

# РОЗРОБКА АЛГОРИТМІВ МЕТОДОМ ПОКРОКОВОЇ ДЕТАЛІЗАЦІЇ

Практичне заняття №1



Порядок роботи:

1. Вивчити основні засади розробки алгоритмів методом покрокової деталізації.
2. Ознайомитися з прикладами розробки алгоритмів методом покрокової деталізації.
  - 2.1. Обчислення коренів квадратного рівняння.
  - 2.2. Друк шахової дошки.
  - 2.3. Прості числа.
3. Домашнє завдання



### Домашнє завдання

Методом покрокової деталізації розробити алгоритм для сортування масиву. Результат представити у вигляді діаграм Нассі-Шнейдермана для кожного кроку деталізації з поясненням, що деталізується на поточному кроці.

Варіанти завдання:

- 1. Сортування бульбашкою (Bubble sort)
- 2. Сортування вставками (Insertion sort)
- 3. Сортування вибором (Selection sort)



## ПРАКТИЧНЕ ЗАНЯТТЯ №1. РОЗРОБКА АЛГОРИТМІВ МЕТОДОМ ПОКРОКОВОЇ ДЕТАЛІЗАЦІЇ

Розробка програми – складний процес, тому практично з будь-якого етапу можливе повернення на попередні етапи для виправлення помилок або прийняття інших проектних рішень. Найчастіше такого роду повернення є наслідком помилок, допущених при логічному проектуванні програми. Тому в процесі програмування необхідно особливу увагу приділяти розробці алгоритмів.

Складання алгоритмів розв'язання задач – це робота творча. Немає універсального способу, що дозволяє без особливих зусиль скласти будь-які алгоритми. Тим не менш, деякі рекомендації, що стосуються методики розробки алгоритмів, можна дати. Вибір конкретного алгоритму для розв'язання задачі на комп'ютері зазвичай проводиться за наступними критеріями:

- забезпечення оптимального часу рішення задачі;
- забезпечення оптимального використання наявних ресурсів (пам'яті);
- забезпечення необхідної точності обчислень.



## ПРАКТИЧНЕ ЗАНЯТТЯ №1. РОЗРОБКА АЛГОРИТМІВ МЕТОДОМ ПОКРОКОВОЇ ДЕТАЛІЗАЦІЇ

Від 50 до 100% часу програміст витрачає на виправлення і модифікацію програм. У зв'язку з цим індустрія програмування пропонує використовувати систематичні підходи до програмування (а тим самим і до алгоритмізації задач), тобто пропонує методики, використання яких зменшує ймовірність помилок в програмах, спрощує їх розуміння і полегшує модифікацію. Програми повинні мати просту структуру. Складні, заплутані програми, як правило, є непрацездатними, а їх тестування вимагає великих витрат.

Основу сучасної технології розробки алгоритму складають:

- проектування алгоритму методом покрокової деталізації;
- застосування уніфікованих алгоритмічних структур (слідування, розгалуження, повторення).

Метод покрокової деталізації дозволяє звести складну задачу до сукупності простих задач.

Застосування уніфікованих алгоритмічних структур дозволяє стандартизувати процес розробки алгоритму.

Обидві складові технології сприяють мінімізації помилок і одержанню правильного результату.



## ПРАКТИЧНЕ ЗАНЯТТЯ №1. РОЗРОБКА АЛГОРИТМІВ МЕТОДОМ ПОКРОКОВОЇ ДЕТАЛІЗАЦІЇ

Розробка алгоритму за методом покрокової деталізації (уточнення) є чітким логічним процесом. Спочатку завдання розглядається в цілому, виділяються найбільш великі його частини.

Алгоритм, який вказує порядок виконання цих частин, описується в структурованій формі, не вдаючись у дрібні деталі.

Потім від загальної структури переходять до опису окремих частин.

Таким чином, розробка алгоритму складається з послідовності кроків в напрямку уточнення алгоритму.

Процес побудови алгоритму за методом покрокової деталізації виглядає наступним чином.

На першому кроці ми вважаємо, що перед нами досконалий виконавець, який «все знає і все вміє». Тому досить визначити вхідні дані і результати алгоритму, а сам алгоритм представити у вигляді єдиної абстрактної команди – постановки задачі.



## ПРАКТИЧНЕ ЗАНЯТТЯ №1. РОЗРОБКА АЛГОРИТМІВ МЕТОДОМ ПОКРОКОВОЇ ДЕТАЛІЗАЦІЇ

Якщо виконавець не навчений виконувати задану команду, то необхідно представити цей припис у вигляді сукупності більш простих команд, можливо теж абстрактних.

Для цього:

- завдання розбивають на кілька частин, кожна з яких простіше всього завдання;
- рішення кожної частини завдання формулюють в окремій команді, яка також може виходити за рамки системи команд виконавця (абстрактні команди);
- при наявності в алгоритмі команд, що виходять за межі можливостей виконавця, такі команди знову представляються у вигляді сукупності ще більш простих команд.

