції: «Система вводу-виводу С++»

План:

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 55 із 153	

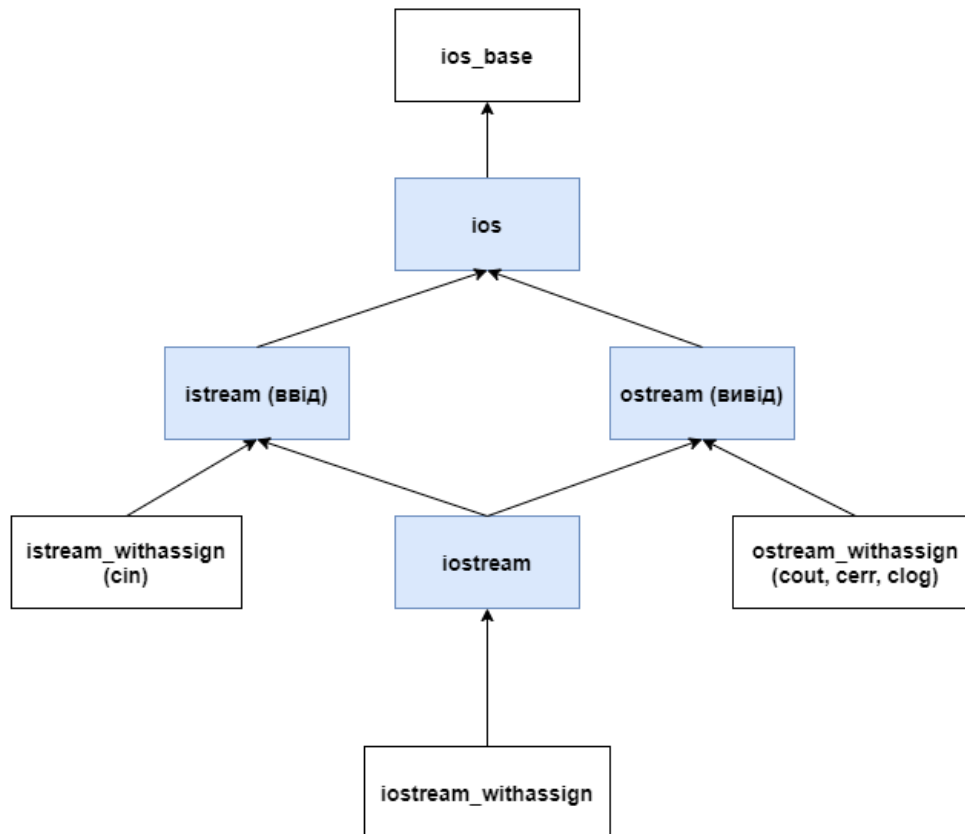
1. Базові положення системи вводу-виводу C++.
2. Ввід даних за допомогою глобального об'єкту cin.
3. Вивід даних за допомогою глобального об'єкту cout.

Список джерел:

Функціонал потоків вводу/виводу не визначений як частина мови C++, а надається Стандартною бібліотекою C++ (і знаходиться в просторі імен std). На попередніх уроках ми підключали заголовковий файл бібліотеки iostream і використовували об'єкти cin і cout для простого вводу та виводу даних. На цьому уроці ми детально розглянемо бібліотеку iostream.

Бібліотека iostream

При підключенні заголовку iostream, ми отримуємо доступ до всієї ієрархії класів бібліотеки iostream, які відповідають за функціонал вводу/виводу даних (включаючи клас, який називається iostream). Ієрархія цих класів виглядає приблизно наступним чином:



Перше, що ви можете помітити в цій ієрархії — множинне спадкування (те, що насправді не рекомендується використовувати). Проте, бібліотека `iostream` була розроблена і ретельно протестована відповідним чином, щоб уникнути типових помилок, які виникають при роботі з множинним спадкуванням, тому ви можете спокійно використовувати цю бібліотеку.

Потоки в C++

Друге, що ви могли б помітити — це часте використання слова «stream» (тобто «потік»). По суті, ввід/вивід в мові C++ реалізований за допомогою потоків. Абстрактно, потік — це послідовність символів, до яких можна отримати доступ. З часом потік може надавати або споживати потенційно необмежені обсяги даних.

Ми матимемо справу з двома типами потоків:



Потік вводу (або «вхідний потік») використовується для зберігання даних, отриманих від джерела даних: клавіатури, файлу, мережі тощо. Наприклад, користувач може натиснути клавішу на клавіатурі в той час, коли програма не очікує вводу. Замість ігнорування натискання клавіші, дані поміщаються у вхідний потік, де потім очікують відповіді від програми.

Потік виводу (або «вихідний потік») використовується для зберігання даних, наданих конкретному споживачеві даних: монітору, файлу, принтеру тощо. При запису даних на пристрій виводу, цей пристрій може бути не готовим прийняти дані негайно. Наприклад, принтер все ще може прогріватися, коли програма вже записує дані в вихідний потік. Таким чином, дані перебуватимуть в потоці виводу до тих пір, поки принтер не почне їх використовувати.

Деякі пристрої, такі як файли і мережі, можуть бути джерелами як вводу, так і виводу даних.


Гарна новина полягає в тому, що програмісту не потрібно знати деталі взаємодії потоків з різними пристроями і джерелами даних, йому потрібно тільки навчитися взаємодіяти з цими потоками для читання і запису даних.

Ввід/вивід в C++

Хоча клас `ios` є дочірнім класу `ios_base`, дуже часто саме цей клас буде найбільш батьківським класом, з яким ви працюватимете/взаємодіятимете напряму. Клас `ios` визначає купу різних речей, які є загальними для потоків вводу/виводу.

Клас `istream` використовується для роботи з вхідними потоками. Оператор вилучення `>>` використовується для вилучення значень з потоку. В цьому є сенс: коли користувач натискає на клавішу клавіатури, код цієї клавіші поміщається у вхідний потік. Потім програма витягує це значення з потоку і використовує його.

Клас `ostream` використовується для роботи з вихідними потоками. Оператор вставки `<<` використовується для вставки значень в потік. В цьому також є сенс: ви

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 58 із 153	

вставляєте свої значення в потік, а потім споживач даних (наприклад, монітор) використовує їх.

Клас `iostream` може обробляти як ввід, так і вивід даних, що дозволяє йому здійснювати двонаправлений ввід/вивід.

Нарешті, залишилися 3 класи, які закінчуються на `_withassign`. Ці потокові класи є дочірніми класами `istream`, `ostream` і `iostream` (відповідно). У більшості випадків ви не працюватимете з ними напряму.

Стандартні потоки в C++

Стандартний потік — це попередньо підключений потік, який надається програмі її оточенням. Мова C++ поставляється з чотирма попередньо визначеними стандартними об'єктами потоків, які ви можете використовувати (перші три ви вже зустрічали):

`cin` — клас `istream_withassign`, пов'язаний зі стандартним вводом (зазвичай це клавіатура);

`cout` — клас `ostream_withassign`, пов'язаний зі стандартним виводом (зазвичай це монітор);

`cerr` — клас `ostream_withassign`, пов'язаний зі стандартною помилкою (зазвичай це монітор), який забезпечує небуферизований вивід;

`clog` — клас `ostream_withassign`, пов'язаний зі стандартною помилкою (зазвичай це монітор), який забезпечує буферизований вивід.

Небуферизований вивід зазвичай обробляється відразу ж, тоді як буферизований вивід зазвичай зберігається і виводиться як блок. Оскільки `clog` використовується рідко, то його зазвичай ігнорують.

Приклад на практиці

Ось приклад використання вводу/виводу даних зі стандартними потоками:

```
#include <iostream>
```



```
#include <cstdlib> // для exit()

int main()
{
    // Спочатку ми використовуємо оператор вставки з об'єктом cout для виводу
    тексту на монітор
    std::cout << "Enter your age: " << std::endl;

    / Потім ми використовуємо оператор вилучення з об'єктом cin для отримання
    користувачького вводу
    int nAge;
    std::cin >> nAge;
    if (nAge <= 0)
    {
        // В цьому випадку ми використовуємо оператор вставки з об'єктом cerr для
        виводу повідомлення про помилку
        std::cerr << "Oops, you entered an invalid age!" << std::endl;
        exit(1);
    }

    // А тут ми використовуємо оператор вставки з об'єктом cout для виводу
    результату
    std::cout << "You entered " << nAge << " years old" << std::endl;
    return 0;
}
```

Бібліотека `iostream` досить складна, тому ми не зможемо охопити її повністю в рамках даних уроків. Проте, ми можемо розглянути її основний функціонал. На цьому уроці ми розберемося з класом `istream`.



Примітка: Весь функціонал об'єктів, які працюють з потоками вводу/виводу, знаходиться в просторі імен `std`. Це означає, що вам потрібно або додавати префікс `std::` до всіх об'єктів і функцій вводу/виводу, або використовувати рядок `using namespace std;`

Оператор вилучення

Як ви вже дізналися на попередньому уроці, ми можемо використовувати оператор вилучення `>>` для зчитування інформації з вхідного потоку. Однією з найбільш поширених проблем при зчитуванні рядків з вхідного потоку є запобігання переповнення. Наприклад:

```
#include <iostream>
int main()
{
    char buf[12];
    std::cin >> buf;
}
```

Що відбудеться, якщо користувач введе 20 символів? Правильно, переповнення. Одним із способів вирішення цієї проблеми є використання маніпуляторів. Маніпулятор — це об'єкт, який застосовується для зміни потоку даних з використанням операторів вилучення (`>>`) або вставки (`<<`).

Ми вже працювали з одним з маніпуляторів — `endl`, який одночасно виводить символ нового рядка і видаляє поточні дані з буфера. Мова C++ надає ще один маніпулятор — `setw()` (з заголовку `iomanip`), який використовується для обмеження кількості символів, які зчитуються з потоку. Для використання `setw()` вам потрібно просто передати в якості параметра максимальну кількість символів для вилучення і вставити виклик цього маніпулятора наступним чином:



```
#include <iostream>
#include <iomanip>

int main()
{
    char buf[12];
    std::cin >> std::setw(12) >> buf;
}
```

Ця програма тепер прочитає тільки перші 11 символів з вхідного потоку (+ один символ для нуль-термінатора). Всі інші символи залишаться в потоці до наступного вилучення.

Вилучення і пробіли

Важливий момент: оператор вилучення працює з «відформатованими» даними, тобто він ігнорує всі пробіли, символи табуляції і символ нового рядка.

Наприклад:

```
#include <iostream>

int main()
{
    char ch;
    while (std::cin >> ch)
        std::cout << ch;
    return 0;
}
```

Якщо користувач введе наступне:

Hello! My name is Anton



То оператор вилучення пропустить всі пробіли і символи нового рядка. Отже, результат виконання програми:

```
Hello!MynameisAnton
```

Часто користувацький ввід все ж потрібен з усіма його пробілами. Для цього клас `istream` надає безліч функцій. Однією з найбільш корисних є функція `get()`, яка вилучає символ з вхідного потоку. Ось вищенаведена програма, але вже з використанням функції `get()`:

```
#include <iostream>

int main()
{
    char ch;
    while (std::cin.get(ch))
        std::cout << ch;
    return 0;
}
```

Тепер, якщо ми введемо наступне:

```
Hello! My name is Anton
```

То отримаємо:

```
Hello! My name is Anton
```

Функція `get()` також має рядкову версію, в якій можна вказати максимальну кількість символів для вилучення. Наприклад:

```
#include <iostream>

int main()
{
    char strBuf[12];
    std::cin.get(strBuf, 12);
}
```



```
std::cout << strBuf << std::endl;  
return 0;  
}
```

Якщо ми введемо наступне:

Hello! My name is Anton

То отримаємо:

Hello! My n

Зверніть увагу, програма зчитує тільки перші 11 символів (+ нуль-термінатор).

