

§ 60. Алгоритми і програми з розгалуженнями

Вивчивши цей параграф, ми:

познайомимося із засобами реалізації базової алгоритмічної структури розгалуження в

НАМ і мові програмування Паскаль;

дізнаємося, що таке складений оператор;

довідаємося, за якими правилами описуються складені умови;

з'ясуємо на прикладах особливості застосування команди розгалуження й умовного оператора.

====60.1. Засоби реалізації базової алгоритмічної структури розгалуження =====

Базова алгоритмічна структура розгалуження має дві форми — повну й коротку. В навчальній алгоритмічній мові і в мові програмування Паскаль є відповідні засоби реалізації розгалуження: **команда розгалуження** і **умовний оператор**, які також мають дві форми — повну і коротку. Схема реалізації розгалуження в НАМ і мові програмування Паскаль подана на рис. 60.1.

Блок-схема Базова структура розгалуження	НАМ Команда розгалуження	Мова програмування Паскаль Умовний оператор
	<u>якщо</u> <умова> —	<i>if</i> <умова>
<div style="border: 1px solid black; width: 50px; height: 20px; margin: 0 auto; text-align: center; padding: 2px;">Дія 1</div>	<u>то</u> < команда 1>	<i>then</i> <оператор 1>
<div style="border: 1px solid black; width: 50px; height: 20px; margin: 0 auto; text-align: center; padding: 2px;">Дія 2</div>	<u>інакше</u> <команда 2>	<i>else</i> <оператор 2>
		;

а) повна форма;

Блок-схема Базова структура розгалуження	НАМ Команда розгалуження	Мова програмування Паскаль Умовний оператор
	<u>якщо</u> <умова>	<i>if</i> <умова>
<div style="border: 1px solid black; width: 50px; height: 20px; margin: 0 auto; text-align: center; padding: 2px;">Дія 1</div>	<u>то</u> < команда 1>	<i>then</i> <оператор 1>
		;

б) коротка форма

Рис. 60.1. Схема реалізації розгалуження в НАМ і мові програмування Паскаль

Команда розгалуження (або **команда якщо**) записується за допомогою службових слів: якщо, то й інакше. Після якщо розміщується умова, після то — команда на виконання дії 1, після інакше — команда на виконання дії 2. Зверніть увагу на те, що кожне службове слово займає окремий рядок. Це потрібно для того, щоб конструкція команди була абсолютно зрозумілою.

За командою розгалуження в залежності від результату перевірки умови буде виконано одну з двох команд: команду 1, якщо результатом перевірки є «так», команду 2 — якщо «ні».

Команда на виконання дії 2 може бути відсутньою.

Аналогом команди розгалуження в мові Паскаль є *умовний оператор* (або *оператор if*). Цей оператор також має дві форми — повну і коротку — й утворюється так само, як і команда розгалуження, тільки із заміною українських слів якщо, то, інакше на їх англійські еквіваленти: **if** (якщо), **then** (то), **else** (інакше).

Наведемо приклади виконання команди розгалуження й аналогічного умовного оператора.

Значення змінних до виконання команди (оператора)		Команда розгалуження	Умовний оператор	Значення змінних після виконання команди (оператора)	
x	y			x	y
-1	5	<u>якщо</u> $x > 1$	if $x > 1$	-1	1
1	5	<u>то</u> $y := 2 * x$	then $y := 2 * x$	1	1
2	5	<u>інакше</u> $y := 1$	else $y := 1;$	2	4
9	3	<u>якщо</u> $ x < 3$	if $\text{abs}(x) < 3$	9	3
-2	3	<u>то</u> $y := 9 - x^2$	then $y := 9 - \text{sqr}(x);$	-2	5
3	3			3	3
0	-1	<u>якщо</u> $2 * x > 3 * y$	if $2 * x > 3 * y$	1	-1
2	5	<u>то</u> $x := x - y$	then $x := x - y$	2	9
3	2	<u>інакше</u> $y := y + 2 * x$	else $y := y + 2 * x;$	3	8
5	3			2	3

====60.2. Програма «Два числа»=====

Розробимо програму порівняння двох цілих чисел, які вводяться користувачем із клавіатури. Позначимо ці числа через a і b .

