

§ 57. Уведення і виведення значень величин

Вивчивши цей параграф, ми:

познайомимося з основними складовими алгоритму опрацювання величин;
дізнаємося, як значення величин вводяться з клавіатури і виводяться на екран;
з'ясуємо, як зробити роботу з екраном приємною і зручною.

====57.1. Складові алгоритму опрацювання величин=====

В алгоритмах опрацювання величин зазвичай можна виділити такі три складові: введення значень вхідних величин (вхідних даних); обчислення значень інших величин через вхідні дані, тобто їх опрацювання; виведення результатів роботи алгоритму (вихідних даних).

Значення вхідних даних у процесі виконання алгоритму змінювати не прийнято. Вони звичайно зберігаються до завершення роботи алгоритму такими, якими були надані виконавцеві.

Опис алгоритму засобами НАМ зовсім не передбачає команд на введення або виведення даних. Вважається, що значення аргументів будуть надані до виконання алгоритму, а значення результатів будуть виведені після виконання алгоритму, і опису піддається лише процес перетворення аргументів на результати.

Якщо алгоритм створюється для його виконання комп'ютером, тобто розробляється програма, то всі три складові алгоритму описуються явно: як саме вхідні значення змінних повідомляються комп'ютеру, як вони мають бути ним опрацьовані, у який спосіб результати роботи програми будуть виведені на екран комп'ютера.

Далі ми розглянемо засоби введення і виведення даних у мові програмування Паскаль.

====57.2. Оператори введення значень з клавіатури=====

При роботі з комп'ютером користувачеві надається можливість задавати значення величин із клавіатури. У мові Паскаль для цього застосовуються **оператори введення** **read** (від англ. read — читати) і **readln** (скорочення від англ. read line — читати рядок), які мають такі конструкції:

```
read(список імен змінних);  
readln(список імен змінних);
```

При виконанні оператора **read** (**readln**) комп'ютер переходить у стадію очікування введення даних із клавіатури. Значення, що вводяться, відображаються на екрані комп'ютера. Кожне наступне значення відокремлюють від попереднього пропуском. За кількістю і типом уведених значень мають відповідати змінним у списку оператора. Завершення введення всіх очікуваних значень позначають натисненням клавіші **[Enter]**. Перше введенне значення буде надане першій у списку величині, друге — другій і т. д.

«Порожній» оператор **readln** (тобто **readln;**) застосовують для переведення програми в режим очікування натиснення клавіші **[Enter]** (наприклад, якщо потрібно зробити паузу для ознайомлення з повідомленнями, виведеними на екран комп'ютера).

Розглянемо приклади застосування оператора **read**. Припустимо, що змінна *c* є цілою, *d* — дійсною, *r* — рядковою.

Оператор	Введено з клавіатури	Значення, одержані змінними		
		<i>c</i>	<i>d</i>	<i>r</i>
read (<i>c, d, r</i>);	3 15.7 слово	3	15.7	слово
read (<i>c, d, r</i>);	-125 14 два слова	-125	14.0	два слова
read (<i>d, c, r</i>);	2.1 25 25	25	2.1	25
read (<i>c, d, r</i>);	025 -16 1 ***** 1	25	-16.0	1 ***** 1

Якщо значення 25 уводиться як значення текстової змінної, то воно сприймається як послідовність із двох символів 2 і 5, а не як число, і арифметичні операції над ним виконувати не можна.

====57.3. Оператори виведення значень на екран=====

Результати роботи програми можна побачити лише на екрані (або на папері, якщо їх буде надруковано). Для виведення значень величин на екран у мові Паскаль існують *оператори виведення* **write** (від англ. write — писати) і **writeln** (скорочення від англ. write line — писати рядок). Конструкція цих операторів має такий вигляд:

write (список виразів) ;
writeln (список виразів) ;

Вираз задає правило для винайдення значення.

При виконанні оператора **write** або **writeln** на екран із *поточної* позиції курсора будуть виведені один за одним *значення виразів*, розміщених у списку.

