Способи опису алгоритмів. Складання й запис алгоритмів. Базові алгоритмічні конструкції. Поняття оператора. Різновиди операторів. Оператори введення й виведення даних. Структура й складові елементи програм, записаних об' єктно-орієнтованою мовою програмування

АЛГОРИТМИ ТА ЇХ ПРОГРАМНА РЕАЛІЗАЦІЯ

Візуальне програмування суттєво полегшує написання складних програм, оскільки творці візуального середовища Delphi вже запрограмували цілий ряд операцій у відеокомпонентах. Проте й у візуальному середовищі слід уміти розробляти алгоритми програмованих задач. У цьому розділі ми познайомимося зі способами описування алгоритмів і правилами побудови їхніх структурних схем.

Способи опису алгоритмів. Складання й запис алгоритмів. Базові алгоритмічні конструкції

Нині поширилась і бурхливо розвивається наука, що вважається частиною дискретної математики,— теорія алгоритмів. Вона вивчає загальні властивості алгоритмів, порівнює час їх виконання, вводить числові оцінки їхньої складності. Цю теорію заклали Е. Борель (1912) і Г. Вейль (1921). В обчислювальній техніці найчастіше використовується теорія алгоритмів в інтерпретації А. Тьюрінга й Е. Поста (1936), їхні конструкції багато в чому передбачили ідеї, покладені в основу сучасних цифрових обчислювальних машин (машина Тьюрінга).

Прийнято вважати, що існують чотири способи опису алгоритмів:

- *словесний (мама каже синові: «Віднеси взуття в ремонт, по дорозі назад купи хліба й подивись, чи немає листів у поштовій скриньці»,— цілком однозначний алгоритм);*
- *формульний (так, запис формул a = 1; 6 = 2; c = -8;* $x_{1,2} = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$ *– також цілком однозначний алгоритм, хоча в ньому не застосовано жодного слова);*
- *за допомогою алгоритмічних схем (так званих структурних схем алгоритмів, які буде розглянуто нижче);*
- *алгоритмічними мовами (багато хто називає їх мовами програмування, хоча, нагадаємо, мова Pascal, що лежить в основі Delphi, запатентована не для програмування, а для опису алгоритмів — мовою програмування зі своїм транслятором вона стала пізніше).*

Найпоширенішою формою запису алгоритмів є структурні схеми. Умовні позначки у вигляді прямокутників, ромбів та інших

фігур на таких схемах показують елементарні кроки обчислювального процесу — «обчислення», «логічне розгалуження», «початок/кінець», «ручна операція» та ін. Стрілки ж відображують напрямок ходу обчислень.

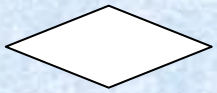
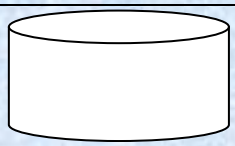
Перш ніж описувати елементи алгоритмічних структур, зупинімося на загальних правилах їх використання:

- усі елементи структурних схем алгоритмів відповідають міжнародним стандартам, тому слід застосовувати наявні значки, а не придумувати власні зображення команд;
- елементи розроблено стосовно мови Algol-60, у зв'язку з чим деякі алгоритмічні конструкції зображуються досить складно (наприклад, розгалуження в різних напрямках — CASE);
- серед значків структурних схем чимало архаїчних — «запис на перфострічку», «виведення на перфокарти» тощо, тому доцільно описувати обчислювальний процес у загальному вигляді, не заглиблюючись у «машинні» подробиці;
- запис усередині значків слід виконувати математичною мовою. Використання операторів Delphi або C призводить до абсурдного висновку, що краще читати програму, а не ваш алгоритм;
- усі елементи алгоритму мають бути пронумеровані.

Умовні позначки елементів структурних схем алгоритмів наведено в табл. 3.1.

