

ЗАТВЕРДЖУЮ
Директор ЗПО «ДПТК»
Олександр СТРИЛЕЦЬ
31.08.2026 р.
ПОГОДЖЕНО
Заступник директора з навчально-методичної
роботи ЗПО «ДПТК»
Сергій ГОРДІЄНКО
31.08.2026 р.

КОНСПЕКТ ЛЕКЦІЙ З ДИСЦИПЛІНИ: «ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНЖЕНЕРНИХ РОЗРАХУНКІВ»

ВСТУП

Вивчення здобувачами освіти дисципліни «Програмне забезпечення інженерних розрахунків» має на меті поєднати теоретичний і практичний навчальний матеріал з математики, фізики, інженерної та комп'ютерної графіки з ефективним застосуванням комп'ютера для використання цих знань і вмінь у фахових дисциплінах.

Дисципліна «Програмне забезпечення інженерних розрахунків» входить до циклу загальної підготовки навчальних дисциплін обов'язкової підготовки згідно з навчальним планом підготовки фахових молодших бакалаврів зі спеціальності **G19 «Будівництво та цивільна інженерія»** освітньо-професійної програми «Будівництво та експлуатація будівель і споруд».

Згідно з освітньо-професійною програмою дисципліна «Програмне забезпечення інженерних розрахунків» формує у здобувачів освіти відповідні здатності:

- застосовувати типові аналітичні методи та комп'ютерні програмні засоби для розв'язування інженерних завдань у галузі будівництва та цивільної інженерії, ефективні кількісні методи математики, фізики, інженерних наук, а також відповідне комп'ютерне програмне забезпечення для розв'язування типових спеціалізованих задач;
- виконувати інженерні розрахунки на ПЕОМ з використанням алгоритмічної мови високого рівня (Фортран, ін.) та сучасних прикладних програмних засобів (Excel, MathCAD, VBA).

Після засвоєння навчальної дисципліни «Програмне забезпечення інженерних розрахунків» здобувачі освіти мають продемонструвати такі результати навчання:

- здійснювати інженерні розрахунки для вирішення типових спеціалізованих задач і практичних проблем у галузі будівництва та цивільної інженерії;
- аналізувати інженерні об'єкти, процеси та методи;
- читати, аналізувати, редагувати вихідний програмний код, складати програми для інженерних розрахунків на ПЕОМ, користуючись алгоритмічною мовою високого рівня (Фортран, ін.) та відповідними компіляторами, отримувати з вихідного програмного коду об'єктний файл і файл, що виконується, та результати його роботи.

Дисципліні «Програмне забезпечення інженерних розрахунків» передують навчальні дисципліни «Інженерна та комп'ютерна графіка», «Вища математика», «Фізика», «Хімія». Кредитний модуль «Програмне забезпечення інженерних розрахунків» забезпечує такі дисципліни, як «Основи розрахунку будівельних конструкцій», «Будівельні конструкції», «Технологія і організація будівельного виробництва».

Лекції з дисципліни «Програмне забезпечення інженерних розрахунків» проводяться для підготовки фахових молодших бакалаврів зі спеціальності **G19 «Будівництво та цивільна інженерія»** освітньо-професійної програми **«Будівництво та експлуатація будівель і споруд»**.

У практичній діяльності фахівців з будівництва та цивільної інженерії все більш необхідними стають компетентності щодо забезпечення інженерних розрахунків. Вони реалізуються при обробці експериментальних даних, визначенні міцності та жорсткості будівельних конструкцій, при розв'язанні багатьох інших інженерних завдань сучасної будівельної галузі.

Тому основна мета лекцій з дисципліни «Програмне забезпечення інженерних розрахунків» — ознайомити здобувачів освіти з теоретичними положеннями та можливостями розповсюдженого і затребуваного базового програмного забезпечення, як-от прикладний пакет Excel, програма MathCAD, мови VBA і Фортран, середовища IDE.

