**Тема 2. Етапи розв’язання задач на ЕОМ. Основи алгоритмізації. Основні поняття про життєвий цикл програмного забезпечення.**

**Лабораторна робота 2**.

**Тема : Запис алгоритмів за допомогою псевдокоду**

**Мета :**

1. Зрозуміти концепцію псевдокоду та його використання в процесі розробки алгоритмів.
2. Розвинути навички запису алгоритмів використовуючи псевдокод.
3. Оволодіти методами перевірки та оптимізації псевдокоду.
4. Підвищити здатність абстрактного та логічного мислення через процес побудови алгоритмів.

Основне завдання цієї лабораторної роботи - вивчити, як ефективно використовувати псевдокод для планування, опису та візуалізації алгоритмів перед їх реалізацією на конкретній мові програмування.

**Теоретична частина**

Псевдокод - це спрощений мова програмування, яка використовується для запису алгоритмів та ідеї за їхньої розробки. Він не має строгого синтаксису, як код на конкретній мові програмування, і в основному зрозумілий для людини.

Інженери часто використовують псевдокод в якості проміжного етапу в програмуванні-між стадією планування і стадією написання працюючого коду. Хороший псевдокод може перетворитися в коментарі до фінальної версії програми і буде допомагати програмісту виправляти помилки в майбутньому або коригувати код.

Пам'ятайте, що псевдокод суб'єктивний і не стандартизований.у ньому немає чітко встановленого синтаксису-є тільки негласне правило використовувати стандартні конструкції, які інші програмісти могли б зрозуміти без особливих проблем. Якщо ви пишете код самостійно, псевдокод допоможе вам упорядкувати думки і розробити план. Якщо ви працюєте разом з іншими інженерами (і не має значення, який їх рівень володіння предметом), важливо використовувати як мінімум основні конструкції, щоб всі інші розуміли, що ви хотіли зробити.

**Процес запису псевдокоду складається з декількох кроків:**

1. **Визначте основні етапи алгоритму:** Подумайте про основні кроки, необхідні для виконання вашого алгоритму, та запишіть їх у вигляді списку.
2. **Почніть із запису псевдокоду:** Почніть записувати кожен етап використовуючи псевдокод. Ви можете використовувати просту мову та терміни, що зазвичай використовуються в програмуванні, наприклад, "if" (якщо), "for" (для), "while" (поки), "function" (функція) і так далі.
3. **Перевірте псевдокод:** Перевірте свій псевдокод, щоб впевнитися, що він виконує правильні кроки в правильному порядку. Це допоможе вам виявити будь-які помилки або недоліки в вашому алгоритмі.
4. **Оптимізуйте псевдокод:** Подумайте, чи можна поліпшити або оптимізувати ваш алгоритм. Чи є кроки, які можна об'єднати або упростити?

**Приклад псевдокоду**

Розберіть приклад простої програми. Уявіть, що програма повинна замінювати поєднання букв " foo " в текстовому файлі. Програма прочитає кожен рядок в цьому файлі, пошукає потрібне поєднання в кожному рядку і замінить його на інше. Повторювані кроки починаються з прогалин - в ідеалі так повинно бути і в цьому коді. Початковий начерк псевдокода може виглядати так:

* відкрити файл
* У кожному рядку файлу:
* знайти поєднання
* видалити поєднання
* вставити інше поєднання
* закрити файл

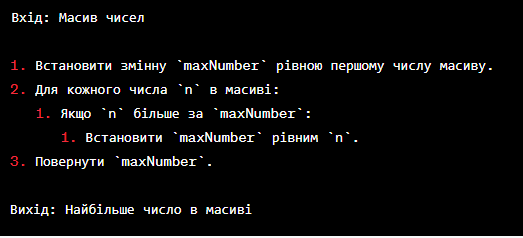
**Пишіть псевдокод ітераційно**: напишіть його один раз, а потім міняйте в ньому дані . Один з плюсів псевдокода полягає в тому, що ви можете накидати тільки основу, а все складне залишити на потім. Зверніть увагу на те, що в прикладі, наведеному вище, немає вказівок на те, яким повинно буде поєднання букв. Ви як програміст можете переписати псевдокод, щоб в ньому були присутні алгоритми для видалення окремих букв і заміни їх іншими. Другий начерк може мати наступний вигляд:

* відкрити файл
* У кожному рядку файлу:
* знайти слово наступним чином:
* прочитати символ у рядку
  + Якщо символ збігається, то:
* Якщо всі з наступних символів збігаються
* то це правильний вибір
* видалити символи слова
* вставити символи нового слова
* закрити файл

**Приклад псевдокоду 2**

Давайте розглянемо завдання: "Розробіть алгоритм, який знаходить найбільше число в масиві чисел".

Псевдокод для цього алгоритму може виглядати так:



Цей псевдокод пройшов через основні етапи: визначення алгоритму (знайти найбільше число), запис алгоритму в псевдокоді, перевірка алгоритму (переконатися, що він відповідає завданню) та оптимізація (простота та читабельність). Його можна легко перетворити на реальний код на більшості мов програмування.

**Основні структурні елементи псевдокоду включають:**

Вхідні дані: Визначення даних, які будуть використовуватися алгоритмом. Зазвичай вказується на початку псевдокоду.

Вхід: a, b

Вихідні дані: Що саме має вивести алгоритм або повернути як результат.

Вихід: c

Присвоєння: Встановлення значення змінної.

c = a + b

Умовні оператори: Умовні переходи типу if, else, else if.

якщо a > b то

вивести "a більше b"

інакше

вивести "b більше або рівне a"

Цикли: Оператори циклу, такі як for, while.

для i = 1 до n

вивести i

Функції та процедури: Виклик підпрограм або функцій.

Функція find\_max(a, b)

якщо a > b то

повернути a

інакше

повернути b

Коментарі: Іноді в псевдокоді використовують коментарі для пояснення деяких кроків

// Це коментар

Операції арифметичні, логічні, порівняння: Ці оператори допомагають у математичних та логічних операціях.

= a && b // логічна e = a == b // порівняння

Блоки коду: Зазвичай в псевдокоді блоки коду виділяють відступами або ключовими словами, такими як початок і кінець, для групування декількох інструкцій.

Масиви та списки: Часто використовуються для структурування даних.

список = [1, 2, 3, 4]

Ці елементи можуть комбінуватися для формування більш складних структур і алгоритмів.

**Практична частина**

**Завдання 1**: Напишіть псевдокод для алгоритму, який визначає, чи є число простим.

**Завдання 2**: Сформулюйте псевдокод для алгоритму, який знаходить найбільший спільний дільник двох чисел.

**Завдання 3**: Опишіть псевдокодом процес обчислення факторіалу числа.

**Завдання 4**: Сформулюйте псевдокод для алгоритму, який визначає, чи містить рядок дублікати.

**Висновки**

Ці завдання спрямовані на розробку навичок логічного мислення та розуміння основних алгоритмічних концепцій. Зауважте, що відповіді на ці завдання будуть залежати від конкретного способу викладення псевдокоду.

Звіт повинен містити всі псевдокоди з описом.