Різниця між операторами **write** і **writeln** полягає в тому, що після виконання оператора **write** курсор залишається на поточному місці, а після виконання **writeln** переходить на наступний рядок.

«Порожній» оператор **writeln** (тобто **writeln;**) застосовують для пропуску рядка на екрані.

Розглянемо виведення на екран числових значень. Вираз, який задає правило винайдення числового значення, називається *арифметичним*. Арифметичний вираз може бути простим (числова константа, числова змінна або числова функція) і складним, тобто утвореним із простих за допомогою знаків арифметичних операцій. При обчисленні значення складного виразу спочатку обчислюються значення його простих складових, а потім виконуються арифметичні операції над ними з урахуванням їх пріоритету. Для встановлення іншого порядку виконання дій застосовуються круглі дужки.

Наприклад:

writeln (3) ; — виведе на екран значення 3;
writeln (b) ; — виведе на екран те значення, яке на момент виконання оператора зберігає змінна b;
writeln (abs (1+2*b)) ; — виведе на екран модуль значення суми 1+2b;
writeln ((a+b) /2) ; — виведе на екран середнє арифметичне значень змінних a і b.

Вираз в операторі **writeln** може бути не тільки арифметичним, а й рядковим. Наприклад, оператор

writeln ('Я дуже люблю інформатику!');

виведе на екран рядок:

Я дуже люблю інформатику!

Оператор виведення може містити список різних виразів, тобто їх перелік через кому. Наприклад, оператор:

writeln (s, sqr(s), s*sqr(s));

для s=8 виведе на екран такий рядок:

864512

Для розділення значень при їх виведенні застосовують рядкові константи, наприклад константу ' ', яка містить пропуск.

Оператор

writeln (s, ' ',sqr(s), ' ',s*sqr(s));

виведе на екран такий рядок:

8 64 512

====57.4. Формати виведення=====

Для того щоб повідомлення, які виводяться програмою на екран, були зручними для сприйняття, доцільно скористатися *форматом виведення* значень.

Для цілочисельних і текстових значень формат задає загальну кількість позицій, відведену для розміщення значення на екрані. Кількість позицій зазначається через двокрапку після виразу, значення якого виводиться. Зайві позиції дадуть пропуски *перед* значенням.

Наприклад, оператор **write**(2:5) виведе значення 2 з пропуском перед ним чотирьох позицій. Оператор **write**('Квадрат':10) виведе слово «Квадрат» з пропуском трьох позицій перед словом.

Формат для виведення дійсних чисел складається з двох частин — загальної кількості позицій і кількості позицій після десяткової крапки. У загальну кількість входять також позиції для знака від'ємного числа і для десяткової крапки. Дробова частина дійсного числа при виведенні буде округленою до заданої кількості знаків. Якщо вказаних позицій не вистачає для виведення цілої частини числа або його знака, комп'ютер додасть необхідну кількість.

Наприклад, і оператор **write**(2/3:6:2), і оператор **write**(2/3:3:2) виведуть на екран значення результату ділення 2 на 3, тобто 0.67, хоча перший з двома пробілами перед числом, а другий — без пропусків.

====57.5. Коментоване введення й виведення=====

Для того щоб робота користувача з програмою відтворювалася на екрані у зрозумілому вигляді, застосовують пояснення (коментарі), які допомагають користувачеві усвідомити, що від нього вимагається, які дані очікуються програмою, які значення виводяться нею на екран.

Для коментованого введення використовують комбінацію операторів виведення і введення. Спочатку виводять на екран пояснювальний текст, а потім зчитують уведені користувачем значення. Текст подають як рядкову константу в операторі виведення.

Наприклад, якщо користувач має ввести своє прізвище та ім'я, то це можна пояснити йому в такий спосіб:

```
write('Введіть ваше прізвище: ');
readln(f);
write('Введіть ваше ім'я: ');
readln(name);
```

Зверніть увагу на використання операторів **write** і **readln**. Кожна пропозиція користувачеві розпочинається з нового рядка, свої дані він вводить безпосередньо після двокрапки та пропуску.