У процесі вивчення дисципліни «Програмне забезпечення інженерних розрахунків» здобувач освіти отримує базові знання з інформатики, інженерних розрахунків на ПЕОМ, правил роботи з комп'ютером, основ програмування; зможе виконувати набір текстів, вставки об'єктів, будувати графіки, таблиці та діаграми в редакторі Excel, виконувати обчислення та оцінювати отримані результати в програмних середовищах MathCAD, VBA, MSDev, FPS або аналогічних.

Рейтинг здобувача освіти з дисципліни «Програмне забезпечення інженерних розрахунків» складається з балів, що він отримує за: виконання практичних робіт, тестування та модульний контроль; семестровим контролем є **диференційований залік**.

ЛЕКЦІЯ 1

Структурування даних – інформація, одиниці вимірювання даних та одиниці зберігання даних

Дисципліна «Програмне забезпечення інженерних розрахунків» є складовою початковим курсом для опанування системи дисциплін з інформаційних технологій для фахівця будівельної галузі, в основі якої лежить інформатика. «Інформатика» походить від французьких слів *information* (інформація) і *automatique* (автоматика) і дослівно означає «інформаційна автоматика». Англomовний варіант цього терміну – «Computer science», що означає «комп'ютерна наука», охоплює такі галузі наукової діяльності:

– розробка, створення, використання та обслуговування систем обробки інформації, (включаючи комп'ютери та їх програмне забезпечення);

– організаційні, комерційні, адміністративні та соціально-політичні проблеми пов'язані з комп'ютеризацією; тобто як масове впровадження комп'ютерної техніки впливає на всі сфери життя людей.

Основні напрямки розвитку «комп'ютерних наук»:

1. розробка обчислювальних систем і програмного забезпечення;
2. теорія інформації, що вивчає процеси, пов'язані з передачею, прийомом, перетворенням і зберіганням інформації;
3. методи штучного інтелекту, що дозволяють створювати програми для вирішення завдань, що вимагають певних інтелектуальних зусиль при виконанні їх людиною (логічний висновок, навчання, розуміння мови, візуальне сприйняття, ігри та ін);
4. системний аналіз, що полягає в аналізі призначення проектованої системи і у встановленні вимог, яким вона повинна відповідати;
5. методи машинної графіки, анімації, засоби мультимедіа;
6. засоби телекомунікації, в тому числі, глобальні комп'ютерні мережі, об'єднуючі все людство в єдине інформаційне співтовариство;
7. різноманітні прикладання, що охоплюють виробництво, науку, освіту, медицину, торгівлю, сільське господарство і всі інші види господарської та громадської діяльності.

Отже маємо суперпозицію з двох частин: технічні засоби та програмні засоби.

1. Технічні засоби, тобто апаратура комп'ютерів, в англійській мові позначаються словом Hardware, яке буквально перекладається як «тверді вироби», або в побуті – «залізо».
2. А для програмних засобів вибрано дуже вдале слово Software (в перекладі з англійської мови буквально – «м'які вироби»), яке підкреслює рівнозначність програмного забезпечення та самої машини, і разом з тим підкреслює здатність програмного забезпечення модифікуватися, пристосовуватися, розвиватися.

Крім цих двох частин можна помітити ще один елемент – алгоритмічні засоби. Для нього використовується назва «Brainware» (від англ. «Brain» – інтелект). Ця гілка пов'язана з розробкою алгоритмів і вивченням методів та прийомів їхньої побудови.

В термін «інформація» вкладають різний зміст у техніці, науці і в життєвих ситуаціях. Його не можна пояснити однією фразою.

У побуті ми називаємо інформацією будь-які дані або відомості, які когось цікавлять. Наприклад, повідомлення про які-небудь події, про чийсь діяльність і т.п. «Інформувати» в цьому сенсі означає "повідомити щось, невідоме раніше". Інформація – це набір різних даних у вигляді малюнків, схем, словесного опису і т.д. про об'єкти і явища, їх параметрах, властивості і стан. Важливо, щоб ці дані могли бути сприйняті суб'єктами життєдіяльності (живими організмами, управляючими машинами та ін.).

