

ББК 32.973я7  
Г54  
УДК 681.3

## Навчальний посібник

**Глинський Я.М., Анохін В.Є., Ряжська В.А.**  
Г54 Паскаль. Turbo Pascal і Delphi. 3-є вид. — Львів: «Деол»,  
2002. — 144 с.  
ISBN 966-7449-15-7

Третє стереотипне видання книжки призначене для методичної підтримки курсу програмування мовою Паскаль у середовищах Turbo Pascal 7.0 для MS-DOS чи Borland Pascal для Windows і Delphi x, який читається для студентів різних напрямків підготовки, а також є складовою шкільного курсу інформатики.

Наведено теоретичні відомості та збірник задач, а значна кількість прикладів і зразків розв'язування типових завдань стане у нагоді під час підготовки до практичних занять.

Для студентів та учнів шкіл.

Г 1404000000-015 Без оголошення **ББК 32.973я7**  
2002

Усі назви програмних продуктів є зареєстрованими товарними марками відповідних фірм.

Ніяка частина цієї книжки не може бути відтворена у жодній формі будь-якими засобами, включаючи ксерокопіювання, без дозволу видавництва. Право на видання книжки російською мовою належить видавництву «ДіаСофт».

ISBN 966-7449-15-7

© «Деол», 2002

## Розділ 1. Turbo Pascal 7.0

### § 1. Основні поняття та елементи мови

1. **Опис мови.** Мову Паскаль створив на початку 70-х років професор Н.Вірт зі Швейцарії. Вона названа на честь французького математика і філософа Блеза Паскаля (1623-1662) — винахідника першої у світі механічної обчислювальної машини. Мова вважається найбільш досконалою порівняно з іншими мовами програмування. Її використовують для розв'язування різноманітних задач.

Програми складаються з синтаксичних конструкцій, які називають командами (операторами, вказівками, реченнями). Команди будуються з *лексем* — неподільних елементів мови: слів, чисел, символів операцій тощо. Слова поділяються на службові, стандартні імена та імена (ідентифікатори), які користувач дає різним об'єктам. Розглянемо основні *службові слова* мови. Їх можна записувати як великими, так і малими літерами:

<b>and</b> — і	<b>mod</b> — остача
<b>array</b> — масив	<b>nil</b> — нуль
<b>begin</b> — початок	<b>not</b> — ні
<b>case</b> — вибір	<b>of</b> — з
<b>const</b> — стали	<b>or</b> — або
<b>div</b> — ділення без остачі	<b>packed</b> — стиснутий
<b>do</b> — виконати	<b>procedure</b> — процедура
<b>downto</b> — униз до	<b>program</b> — програма
<b>else</b> — інакше	<b>record</b> — запис
<b>end</b> — кінець	<b>repeat</b> — повторювати
<b>file</b> — файл	<b>set</b> — множина
<b>for</b> — для	<b>then</b> — то
<b>function</b> — функція	<b>to</b> — до
<b>goto</b> — перейти до	<b>type</b> — тип
<b>if</b> — якщо	<b>until</b> — доки
<b>in</b> — в	<b>var</b> — змінні
<b>label</b> — позначка	<b>while</b> — доки та інші.

**Стандартні імена.** Великий набір слів є проміжним між службовими словами та іменами користувача. Це стандартні імена. Їх є декілька груп:

- 1) *назви стандартних типів даних*: boolean (логічний), char (символьний), integer (цілий), real (дійсний), text (текстовий файл) тощо;
- 2) *назви стандартних сталих*: false (хибність), true (істинність), maxint (максимальне ціле), pi (число  $\pi$ ) тощо;
- 3) *назви стандартних функцій*: abs, arctan, cos, exp, ln, sin тощо;
- 4) *назви стандартних процедур*: read, readln, write, writeln тощо.

Стандартні імена користувач може використати для позначення інших об'єктів, однак тоді вони втрачають основне призначення.

Імена, які користувач надає величинам (сталим, змінним), програмі, підпрограмам тощо, мають складатися з *латинських* літер, цифр, символу “\_”. В імені цифра не може бути першим символом. Наприклад, програму можна назвати так: myprogram\_1.