Таблиця 3.1

Елементи структурних схем алгоритмів на їхнє призначення

Елемент	***Призначення елемента***
	Обчислювальний процес. Фактично це один або кілька нероздільних операторів присвоювання, що здійснюють обчислювальні перетворення
	Операції введення-виведення без конкретизації типу периферійних пристроїв
	Розгалуження дво- або багатоальтернативне
	Початок і кінець алгоритму. Формально може і не відповідати жодному з операторів програми
	Запис на «магнітний барабан» (тобто вінчестер). Елемент потрібен для тих мов, де запис на диск — проблема

	Міжсторінкове перенесення
	Перенесення в межах сторінки
	Процедура. «Наперед визначений процес»
	«Модифікація». Зазвичай використовується для запису параметричного циклу

Як приклад на рис. 3.1 наведено структурну схему алгоритму знаходження коренів квадратного рівняння.

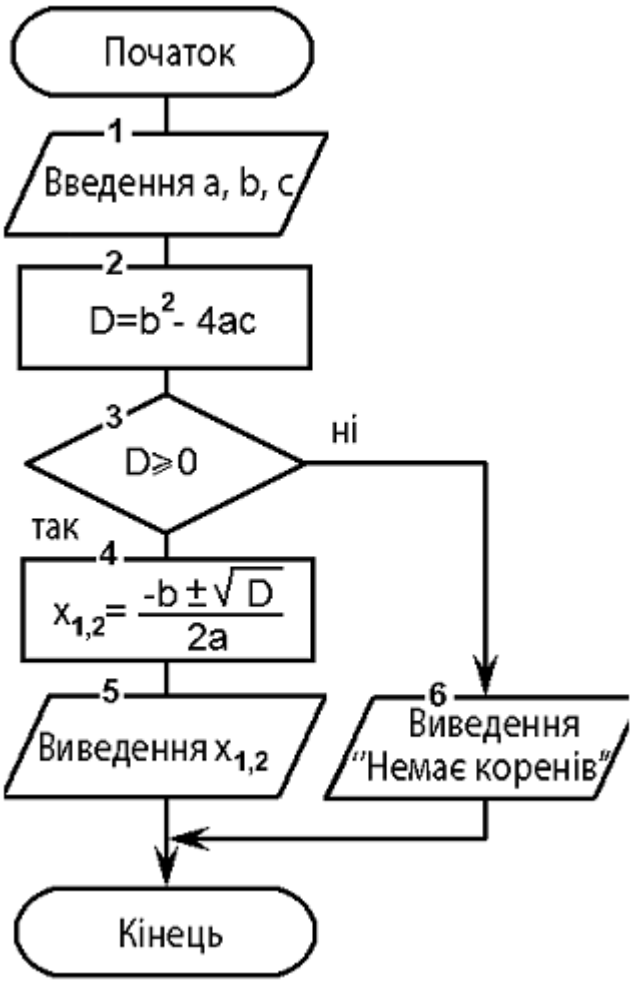


Рис. 3.1. Структурна схема алгоритму знаходження коренів квадратного рівняння

Зазначимо правила побудови структурних схем алгоритмів. Елементи таких схем мають вертикальну структуру. Передача керування від одного елемента до іншого, спрямована згори вниз або зліва направо, вважається природною і може не позначатися стрілками. Передача керування в інших напрямках, ламані лінії зв'язку або злиття зв'язків обов'язково позначаються стрілками.

Зверніть увагу: одна й та сама конфігурація компонентів структурних схем може відповідати різним алгоритмам (у графічному задаванні алгоритмів важливе не тільки геометричне зображення елементів, але й записаний у них текст).

Запис алгоритмів за допомогою динамічних змінних вимагає великої праці, а написання програми стековою мовою Forth — іще більшої.

Структура и складові елементи програм, записаних об'єктозорієнтованою мовою програмування

Останнім часом набула поширення мова UML *(Unified Modding Language)* — загальноцільова мова візуального моделювання, розроблена для візуалізації, проектування й документування

компонентів програмного забезпечення, а також бізнес-процесів та інших систем, що не належать до програмного забезпечення. Ця мова ввібрала в себе найбільш прогресивні методи програмної інженерії, що з успіхом використовувалися протягом останніх років у моделюванні великих і складних систем. Мова UML дуже різнобічна, причому включений у неї механізм так званих діаграм діяльності (activity diagram) дозволяє ефективно описувати алгоритми. Із цієї причини дедалі частіше в літературі замість структурних схем алгоритмів наводять діаграми діяльності з мови UML.