Процес триває до тих пір, поки всі команди не будуть зрозумілі виконавцю.

Об'єднуючи отримані команди в єдину сукупність виконуваних у певній послідовності команд, отримуємо необхідний алгоритм розв'язання поставленої задачі.

Деталізація стосується не тільки алгоритму розв'язання завдання, але і структур даних при розробці програми. На початку звичайно працюють з більш абстрактними інформаційними одиницями, детальна структура яких уточнюється в подальшому.



## ПРАКТИЧНЕ ЗАНЯТТЯ №1. РОЗРОБКА АЛГОРИТМІВ МЕТОДОМ ПОКРОКОВОЇ ДЕТАЛІЗАЦІЇ

### Правила деталізації:

- Не поспішайте зайнятися дрібницями.
- За один крок робіть тільки невеликі розширення.
- Не треба вдаватися в деталі занадто рано, концентруйте увагу перш за все на самому істотному.
- Ретельно зважуйте прийняті рішення.
- З кожним розширенням можуть бути пов'язані як досить очевидні рішення, так і досить підступні.
- Не поспішайте, задавайте собі масу питань.
- Намагайтеся зрозуміти наслідки того, що ви робите.
- Уважно стежте за даними. Вони можуть істотно вплинути на прийняте рішення, аж до його заміни.
- Будьте готові скасувати раніше прийняті рішення. Бувають моменти, коли виникають непередбачені проблеми. Якщо це якимось пов'язано з прийнятим раніше рішенням, спробуйте реалізувати інший варіант на одному або декількох рівнях деталізації.





Метод покрокової деталізації :

- реалізує спадний (зверху вниз) підхід до програмування;
- передбачає покрокову розробку алгоритму.

Протягом всього процесу покрокової деталізації логіка процесу виражається основними конструкціями структурного програмування:

- слідування,
- розгалуження,
- повторення.



В методі покрокової деталізації можна виділити наступні етапи:

1. Створюється опис програми в цілому. Визначаються основні логічні шаги, необхідні для вирішення завдання, навіть якщо поки невідомо, як їх виконати. Ці логічні кроки можуть відображати різні фізичні способи вирішення або можуть бути груповими іменами для тих дій, виконання яких видається досить смутно (абстрактні команди). Послідовності кроків, необхідних для вирішення задачі, записуються на звичайній мові.
2. У загальних термінах деталізується опис кроків, введених на етапі 1. У деталізованій опис може входити позначення циклічних структур, в той час як дії всередині циклів можуть як і раніше залишатися неясними. Таким чином, виконуються тільки загальні ескізи складних дій.
3. На цьому і наступних рівнях у вигляді послідовних ітерацій виробляються ті ж дії, що описані на етапі 2. При кожній новій ітерації уточнюються деталі, що залишилися неясними після попередніх ітерацій. У міру виконання ітерацій невизначені деталі стають все простіше і простіше, так що на якомусь етапі можуть бути повністю описані.



## ПРАКТИЧНЕ ЗАНЯТТЯ №1. РОЗРОБКА АЛГОРИТМІВ МЕТОДОМ ПОКРОКОВОЇ ДЕТАЛІЗАЦІЇ

### Переваги методу покрокової деталізації:

- Зберігається цілісність програми: від складного до простого.
- Проектування програми, кодування, перевірку та документування можна робити паралельно.
- В кожен момент часу, навіть на початку розробки є працюючий варіант програми.

### Увага!

Доречно зауважити, що використання методу покрокової деталізації зовсім не призводить автоматично до вироблення правильного алгоритму. Насправді цей метод є всього лише засобом, за допомогою якого виконується і контролюється робота щодо конкретизації алгоритму. Що стосується пошуку ефективного і надійного алгоритму, то це як правило ітераційний процес, який неможливий без творчих зусиль програміста.



## ПРАКТИЧНЕ ЗАНЯТТЯ №1. РОЗРОБКА АЛГОРИТМІВ МЕТОДОМ ПОКРОКОВОЇ ДЕТАЛІЗАЦІЇ

Завдання 1.

Обчислити корені квадратного рівняння  $ax^2 + bx + c = 0$ .

Завдання 2. Друк шахової дошки.

Надрукувати зображення шахової дошки.

Розробити алгоритм для виконувача, який розуміє і вміє виконувати команди:

- 1) друк порожнього чорного, білого і сірого квадратів,
- 2) друк сірих квадратів з символами від  $a$  до  $h$  і цифрами від 1 до 8
- 3) перехід на новий рядок.

	a	b	c	d	e	f	g	h	
8		■		■		■		■	8
7	■		■		■		■		7
6		■		■		■		■	6
5	■		■		■		■		5
4		■		■		■		■	4
3	■		■		■		■		3
2		■		■		■		■	2
1	■		■		■		■		1
	a	b	c	d	e	f	g	h	

Завдання 3. Прості числа.

Визначте кількість простих чисел серед першої тисячі натуральних чисел. Простим є число, яке ділиться без остачі тільки на себе і на 1.