Одне і те ж інформаційне повідомлення (стаття в газеті, оголошення, лист, телеграма, довідка, розповідь, креслення, радіопередача і т.п.) може містити різну кількість інформації для різних людей – в залежності від їх попередніх знань, від рівня розуміння цього повідомлення і інтересу до нього. В принципі, деякий набір даних взагалі може не являти собою ніякої інформації. Так, повідомлення, складене іноземною мовою, не несе ніякої інформації людині, яка

не знає цієї мови. Ніякої інформації не містить і повідомлення знайомою мовою, якщо його зміст незрозумілий або вже відомо.

Тобто можна сказати, що інформація може бути, або може бути відсутньою або наявною (відповідно крайні випадки – «0» або «1»).

Тобто для того, щоб дані стали інформацією, суб'єкт інформаційного обміну повинен сприймати адекватно отримані відомості. В іншому випадку можна говорити про те, що інформація відсутня.

Звідси випливає важливий висновок про те, що інформація не є характеристикою даних або повідомлення, а являє собою продукт взаємодії повідомленням і його споживачем (суб'єктивність і об'єктивність інформації). Без наявності споживача, говорити про інформацію безглуздо.

У випадках, коли говорять про автоматизовану роботу з інформацією за допомогою будь-яких технічних пристроїв, зазвичай в першу чергу цікавляться не змістом повідомлення, а тим, скільки символів це повідомлення містить.

Якщо мова йде про комп'ютерну техніку, то під інформацією розуміють деяку послідовність символічних позначень (букв, цифр, закодованих графічних образів і звуків і т.п.), в зрозумілому для комп'ютера вигляді. Якщо при цьому набір зазначених символів є випадковим і не має смислового навантаження, то він не є інформацією для комп'ютерних засобів.

Інформація в техніці може існувати в найрізноманітніших формах:

- у вигляді текстів, малюнків, креслень, фотографій;
- у вигляді світлових або звукових сигналів;
- у вигляді радіохвиль;
- у вигляді електричних та нервових імпульсів;
- у вигляді магнітних записів.

Необхідно відзначити, що властивостями інформації володіють як матеріальні предмети, процеси і явища, так і нематеріальні (поняття, характеристики, слова і т.п.) Всі вказані параметри називаються інформаційними об'єктами.

Інформація являє собою деяке повідомлення (набір даних). Це повідомлення передається у вигляді деяких сигналів (звукових, електричних, світлових) від деякого джерела інформації до її приймача по деякому каналу зв'язку між ними. Як правило, повідомлення кодується джерелом в сигнал, що передається. Цей сигнал посиляється по каналу зв'язку. Приймач приймає сигнал, який декодується і стає прийнятим повідомленням.

Приклади:

1. Повідомлення, що містить інформацію про погоду, передається приймачу (телеглядачеві) від джерела – фахівця-метеоролога допомогою каналу зв'язку - телевізійної передавальної апаратури і телевізора;
2. Приймач інформації (телеглядач) своїми органами почуттів (око, вухо) сприймає інформацію із зовнішнього світу, переробляє її в певну послідовність нервових імпульсів, передає імпульси по нервових волокнах, зберігає в пам'яті у вигляді стану нейронних структур мозку, відтворює у вигляді звукових сигналів, рухів і т.п., використовує в процесі своєї життєдіяльності.
3. Звичайний розмова між співрозмовниками (інформаційний обмін) відбувається за тими ж принципами. Якщо один зі співрозмовників (джерело інформації) хоче розповісти про щось слухачеві (приймач

інформації), він кодує це поняття певним поєднанням звукових сигналів (словом).

Дії, які можна виконувати з інформацією: поширювати; передавати; сприймати; перетворювати; комбінувати; використовувати; запам'ятовувати; обробляти; ділити на частини; приймати; копіювати; спрощувати; збирати; зберігати; шукати; вимірювати; руйнувати та ін.

Властивості інформації: достовірність; повнота; цінність; своєчасність; зрозумілість; доступність; стислість та ін.

Необхідно ввести деякі кількісні характеристики поняття «інформація». Отже, питання: чи можна кількісно виміряти інформацію? Для чого це потрібно? Справа в тому, що в багатьох практичних випадках якісними особливостями інформації можна знехтувати.