Наведемо програму і три варіанти її виконання для різних значень a і b .

Програма	Приклади виконання програми		
program two_numbers;	Приклад 1	Приклад 2	Приклад 3
var a, b: integer ;			
begin	<i>Введіть два цілих числа:</i>		
writeln ('Введіть два цілих числа:');	3 5	7 5	5 5
readln (a,b);	3 = 5 ні	7 = 5 ні	5 = 5 так
if a = b	-	-	5 дорівнює 5
then writeln (a, ' дорівнює ', b);	-	-	-
if a < b	3 < 5 так	7 < 5 ні	5 < 5 ні
then writeln (a, ' менше ', b);	3 менше 5	-	-
if a > b	3 > 5 ні	7 > 5 так	5 > 5 ні
then writeln (a, ' більше ', b);	-	7 більше 5	-
readln			
end.			

Курсивом виділено повідомлення, які в процесі виконання програми будуть виведені на екран.

====60.3. Серія команд, складений оператор=====

У всіх розглянутих нами схемах і прикладах конструкція розгалуження використовувалася для вибору *однієї з двох дій*. Якщо потрібно здійснювати вибір *однієї з двох сукупностей дій*, то такі сукупності подають як *одну складену дію*.

У НАМ *серія команд*, яка має сприйматися як одна *складена команда*, позначається службовими словами: пс (скорочення від *початок серії*) і кс (скорочення від *кінець серії*). Ці службові слова розташовуються відповідно перед першою і після останньої команди серії.

У мові Паскаль серія операторів, яка має сприйматися як один *складений оператор*, позначається службовими словами **begin** (початок) і **end** (кінець). Ці слова наче дужки обмежують серію операторів з обох сторін.

Наприклад, змінній z потрібно надати значення за таким правилом:

$z = x - y$, де $y = \sqrt{x}$, — для додатного значення x ;

$z = 1$ — для від'ємного або нульового значення x .

Це можна зробити у такий спосіб:

<u>якщо</u> $x > 0$	if $x > 0$
<u>то</u> <u>пс</u>	then begin
$y := \sqrt{x}$	$y := \text{sqrt}(x);$ {“;” розділяє оператори-складники}
$z := x - y$	$z := x - y;$ {“;” можна пропустити}
<u>кс</u>	end {“;” немає!}
<u>інакше</u> $z := 1$	else $z := 1;$ {“;” позначає завершення оператора if }

Зверніть увагу на викристання символів “;” у наведеному фрагменті програми.

====60.4. Програма «Котра година?»=====

Як відомо, наша Земля поділена на 24 пояси, у кожному з яких усі годинники показують однаковий час. Між різними часовими поясами час відрізняється на певну кількість годин. Якщо ви знаєте, котра година у вашому місті, то дуже просто можете визначити час у будь-якій іншій точці Землі. Для цього вам потрібні такі дані: номер годинного пояса, де знаходиться ваше місто (позначимо його через n_0), і номер годинного пояса в точці, що вас цікавить (позначимо його через n). Час t_n у поясі n визначається через час t_0 у поясі n_0 за формулою:

$$t_n = t_0 + n - n_0.$$

Складемо програму визначення часу в заданому пункті нашої планети за відомим часом в даному пункті.

Проаналізуємо задачу. Номери годинних поясів — це цілі числа, які змінюються від -11 до $+12$. Наприклад, $+2$ для України, $+9$ для Токію, -5 для Нью-Йорка. Можна переконатися, що наша формула може нам дати несподівані результати. Так, якщо в Києві п'ята година, то за формулою одержимо в Нью-Йорку $t_n = -2$ години, а якщо в Києві 23 година, то в Токію $t_n = 30$ годин. У першому випадку ($t_n < 0$) маємо, що в Нью-Йорку ще вчорашня доба, там не -2 години, а $24 - 2$, тобто 22 година. У другому випадку ($t_n > 24$) — навпаки, в Токію вже настав завтрашній день, там $30 - 24$, тобто шоста година ранку.