Для виведення на екран апострофа його подвоюють. Так, для того щоб вивести на екран слово «ім'я», яке містить апостроф, ми записали його у такому вигляді: «ім''я».

Для коментованого виведення результатів, одержаних у процесі роботи програми, в операторі **write** застосовують рядкові константи, де розміщують потрібні пояснення.

Наприклад, для виведення результату *rez* тестування учня, прізвище якого є значенням змінної *f*, а ім'я — змінної *name*, можна скористатися таким оператором:

```
writeln('Учень ', f, ' ', name, ' за виконання тесту має оцінку ', rez);
```

Для значень *f* = 'Петров', *name* = 'Олексій', *rez* = 10 на екран буде виведено:
Учень Петров Олексій за виконання тесту має оцінку 10

====57.6. Програма «Сума чисел»=====

Програма «Сума чисел» демонструє використання операторів введення/виведення.

Програма пропонує користувачеві ввести два цілих числа, обчислює їх суму і виводить

результат у вигляді арифметичного виразу.

Наприклад, якщо користувач уведе числа 3 та 5, на екрані з'явиться такий запис:
3+5=8

Наведемо текст програми:

```

program example;                                {Заголовок програми}
    var a, b: integer;                          {Оголошення змінних}
begin                                            {Початок програми}
    write('Уведіть два цілих числа: '); {Пропозиція користувачеві}
    readln(a, b);                                {Уведення чисел}
    writeln(a, '+', b, '= ', a+b);             {Виведення їх суми}
    readln;                                     {Очікування натиснення клавіші [Enter]}
end.                                           {Кінець програми}

```

До кожного рядка програми у фігурних дужках наведено пояснення. *Коментарі*, розміщені у фігурних дужках, комп'ютер ігнорує, але вони зберігаються в тексті програми. Такі коментарі допомагають зрозуміти програму тому, хто з нею працює, створює на її основі інші програми.

Отже, програма оздоблюється коментарями і ззовні, і зсередини — і для того, хто її використовує як готовий продукт, і для того, хто її майструє.

Якщо один або декілька операторів програми взяти у фігурні дужки, комп'ютер їх «не помітить». Такий прийом часто використовують під час налагодження програми.

====57.7. Створення форм для введення й виведення даних=====

Бажано, щоб діалог користувача з програмою на екрані комп'ютера відбувався за певним заздалегідь розробленим сценарієм. Для цього потрібно продумати, яка інформація буде виводитися на екран, у якому місці екрана мають з'являтися коментарі, підсумкові дані тощо. Діалог можна прикрашати кольоровим забарвленням, рамочками із спецсимволів і т. п.

Розмістимо на початку програми перед розділом опису змінних оголошення про використання засобів роботи з екраном — *бібліотеки* `Crt`. Це оголошення має такий вигляд:

```
uses Crt;
```

Тепер ми можемо скористатися такими операторами:

- *оператором очищення екрана* `clrscr` (від англ. clear screen — очистити екран);
- *оператором установки курсора в задану позицію* екрана `gotoxy` (`p`, `r`) (від англ. go to — перейти до). Значення `p` задає позицію в екранному рядку (від 1 до 80), значення `r` — номер рядка екрана (від 1 до 25);
- *оператором задання кольору символів*, що виводяться на екран, `textcolor` (`n`) (від англ. text — текст і color — колір). Значення `n` задає номер кольору і є цілим від 0 до 15. Таблиця кольорів наведена нижче. Заданий колір зберігатиметься, доки не буде подана команда на зміну кольору;
- *оператором задання кольору фону* `textbackground` (`n`) (від англ. text — текст і background — фон). Значення номера кольору фону є цілим від 0 до 7. Після виконання цієї команди символи будуть виводитися на тлі заданого кольору.

Послідовність операторів `textbackground` (`n`) і `clrscr` дозволяє надати екрану заданого кольору `n`.

Зверніть увагу на останні рядки програми. Використовуючи кольори, візьміть за правило обов'язково в кінці програми поновлювати первісну кольорову гаму екрана!