Приклад:

Необхідно передати дані про студентів першого курсу, присутніх на лекції, в деканат. Як це зробити?

1 спосіб. Викладач може сфотографувати аудиторію і передати це зображення по комп'ютерній мережі на сервер деканату;

2 спосіб. Викладач може написати список студентів, може зазначити тільки присутніх або тільки відсутніх і т.п.

Результат може бути один: деканат зробить висновок з отриманої інформації і, наприклад, зніме стипендію відсутнім на лекції.

Тобто зі сказаного випливає, що найважливішим в інформаційному процесі є отримання ВИСНОВКУ: присутній – не присутній.

В якості одиниці інформації умовилися прийняти один біт (з англ. bit – «binary, digit» – двійкова цифра).

Один біт виражається саме в висновку: присутній – не присутній. Прийнято умовно виражати біти у цілих числах двійкової системи числення: 0 або 1 («ні» або «так»). Тобто 1 біт – це «розрахунок до двох». Він займає один розряд.

Два біта, тобто два розряди двійкової системи числення в різних поєднаннях, це розрахунок до 4. Три біта – до 8.

Якщо присвоїти кожному з поєднань цифр яке-небудь поняття, то можна закодувати певну інформацію.

Отже, при вживанні двійкового кодування використовується поняття розрядності кодування: використання двох цифр відповідає двохраньковому кодуванню, використання трьох цифр – трьоххраньковому. Відзначимо особливість впливу розрядності кодування на кількість понять, які можна закодувати: збільшення на одиницю розрядності (або що те ж саме – кількості бітів) подвоює кількість кодованих понять.

Загальна формула, яка виражає співвідношення між кодованими поняттями і розрядністю кодування (тобто кількістю бітів, які використовуються для кодування), має вигляд:

$$N = 2^m$$

де N – кількість незалежних кодованих значень (стільки понять можна закодувати різними бітами);

m – розрядність двійкового кодування (кількість бітів).

Біт – занадто дрібна одиниця виміру. На практиці частіше застосовується більш велика одиниця – байт, 1 байт = 8 бітам. В деяких випадках саме вісім бітів потрібно для того, щоб закодувати будь-який з 256 символів «алфавіту комп'ютера».

Широко також використовуються інші похідні одиниці інформації:

- 1 Кілобайт (кбайт) = 1024 байт,
- 1 Мегабайт (Мбайт) = 1024 Кбайт,
- 1 Гігабайт (ГБ) = 1024 Мбайт.

Останнім часом у зв'язку зі збільшенням обсягів оброблюваної інформації входять у вживання такі похідні одиниці, як:

- 1 Терабайт (Тбайт) = 1024 Гбайт,
- 1 Петабайт (Пбайт) = 1024 Тбайт.

Основою системи збереження інформації в комп'ютері є спеціальні пристрої. Їхня історична назва – диски. На комп'ютері їх може бути декілька. Кожен диск має назву (ім'я диску). Ім'я диску складається з латинської букви і двокрапки. Наприклад: С: або А:. Будь-який диск (незалежно від його фізичної суті) для користувача є сховищем інформації, до якого він звертається за ім'ям.

Інформація в комп'ютері зберігається у вигляді файлів. Файл – це сукупність даних, яка зберігається на дисках та має ім'я.

Ім'я файлу складається з двох частин. Перша частина – це безпосереднє ім'я файлу. Друга – розширення імені, яке вказує на тип файлу. Ім'я файлу може бути довжиною до 256 символів і складатися з довільної послідовності латинських літер, літер кирилиці, цифр, знаків підкреслення, тире, пробілу та таке інше. Існують символи, які не можна включати до імені файлу: * : " ? / \ < >.

Розширення імені може бути довжиною до 4 символів (зазвичай – 3) вказує на тип файлу. Тип файлу, в свою чергу, вказує операційній системі, яким чином треба опрацьовувати цей файл. Основні типи файлів наведені в табл. 1. Система збереження інформації передбачає її структурування у папках на диску. В папках можуть розташовуватись файли, а також інші папки. Розміщення одних папок усередині інших утворюють ієрархію («дерево») системи збереження інформації на комп'ютері.