Решта символів залишаються у вхідному потоці.

Один важливий нюанс: функція `get()` НЕ зчитує символ нового рядка!

Наприклад:

```
#include <iostream>  
  
int main()  
{  
    char strBuf[12];  
    // Зчитуємо перші 11 символів  
    std::cin.get(strBuf, 12);  
    std::cout << strBuf << std::endl;  
    // Зчитуємо додатково ще 11 символів  
    std::cin.get(strBuf, 12);  
    std::cout << strBuf << std::endl;  
    return 0;  
}
```

Якщо користувач введе наступне:

Hello!

То отримає:



Hello!

І програма відразу ж завершить своє виконання! Чому так? Чому не спрацює другий ввід даних? Справа в тому, що перший `get()` зчитує символи до символу нового рядка, а потім зупиняється. Другий `get()` бачить, що у вхідному потоці все ще є дані і намагається їх отримати. Але перший символ, на який він натикається — символ нового рядка, тому відбувається другий «Стоп!».

Для вирішення даної проблеми клас `istream` надає функцію `getline()`, яка працює точно так само, як і функція `get()`, але при цьому може зчитувати символи нового рядка:

```
#include <iostream>

int main()
{
    char strBuf[12];
    // Зчитуємо 11 символів
    std::cin.getline(strBuf, 12);
    std::cout << strBuf << std::endl;
    // Зчитуємо додатково ще 11 символів
    std::cin.getline(strBuf, 12);
    std::cout << strBuf << std::endl;
    return 0;
}
```

Цей код працює точно так, як очікується, навіть якщо користувач введе рядок з символом нового рядка.

Якщо вам потрібно дізнатися кількість символів, вилучених останнім `getline()`, використовуйте функцію `gcount()`:

```
#include <iostream>
```




```
int main()
{
    char strBuf[100];
    std::cin.getline(strBuf, 100);
    std::cout << strBuf << std::endl;
    std::cout << std::cin.gcount() << " characters were read" << std::endl;
    return 0;
}
```

Результат:

Hello! My name is Anton


24 characters were read

Спеціальна версія функції `getline()` для `std::string`

Є спеціальна версія функції `getline()`, яка знаходиться поза класом `istream` і використовується для зчитування змінних типу `std::string`. Вона не є членом ні `ostream`, ні `istream`, а підключається заголовком `string`. Наприклад:

```
#include <iostream>
#include <string>
int main()
{
    using namespace std;
    string strBuf;
    getline(cin, strBuf);
    cout << strBuf << endl;
    return 0;
}
```

Ще декілька корисних функцій класу `istream`

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯНАУ НМК 07.01.07–01–2022
		Стор. 66 із 153	

Є ще кілька корисних функцій класу `istream`, які ви можете використовувати:

функція `ignore()` — ігнорує перший символ з потоку;


функція `ignore(int nCount)` — ігнорує перші `nCount` (кількість) символів з потоку;

функція `peek()` — зчитує символ з потоку, при цьому не видаляючи його з потоку;

функція `unget()` — повертає останній зчитаний символ назад в потік, щоб його можна було вилучити в наступний раз;

функція `putback(char ch)` — поміщає обраний вами символ назад в потік, щоб його можна було вилучити наступного разу.

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. Трофименко О.Г. С++. 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.
6. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 67 із 153	

7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К.: НАУ, 2023. – 48 с. (електронна версія).

Лекція №8

Тема лекції: «Система вводу-виводу C++ (продовження)»

План:

1. Використовування файлів для вводу-виводу даних.
2. Файли з довільним доступом.

Зміст лекції:

Перш ніж почати вивчати операції з файлами на C++, необхідно ознайомитися з концепцією потокового *вводу-виводу*. Потік нагадує «канал» або «трубу», по якій дані надходять від передавача до приймача. Надзвичайно важливою особливістю обробки поточкових даних є те, що елементи даних можна відправляти або читати з потоку тільки по одному, тобто *послідовно*.

Цей курс охоплює тільки односпрямовані потоки, в яких дані завжди передаються в одному напрямку. З програми дані можуть відправлятися (записуватися) у *потік виведення*, а прийматися (читатися) в програмі з *потіку введення*. Наприклад, відразу після запуску програми до клавіатури підключається стандартний вхідний *потік «cin»*, а стандартний *потік виведення «cout»* - до екрану. потоки вводу-виводу, такі як «cin» і «cout», є прикладами об'єктів класу "stream". Тому вивчення потоків корисно з цієї причини, що дозволяє ознайомитися з елементами синтаксису і деякими об'єктно-орієнтованими поняттями C++.

Список функцій для роботи з файловими потоками зберігається в заголовному файлі "fstream.h". Тому у всіх розглянутих нижче фрагментах



програми передбачається, що на початку програми є відповідна директива «`#include`»:

```
#include<fstream.h>
```

2.1 Створення потоків


У програмі необхідно «створити» потік, перш ніж вперше отримати доступ до вхідного або вихідного потоку. Оператори для створення потоків схожі на описи змінних, і вони зазвичай розміщуються на початку програми або функції поруч з описами змінних. Наприклад, оператори

```
ifstream in_stream;  
ofstream out_stream;
```

створюють потік з іменем "in_stream", який є об'єктом класу (як тип даних) "ifstream" (output-file-stream, файловий потік введення), та потік з іменем "out_stream", який є об'єктом класу "ofstream" (output-file-stream, файловий потік виведення). Аналогію між потоками і звичайними змінними (такими як "int", "char" і т.д.) не варто сприймати занадто буквально. Наприклад, не можна застосувати оператор присвоєння до потоків (наприклад, не можна написати "in_stream1 = in_stream2").

2.2 Підключення та відключення потоків від файлів

Після створення потоку його можна підключити до файлу (відкрити файл) за допомогою функції-члена "open(...)". (У попередніх лекціях вже використовувалося кілька функцій-членів потоків виводу, наприклад, у Лекції 2 використовувалися "precision(...)" та "width(...)".) Функція "open(...)" працює по-

	Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 69 із 153	

різному для потоків `ifstream` та `ofstream` (тобто це поліморфна функція).

Щоб підключити потік `ifstream` з іменем `"in_stream"` до файлу з іменем `"Lecture_4.txt"`, скористайтеся таким викликом:

```
in_stream.open("Lecture_4.txt");
```

Ця інструкція підключить потік `"in_stream"` до початку файлу `"Lecture_4.txt"` (графічно стан програми після виконання оператора показано на рис. 2).

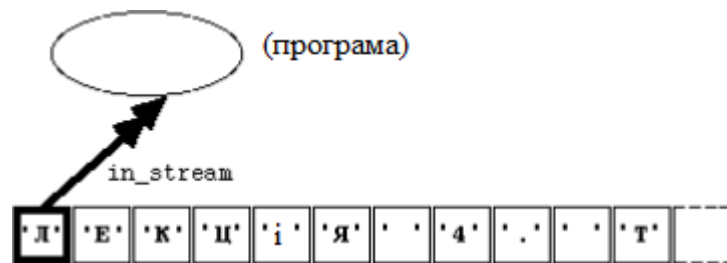



Рис. 2. Стан програми після підключення потоку виводу до файлу.

Щоб підключити вихідний потік `ofstream` з іменем `"out_stream"` до файлу `"Lecture_4.txt"`, потрібно виконати аналогічну інструкцію:

```
out_stream.open("Lecture_4.txt");
```

Ця інструкція підключить потік `"out_stream"` до файлу `"Lecture_4.txt"`, але попередній вміст файлу буде видалено. Файл буде підготовлений до отримання нових даних (рисунок 3).

	Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 70 із 153	

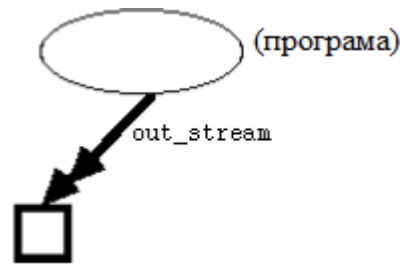


Рис. 3. Стан роботи програми після підключення вихідного потоку до файлу.

Щоб відключити потік "in_stream" від файлу, до якого він підключений (закрити файл), потрібно викликати функцію члена "close()":

```
in_stream.close();
```

Після цього стан програми по відношенню до файлу буде таким же, як на рис. 4.

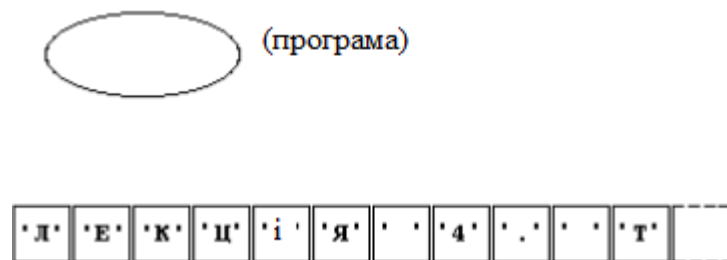



Рис. 4. Стан програми після відключення вхідного потоку від файлу.

Функція-член відключення від файлу вихідного потоку:

```
out_stream.close();
```

виконує аналогічні дії, але, крім того, в кінець файлу додається службовий символ «end-of-file». Таким чином, навіть якщо у вихідний потік не було записано жодних

	Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»	Шифр документа	СМЯНАУ НМК 07.01.07–01–2022
		Стор. 71 із 153	

даних, то після відключення потоку «out_stream» у файлі «Lecture_4.txt» залишиться один службовий символ (рис. 5). В цьому випадку файл «Lecture_4.txt» залишиться на диску, але він буде *порожній*.

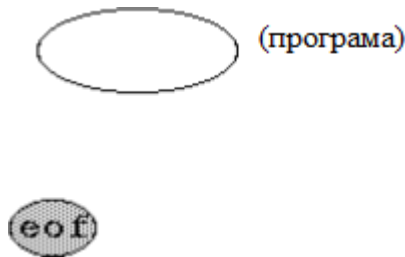


Рис. 5. Стан програми після відключення вихідного потоку від файлу, в який не були записані символи.


3. Перевірка на помилки при виконанні файлових операцій

Операції з файлами, такі як відкриття та закриття файлів, відомі як одне з найбільш ймовірних джерел помилок. Надійні комерційні програми завжди перевіряють, чи була операція з файлом успішною чи ні. У разі помилки викликається спеціальна функція обробника помилок. Найпростіший спосіб перевірити наявність помилок в файлових операціях - викликати функцію-член "fail()". Виклик

```
in_stream.fail();
```

повертає значення true (True), якщо остання операція потоку "in_stream" призвела до помилки (можливо, була спроба відкрити файл, якого не існує). Після помилки потік «in_stream» може бути пошкодженим, тому краще не продовжувати роботу з ним.