Отже, з аналізу задачі випливає, що вихідними даними мають бути час у поясі t_n і повідомлення, який там день — вчорашній, сьогоднішній чи завтрашній. Для зберігання повідомлення виділимо змінну d рядкового типу.

Для складання алгоритму скористуємося таким прийомом. На першому етапі розв'язання задачі визначимо час t_n за формулою й позначимо день як сьогоднішній. На другому етапі перевіримо, чи не є від'ємним значення t_n . Якщо так, то змінимо відповідним чином значення t_n і d . На третьому етапі перевіримо, чи не є значення $t_n \geq 24$. Якщо так, то змінимо t_n і d .

Запишемо програму мовою програмування Паскаль для випадку, коли відомим нам часом є час в Україні. Номер часового пояса України задамо як константу.

```

program what_oclock;
  uses Crt;
  const n0 = 2;
  var n, t0, tn: integer; d: string;
begin
  clrscr;
  write('Котра година в Україні?: ');           {Уводимо вхідні дані}
  readln(n);
  write('Уведіть номер часового поясу, де вас цікавить час: ');
  readln(t0);
  tn := t0 + n - n0;                             {Визначаємо час за формулою}
  d := 'день сьогоднішній';                     {День визначаємо як сьогоднішній}
  if tn < 0                                       {Виконуємо першу перевірку часу}
  then begin                                     {Якщо потрібно, коригуємо час і день}
    tn := tn + 24;
    d := 'день учорашній';
  end;
  if tn >= 24                                    {Виконуємо другу перевірку часу}
  then begin                                     {Якщо потрібно, коригуємо час і день}
    tn := tn - 24;
    d := 'день завтрашній';
  end;
  writeln ('Там ', d, ', ', ', ', tn, ' год. '); {Виводимо результати}
  readln;
end.

```

====60.5. Складені умови=====

Умова, яка перевіряється при виконанні розгалуження, може бути простою і складеною. У ролі *простой умови*, як ми знаємо, виступає будь-яке висловлювання, яке може набувати одного з двох значень — «істина» («так») або «хибність» («ні»). Звичайно в алгоритмах і програмах прості умови подаються у вигляді двох арифметичних виразів, з'єднаних знаком операції відношення (порівняння). Саме такі умови ми й використовували в раніше наведених прикладах.

Іноді для вибору тих чи інших дій потрібно перевіряти декілька простих умов. У такому разі ми маємо *складену умову*. Складена умова утворюється з простих за допомогою знаків логічних операцій і круглих дужок.

Розглянемо конструювання складеної умови з двох простих умов.

- Якщо потрібно, щоб виконувалися дві умови водночас, їх поєднують знаком операції логічного множення – операції **і**. Результатом виконання операції **і** є «істина» («так») лише в одному випадку: *обидва операнди (множники) мають значення «істина»*. Наприклад, якщо потрібно із списку учнів відібрати тих дівчат, які є відмінницями, то умову відбору потрібно записати у вигляді: «учень є дівчиною» **і** «учень є відмінником». До відібраних потраплять тільки учениці-відмінниці.
- Якщо достатньо, щоб виконувалася хоча б одна умова з двох, то їх поєднують знаком операції логічного додавання – операції **або**. Результатом виконання операції **або** є «істина» («так»), якщо *хоча б один із операндів (доданків) має значення «істина»*.

Наприклад, якщо потрібно серед учнів відібрати всіх відмінників і всіх спортсменів, то умова відбору має такий вигляд: «учень є відмінником» *або* «учень є спортсменом». До відібраних потраплять учні, які є і відмінниками, і спортсменами, а також учні-відмінники, які не є спортсменами, і учні-спортсмени, які не є відмінниками.

- Для запису складених умов розглянутого виду в НАМ використовують службові слова i та або, а в мові програмування Паскаль — їх англійські еквіваленти: **and** (і) та **or** (або). Прості умови, що входять у складені, обов'язково беруть у круглі дужки.

Наприклад, якщо потрібно перевірити, чи належить змінна x відрізку $[1;5]$, то це записується так:

$(x \geq 1) \text{ i } (x \leq 5)$ в НАМ;

$(x \geq 1) \text{ and } (x \leq 5)$ мовою Паскаль.