ВИСНОВКИ

У програмах опрацювання величин вхідні значення звичайно вводять з клавіатури, а вихідні виводять на екран комп'ютера. Для введення значень із клавіатури застосовують оператор **read** або **readln**, для виведення на екран — оператор **write** або **writeln**. Оператори **write** і **writeln** використовують також для створення на екрані спеціальних форм для розміщення в них повідомлень. За допомогою операторів бібліотеки **Crt** можна управляти кольором символів і екрана, виводити повідомлення на задану позицію на екрані, очищати екран тощо

Контрольні питання та вправи

1. В алгоритмах опрацювання величин звичайно є три складові:
 - а) уведення значень ... величин;
 - б) опрацювання ... величин;
 - в) виведення значень ... величин.
 Вставте пропущені слова «вхідних» чи «вихідних».
2. Призначенням оператора **read** є:
 - а) виведення значення величин на екран;
 - б) уведення значень величин з екрана;
 - в) уведення значень величин з клавіатури.
3. Призначенням оператора **writeln** є виведення на екран:
 - а) імен величин;
 - б) виразів;
 - в) значень виразів.
4. Що є результатом виконання оператора **gotoxy** (10, 5)?
 - а) установлення курсора на п'яту позицію десятого рядка;
 - б) установлення курсора на десяту позицію п'ятого рядка;
 - в) надання значень $x = 10, y = 5$;
 - г) надання значень $y = 10, x = 5$.
5. У результаті виконання яких операторів на екран буде виведено значення 4?
 - а) **writeln** (2*2);
 - б) **writeln** (sqrt(4));
 - в) **writeln** (8 div 2);
 - г) **writeln** (sqr(2));
 - д) **writeln** (sqrt(2));
 - е) **writeln** ('2+2');
 - є) **writeln** (8 mod 2);
 - ж) **writeln** (4);
 - з) **writeln** (sqr(3) mod 5).
6. У результаті виконання яких операторів на екран буде виведено значення x , яке дорівнює $17/11$, у вигляді 1.54545?

а) writeln (x:6:1);	б) writeln (x:9:6);
в) writeln (x:8:7);	г) writeln (x:8:5);

д) `writeln(x:10:5);` е) `writeln(x:5:2)` .

7. Складіть програму «Моє ім'я». Програма має виводити на екран літери вашого імені, утворені з кольорових зірочок або інших символів на ваш вибір.

8. Складіть програму «Журналіст». Програма задає користувачеві три запитання: «Який об'єкт будеється?»; «Коли розпочалося будівництво?»; «Як ведуться роботи?». Надані відповіді використовуються для вставляння в стандартний текст-репортаж: «У нашому місті будеється Роботи розпочалися в ... році. Хоча вони тривають ...-ий рік, немає жодного сумніву: ... буде здано вчасно. Адже бригада працює ... !»

9. Складіть програму, яка вгадує задумане число. Програма виводить на екран пропозицію користувачеві: «Задумайте число, а я його відгадаю. Подвойте своє число. Додайте до нього 4. Поділіть суму на 2. Додайте 7. Помножьте на 3. Уведіть число, яке ви отримали:» і після введення числа повідомляє «Ви задумали число ...».

10. Складіть програму «Замовлення», яка виводить на екран бланк замовлення квитка на авіарейс. Пасажир у відведених полях вставляє своє прізвище, початковий і кінцевий пункти рейсу, вибирає клас салону (перший клас, бізнес-клас чи VIP-клас), поставивши «хрестик» проти вибраного варіанта.

11. Складіть програму виведення на екран запрошення. Текст запрошення складається стандартним способом за відповідями користувача на запитання програми. Оформіть запрошення умовним малюнком із застосуванням різних кольорів.

12. Складіть програму «Візитна картка». Програма виводить на екран бланк-запит для збирання даних користувача і виводить на екран його візитну картку.

арифметичний вираз, бібліотека `Str`, вираз, коментар, оператор введення, оператор виведення, оператор встановлення кольору екрана, оператор встановлення кольору фону, оператор очищення екрана, оператор установки курсору в задану позицію, формат виведення