*Зауваження.* У цій книжці службові слова будемо записувати малими літерами і виокремлюватимемо товстим шрифтом.

## 2. Структура програми. Програма складається із заголовка

```
program <ім'я програми>;
```

розділів описової частини

<b>uses</b>	— приєднання бібліотек та модулів;
<b>label</b>	— оголошення міток (позначок);
<b>const</b>	— оголошення сталих;
<b>type</b>	— опис типів;
<b>var</b>	— оголошення змінних;
<b>procedure</b>	— оголошення процедур користувача;
<b>function</b>	— оголошення функцій користувача

та виконуваної частини

```
begin  
  <розділ команд>  
end.
```

Заголовок та усі розділи, окрім останнього, є необов'язковими. Розділювачем між конструкціями (командами) програми є символ “;”. У кінці програми завжди має стояти крапка.

Заголовок програми надає програміст. В *іменах*, які користувач дає своїм програмам та змінним, великі і малі букви рівноправні: імена A та a (або MyName та myname) позначають один і той самий об'єкт.

У програму можуть входити коментарі. *Коментар* — фрагмент тексту програми, взятий в фігурні дужки або записаний так: (\* коментар \*). Коментар слугує для пояснення роботи програми і не впливає на виконання команд. Він може бути розташований у довільному місці програми.

*Зауваження.* Особливим видом коментаря є директива. Директиви призначені для задавання режимів функціонування компілятора. Як і коментар, директиву записують у фігурних дужках, де після відкриваючої дужки зазначають символ \$, наприклад, директива {\$F+} підтримує так звану fag-модель пам'яті для роботи з процедурами та функціями, {\$N+} забезпечує застосування математичного співпроцесора тощо (див. § 14).

**3. Розділи оголошення сталих і змінних.** Усі величини, які входять у програму, повинні бути описані у розділі сталих (констант), якщо вони не мінятимуть значення протягом виконання програми:

```
const <стала 1> = <значення 1>;
...
<стала n> = <значення n>;
```

або у розділі оголошення змінних, якщо вони обчислюватимуться:

```
var <список змінних 1> : <тип змінних 1>;
...
<список змінних n > : <тип змінних n>;
```

Елементи списків записують через кому. Кутові дужки <...> — це засіб формалізованого описування конструкцій мови. У конкретних програмах їх не використовують.

**4. Перша програма.** Програма — це послідовність команд, за допомогою яких записують алгоритм розв'язування задачі. Програми (алгоритми) складають за таким принципом: *вводять* дані, *визначають* потрібне, *виводять* результати. Аналогічно розв'язують задачі з математики та фізики, але тут обчислення вручну виконувати не потрібно — їх виконає комп'ютер.

Розглянемо програму з назвою Трукутрук для розв'язування задачі обчислення периметра  $p$  та площі  $s$  трикутника зі сторонами  $a=5$ ,  $b=3.6$ ,  $c=4.2$  за формулою Герона. Усі команди, наведені в програмі, будуть детально розглянуті нижче.

```
program Трукутрук;
uses Crt;           {Приєднуємо модуль Crt}
const a=5; b=3.6; c=4.2; {Вводимо довжини сторін}
var p,s: real;     {Оголошуємо змінні для}
begin             {периметра та площі}
  clrscr;         {Очищуємо екран}
  p:=a+b+c;      {Обчислюємо периметр}
  writeln('p=', p:5:2); {Виводимо значення периметра}
  p:=p/2;        {Обчислюємо півпериметр}
  s:=sqrt(p*(p-a)*(p-b)*(p-c)); {Визначаємо площу}
  writeln('s=', s:5:2); {Виводимо значення площі}
```

```
writeln('Виконав Панчишин Ю. ');
readln
end.
```

Символ \* позначає операцію множення, / — ділення, а  $\text{sqrt}(x)$  — це функція обчислення квадратного кореня з  $x$ . Два символи := позначають операцію присвоєння. Команда writeln призначена для виведення результатів. Після виконання програми на екрані отримаємо:

```
p= 12.80
s= 7.43
```

Виконав Панчишин Ю.

*Завдання.* Ознайомтеся з середовищем програмування Turbo Pascal (див. § 14). Розглянувши програму Трукутрук як зразок, розв'яжіть задачу № 1 свого варіанта з розділу "Задачі".