Таблиця 1 – Основні типи файлів

Розширення імені файлу (тип файлу)	Призначення
.txt	Текстовий файл. Відкривається програмою Блокнот (Notepad).
.jpg, .jpeg, .bmp	Файли малюнків.
.doc, .docx	Текстовий файл. Відкривається програмою Microsoft Word.
.xls, .xlsx	Файл електронних таблиць. Відкривається програмою Microsoft Excel.

Розширення імені файлу (тип файлу)	Призначення
.ppt	Файл презентації. Відкривається програмою Microsoft PowerPoint.
.exe, .com	Файли програм. Запускаються на виконання.
.zip, .rar	Архівні файли. Зберігають інші файли у стислому стані.
.html	Файл Web-сторінки. Переглядається у вікні Web-оглядача.

ЛЕКЦІЯ 2

Алгоритми. Властивості алгоритмів. Типові структури алгоритмів

Алгоритми – це правила, що пропонують виконання послідовностей дій, що призводять до розв'язання завдання.

Не обов'язково алгоритм – це математика. Це перелік команд або інструкцій, за якими потрібно виконати деяку послідовність дій.

Приклад:

Алгоритм переходу вулиці: 1) «подивися наліво, переконайся, що ліворуч немає транспорту»; 2) «переходи вулицю»; 3) «дійшовши до середини, подивися направо, переконайся, що праворуч немає транспорту»; 4) «продовжуй переходити вулицю».

Отже, алгоритм – це послідовна система команд, яка адресована виконавцю чітко і однозначно задавати процес вирішення завдання.

Одиницею вимірювання в алгоритмі є крок. Крок алгоритму є окремою закінченою дією. Як правило алгоритм складається з декількох кроків.

Властивості алгоритмів:

1. **Дискретність** — процес розв'язку розбивається на кроки. Кожен крок — це одна дія або підпорядкований алгоритм (метод покрокової деталізації). Таким чином полегшується процес знайдення помилок і редагування алгоритму.
2. **Визначеність (точність)** — кожен крок алгоритму має бути однозначно описаною дією і не містити двозначностей.
3. **Зрозумілість** — усі дії, включені до алгоритму, мають бути у межах компетенції виконавця алгоритму.
4. **Універсальність (масовість)** — алгоритм має виконуватися при будь-яких значеннях вхідних даних та початкових умов.
5. **Скінченність** — алгоритм має бути реалізований за скінченне число кроків і повинен користуватися кінцевим набором вхідних значень.
6. **Результативність** — алгоритм має якнайскоріше привести до отримання результату.

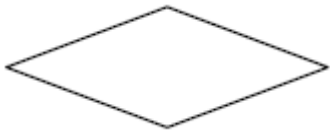
Якщо алгоритму притаманні перелічені вище властивості, то його виконання провадиться формально, тобто точно за схемою алгоритму, без будь-яких втручань у послідовність дій та у самі дії.

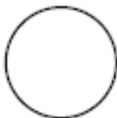
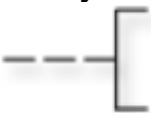
Розрізняють такі види алгоритмів:

1. **Лінійні алгоритми** – складаються з декількох кроків, які виконуються послідовно один за одним.
2. **Розгалужений алгоритм** – має варіанти вибору умови, яка виконується. В такому алгоритмі виконання кроків змінюється від певних умов.
3. **Циклічні алгоритми** – це коли певна дія або група дій виконується декілька разів, циклічне повторення алгоритму декілька разів.

До етапу розроблення програм за допомогою любых мов програмування основним підготувальним етапом являється створення блок-схем за алгоритмом. Блок-схема алгоритму – це графічне представлення логічної структури алгоритму, де кожний етап обробки інформації зображується у вигляді геометричних символів – блоків (представлено у табл. 1).