У класичних діаграмах діяльності наявні позначення станів і переходів, репрезентація дій, однак відсутня сигнатура подій, які ініціювали ці дії. (Для репрезентації даних подій в UML існує інший тип діаграм.)

Кожен стан на діаграмі діяльності відповідає виконанню певної елементарної операції, а перехід у наступний стан здійснюється тільки після завершення операції в попередньому стані. Зовнішній вигляд простої діаграми діяльності подано на рис. 3.2.

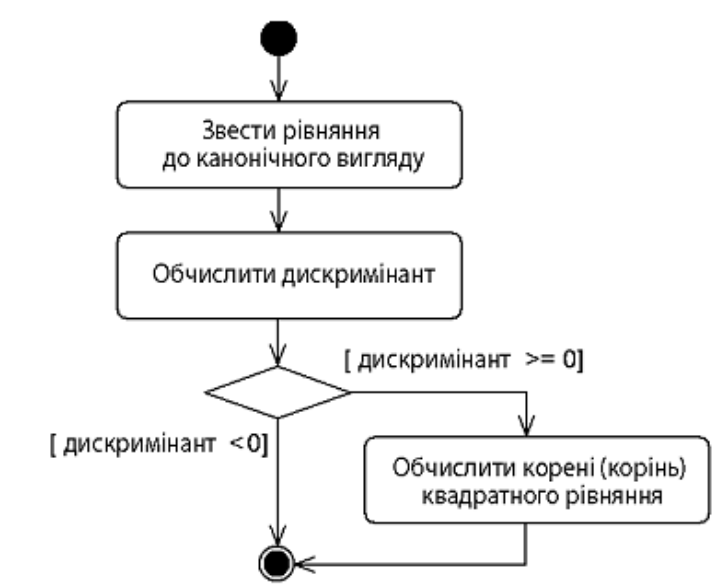


Рис. 3.2. Зображення алгоритму знаходження коренів квадратного рівняння у вигляді діаграми діяльності

Звернемо увагу на відмінності діаграм діяльності від традиційних структурних схем. Вони полягають у такому:

- усі стани на діаграмах діяльності зображуються прямокутниками зі скругленими кутами;
- нумерації цих прямокутників немає, однак внутрішній запис має бути унікальним у межах даного алгоритму (інакше необхідний прямокутник складно знайти);
- переходи зображуються стрілками, які прорисовуються завжди, навіть якщо перехід іде в «природному» напрямку згори вниз або зліва направо;
- оператор розгалуження внутрішніх написів не має: умови розгалуження записані над його вихідними стрілками (діями);
- початок і кінець алгоритму зображуються чорними кружками (закінчення алгоритму — з білою облямівкою)

Дія може бути записана природною мовою, певним псевдокодом або мовою програмування. Ніяких обмежень щодо записування дій не накладено. Рекомендується як ім'я простої дії використовувати

дієслово з пояснювальними словами. Якщо ж дія може бути подана в певному формальному вигляді, то доцільно записати її тією мовою програмування, якою передбачається реалізувати конкретний проект.

На відміну від звичайних структурних схем алгоритмів, діаграми діяльності дозволяють описувати паралельні обчислювальні процеси, як це показано на рис. 3.3.