## § 2. Типи даних

**1. Стандартні типи даних.** Одним з найважливіших понять у програмуванні є змінна. *Змінна* — це поіменована ділянка оперативної пам'яті комп'ютера, де зберігається значення деякої величини. Змінна має такі властивості: назву (ім'я), значення, тип. Кількість змінних та їхні властивості задає користувач. Тип змінної визначає її допустимі значення, а також операції, які можна над нею виконувати. У програмі Трукутрук з § 1 змінні  $p$  та  $s$  оголошені як змінні числового дійсного типу.

Розглянемо стандартні типи даних: *числові цілі* (табл.1), *числові дійсні* (табл.2), *символьний* та *логічний*. Інші типи вивчатимемо далі.

Табл.1. Цілочислові типи

Назва типу	Діапазон значень
byte	0..255
shortint	-128..127
word	0..65535
integer (основний тип)	-32768..32767
longint	-2147483648..2147483647

Розглянемо такі дві змінні: `var x:byte; y:integer`. Цим змінним можна надати, наприклад, таких значень:  $x=65$ ;  $y=-5$ . Зауважимо, що змінна  $x$  не може набувати від'ємних значень, оскільки вони не входять до діапазону допустимих значень типу `byte`.

Табл.2. Дійсні типи

Назва типу	Діапазон
single	$1.5 \cdot 10^{-45}$ – $3.4 \cdot 10^{38}$
real (основний тип)	$2.9 \cdot 10^{-39}$ – $1.7 \cdot 10^{38}$

double	5.0 · 10 <sup>-324</sup> –1.7 · 10 <sup>308</sup>
extended	3.4 · 10 <sup>-4951</sup> –1.1 · 10 <sup>4932</sup>

Дійсні числа можна записувати у форматі з фіксованою крапкою, наприклад, 5.04, -12.109, або у форматі з плаваючою крапкою, наприклад, -5.2E+2 (це є -520.0), 16.1E-3 (це є 0.0161). Символи nE+2 означають множення числа n на 10 у степені +2, тобто

$$nEm = n \cdot 10^m$$

**Символьний** тип (char) — це множина символів кодової таблиці комп'ютера ASCII. Символьна стала — це символ, взятий у лапки, наприклад, 'a', '5', '%'.

**Логічний** тип (boolean) характеризується двома значеннями: **false** (хибність) та **true** (істинність).

Усі стандартні типи (окрім дійсного) є *впорядкованими*, тобто для кожного даного визначені наступне та попереднє значення.

**2. Нестандартні прості типи.** Усі наведені вище типи даних є стандартними. Їх не треба описувати у розділі типів. Користувач може визначити свої власні типи даних у розділі **type** так:

```
type <ім'я типу 1> = <опис типу 1>;
...
<ім'я типу n> = <опис типу n>;
```

До простих нестандартних типів належать перерахований, діапазонний та рядковий типи.

**Перерахований** тип утворюють з ідентифікаторів (імен користувача) шляхом їх об'єднання у список, який записують у круглих дужках:

```
type <ім'я типу> = (<значення 1>, <значення 2>, ...,
<значення n>);
```

*Приклад.* Опишемо два перераховані типи (дні тижня та кольори):

```
type week = (mon, tue, wed, fri, sat, sun);
colors = (red, green, yellow, white);
```

та оголосимо дві змінні — day та mycolor цих типів:

```
var day : week; mycolor : colors;
```

Цим змінним можна надати, наприклад, такі значення:

```
day := fri, mycolor := green, mycolor := yellow.
```

Номер першого елемента списку завжди є 0. Дані перерахованого типу не можна вводити з клавіатури чи виводити. Даними перерахованого типу не можуть бути числові чи символні значення.