Якщо потрібно перевірити, чи має пасажир право на безкоштовний проїзд за віком (безкоштовний проїзд дозволяється дітям до 7 років і дорослим старше 60 років), то умову перевірки віку пасажирів (vik) можна записати у вигляді:

$(vik < 7) \text{ або } (vik > 60)$ в НАМ;

$(vik < 7) \text{ or } (vik > 60)$ мовою Паскаль.

====60.6. Програма «Діалог»=====

Як приклад застосування розгалужень зі складеними умовами розробимо програму, яка імітує діалог із користувачем: виводить на екран запитання і використовує надані відповіді для підтримки «бесіди».

```

program dialog;
  uses Crt;
  var vik, v: integer; name, p: string;
begin
  clrscr;
  writeln('Давайте познайомимося! Як вас звуть?');
  readln(name);
  writeln('Дуже приємно, ', name, '! А скільки вам років?');
  readln(vik);
  v := vik mod 10;                                {Знаходимо останню цифру числа років}
  p := ' років';                                  {Добираємо відповідну форму}
  if v=1                                          {слова «рік» за загальним правилом}
  then p:= ' рік';
  if (v>1) and (v<5)
  then p:= ' роки';
  if (vik>10) and (vik<15)                        {Числівники 11, 12, 13 і 14}
  then p:= ' років';                              {не підпадають під загальне правило}
  writeln(vik, p, ' – чудова життєва пора. ');
  writeln('Хай вам щастить!');
  readln;
end.

```

====60.7. Вкладені умовні оператори=====

Умовний оператор призначається для альтернативного вибору одного з двох операторів будь-якого виду. Такими операторами в розглянутих прикладах виступали оператори присвоєння або виведення. Проте здійснювати вибір можна і між умовними операторами. Це означає, що в умовному операторі після службових слів **then** і **else** можна розміщувати інший умовний оператор. Така конструкція називається вкладенням умовних операторів, або **вкладеними умовними операторами**. Аналогічні конструкції в НАМ називаються *вкладеними розгалуженнями*.

Наприклад, для обчислення найбільшого (\max) серед чисел a , b , c можна скористатися таким оператором:

```
if a > b
  then if a > c then max := a else max := c
  else if b > c then max := b else max := c;
```

Звичайно конструкції з вкладеними умовними операторами не є простими для сприйняття, тому їх краще уникати й застосовувати ланцюжок коротких умовних операторів. Для нашого випадку краще спочатку надати \max значення одного із чисел, наприклад a , а потім «просіяти» його крізь сито перевірок:

```
max := a;
if b > max then max := b;           {Тепер max є найбільшим з двох – a і b}
if c > max then max := c;           {max стає найбільшим з трьох – a, b, c}
```

У такому варіанті маємо цілком прозорий і зрозумілий текст програми.

ВИСНОВКИ

Для подання базової алгоритмічної структури розгалуження в НАМ існує команда розгалуження якщо, а в мові Паскаль — умовний оператор **if**. Для того щоб можна було здійснювати вибір не між двома командами (операторами), а між двома групами команд (операторів), їх об'єднують в одну складену команду (оператор), розміщуючи між службовими словами пс і кс (**begin** і **end**). Умову, за якою здійснюється вибір, можна утворювати з використанням операцій порівняння. Для перевірки, чи виконуються декілька умов водночас, застосовується операція і (**and**). Якщо достатньо, щоб виконувалася хоча б одна з умов, застосовують операцію або (**or**). Складовою частиною умовного оператора може виступати інший умовний оператор, проте такі конструкції є громіздкими і їх краще замінити послідовною перевіркою умов.

Контрольні питання та вправи

1. Базова алгоритмічна структура розгалуження в НАМ подається ..., а в мові Паскаль — Вставте пропущені слова.

- умовною командою;
- командою розгалуження;
- умовним оператором;
- оператором розгалуження.

2. Які оператори записані без помилок?

- if a>0 then x := a else y := b;**
- if a>0 then x := a; else y := b;**
- if a>0 then x := a; writeln(x) else y := b; writeln(y);**

3. Яке значення буде надано змінній x в результаті виконання оператора:

```
if a>0 then x := a+1 else x := 2*a;
```

якщо значення змінної a дорівнює:

- а) 5;
- б) -15;
- в) 0;
- г) 24?