У наступному фрагменті програми в разі помилки при відкритті файлу на

	Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 72 із 153	

екран виводиться повідомлення і програма завершується за допомогою бібліотечної функції "exit()" (вона описана в файлі "stdlib.h"):

```
#include <iostream>
#include <fstream.h>
#include <stdlib.h>
using namespace std;

int main()
{
    ifstream in_stream;
    in_stream.open( "Lecture_4.txt" );
    if ( in_stream.fail() )
    {
        cout << "Вибачте, але файл не вдалося відкрити!\n";
        exit(1);
    }
    ...
}
```

4. Символьний ввід/вивід

4.1 Функція введення "get(...)"

Після того, як файл для введення даних відкритий, з нього можна прочитати окремі символи. Для цього використовується функція "get(...)". вона має параметр типу "char&". Якщо програма знаходиться в стані, як на рис. 2, то після виклику:

```
in_stream.get(ch);
```


відбудеться наступне: (а) змінній "ch" буде присвоєно значення "'Л'", і (б) потік "in_stream" буде підготовлений до читання наступного символу (рис. 6).

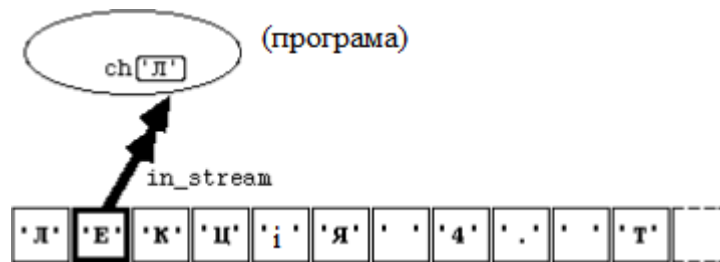


Рис. 6. Стан програми після прочитання першого символу з файлу.

4.2 Функція виводу "put(...)"

За допомогою вихідного потоку класу `ofstream` можна записувати окремі символи у відкритий файл. Для цього клас `ofstream` має функцію-член `"put(...)"`. Записуваний символ передається йому як параметр типу `"char"`. Якщо програма знаходиться в стані, показаному на рис.3, то оператор `out_stream.put('Л');` змінить стан на той, що показаний на рис.7:

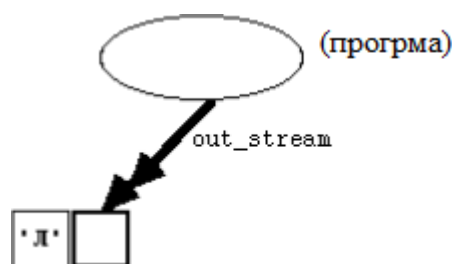


Рис. 7. Стан програми після запису першого символу в файл.

4.3 Функція "putback(...)"

У C++ потік `ifstream` має функцію-член `"putback(...)"`. По суті, вона не



«повертає символ назад» (тобто не змінює вміст файлу), а веде себе так, як ніби це робить. На рис. 8 показує стан, в якому програма перейде зі стану рис. 6 після виконання оператора:

```
in_stream.putback(ch);
```

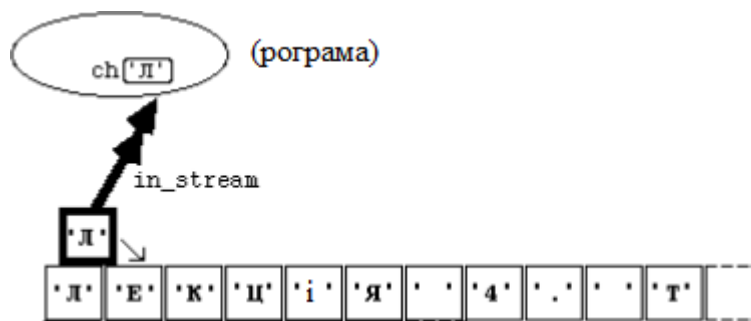


Рис. 8. Стан програми після виклику функції "putback('Л')".

«Повернути назад» можна будь-який символ. Стан програми після виклику

```
in_stream.putback('7');
```

показано на рис. 9.

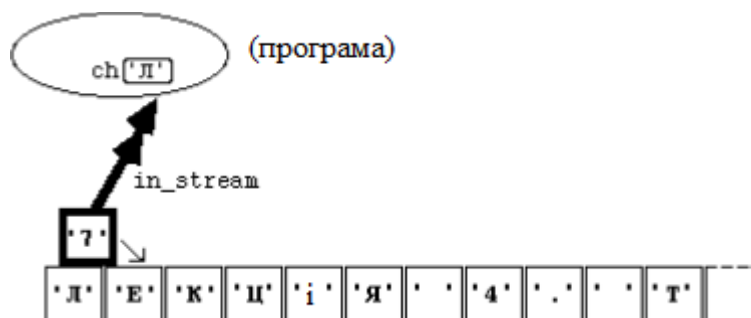



Рис. 9. Стан програми після виклику функції "putback('7')".

	<p align="center">Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 75 із 153	

Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. Трофименко О.Г. С++. 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.
6. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.
7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).

Лекція №9

Тема лекції: «Структури»

План:

1. Структури.
2. Структури з бітовими полями.
3. Масиви структур.



4. Показчики на структури.
5. Структури як параметри функцій.
6. Передача за посиланням масивів структур.
7. Функції роботи з датою та часом.


Зміст лекції:

У мові програмування C++ визначено декілька складених типів даних, тобто типів, які складаються з двох або більше елементів. Такі складені типи даних як масив і перерахунок розглядалися раніше відповідно у розд. 6.4 і 9.4. З двома іншими – структурами і об'єднаннями – ми познайомимося у цьому розділі, а знайомство з ще одним складеним типом – класом – ми відкладемо до розд. 12. І, хоча структуровані та об'єднані типи даних призначено для задоволення різних програмних потреб, обидва вони є зручними засобами керування групами взаємопов'язаних між собою змінних. При цьому важливо розуміти, що створення структури (або об'єднання) означає створення нового визначеного програмістом типу даних. Сама можливість створення власних типів даних є ознакою потужності мови C++.

У мові програмування C++ структури та об'єднання мають як об'єктноорієнтовані, так і не об'єктно-орієнтовані атрибути. У цьому розділі детально проаналізуємо тільки останні. Об'єктно-орієнтовані їх властивості будемо розглядати в наступному розділі після введення таких понять як класи і об'єкти.

Організація роботи зі структурами даних

У програмуванні структури даних – способи організації даних у комп'ютерах. Часто разом зі структурою даних пов'язується і специфічний перелік операцій, що можуть бути виконаними над даними, організованими в таку структуру. Правильний підбір структур даних є надзвичайно важливим для ефективного функціонування відповідних алгоритмів їх оброблення. Добре побудовані структури даних дають

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯНАУ НМК 07.01.07–01–2022
		Стор. 77 із 153	

змогу оптимізувати використання машинного часу та пам'яті комп'ютера для виконання найбільш критичних операцій.

Відома формула "Програма = Алгоритми + Структури даних" дуже точно виражає потребу відповідального ставлення до такого підбору. Тому іноді навіть не обраний алгоритм для оброблення масиву даних визначає вибір тої чи іншої структури даних для їх збереження, а навпаки. Підтримка базових структури даних, які використовуються в програмуванні, включена в комплекти стандартних бібліотек мови програмування C++.

Основні положення

У мові програмування C++ структура представляє колекцію змінних, об'єднаних загальним іменем, яка забезпечує зручний засіб зберігання споріднених даних в одному місці. Структура – сукупність різних типів даних, оскільки вони складаються з декількох різних, але логічно взаємопов'язаних між собою змінних. З цих самих причин структури іноді називають складеними або конгломератними типами даних.

Структура — група взаємопов'язаних між собою змінних.

Перед визначенням структурних змінних, необхідно визначити формат структури. Це робиться за допомогою оголошення структури. Оголошення структури дає змогу компілятору зрозуміти, змінні якого типу вона містить. Змінні, які належать до структури, називаються її членами. Члени структури також називають полями.

Член структури — змінна, яка є частиною структури.

У загальному випадку всі члени структури мають бути логічно пов'язані одна з одною! Наприклад, структури зазвичай використовують для зберігання такої інформації, як поштові адреси, банківські реквізити, елементи книжкової бібліографії і т.ін. Безумовно, відносини між членами структури абсолютно



суб'єктивні і визначаються програмістом. Компілятор "нічого про них не знає" (або "не хоче знати").

Ключове слово `struct` означає початок оголошення структури.

Загальний формат оголошення структури має такий вигляд:

```
struct ім'я_типу_структури {  
тип ім'я_члена_1;  
тип ім'я_члена_2;  
тип ім'я_члена_3;  
.....  
тип ім'я_члена_n;  
} структурна_змінна_1, структурна_змінна_2, ..., структурна_змінна_m;
```

Розглянемо деякі приклади оголошення структур. Визначимо структуру, яка може містити інформацію про продукцію, що зберігаються на складі приватної фірми. Один запис інвентарної відомості зазвичай складається з декількох даних, наприклад: назву продукції, вартості та наявної кількості. Тому для керування такою інформацією зручно використовувати саме таку структуру. У наведеному нижче коді програми оголошується структура, яка визначає такі елементи: назву продукції, її вартість, роздрібну ціну, наявну кількість, кількість днів до поновлення запасів. Цих даних часто цілком достатньо для керування складським господарством. Про початок оголошення структури компіляторіві повідомляє ключове слово `struct`:

```
struct invStruct { // Попереднє оголошення типу структури  
char nameProd[40]; // Назва продукції  
double vartProd; // Вартість продукції  
double rozdrCina; // Роздрібна ціна  
int наявKilk; // Наявна кількість  
int kilkDniv; // Кількість днів до поновлення запасів
```



Ім'я типу структури — її специфікатор типу даних.

Зверніть увагу на те, що оголошення структури завершується крапкою з комою, тобто вона може бути настановою. Ім'ям типу структури тут є `invStruct`. Іншими словами, ім'я `invStruct` ідентифікує конкретну структуру даних і є її специфікатором типу.

У нашому оголошенні структури насправді не було створено жодної структурної змінної, а визначено тільки формат типу даних. Щоб за допомогою цієї структури визначити реальну структурну змінну (тобто фізичний об'єкт), потрібно записати таку настанову:

```
invStruct InvVidom;
```

Ось тепер визначається структурна змінна типу `invStruct` з іменем `InvVidom`.

Під час визначення структурної змінної компілятор мови програмування C++ автоматично виділить об'єм пам'яті, достатній для зберігання всіх членів структури. На рис. 10.1 показано, як змінну `InvVidom` буде розміщено в пам'яті комп'ютера (у припущенні, що `double`-значення займає 8 байтів, а `int`-значення – 4 байти).

Одночасно з оголошенням імені типу структури можна визначити одну або декілька структурних змінних:

```
struct invStruct { // Попереднє оголошення типу структури  
    char nameProd[40]; // Назва продукції  
    double vartProd; // Вартість продукції  
    double rozdrCina; // Роздрібна ціна  
    int nayavKilk; // Наявна кількість  
    int kilkDniv; // Кількість днів до поновлення запасів  
} InvVidomA, InvVidomB, InvVidomC; // Визначення структурної змінної
```

Цей код програми оголошує структурний тип `invStruct` і визначає структурні змінні `InvVidomA`, `InvVidomB` і `InvVidomC` цього типу. Важливо розуміти, що кожна

	<p style="text-align: center;">Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 80 із 153	

структурна змінна містить власні копії членів структури. Наприклад, поле `varProd` структури `InvVidomA` ізольовано від поля `varProd` структури `InvVidomB`. Отже, зміни, що вносяться в певне поле однієї структурної змінної, ніяк не впливають на вміст такого самого поля іншої структурної змінної.

Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. Трофименко О.Г. С++. 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.
6. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.
7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).

Лекція №10

Тема лекції: «Структури (продовження)»

План:



1. Об'єднання.
2. Переліки.
3. Операції з об'єднаннями.

Зміст лекції:

Об'єднання (англ. union) – тип даних користувача, який дуже схожий на структуру, проте тут всі дані займають одну і ту ж область пам'яті. Тому розмір об'єднання дорівнює розміру його найбільшого члена. У будь-який момент часу об'єднання зберігає значення тільки одного з членів. Тобто на якомусь етапі Вам потрібно один тип даних, на іншому – інший. Тому, загалом, об'єднання економить пам'ять комп'ютера від непотрібних на даному етапі змінних.

Оскільки об'єднання зберігає і використовує завжди одне поле їх множини на вибір, то виникає питання про виділення пам'яті під це поле. Тут принцип зрозумілий – вибирається найбільший з типів даних. У мові програмування C++ до об'єднання звертаються так само, як і до структури: через символ "->" при використанні покажчика, або "." при використанні звичайної змінної.

Оголошення об'єднання

Об'єднання складається з декількох змінних, які розділяють між собою одну і ту саму область пам'яті. Це означає, що об'єднання забезпечує можливість інтерпретації однієї і тієї ж самої конфігурації бітів двома (або більше) різними способами. Оголошення об'єднання, як неважко переконатися на наведеному нижче прикладі, є подібним до оголошення структури:

```
union myUnion { // Попереднє оголошення типу об'єднання
    short int c;
    char ch;
};
```

	Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 82 із 153	

У наведеному прикладі оголошується об'єднання, у якому значення типу `short int` і значення типу `char` розділяють одну і ту саму область пам'яті. Необхідно відразу ж з'ясувати один момент: неможливо зробити так, щоб це об'єднання зберігало і цілочисельне значення, і символ одночасно, оскільки змінні `s` та `ch` накладаються (у пам'яті) одне на друге. Але програма у будьякий момент може обробляти інформацію, що міститься у цьому об'єднанні, як цілочисельне значення або як символ. Отже, об'єднання забезпечує два (або більше) способи представлення однієї і тієї ж самої множини даних. Як видно з цього прикладу, об'єднання оголошується за допомогою ключового слова `union`.

Як і під час оголошення структур, так і під час оголошення об'єднання не визначається жодна змінна. Змінну можна визначити, розмістивши її ім'я в кінці оголошення або скориставшись окремою настановою визначення. Щоб визначити змінну об'єднання іменем `uVar` типу `myUnion`, достатньо записати:

```
myUnion uVar;
```

У змінній об'єднання `uVar` як змінна `s` типу `short int`, так і символна змінна `ch` розділяють одну і ту саму область пам'яті.

Під час оголошення об'єднання компілятор автоматично виділяє область пам'яті, достатню для зберігання в об'єднанні змінних найбільшого за об'ємом типу.

Щоб отримати доступ до елемента об'єднання, використовують такий самий синтаксис, який застосовується і для структур: оператори "крапка" і "стрілка". При безпосередньому зверненні до об'єднання (або за допомогою посилання) використовують оператор "крапка". Якщо ж доступ до змінної об'єднання здійснюється через покажчик, використовують оператор "стрілка". Наприклад, щоб присвоїти букву "A" елемента `ch` об'єднання `uVar`, достатньо використати таку настанову:

```
uVar.ch = "A";
```



У наведеному нижче прикладі функції Fun1() передається посилання на об'єднання uVar. У тілі цієї функції за допомогою покажчика змінної с присвоюється значення 10:

```
Fun1(&uVar); // Передаємо функції Fun1() покажчик
// на об'єднання uVar.
//...
}
void Fun1(myUnion *un)
{
un->c =10; // Присвоюємо число 10 члену об'єднання uVar через покажчик.
}
```


Оскільки об'єднання дають змогу складеній програмі інтерпретувати одні і ті ж самі дані по-різному, то вони часто використовують у випадках, коли потрібне незвичайне перетворення типів. Наприклад, наведений нижче код програми використовує об'єднання для переставляння двох байтів, які становлять коротке цілочисельне значення. Тут для відображення вмісту цілочисельних змінних використовується функція DispBit(), розроблена в розд. 91.

Код програми 10.8. Демонстрація механізму використання об'єднання для переставляння двох байтів, які становлять коротке цілочисельне значення

```
#include <vcl>
#include <iostream> // Потокowe введення-виведення
#include <conio> // Консольний режим роботи
using namespace std; // Використання стандартного простору імен
void DispBit(unsigned u);
union swapBytes { // Попереднє оголошення типу об'єднання
```



```
short int num;
char ch[2];
};
int main()
{
swapBytes bitS;
char tmp;
bitS.num = 15; // Двійковий код: 0000 0000 0000 1111
cout << "Початкові байти:";
DispBit(bitS.ch[1]); cout << " ";
DispBit(bitS.ch[0]); cout << endl << endl;
// Обмін байтів.
tmp = bitS.ch[0]; bitS.ch[0] = bitS.ch[1]; bitS.ch[1] = tmp;
cout << "Байти після переставляння:";
DispBit(bitS.ch[1]); cout << " ";
DispBit(bitS.ch[0]); cout << endl << endl;
getch(); return 0;
}
// Відображення бітів, з яких складається байт.
void DispBit(unsigned u)
{
register int t;
for(t=128; t>0; t=t/2)
if(u & t) cout << "1 ";
else cout << "0 ";
}
```

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 85 із 153	

У процесі виконання програма відображає на екрані такі результати:

Початкові байти : 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

Байти після переставляння : 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0

У цьому коді програми цілочисельній змінній bitS.num присвоюється число 15. Перестановка двох байтів, що становлять це значення, здійснюється шляхом обміну двох символів, які утворюють масив ch. Як наслідок, старший і молодший байти цілочисельної змінної num міняються місцями. Ця операція можлива тільки тому, що як змінна num, так і масив ch розділяють одну і ту саму область пам'яті.

Список джерел:

1. Труніна Г.О., Настенко Д.В., Нестерко А.Б. Обчислювальна техніка та програмування : навч. посіб. – Київ : НТУУ «КПІ», 2020. – 117 с.
2. Вступ до алгоритмів. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест і Кліфорд Стайн. Переклад з англійської (третього видання). - К.: К.І.С., 2019 - 1288 с.
3. Івашко В.В. Основи програмування : конспект лекцій. Чернівці : Чернівецький нац. універс. ім. Ю.Федьковича 2021. – 177 с.
4. Васильєв О. М. Програмування мовою Python. Тернопіль : Навчальна книга – Богдан, 2019. 504 с.
5. Трофименко О.Г. С++. 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.
6. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. Київ :Видавництво Ліра-К, 2020.
7. Єнчев С.В., Прохоренко І.В., Тимошенко Н.А. Обчислювальна техніка та алгоритмічні мови. Робота з офісними програмами: лабораторний практикум. – К. : НАУ, 2023. – 48 с. (електронна версія).




**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Аерокосмічний факультет**

Кафедра комп'ютеризованих електротехнічних систем та технологій

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ З ПІДГОТОВКИ СТУДЕНТІВ ДО
ЛАБОРАТОРНИХ РОБІТ**

з дисципліни **«Обчислювальна техніка та алгоритмічні мови»**
за напрямом 141 «Електроенергетика, електротехніка та електромеханіка»

Укладач: к.т.н., доц. С.В. Єгоров
Методичні рекомендації розглянуті та
схвалені на засіданні кафедри
комп'ютеризованих електротехнічних систем
та технологій
Протокол № від « »20 ____ р.

	Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 87 із 153	

ЗАГАЛЬНІ ВКАЗІВКИ

Метою проведення лабораторних занять з навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови» є вивчення принципів і методів розробки програмного забезпечення для ОС Windows, ознайомлення із програмними засобами розробки, налагодження і програмування в середовищі сучасних ОС які використовуються у якості засобів контролю за процесами, які відбуваються у засобах вимірювальної техніки, мікропроцесорних засобах автоматизації, побутових пристроях та промисловому обладнанні.

Основними *завданнями* вивчення дисципліни є ознайомлення з галузями використання, класифікацією та можливостями сучасних мов програмування, а програмними засобами для програмування ОС; формування уявлень про принципи та типові алгоритми роботи сучасного ПЗ.

Результати навчання, які дає можливість досягнути навчальна дисципліна **наступні:**

В результаті вивчення дисципліни студенти повинні мати здатність розв'язувати спеціалізовані задачі та вирішувати практичні проблеми під час професійної діяльності у галузі електроенергетики, електротехніки та електромеханіки або у процесі навчання, що передбачає застосування теорій та методів фізики та інженерних наук і характеризуються комплексністю та невизначеністю умов.

Програмні результати:

ОПП «Енергетичний менеджмент»:

- ПР13. Розуміти значення традиційної та відновлюваної енергетики для успішного економічного розвитку країни.

ОПП «Електротехнічні системи електроспоживання»:



- ПРН6. Застосовувати прикладне програмне забезпечення та мікропроцесорну техніку для вирішення практичних проблем у професійній діяльності.
- ПРН18. Вміти самостійно вчитися, опановувати нові знання і вдосконалювати навички роботи з сучасним обладнанням, вимірювальною технікою та прикладним програмним забезпеченням.

У результаті вивчення навчальної дисципліни студент повинен набути наступні:

Загальнонаукові компетенції.

ОПП «Енергетичний менеджмент»:

ЗК1. Здатність до абстрактного мислення, аналізу і синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК4. Здатність спілкуватися іноземною мовою.

ЗК5. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.


ЗК6. Здатність виявляти, ставити та вирішувати проблеми.

ЗК7. Здатність працювати в команді.

ЗК8. Здатність працювати автономно.

ЗК9. Здатність реалізувати свої права і обов'язки як члена суспільства, усвідомлювати цінності громадянського (вільного демократичного) суспільства та необхідність його сталого розвитку, верховенства права, прав і свобод людини і громадянина в Україні.

ЗК10. Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 89 із 153	

форми рухової активності для активного відпочинку та ведення здорового способу життя

ЗК11. Навички використання інформаційних і комунікаційних технологій.

ЗК12. Здатність планувати та управляти часом.

ЗК13. Здатність приймати обґрунтовані рішення.

ОПП «Електротехнічні системи електроспоживання»:

ЗК5. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК11. Здатність використання інформаційних і комунікаційних технологій.

Фахові компетенції.

ОПП «Електротехнічні системи електроспоживання»:

ФК9. Усвідомлення необхідності підвищення ефективності електроенергетичного, електротехнічного та електромеханічного устаткування.

Лабораторна робота №1.

Тема: «Програмування лінійних алгоритмів»


Мета роботи: дослідити виконання арифметичних, логічних команд та лінійні алгоритми.

Методичні вказівки. Дослідити функції математичних виразів, що знаходяться в бібліотеці `math.lib`. Вивчити основні типи даних та приведення типів.

Лабораторна робота №2.

Тема: «Програмування алгоритмів, що розгалужуються»

Мета роботи: користуючись середовищем розробника написати програму,

	<p style="text-align: center;">Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 90 із 153	

що використовує алгоритм, що розгалужується.

Методичні вказівки. Вивчити логічні та порівняльні оператори, що використовуються в C++.