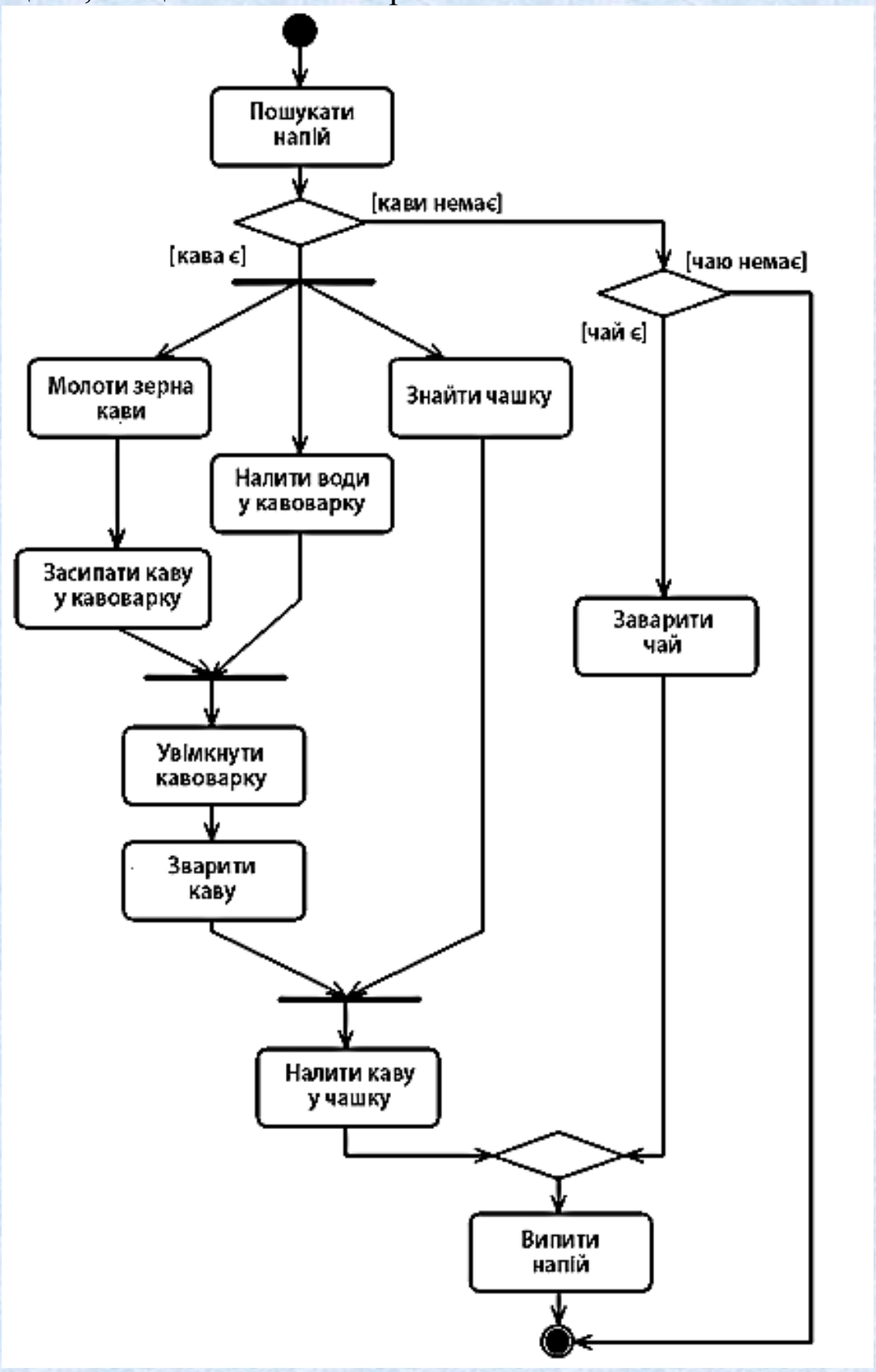


Рис. 3.3. Діаграма діяльності алгоритму, який має паралельні вітки

Очевидно, що в системах із багатоядерними процесорами й розпаралелюванням завдань діаграми діяльності дають більше можливостей записувати алгоритми, ніж звичайні структурні схеми.

ТЕОРЕТИЧНІ ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

- Що вивчає теорія алгоритмів? Хто заклав основи цієї науки?
- Перелічіть два способи опису алгоритмів.
- Які елементи використовуються в побудові структурних схем?
- У яких випадках лінії на структурних схемах обов'язково позначаються стрілками?