4. У ящику знаходяться N кульок — декілька чорних та 20 білих. Відомо, що кількості чорних і білих кульок задовольняють умові:

(чорних не більше, ніж білих) або (чорних не менше 25)

Чи може загальна кількість кульок дорівнювати:

- а) 45;
- б) 25;
- в) 42;
- г) 40;
- д) 43;
- е) 50 ?

5. Для визначення, чи є ціле число a додатнім, від'ємним чи нулем, складено три алгоритми. Унаслідок виконання яких з них змінна y одержить потрібне значення?

а) якщо $a = 0$
то $y := \text{"нуль"}$
якщо $a > 0$
то $y := \text{"додатне"}$
інакше $y := \text{"від'ємне"}$

б) якщо $a > 0$
то $y := \text{"додатне"}$
інакше $y := \text{"від'ємне"}$
якщо $a = 0$
то $y := \text{"нуль"}$

в) якщо $a = 0$
то $y := \text{"нуль"}$
якщо $a > 0$
то $y := \text{"додатне"}$
якщо $a < 0$
то $y := \text{"від'ємне"}$

6. Складіть програму, яка виводить на екран рекламу гамбургерів і пропонує користувачеві зробити замовлення. Програма приймає замовлення й повідомляє користувача про те, який безкоштовний сюрприз він отримає. До одного замовленого гамбургера додається пляшка мінеральної води; до двох — пляшка кока-коли; до трьох або більше — ще один гамбургер.

7. Складіть програму, яка запитує користувача, котра година, і виводить на екран відповідне побажання — «Доброго ранку!», «Доброго дня!», «Доброго вечора!» або «Доброї ночі!». Для кожного побажання виберіть відповідний колір.

8. Глядач, який знаходиться в середині n -го ряду концертної зали, і радіослухач, який знаходиться в іншому місті на відстані r , слухають концерт. Складіть програму, яка визначає, хто з них першим почує мелодію, якщо відомі відстань від сцени до першого ряду d , відстань між рядами h . Дані про швидкості розповсюдження звуку й радіохвилі знайдіть самостійно. Користуючись складеною програмою, визначте, у якому приблизно місті знаходиться слухач, який сприймає мелодію водночас із глядачем, що сидить у першому ряді концертної зали в центрі Києва ($d = 5$ м).

9. У супермаркеті оголосили акцію. Покупець, який придбає за три покупки товар на суму, що перевищує 1000 грн., одержує дисконтну картку за умови, що вартість кожної наступної покупки, починаючи з другої, не менше як на півціни перевищує сумарну вартість попередніх. Складіть програму, яка за вартістю трьох покупок визначає, чи заробив покупець дисконтну картку.

10. Складіть програму, яка моделює діалог агента туристичної фірми з користувачем. «Агент» розпитує користувача про його наміри, фінансові обмеження тощо і пропонує відповідний варіант відпочинку.

11. Складіть програму перевірки, чи може шахова фігура, яка знаходиться на шахівниці у клітинці з номером i по горизонталі й номером k по вертикалі, зробити хід у клітинку з відповідними номерами $i1$, $k1$. Розробіть варіанти програми для випадків, коли шаховою фігурою є:

а) ладья; б) слон; в) ферзь; г) король; д) кінь.

12. Складіть програму розв'язання задачі про вигідну покупку. Пара шкарпеток на ринку коштує 1 грн., дюжину віддають за 10,5 грн., а дюжину дюжин — в'язку — за 111 грн. Покупець хоче придбати n шкарпеток. Як йому зробити вигідну покупку? Наприклад, вигідніше купити дюжину шкарпеток, ніж 11 пар, — одна пара даром. Програма має за заданим значенням n визначити, яку кількість в'язок, дюжин і пар шкарпеток слід придбати покупцеві.

⚡ вклені умовні оператори, команда розгалуження, операції відношення, операції порівняння, операція “або”, операція “і”, серія команд, складена умова, складений оператор, умовний оператор