**Діапазонний** тип — це звуження деякого базового упорядкованого типу. Його описують наступним чином:

```
type <ім'я типу> = <значення 1> .. <значення 2>;
```

*Приклад.* Розглянемо три типи, які описують усі дні тижня, робочі дні та номери місяців, і оголосимо змінні day1, day2, day3 та month цих типів:

```
type week = (mon,tue,wed,the,fri,sat,sun);
workdays = mon..fri;
months = 1..12;
var day1,day2 : week; day3 : workdays; month: months;
```

**Рядковий** тип даних (string). Значенням змінної рядкового типу може бути довільна послідовність, яка складається не більше, ніж з 255 символів. Змінні рядкового типу можна описати у розділі опису констант, типів, або оголосити у розділі змінних. Наприклад,

```
const s = 'Bye!'; type t = string[10]; var top : t; w : string.
```

У квадратних дужках зазначають довжину рядка. Сталі типу рядок використані в програмі Трукітрук у команді виведення writeln.

**3. Типовані сталі.** Окрім звичайних, є ще типовані сталі. Вони дають змогу оголошувати змінну й відразу надавати їй значення:

```
const <ім'я сталої 1> : <тип 1> = <значення 1>;
...
<ім'я сталої n> : <тип n> = <значення n>;
```

На відміну від звичайних, значення типованих сталих у програмі можна змінювати. Приклади типованих сталих:

```
const mysymbol : char = 'a'; n : integer = 5; p : real = 1.73.
```

### § 3. Прості програми

Прості (інша назва – лінійні) програми складаються з команд присвоєння, введення-виведення даних та викликів процедур.

#### 1. Команда присвоєння має вигляд

```
<ім'я змінної> := <вираз>;
```

*Дія команди.* Обчислюється вираз і його значення надається змінній. Вираз призначений для описування формул, за якими виконуються обчислення. Вираз може містити числа, змінні, сталі, назви функцій, з'єднані символами операцій.

Змінна і вираз мають бути одного типу або узгодженими: змінним дійсного типу можна надавати значення виразів цілого типу, а змінним рядкового типу присвоювати значення виразів символічного типу, але не навпаки.

*Приклад.* Розглянемо дію команд присвоєння в програмі Трукітрук: p:=a+b+c; p:=p/2; s:=sqrt(p\*(p-a)\*(p-b)\*(p-c)). Тут обчислюється значення периметра і воно надається змінній p, півпериметра (надається також змінній p) та площі (надається змінній s).

2. Основні операції, стандартні функції та деякі процедури мови Паскаль описані в табл. 3 та 4 відповідно.

Табл. 3. Основні операції

Пріоритет	Операції	Зміст операції
1	+, - not	Присвоєння знаку Логічне заперечення
2	*, / div, mod and	Множення, ділення Цілочислові операції Логічний добуток
3	+,- or, xor	Додавання, віднімання Логічні суми
4	=,<,>,<=,>= in	Відношення (порівняння) Належність до множини

Виконання кожної операції здійснюється з урахуванням її пріоритету (1 - найвищий). Для зміни звичайного порядку виконання операцій використовують круглі дужки, наприклад:  $2*3-2=-8$ ;  $2*(-3-2)=-10$ ;  $5*(2+13)=75$ ;  $20+100/20*5=45$ ;  $20+100/(20*5)=21$ .

Результатом операції div є ціла частина від ділення двох чисел, а mod - остача, наприклад:  $(9-2) \text{ div } 2=3$ ;  $7 \text{ mod } 2 = 1$ .

Означення логічних операцій будуть подані далі.

Табл. 4. Основні стандартні функції та процедури

Функція	Тип аргументу	Тип результату	Математичний запис, коментар
abs(x)	integer, real	integer, real	$ x $
arctan(x)	integer, real	real	$\arctg x$
cos(x)	integer, real	real	$\cos x$
sin(x)	integer, real	real	$\sin x$
exp(x)	integer, real	real	$e^x$
ln(x)	integer, real	real	$\ln x$
sqrt(x)	integer, real	real	$\sqrt{x}$
sqr(x)	integer, real	integer, real	$x^2$
ord(x)	char упорядкований	integer	ASCII-код сим্ব., номер елемента
succ(x)	упорядкований	упорядкований	повертає наступне значення x
pred(x)	упорядкований	упорядкований	повертає попереднє значення x

round(x)	real	integer	заокруглює число $x$ до цілого
trunc(x)	real	integer	відкидає дробову частину числа $x$
int(x)	real	real	відкидає дробову частину числа $x$
frac(x)	real	real	дробова частина числа $x$
odd(x)	integer	boolean	true ( $x$ - непарне), false ( $x$ - парне)
random(x)	integer	integer	Генерує випадкове число з діапазону від 0 до $x$
upcase(x)	char	char	замінює малу літеру латинської абетки на велику
<b>процедури:</b>			
inc(x,y)	integer	integer	збільшує $x$ на $y$
inc(x)	integer,char	integer,char	збільшує $x$ на 1
dec(x,y)	integer	integer	зменшує $x$ на $y$
dec(x)	integer,char	integer,char	зменшує $x$ на 1