Лабораторна робота №3.

Тема: «Програмування циклічних алгоритмів»

Мета роботи: користуючись середовищем розробника написати програму, що використовує циклічний алгоритм.

Методичні вказівки. Вивчити оператори, що використовуються для організації циклів в C++.

Лабораторна робота №4.

Тема: «Програмування з використанням одномірних масивів»

Мета роботи: користуючись середовищем розробника написати програму, що використовує одновірні масиви.

Методичні вказівки. Вивчити оператори, що використовуються для оголошення одномірних масивів, та забезпечують роботу з масивами.

Лабораторна робота №5.

Тема: «Показчики. Програмування з використанням динамічних двомірних масивів»

Мета роботи: користуючись середовищем розробника написати програму, що використовує динамічні двомірні масиви.

Методичні вказівки. Вивчити оператори, що використовуються для оголошення двомірних масивів, та забезпечують роботу з масивами.



Лабораторна робота №6.

Тема: «Програмування з використанням рядків»

Мета роботи: користуючись середовищем розробника написати програму, що використовує рядки.

Методичні вказівки. Вивчити оператори, що використовуються для оголошення рядків, та функції для роботи з рядками.

Лабораторна робота №7.

Тема: «Програмування з використанням структур»

Мета роботи: користуючись середовищем розробника написати програму, що використовує структури.

Методичні вказівки. Вивчити оператори, що використовуються для оголошення структур, та методи роботи зі структурами.

Лабораторна робота №8.

Тема: «Програмування з використанням функцій»

Мета роботи: користуючись середовищем розробника написати програму, що використовує функції.


Методичні вказівки. Вивчити методи оголошення функцій, методи передачі аргументів у функції.

Лабораторна робота №9.

Тема: «Програмування з використанням файлів»

Мета роботи: користуючись середовищем розробника написати програму, що використовує файли.

Методичні вказівки. Вивчити методи та функції роботи з файлами.


	<p align="center">Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 92 із 153	

Лабораторна робота №10.

Тема: «Сортування за ключом одновимірних масивів структур»

Мета роботи: користуючись середовищем розробника написати програму, що використовує методи сортування масивів.

Методичні вказівки. Вивчити методи сортування масивів.

	Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 93 із 153	

Аерокосмічний факультет
Кафедра комп'ютеризованих електротехнічних систем та технологій

ЗАТВЕРДЖУЮ

Зав. кафедри _____ В.П. Квасніков

« » _____ 2022 р.

МОДУЛЬНА КОНТРОЛЬНА РОБОТА №1
з дисципліни «Обчислювальна техніка та алгоритмічні мови»

Розробник
к.т.н., доц. Сергій Єгоров



Перелік завдань

для проведення модульної контрольної роботи №1

$$1. s = \frac{|x| + \frac{1}{y^2 + 1}}{y \left(\operatorname{arctg} z - \frac{1}{3} \right)} \left(1 + \frac{z^2}{3 - z^2/5} \right)$$

$$2. s = \frac{\sqrt{10 \left(\sqrt[3]{x} + x^{y+2} \right)}}{x^2 + y^2 + 2} - \sqrt[3]{x^6 + \ln^2 y}$$

$$3. s = \frac{1 + \sin^2(x + y)}{\operatorname{arctg}(x) + \operatorname{arctg}(z)} x^{|y|} + \cos^2 \left(\operatorname{arctg} \frac{1}{z} \right)$$

$$4. s = |\cos x - \cos y|^{-1/\sin z} + \frac{|y - x|^2}{2} - \frac{2^{(y^x)}}{3}$$

$$5. s = 2 \cos \left(x - \frac{2}{3} \right) (1 + |y - x|) + e^{|x-y|} \operatorname{tg}^3 z$$



$$6. s = \sqrt[3]{9 + (x - y)^2} \frac{\cos^3(y)}{e^{|x-y|} + \frac{x}{2}} y^{x+1}$$

$$7. s = 5 \operatorname{arctg}(x) - \frac{1}{4} \arccos(x) \left| x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right|$$

$$8. s = \frac{\sqrt[3]{|y-2|} + 3}{\left| x - \frac{2y}{1+x^2y^2} \right|} + \left(x - \frac{y}{2} \right)$$

$$9. s = \frac{x+3|x-y|+x^2}{|x-y|z+x^2} (y-x) \frac{\cos y - \frac{z}{(y-x)}}{1+(y-x)^2}$$

$$10. s = 2^{-x} \sqrt{x + \sqrt[4]{|y|}} \sin^2(\operatorname{arctg}(z))$$

$$11. s = y^{\sqrt[3]{|x|}} + \left(\arcsin^2 z - |x-y| \right) \cdot |x-y| \left(1 + \frac{\sin^2 z}{\sqrt{x+y}} \right)$$

$$12. s = |y-x|^3 + (3^x)^y - \frac{1+x|y-\operatorname{tg}z|}{\ln\left(y^{-\sqrt{|x|}}\right)}$$

$$13. s = \frac{\sqrt[4]{y + \sqrt[3]{x-1}}}{|x-y|(\sin^2 z + \operatorname{tg}z)}$$

$$14. s = \frac{\sqrt[3]{e^{x-1/\sin z}}}{e^{|x-y|}|x-y|^{x+y}} + \frac{x + \frac{y}{2}}{2|x+y|} (x+1)^{(1+2\sin^2 y)}$$



$$15. s = \frac{x^{y+1} + e^{y-1}}{\frac{1}{2} + \sin^2 y} \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4} \right)$$

$$16. s = \begin{cases} \ln(|x| + |y|), & xy > 0 \\ (x + y)^2 + y^3, & xy < 0 \\ \sqrt[3]{\sin^2(y)}, & \text{інакше} \end{cases}$$

$$17. s = \begin{cases} \ln|x^3| + \cos(y) & x/y > 0 \\ 2/3 + \ln(|\sin(y)|), & x/y < 0 \\ e^{x-|y|}, & \text{інакше} \end{cases}$$

$$18. s = \begin{cases} x^2 + \sqrt[3]{y} + \sin(y), & x - y = 0 \\ 2x^3 + 3y^2, & x - y > 0 \\ \sqrt[3]{x} \cdot \sin(x), & \text{інакше} \end{cases}$$

$$19. s = \begin{cases} (y + x)^2 + x^3, & x > y \\ (y - x)^3 + \cos(x), & x < y \\ \sqrt[3]{|x|} + x^3/y, & \text{інакше} \end{cases}$$

$$20. s = \begin{cases} y\sqrt{|x|} + 3\sin(x), & x > y \\ x\sqrt{|x|}, & x < y \\ y \cdot x^2, & \text{інакше} \end{cases}$$


$$21. s = \begin{cases} (x^2 + y)^3, & 0,5 < xy < 10 \\ \sqrt[3]{|x + y|}, & 0,1 < xy < 0,5 \\ \sqrt[3]{|x|} + y, & \text{інакше} \end{cases}$$

$$22. s = \begin{cases} e^{-x}, & 1 < xb < 10 \\ |x + y|^3, & 12 < xb < 40 \\ \ln|x/y| + x/y, & \text{інакше} \end{cases}$$

$$23. s = \begin{cases} (x^2 + y)^3, & x/y < 0 \\ \sqrt[3]{|x - y|} + \operatorname{tg}(x), & x/y > 0 \\ \sqrt[3]{|\sin(y)|}, & \text{інакше} \end{cases}$$

$$24. s = \begin{cases} \ln|x^2 \cdot y|, & x > |y| \\ |x - y|, & 3 < x < |y| \\ x^3 + \sin^2(y), & \text{інакше} \end{cases}$$

$$25. s = \begin{cases} (x + y)^2 - \sqrt[3]{|x|}, & |xy| > 10 \\ \ln(x) + \sqrt[3]{|y|}, & |xy| < 10 \\ 2x^2, & \text{інакше} \end{cases}$$

	Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»	Шифр документа	СМЯНАУ НМК 07.01.07–01–2022
		Стор. 97 із 153	

Аерокосмічний факультет
Кафедра комп'ютеризованих електротехнічних систем та технологій

ЗАТВЕРДЖУЮ

Зав. кафедри _____ В.П. Квасніков

« » _____ 2022 р.

МОДУЛЬНА КОНТРОЛЬНА РОБОТА №2
з дисципліни «Обчислювальна техніка та алгоритмічні мови»

Розробник
к.т.н., доц. Сергій ЄГОРОВ



Перелік завдань

для проведення модульної контрольної роботи №2

Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,1$ до $b = 1,2$ з кроком $h = 0,1$.

1.
$$y = \sum_{n=1}^{20} \frac{x^{n-1}}{2n+1}.$$

2.
$$y = \sum_{n=0}^{20} \frac{(2x)^n}{n+1}.$$

3.
$$y = \sum_{n=1}^{20} \frac{x^{n-1}}{\sin(nx)}.$$

4.
$$y = \sum_{n=1}^{20} \frac{n^2+1}{n} \left(\frac{x}{2}\right)^n.$$

5.
$$y = \sum_{n=1}^{20} \frac{\cos\left(n \cdot \frac{\pi}{4}\right)}{n+1} x^n.$$

6.
$$y = \sum_{n=1}^{20} \frac{x^{2n-2}}{2n+1}.$$

7.
$$y = \sum_{n=0}^{20} \frac{x^{2n}}{\cos(nx)}.$$

8.
$$y = \sum_{n=1}^{20} \frac{2n^2+1}{2n} x^{2n-2}.$$

9.
$$y = \sum_{n=1}^{20} \frac{2n+1}{\sin(nx)} x^{n-1}.$$

10.
$$y = \sum_{n=0}^{20} \frac{\cos^n(x)}{2n+1}.$$

11.
$$y = \sum_{n=1}^{20} \frac{n \cdot x^{n-1}}{\sin(2n+x)}.$$

12.
$$y = \sum_{n=1}^{20} \frac{(1+x)^{n-1}}{|\sin^n(x)|}.$$

13.
$$y = \sum_{n=1}^{20} \frac{x^{2n-2}}{4 \cos(nx^2)}.$$

14.
$$y = \sum_{n=1}^{20} \frac{n^2}{(2n+1)} x^{n-1}.$$

15.
$$y = \sum_{n=0}^{20} \frac{\cos\left(n \cdot \frac{\pi}{4}\right)}{n^2} x^n.$$

16. Перетворити масив таким чином: всі негативні елементи масиву перенести на кінець, зберігши вихідне взаємне розташування, як серед негативних, так і серед інших елементів масиву.
17. Розташувати елементи масиву у зворотному порядку.
18. Знайти та поміняти місцями елементи, що мають мінімальне та максимальне значення в масиві.
19. Визначити, чи впорядковані елементи масиву зі зростанням.
20. Вивести всі елементи масиву, що не повторюються.



21. Зрушити елементи масиву циклічно на n позицій вліво.
22. Зрушити елементи масиву циклічно на n позицій вправо.
23. Видалити мінімальний та максимальний елементи масиву.
24. Сформувати два нових масиви: у перший записати негативні елементи вихідного масиву, у другий – усі інші.
25. Визначити, чи симетричний масив, тобто він читається однаково зліва направо і праворуч наліво.
26. Знайти кількість елементів масиву, які відрізняються від середнього значення елементів масиву не більше ніж на 3.
27. Визначити кількість інверсій у масиві (таких пар елементів, у яких більше значення знаходиться ліворуч від меншого).
28. Визначити кількість елементів, значення яких більше середнього значення всіх елементів масиву.
29. Видалити елементи, значення яких менше середнього значення всіх елементів масиву.
30. Видалити з масиву елементи, що повторюються.