Таблиця 1 – Графічне зображення блоків у блок-схемах і їх призначення

Графічне зображення блоку	Призначення
Овал 	Початок та кінець алгоритму
Паралелограм 	Блок вводу вхідних даних та виведення результату. Цей символ не визначає носія даних
Прямокутник 	Виконання однієї або кількох операцій, обробка даних. Всередині фігури записують безпосередньо самі операції.
Ромб 	Блок умови. В цьому блоці пишеться умова, в залежності від якої вибираються напрямки дії алгоритму. На лініях, що розгалужуються від блоку, пишеться „Так” або „Ні” – залежно від виконання записаної умови.
Шестикутник (початок/кінець циклу)	Початок і кінець циклу. Межі змінної циклу і крок її збільшення записуються всередині символу початку, а в символі кінця циклу – змінна циклу. Операції, що виконуються всередині циклу, розміщуються між ними.

Графічне зображення блоку	Призначення
	
Прямокутник із подвійними боками 	Виконання процесу, що складається з операцій, які визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і дані, які передані в нього.
Коло 	З'єднання. Відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці. Всередині ставиться позначка переходу.
Коментар (пунктирна лінія з дужкою) 	Коментар. Використовується для інформації про процес, коли текст в символі перевищує його обсяг.

Існують правила зображення блок-схем алгоритмів. Кожен алгоритм має свої особливості побудови.

Приклад: Фрагмент блок-схеми для розгалуженого алгоритму представлений на рисунку 1, якщо задана умова для функції.

$$y = \begin{cases} x, & \text{якщо } x \geq 0 \\ -x, & \text{якщо } x < 0 \end{cases}$$

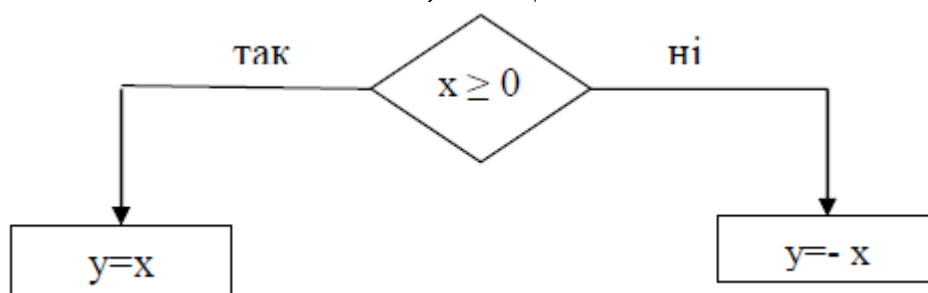


Рис. 1 – Фрагмент блок-схеми розгалуженого алгоритму

Приклад: Блок-схема для лінійного алгоритму, якщо задано знайти суму S з трьох доданків a, b, c (Рис. 2).

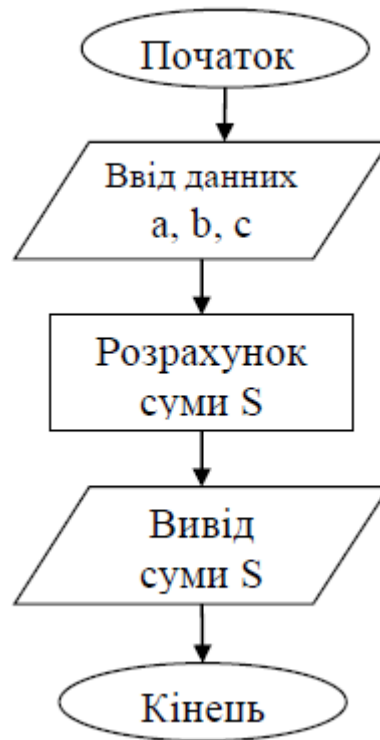


Рис. 2 – Блок-схема для лінійного алгоритму знаходження суми S
Цикл схематично зображується на блок-схемах одним з способів (Рис. 3)



Рис. 3 – Блок-схеми для циклічного алгоритму:
а) цикл з передумовою; б) цикл з постумовою.

Повний текст конспекту лекцій з дисципліни «Програмне забезпечення інженерних розрахунків» доступний зареєстрованим здобувачам освіти на внутрішньому репозиторії ЗПО «ДПТК».