Розглянемо приклади значень функцій і виконання процедур:

round(2.1)=2,            int(2.1)=2.0,            x:=1; inc(x,5); (x=6),  
 round(6.8)=7,            int(6.8)=6.0,            x:='a'; inc(x); (x='b'),  
 trunc(2.1)=2,            frac(2.1)=0.1,            x:=7; dec(x,3); (x=4),  
 trunc(6.8)=6,            frac(6.8)=0.8,            x:='d'; dec(x); (x='c').

Інші математичні функції можна виразити через основні. Наприклад,

$$\operatorname{tg} x = \frac{\sin x}{\cos x}, \quad \arccos x = \operatorname{arctg} \frac{\sqrt{1-x^2}}{x}, \quad \arcsin x = \operatorname{arctg} \frac{x}{\sqrt{1-x^2}},$$

$$x^\alpha = e^{\alpha \ln x}, \quad \log_b a = \frac{\ln a}{\ln b}.$$

**3. Команди введення (read, readln) даних.** Надавати значення змінним можна двома способами: за допомогою команди присвоєння, наприклад  $x:=5$ , або команд уведення даних з клавіатури. Другий спосіб робить програму більш універсальною, оскільки дає змогу розв'язувати задачі для різних значень змінних. Команда read має вигляд

```
read(<змінна 1> ,..., <змінна n>);
```

*Дія команди.* Виконання програми зупиняється. Система переходить у режим очікування введення даних (екран темний, миготить курсор). Значення цих даних користувач набирає на клавіатурі **через пропуск** або натискає після кожного даного на клавішу вводу. У результаті виконання цієї команди відповідним змінним будуть присвоєні конкретні значення.

Команда `readln` має вигляд

```
readln(<змінна 1>, ..., <змінна n>);
```

Вона діє як команда `read` з тою різницею, що зайві дані у рядку введення ігноруються. Наступна команда вводу читатиме дані з нового рядка. Цю команду застосовують під час роботи з текстовими файлами (див. § 10).

Розглянемо програму `Trykutnyk`. Значення сторін трикутника можна ввести, наприклад, так: `readln(a,b); read(c)`.

Тоді під час виконання програми на клавіатурі набираємо

5 3.6 (натискаємо на клавішу вводу)

4.2 (натискаємо на клавішу вводу)

У результаті виконання команд `readln(a,b)` та `read(c)` змінні отримують такі значення:  $a=5$ ;  $b=3.6$ ;  $c=4.2$ .

*Зауваження.* Команду `readln` без параметрів часто використовують у середовищі TP для MS-DOS, щоб оглянути результати виконання програми на екрані. Щоб після цього перейти у режим редагування програми, потрібно натиснути на клавішу вводу. У середовищі TP для Windows система виводить результати у окремому вікні, тому там цю команду писати не потрібно.

*Зауваження.* Значення змінних логічного й перерахованого типу вводити з клавіатури *не можна*.

**4. Команди виведення (`write`, `writeln`) даних.** Для виведення на екран повідомлень та результатів обчислень використовують команди `write` та `writeln`:

```
write(<вираз 1>, <вираз 2>, ..., <вираз n>);
```

У списку виведення можуть бути сталі, змінні або вирази.

*Дія команди.* Сталі, значення змінних та виразів виводяться на екран у вікно виведення, яке можна переглянути за допомогою комбінації клавіш `Alt+F5`.

Команда

```
writeln(<вираз 1>, ..., <вираз n>);
```

діє майже так само як і команда `write`; різниця така: наступна після неї команда `write` чи `writeln` буде виводити значення на екран у новому рядку.

Для переходу на новий рядок екрана чи для пропуску рядка використовують команду `writeln` без параметрів.