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 1

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Поясніть на конкретному прикладі основні етапи рішення задачі

II. Напишіть програму для обчислення наступного виразу:

$$s = |\cos x - \cos y|^{-1/\sin z} + \frac{|y-x|^2}{2} - \frac{2^{(y^x)}}{3}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б.)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 2

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Походження мови C++.

II. Напишіть програму для обчислення наступного виразу:

$$s = \frac{\sqrt{10(\sqrt[3]{x} + x^{y+2})}}{x^2 + y^2 + 2} - \sqrt[3]{x^6 + \ln^2 y}$$

Затверджено на засіданні кафедри

Протокол № _____ від „_____” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б.)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 3

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Стандарт ANSI C++.

II. Напишіть програму для обчислення наступного виразу:

$$s = \frac{|x| + \frac{1}{y^2 + 1}}{y \left(\arctg z - \frac{1}{3} \right)} \left(1 + \frac{z^2}{3 - z^2/5} \right)$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б.)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 4

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Керування порядком виконання команд за допомогою оператора if.

II. Напишіть програму для обчислення наступного виразу:

$$s = 2 \cos\left(x - \frac{2}{3}\right) (1 + |y - x|) + e^{|x-y|} \operatorname{tg}^3 z$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б.)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 5

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Змінні, типи даних і вирази. Ідентифікатори.

II. Напишіть програму для обчислення наступного виразу:

$$s = \frac{1 + \sin^2(x + y)}{\arctg(x) + \arctg(z)} x^{|y|} + \cos^2\left(\arctg\frac{1}{z}\right)$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 6

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Змінні, типи даних і вирази. Типи даних. Цілі числа

II. Напишіть програму для обчислення наступного виразу:

$$s = \frac{\sqrt[3]{|y-2|} + 3}{\left| x - \frac{2y}{1+x^2y^2} \right|} + \left(x - \frac{y}{2} \right)$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 7

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Змінні, типи даних і вирази. Типи даних. Дійсні числа

II. Напишіть програму для обчислення наступного виразу:

$$s = 5 \arctg(x) - \frac{1}{4} \arccos(x) \left| x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right|$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б.)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (П.І.Б.)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 8

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Змінні, типи даних і вирази. Типи даних. Перетворення типів у виразах

II. Напишіть програму для обчислення наступного виразу:

$$s = \sqrt[3]{9 + (x - y)^2} \frac{\cos^3(y)}{e^{|x-y|} + \frac{x}{2}} y^{x+1}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б.)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 9

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Змінні, типи даних і вирази. Типи даних. Символьний тип

II. Напишіть програму для обчислення наступного виразу:

$$s = \frac{x + 3|x - y| + x^2}{|x - y|z + x^2} (y - x) \frac{\cos y - \frac{z}{(y - x)}}{1 + (y - x)^2}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 10

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Операторо циклу for

II. Напишіть програму для обчислення наступного виразу:

$$s = \frac{\sqrt[4]{y + \sqrt[3]{x-1}}}{|x-y|(\sin^2 z + \operatorname{tg} z)}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 11

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Відображення на екрані дійсних чисел.

II. Напишіть програму для обчислення наступного виразу:

$$s = \frac{\sqrt[3]{e^{x-1}/\sin z}}{e^{|x-y|}|x-y|^{x+y}} + \frac{x + \frac{y}{2}}{2|x+y|} (x+1)^{(1+2\sin^2 y)}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 12

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Описи, константи та перерахування.

II. Напишіть програму для обчислення наступного виразу:

$$s = y^{\sqrt[3]{|x|}} + (\arcsin^2 z - |x - y|) \cdot |x - y| \left(1 + \frac{\sin^2 z}{\sqrt{x + y}} \right)$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б.)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 13

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Розташування описів констант і змінних у вихідному коді.

II. Напишіть програму для обчислення наступного виразу:

$$s = |y - x|^3 + (3^x)^y - \frac{1 + x|y - \operatorname{tg} z|}{\ln(y^{-\sqrt{|x|}})}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 14

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Присвоєння та вирази. Коротка форма написання операторів присвоєння.

II. Напишіть програму для обчислення наступного виразу:

$$s = \begin{cases} \ln(|x| + |y|), & xy > 0 \\ (x + y)^2 + y^3, & xy < 0 \\ \sqrt[3]{\sin^2(y)}, & \text{інакше} \end{cases}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 15

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Присвоєння та вирази. Логічні вирази та оператори.

II. Напишіть програму для обчислення наступного виразу:

$$s = \begin{cases} (x+y)^2 - \sqrt[3]{|x|}, & |xy| > 10 \\ \ln(x) + \sqrt[3]{|y|}, & |xy| < 10 \\ 2x^2, & \text{інакше} \end{cases}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 16

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Призначення підпрограм.

II. Напишіть програму для обчислення наступного виразу:

$$s = \begin{cases} e^{-x}, & 1 < xb < 10 \\ |x + y|^3, & 12 < xb < 40 \\ \ln|x/y| + x/y, & \text{інакше} \end{cases}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 17

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Визначення нових функцій.

II. Напишіть програму для обчислення наступного виразу:

$$s = \begin{cases} (y+x)^2 + x^3, & x > y \\ (y-x)^3 + \cos(x), & x < y \\ \sqrt[3]{|x|} + x^3 / y, & \text{інакше} \end{cases}$$

Затверджено на засіданні кафедри
Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б.)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 18

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Способи передачі параметрів всередину функцій. Передавання параметрів за значеннями.

II. Напишіть програму для обчислення наступного виразу:

$$s = \begin{cases} y\sqrt{|x|} + 3\sin(x), & x > y \\ x\sqrt{|x|}, & x < y \\ y \cdot x^2, & \text{інакше} \end{cases}$$

Затверджено на засіданні кафедри

Протокол № _____ від „_____” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 19

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Способи передачі параметрів всередину функцій. Передавання параметрів за посиланням.

II. Напишіть програму для обчислення наступного виразу:

$$s = \begin{cases} (x^2 + y)^3, & x/y < 0 \\ \sqrt[3]{|x - y|} + \operatorname{tg}(x), & x/y > 0 \\ \sqrt[3]{|\sin(y)|}, & \text{інакше} \end{cases}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 20

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Поліморфізм і перевантаження функцій.

II. Напишіть програму для обчислення наступного виразу:

$$s = \begin{cases} x^2 + \sqrt[3]{y} + \sin(y), & x - y = 0 \\ 2x^3 + 3y^2, & x - y > 0 \\ \sqrt[3]{x} \cdot \sin(x), & \text{інакше} \end{cases}$$

Затверджено на засіданні кафедри
Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 21

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Процедурна абстракція і «хороший» стиль програмування.

II. Напишіть програму для обчислення наступного виразу:

$$s = \begin{cases} (x^2 + y)^3, & 0,5 < xy < 10 \\ \sqrt[3]{|x + y|}, & 0,1 < xy < 0,5 \\ \sqrt[3]{|x|} + y, & \text{інакше} \end{cases}$$

Затверджено на засіданні кафедри
Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б.)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 22

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Модульне програмування.

II. Напишіть програму для обчислення наступного виразу:

$$s = \begin{cases} \ln|x^2 \cdot y|, & x > |y| \\ |x - y|, & 3 < x < |y| \\ x^3 + \sin^2(y), & \text{інакше} \end{cases}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 23

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Текстові файли та потоки вводу-виводу. Призначення файлів.

II. Напишіть програму для обчислення наступного виразу:

$$s = \begin{cases} \ln|x^3| + \cos(y) & x/y > 0 \\ 2/3 + \ln(|\sin(y)|), & x/y < 0 \\ e^{x-y}, & \text{інакше} \end{cases}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 24

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Текстові файли та потоки вводу-виводу. Потоки вводу-виводу. Створення потоків.

II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=1}^{20} \frac{n^2}{(2n+1)} x^{n-1}$$

Затверджено на засіданні кафедри
Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б.)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (П.І.Б)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 25

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

- I. Текстові файли та потоки вводу-виводу. Потоки вводу-виводу. Підключення та відключення потоків від файлів.
- II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=1}^{20} \frac{n^2 + 1}{n} \left(\frac{x}{2}\right)^n.$$

Затверджено на засіданні кафедри
Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (П.І.Б.)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 26

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

- I. Текстові файли та потоки вводу-виводу. Перевірка на помилок при виконанні файлових операцій.
- II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=1}^{20} \frac{\cos\left(n \cdot \frac{\pi}{4}\right)}{n+1} x^n$$

Затверджено на засіданні кафедри
Протокол № _____ від „_____” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б.)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 27

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Текстові файли та потоки вводу-виводу. Символьний ввід/вивід.

II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=1}^{20} \frac{x^{2n-2}}{2n+1}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б.)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 28

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

- I. Текстові файли та потоки вводу-виводу. Функція виводу "put(...)" та функція "putback(...)".
- II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=1}^{20} \frac{x^{n-1}}{2n+1}$$

Затверджено на засіданні кафедри
Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 29

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

- I. Текстові файли та потоки вводу-виводу. Перевірка досягнення кінця файлу під час операцій введення. Перевірка кінця файлу за допомогою функції "eof()"
- II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=0}^{20} \frac{\cos^n(x)}{2n+1}$$

Затверджено на засіданні кафедри
Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 30

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Текстові файли та потоки вводу-виводу. Передача потоків до функцій як параметрів.

II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=1}^{20} \frac{x^{n-1}}{\sin(nx)}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 31

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

- I. Текстові файли та потоки вводу-виводу. Оператори вводу-виводу ">>" и "<<".
- II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=1}^{20} \frac{(1+x)^{n-1}}{|\sin^n(x)|}$$

Затверджено на засіданні кафедри
Протокол № _____ від „_____” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б.)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 32

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Оператори розгалуження та цикли. Логічні значення, вирази та функції

II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=1}^{20} \frac{2n+1}{\sin(nx)} x^{n-1}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 33

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Оператори розгалуження та цикли. Цикли " for ", " while " і " do...while "

II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=1}^{20} \frac{x^{2n-2}}{4 \cos(\pi x^2)}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 34

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Оператори розгалуження та цикли. Множинне розгалуження та оператор
"switch"

II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від
 $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=0}^{20} \frac{(2x)^n}{n+1}.$$

Затверджено на засіданні кафедри
Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 35

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

- I. Оператори розгалуження та цикли. Блоки і область застосування змінних
II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=0}^{20} \frac{x^{2n}}{\cos(nx)}$$

Затверджено на засіданні кафедри
Протокол № _____ від „_____” _____ 20__ року

Викладач _____ Сергій ЄГОРОВ
(підпис) (П.І.Б.)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 36

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Масиви та рядки символів. Призначення масивів та оголошення масивів.

II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=1}^{20} \frac{n \cdot x^{n-1}}{\sin(2n + x)}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 37

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Масиви та рядки символів. Використання елементів масиву у виразах.

II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=0}^{20} \frac{\cos\left(n \frac{\pi}{4}\right)}{n^2} x^n$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
АЕРОКОСМІЧНИЙ ФАКУЛЬТЕТ**

**КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ ЕЛЕКТРОТЕХНІЧНИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Володимир КВАСНІКОВ
(підпис) (ПІБ)

« _____ » _____ 20__ р.