Розглянемо програму `Труктрук`. Якщо в цій програмі задати такі команди виведення: `write('p=',p); writeln('s=',s); writeln('Виконав Панчишин Ю.')`, то для заданих вхідних даних  $a=5$ ,  $b=3.6$ ,  $c=4.2$ , на екрані отримуємо такий результат:

```
p= 1.2800000000E+01s= 7.429239530E+00
```

Виконав Панчишин Ю.

**5. Форматний вивід.** Команди `write` та `writeln` можуть здійснювати форматний вивід даних. Форматування — це подання результатів у наперед заданому користувачем вигляді. Для цього після виразу через двокрапку записують число (`:n`) — кількість позицій на екрані, які треба надати для виведення значення цього виразу. Формат `:n` застосовують для даних *цілого та рядкового* типів. Під час виведення даного *дійсного* типу зазначають загальну кількість позицій для всіх символів (`n`) та кількість позицій для дробової частини (`m`), тобто формат має вигляд `:n:m`.

Розглянемо команду виведення `writeln('p=',p:5:2)` у програмі `Труктрук`. Вона забезпечує виведення на екран рядкової сталої `'p='` без лапок (з метою пояснення, що обчислено) і числа `12,8` у форматі `:5:2`, тобто так: `p= 12.80`.

**Задача 1.** Дано координати трьох вершин трикутника  $A(1;1)$ ,  $B(2;2)$  та  $C(-1;2)$ . Обчислити медіану  $m_b$  та радіус описаного кола  $r$ .

```
program ТруктрукNew;
uses Crt;
var x1,y1,x2,y2,x3,y3,a,b,c,mb,r,x,y,p,s: real;
begin
  clrscr;
  writeln('Введіть координати:');
  readln(x1,y1,x2,y2,x3,y3);
  a:=sqrt(sqrt(x3-x2)+sqrt(y3-y2)); {Обчислимо довжини}
  b:=sqrt(sqrt(x1-x3)+sqrt(y1-y3)); {сторін трикутника}
  c:=sqrt(sqrt(x1-x2)+sqrt(y1-y2));
  x:=(x1+x3)/2; {Обчислимо координати}
  y:=(y1+y3)/2; {середини сторони b}
  mb:=sqrt(sqrt(x-x2)+sqrt(y-y2)); {Обчислимо медіану mb}
  p:=(a+b+c)/2; {Обчислимо півпериметр}
  s:=sqrt(p*(p-a)*(p-b)*(p-c)); {Обчислимо площу}
  r:=a*b*c/(4*s); {Обчислимо радіус}
  writeln('mb=',mb:5:2); {Виведемо результати}
  writeln('r=',r:5:2); {Виведемо радіус}
  readln
end.
```

*Зауваження.* Якщо не використовувати форматний вивід, то для даних цілого та логічного типу на екрані буде надано до 15 позицій, для даних дійсного типу — 18. Дані дійсного типу виводяться у вигляді  $x.xxxxxxxxЕзнакxx$ , де  $x$  — будь-яка цифра, що часто є надлишковим. Такі числа на екрані можуть зливатися.

*Завдання.* Розв'язати задачу № 2 з розділу “Задачі” свого варіанта.

*Довідка.* Для розв'язування типових задач про трикутник наведемо формули обчислення деяких величин:

віддаль між точками  $(x_1, y_1), (x_2, y_2)$ :  $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ ;

координати середини відрізка:  $x = (x_1 + x_2)/2, y = (y_1 + y_2)/2$ ,

півпериметр трикутника:  $p = (a + b + c)/2$ ;

площа трикутника:  $s = \sqrt{p(p - a)(p - b)(p - c)}$ ;

висоти трикутника:  $h_a = 2s/a, h_b = 2s/b, h_c = 2s/c$ ;

бісектриси трикутника:

$W_\alpha = \frac{2}{(b+c)} \sqrt{bc p(p-a)}, W_\beta = \frac{2}{(a+c)} \sqrt{ac p(p-b)}, W_\gamma = \frac{2}{(a+b)} \sqrt{ab p(p-c)}$ ;

радіус описаного кола:  $r = abc/(4s)$ ;

радіус вписаного кола:  $r = s/p$ , де  $a, b, c$  — сторони трикутника.