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 38

Дисципліна «Обчислювальна техніка та алгоритмічні мови»

I. Масиви та рядки символів. Передавання масивів як параметрів функцій.

II. Вивести на екран таблицю значень функції $y(x)$ для x , що змінюється від $a = 0,2$ до $b = 1,6$ з кроком $h = 0,1$.

$$y = \sum_{n=1}^{20} \frac{2n^2 + 1}{2n} x^{2n-2}$$

Затверджено на засіданні кафедри

Протокол № _____ від „ _____ ” _____ 20__ року

Викладач _____
(підпис)

Сергій ЄГОРОВ
(П.І.Б)



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Аерокосмічний факультет**

Кафедра комп'ютеризованих електротехнічних систем та технологій


**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДЛЯ ВИКОНАННЯ КОНТРОЛЬНОЇ
РОБОТИ ДЛЯ СТУДЕНТІВ ЗАОЧНОЇ ФОРМИ**

з дисципліни **«Обчислювальна техніка та алгоритмічні мови»**
за напрямом 141 «Електроенергетика, електротехніка та електромеханіка»

Укладач: к.т.н., доц. С.В. Єгоров

Методичні рекомендації розглянуті та схвалені на засіданні кафедри комп'ютеризованих електротехнічних систем та технологій

Протокол № _____ від « _____ » 20 ____ р.

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		<p>Стор. 139 із 153</p>	

1.1.Обчислювальна система

Обчислювальна система – програмно-апаратний комплекс, призначений для надання послуг користувачу. Складається з апаратних засобів (hardware) та програмного забезпечення (software).

До апаратного забезпечення обчислювальних систем відносять пристрої та прилади, які утворюють апаратну конфігурацію. Це фактично базовий рівень, на якому можуть бути створені багато вищих рівнів розробленням і завантаженням відповідного програмного забезпечення.

Опис сукупності пристроїв та блоків таких систем і зв'язків між ними визначають архітектуру конкретного обчислювального пристрою. Поняття архітектури тісно пов'язане з принципами його роботи і визначає принципи дії, інформаційні зв'язки і взаємодію основних складових, а саме: процесора, внутрішньої і зовнішньої пам'яті та периферійних пристроїв. Уніфікація

архітектури обчислювальних пристроїв забезпечує їх сумісність з погляду користувача.

1.2.Принципи роботи комп'ютера за фон Нейманом


Принципи, згідно з якими функціонує більшість сучасних комп'ютерів, опубліковано в 1946 році американським математиком Джоном фон Нейманом (1903–1057).

Основні принципи архітектури комп'ютерів фон Неймана є такі:

Основні принципи архітектури комп'ютерів фон Неймана є такі:

– принцип двійкового кодування полягає у тому, що усі дані подають у вигляді двійкових кодів;

– принцип програмного керування полягає у тому, що всі операції з опрацюванням даних здійснюють відповідно до програм і ці програми розташовують у пам'яті комп'ютера;

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 140 із 153	

– принцип однорідності пам'яті полягає у тому, що всі дані, у тому числі й програми, зберігають в одному і тому самому пристрої пам'яті;

– принцип адресності полягає у такій організації пам'яті комп'ютера, за якої процесор може безпосередньо звернутись до даних, розташованих у будьякій частині пам'яті, а кожна комірка пам'яті має унікальну назву – адресу;

Незважаючи на еволюцію обчислювальної техніки і незалежно від відмінностей стосовно способів фізичної реалізації, типовий персональний комп'ютер містить у своєму складі такі основні пристрої, наведені ще у класичній роботі фон Неймана, як: пристрій керування (ПК), арифметикологічний пристрій (АЛУ), пам'ять, система введення та система виведення даних. Схематично це можна уявити так, як показано на рисунку 1:

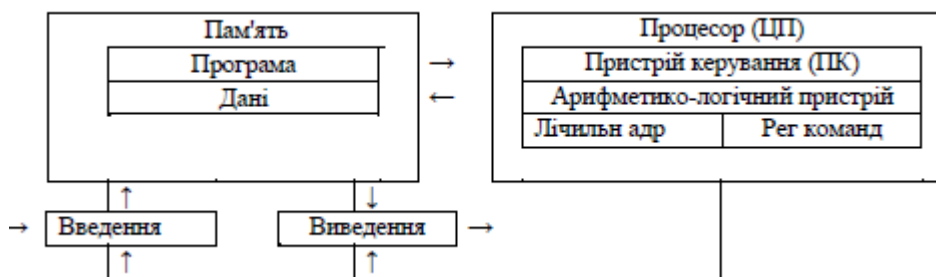


Рисунок 1. Принцип функціонування комп'ютера

Призначення цих пристроїв наступне.

1. За допомогою пристроїв введення дані і програми їх опрацювання потрапляють у пам'ять комп'ютера.
2. З пам'яті комп'ютера дані надсилають до процесора (англійською central processing unit – CPU – модуль центрального процесора), до складу якого і входять АЛУ та ПК.
3. Арифметично-логічний пристрій здійснює опрацювання даних.
4. Пристрій керування керує процесами опрацювання даних, їх збереженням і передаванням.



5. Пристрої виведення даних здійснюють подання результатів опрацювання даних у зручному для користувача вигляді.

1.3. Пам'ять комп'ютера

Пам'ять – це пристрій для збереження інформації з метою оперативного обміну нею із зовнішніми пристроями.

Розрізняють два типи пристроїв пам'яті.

Пристрої внутрішньої швидкої пам'яті – це оперативна пам'ять (ОП чи RAM – read access memory), регістри процесора та надшвидка кеш-пам'ять (cash), потрібні для зберігання опрацьовуваних даних. Вони є енергозалежними і при вимиканні живлення інформація в цих пристроях втрачається.

Пристрої пам'яті для довготривалого збереження інформації енергонезалежні, мають велику ємність, але не можуть забезпечувати швидкий доступ. Найпоширенішими сьогодні є жорсткі диски (вінчестери, HDD – hard disk driver), CD, DVD, USB-driver (“флеш-диски”), memory cards тощо.

Регістри процесора – найбільш швидкісна, але найменша за обсягом пам'ять.

Основна пам'ять підрозділяється на постійну (read only memory, ROM) і оперативну (random access memory, RAM).

Оперативна пам'ять організована у вигляді множини комірок, кожна з яких містить деяку символічну або числову інформацію і має свою унікальну адресу, як показано на рисунку 2.

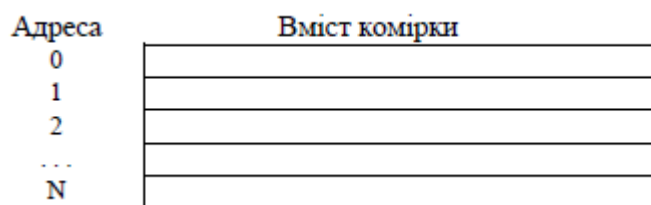



Рисунок 2. Адресація комірок пам'яті

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯНАУ НМК 07.01.07–01–2022
		Стор. 142 із 153	

Усі комірки RAM послідовно пронумеровані з нуля. Якщо j -та комірка містить число m , говорять, що «комірка з адресою j має значення m », або що її «вміст дорівнює m ». Кожна комірка неподільна і містить машинне слово, фізичний розмір якого залежить від розрядності комп'ютера. Кеш-пам'ять розташована між регістрами і процесором і призначена для узгодження швидкостей процесора і основної пам'яті.

Зовнішня пам'ять передбачає наявність накопичувача, який представляє собою пристрій для уведення-виведення інформації. Накопичувачі бувають двох різновидів – для тривалого зберігання інформації та для взаємодії із зовнішнім середовищем (клавіатура, монітор, тощо).

1.4. Системи команд


Кожний процесор має набір вбудованих так званих машинних операцій. Це елементарні дії, виконувані апаратно, реалізовані у вигляді електричних схем.

Їх кількість, перелік і структура визначені ще під час проектування комп'ютера.

Машинна команда – це наказ на виконання такої операції, формат команди – це правило її запису, а перелік усіх машинних операцій разом з форматами команд утворюють систему команд процесора.

Як правило, до системи команд входять арифметичні, пересилання та переходу. Комп'ютер працює по крокам – тактам. Такт роботи процесора – це виконання однієї машинної команди.

Якщо абстрагуватися від роботи зовнішніх пристроїв і вважати, що програма уже якимось чином занесена в комірки пам'яті, адреси комірок з даними відомі, то схема роботи процесора наступна.

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 143 із 153	

Вмикання комп'ютера приводить до запису у лічильник адрес (ЛА) адреси першої команди, розташованої в ОЗП – це завжди фіксоване значення, наприклад, 0.

Після цього ЦП починає послідовно крок за кроком виконувати команди, поки не зустрине команду зупинки.

На кожному такті будуть виконані наступні дії:

1. до реєстру команд (РК) заноситься вміст комірки, що знаходиться за адресою із ЛА;

2. значення ЛА збільшується на 1 і тепер вказує на наступну команду;

3. ПК розшифровує команду із РК і забезпечує її виконання;

Ця послідовність повторюється до команди зупинки ЦП.


1.5. Характеристика програмного забезпечення (ПЗ)

Програмне забезпечення (ПЗ) (software) – сукупність програм та службових даних, призначених для керування роботою комп'ютера. ПЗ комп'ютерів можна поділити на такі основні класи: системне ПЗ, прикладне ПЗ та системи програмування.

Системне ПЗ (system software) – програми та програмні комплекси спільні для використання усіма користувачами технічних засобів комп'ютера, застосовуються і для автоматизації розробки нових програм і для організації виконання існуючих.

До складу системного ПЗ входять:

1. Операційна система (ОС)
2. Система управління файлами (СУФ)
3. Операційне середовище
4. Утіліти
5. Системи програмування

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 144 із 153	

Операційна система (ОС) – комплекс керуючих та обробляючих програм, який виконує функцію інтерфейсу між апаратною складовою та програмами користувача, забезпечує найбільш ефективно використання ресурсів обчислювальної системи й організацію надійних обчислень. Фактично це прошарок між апаратною складовою і рештою ПЗ (функції, класифікація). СУФ призначена для організації зручного доступу до даних, організованих як файли. Файл – це іменованій фрагмент фіксованого розміру на зовнішньому носії, що зберігає, як правило, семантично пов'язану між собою інформацію. Файлова система визначає засоби збереження інформації та правила доступу до неї, регламентує формат файлу.

Для роботи з файлами конкретної файлової системи під управлінням конкретної ОС розробляється відповідна СУФ. Завдяки СУФ замість доступу на низькому рівні з вказанням фізичних адрес комірок, використовують логічний доступ з вказуванням назви файлу і номера запису в ньому.

Домашнє завдання

1. Перевести дане число з десяткової системи числення в двійкову, вісімкову і шістнадцяткову системи числення.
2. Перевести дане число в десяткову систему числення.


Варіант 1

1. а) 777; б) 305; в) 153,25; г) 162,25; д) 248,46.
2. а) 11001110112; б) 10000001112; в) 10110101,12; г) 100000110,101012; д) 671,248; е) 41A,616.

Варіант 2

1. а) 164; б) 255; в) 712,25; г) 670,25; д) 11,89.
2. а) 10011100112; б) 10010002; в) 1111100111,012; г) 1010001100,1011012; д) 413,418; е) 118,8C16.

Варіант 3

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 145 із 153	

1. а) 273; б) 661; в) 156,25; г) 797,5; д) 53,74.

2. а) 11000000002; б) 11010111112; в) 1011001101,000112; г) 1011110100,0112; д) 1017,28; е) 111,В16.

Варіант 4

1. а) 105; б) 358; в) 377,5; г) 247,25; д) 87,27.

2. а) 11000010012; б) 11001001012; в) 1111110110,012; г) 11001100,0112; д) 112,048; е) 334,А16.

Варіант 5

1. а) 500; б) 675; в) 810,25; г) 1017,25; д) 123,72.

2. а) 11010100012; б) 1000111002; в) 1101110001,0110112; г) 110011000,1110012; д) 1347,178; е) 155,6С16.

Варіант 6

1. а) 218; б) 808; в) 176,25; г) 284,25; д) 253,04.

2. а) 1110001002; б) 10110011012; в) 10110011,012; г) 1010111111,0112; д) 1665,38; е) FА,716.

Варіант 7

1. а) 306; б) 467; в) 218,5; г) 667,25; д) 318,87.

2. а) 11110001112; б) 110101012; в) 1001111010,0100012; г) 1000001111,012; д) 465,38; е) 252,3816.

Варіант 8

1. а) 167; б) 113; в) 607,5; г) 828,25; д) 314,71.

2. а) 1100100012; б) 1001000002; в) 1110011100,1112; г) 1010111010,11101112; д) 704,68; е) 367,3816.

Варіант 9


1. а) 342; б) 374; в) 164,25; г) 520,375; д) 97,14.

2. а) 10001101102; б) 1111000012; в) 1110010100,10110012; г) 1000000110,001012; д) 666,168; е) 1С7,6816.

Варіант 10

1. а) 524; б) 222; в) 579,5; г) 847,625; д) 53,35.

2. а) 1011111112; б) 11111001102; в) 10011000,11010112; г) 1110001101,10012; д) 140,228; е) 1DE,5416.

	<p align="center">Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 146 із 153	

Варіант 11

1. а) 113; б) 875; в) 535,1875; г) 649,25; д) 6,52.
2. а) 111010002; б) 10100011112; в) 1101101000,012; г) 1000000101,010112; д) 1600,14; е) 1E9,416.

Варіант 12

1. а) 294; б) 723; в) 950,25; г) 976,625; д) 282,73.
2. а) 100000110012; б) 101011002; в) 1101100,012; г) 1110001100,12; д) 1053,28; е) 200,616.

Варіант 13

1. а) 617; б) 597; в) 412,25; г) 545,25; д) 84,82.
2. а) 1101111012; б) 11100111012; в) 111001000,012; г) 1100111001,10012; д) 1471,178; е) 3EC,516.

Вариант 14

1. а) 1047; б) 335; в) 814,5; г) 518,625; д) 198,91.
2. а) 11011000002; б) 1000010102; в) 1011010101,12; г) 1010011111,11012; д) 452,638; е) 1E7,0816.

Варіант 15

1. а) 887; б) 233; в) 801,5; г) 936,3125; д) 218,73.
2. а) 10101000012; б) 100000101012; в) 1011110000,1001012; г) 1000110001,10112; д) 1034,348; е) 72,616.

Варіант 16


1. а) 969; б) 549; в) 973,375; г) 508,5; д) 281,09.
2. а) 101000102; б) 11100101112; в) 110010010,1012; г) 1111011100,100112; д) 605,028; е) 3C8,816.

Варіант 17

1. а) 163; б) 566; в) 694,375; г) 352,375; д) 288,61.
2. а) 10011010012; б) 1100111012; в) 1000001101,012; г) 1010001001,110112; д) 247,18; е) 81,416.

Варіант 18

1. а) 917; б) 477; в) 74,5; г) 792,25; д) 84,33.

	<p align="center">Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 147 із 153	

2. а) 11100111002; б) 11111011112; в) 111110100,1012; г) 110011110,10000112; д) 1446,628; е) 9C,D16.

Варіант 19

1. а) 477; б) 182; в) 863,25; г) 882,25; д) 75,2.

2. а) 1010111002; б) 10000100112; в) 11100011,12; г) 100101010,000112; д) 1762,78; е) 1B5,616.

Варіант 20

1. а) 804; б) 157; в) 207,625; г) 435,375; д) 30,43.

2. а) 100100002; б) 110010102; в) 1110101100,10112; г) 110110101,101112; д) 1164,368; е) 1D5,C816.

Варіант 21

1. а) 753; б) 404; в) 111,1875; г) 907,0625; д) 62,88.

2. а) 111000112; б) 11110011112; в) 1011111111,010012; г) 1001011101,0112; д) 615,728; е) 3DA,516.

Варіант 22

1. а) 571; б) 556; в) 696,25; г) 580,375; д) 106,67.

2. а) 1100110102; б) 1110010102; в) 1000010011,001012; г) 11010110,000012; д) 1343,668; е) 3C3,616.

Варіант 23

1. а) 244; б) 581; в) 351,6875; г) 1027,375; д) 151,44.

2. а) 10011001112; б) 11000100102; в) 1100110010,11012; г) 1001011,01012; д) 171,38; е) 3A3,416.

Варіант 24


1. а) 388; б) 280; в) 833,5625; г) 674,25; д) 159,05.

2. а) 110011112; б) 1010011012; в) 101001101,0010012; г) 100101011,1012; д) 750,518; е) 90,816.

Варіант 25

1. а) 386; б) 608; в) 398,6875; г) 270,25; д) 317,32.

2. а) 110000012; б) 11111111102; в) 1110100010,101012; г) 1001011001,0112; д) 1335,28; е) 18F,816.

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯНАУ НМК 07.01.07–01–2022
		Стор. 148 із 153	

Варіант 26

1. а) 76; б) 279; в) 572,25; г) 477,375; д) 184,97.
2. а) 10011011112; б) 10110110002; в) 1110100,00112; г) 1000001010,010012; д) 1234,28; е) 1DD,216.

Варіант 27

1. а) 1003; б) 780; в) 74,375; г) 204,25; д) 241,39.
2. а) 10100012; б) 110011012; в) 1010101000,1012; г) 110011001,012; д) 1031,58; е) 158,2416.

Варіант 28

1. а) 262; б) 414; в) 330,5; г) 541,6875; д) 115,41.
2. а) 10010110012; б) 10001012; в) 11101111,1012; г) 111100011,12; д) 150,448; е) 377,716.

Варіант 29

1. а) 775; б) 523; в) 432,25; г) 158,3125; д) 1,09.
2. а) 1011101102; б) 10100102; в) 1001100,1100112; г) 1001000111,100112; д) 236,638; е) 148,616.


Варіант 30

1. а) 149; б) 93; в) 463,6875; г) 184,75; д) 61,52.
2. а) 11001101012; б) 1000010002; в) 1010100111,012; г) 111111001,10112; д) 1636,248; е) C7,7816.

Рекомендована література

Базова література

1. Програмування мікроконтролерів AVR: [навчальний посібник] / С. М. Цирульник, О. Д. Азаров, Л. В. Крупельницький, Т. І. Трояновська. – Вінниця : ВНТУ, 2018. – 111 с. [Електронний ресурс]. – Режим доступу: URL :http://pdf.lib.vntu.edu.ua/books/IRVC/2021/Tsirulnik_2018_111.pdf
2. Програмування для мікропроцесорних систем. Навчальний посібник до виконання лабораторних робіт [Електронний ресурс]: навчальний посібник для здобувачів ступеня магістра за освітньою програмою «Управління, захист та

	<p>Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»</p>	Шифр документа	СМЯНАУ НМК 07.01.07–01–2022
		Стор. 149 із 153	

- автоматизація енергосистем» спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» / КПІ ім. Ігоря Сікорського ; уклад.: О. В. Тимохін, А. О. Тимохіна. -Електронні текстові дані (1 файл: 819.13 Кбайт). - Київ : КПІ ім. Ігоря Сікорського, 2023. - 55 с.
3. Мирончук, О. Ю. Проектування мікроконтролерних пристроїв в радіотехнічних комп'ютеризованих системах. Лабораторний практикум [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за освітніми програмами «Радіотехнічні комп'ютеризовані системи», «Інформаційна та комунікаційна радіоінженерія», «Інтелектуальні технології радіоелектронної техніки» спеціальності 172 «Електронні комунікації та радіотехніка» / О. Ю. Мирончук, О. О. Шпилька ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 2,02 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2023. – 53 с.
 4. Новацький, А. О. Мікропроцесорні та мікроконтролерні системи. Частина 2. Проектування мікропроцесорних систем [Електронний ресурс] : підручник для студентів освітньої програми «Інтегровані інформаційні системи» за спеціальністю 126 «Інформаційні системи та технології» / А. О. Новацький ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 16,34 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 462 с.
 5. Новацький, А. О. Мікропроцесорні та мікроконтролерні системи. Частина 1. Мікропроцесорні системи [Електронний ресурс] : підручник / А. О. Новацький ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 9,73 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 361 с.
 6. Основи проектування систем Інтернету речей. Периферія мікроконтролерів STM32: конспект лекцій [Електронний ресурс] : Навчальний посібник призначено для здобувачів ступеня бакалавра за сертифікатною програмою «Електронні охоронні системи та засоби Інтернету речей» спеціальності 171

	Система менеджменту якості. НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС навчальної дисципліни «Обчислювальна техніка та алгоритмічні мови»	Шифр документа	СМЯ НАУ НМК 07.01.07–01–2022
		Стор. 150 із 153	

«Електроніка» / КПІ ім. Ігоря Сікорського ; уклад. : Ю. О. Оникієнко, А. Р. Рижова. – Електронні текстові дані (1 файл: 4.29 МБ). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 127 с.

Допоміжна література.

1. ATtiny2313A/ATtiny4313 DATASHEET – Atmel Corporation. – 274 с.
2. Новацький, А. О. Мікропроцесорні та мікроконтролерні системи. Частина 2. Проектування мікропроцесорних систем. Лабораторний практикум [Електронний ресурс] : навчальний посібник для студентів освітньої програми «Інтегровані інформаційні системи» спеціальності 126 «Інформаційні системи та технології» / А. О. Новацький ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 12,5 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 268 с.
3. Програмування для мікропроцесорних систем. Навчальний посібник до виконання модульної контрольної роботи [Електронний ресурс] : навчальний посібник для здобувачів ступеня магістра за освітньою програмою «Управління, захист та автоматизація енергосистем» спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» / КПІ ім. Ігоря Сікорського ; уклад.: О. В. Тимохін, А. О. Тимохіна. - Електронні текстові дані (1 файл: 1.02 Мбайт). - Київ : КПІ ім. Ігоря Сікорського, 2023. - 21 с.
4. Мікропроцесори та мікроконтролери. Курс лекцій [Електронний ресурс] : навчальний посібник для студентів спеціальності 153 «Мікро- та наносистемна техніка», освітньої програми «Мікро- та наноелектроніка» / КПІ ім. Ігоря Сікорського ; уклад.: Д. Д. Татарчук, Ю. В. Діденко. – Електронні текстові дані (1 файл: 19,5 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 238 с.



Система менеджменту якості.
НАВЧАЛЬНО-МЕТОДИЧНИЙ КОМПЛЕКС
навчальної дисципліни
«Обчислювальна техніка та алгоритмічні
мови»

Шифр
документа

СМЯ НАУ
НМК 07.01.07–01–2022

Стор. 152 із 153

--	--	--	--	--

