1. ОСНОВНИ ПОНЯТИЯ В ЕЗИКА ПАСКАЛ

1.1. Речник на езика

Всеки език се състои от речник и система от синтактични правила, по които се изграждат конструкциите на езика. Синтактичните правила ще бъдат разгледани по-нататък заедно със съответните езикови конструкции. Речникът на езика Паскал включва *букви, цифри и специални символи*. Към буквите се отнасят всички главни и малки букви от латинската азбука. При това компилаторът на Паскал не прави разлика между главни и малки букви, например **A** и **a** се считат за една и съща буква. Под цифри се разбират арабските цифри от 0 до 9. Специалните символи са:

+	плюс	'	апостроф
-	минус	:	двоеточие
*	звездичка	.	точка
/	наклонена черта	^	стрелка нагоре
:=	символ за присвояване	..	две точки
;	точка и запетая	,	запетая
=	равно	(	лява малка скоба
<>	различно	)	дясна малка скоба
<	по-малко	[	лява средна скоба
>	по-голямо	]	дясна средна скоба
<=	по-малко или равно	{	лява голяма скоба
>=	по-голямо или равно	}	дясна голяма скоба

Към специалните символи спадат и т.н. *"запазени (служебни) думи"*. Те имат точно определено значение в програмата и не могат да бъдат използвани по друг начин. Запазени думи в езика Турбо Паскал са:

and	downto	inherited	or	to
array	else	inline	packed	then
asm	end	interface	procedure	type
begin	exports	label	program	until
case	file	library	record	uses
const	for	mod	repeat	var
constructor	function	nil	set	while
destructor	goto	not	shl	with
div	if	object	shr	
do	in	of	string	

Запазените думи спадат към специалните символи, защото в програмата те се интерпретират именно като символи.

1.2. Имена

За означаване на различните обекти в програмата - данни и части от текста в Паскал се използват *имена (идентификатори).* Те се избират от програмиста. Състоят се от букви и цифри като започват задължително с буква. Дължината им може да бъде произволно голяма, но в различните реализации на езика тя е ограничена до определен брой символи. Запазените думи от речника на Паскал

не могат да бъдат използвани като идентификатори.

Правилни са, например, следните имена: MaxN, FakNom, NomRed, A357N, EGN, FirstDay и End1.

Няколко примера за неправилни имена:

357AN - започва с цифра;
First-Row - съдържа знак минус;
Max N - съдържа празен интервал;
End - запазена дума.

В Паскал са дефинирани имена, наречени *стандартни*, чието предназначение е предварително определено - за означаване на стандартни величини, помощни средства и др. Тези имена могат да се използват от програмиста без да се дефинират. Употребата на същите имена с друго предназначение е допустима, но е нежелателна, защото прави невъзможно използването им по стандартното им предназначение. Стандартни имена са:

abs	boolean	reset	arctan	maxint
rewrite	sin	new	round	char
odd	output	chr	ord	sqr
cos	eoln	sqrt	eof	pack
succ	true	page	text	exp
pred	read	false	put	trunc
get	integer	unpack	input	
write	real	readln	writeln	

Тяхното предназначение ще бъде обсъдено в следващите глави.

1.3. Константи и променливи

Всяка програма е предназначена за обработка на някакви данни. Те биват *константи и променливи*. **Променливите** могат да получават нови стойности по време на изпълнението на програмата, т.е. те могат да се променят докато програмата се изпълнява. За всяка променлива от програмата програмистът задължително избира *име* (*идентификатор*), отговарящо на правилото за избор на име, дадено по-горе. **Константите** не могат да се променят по време на изпълнението на програмата.

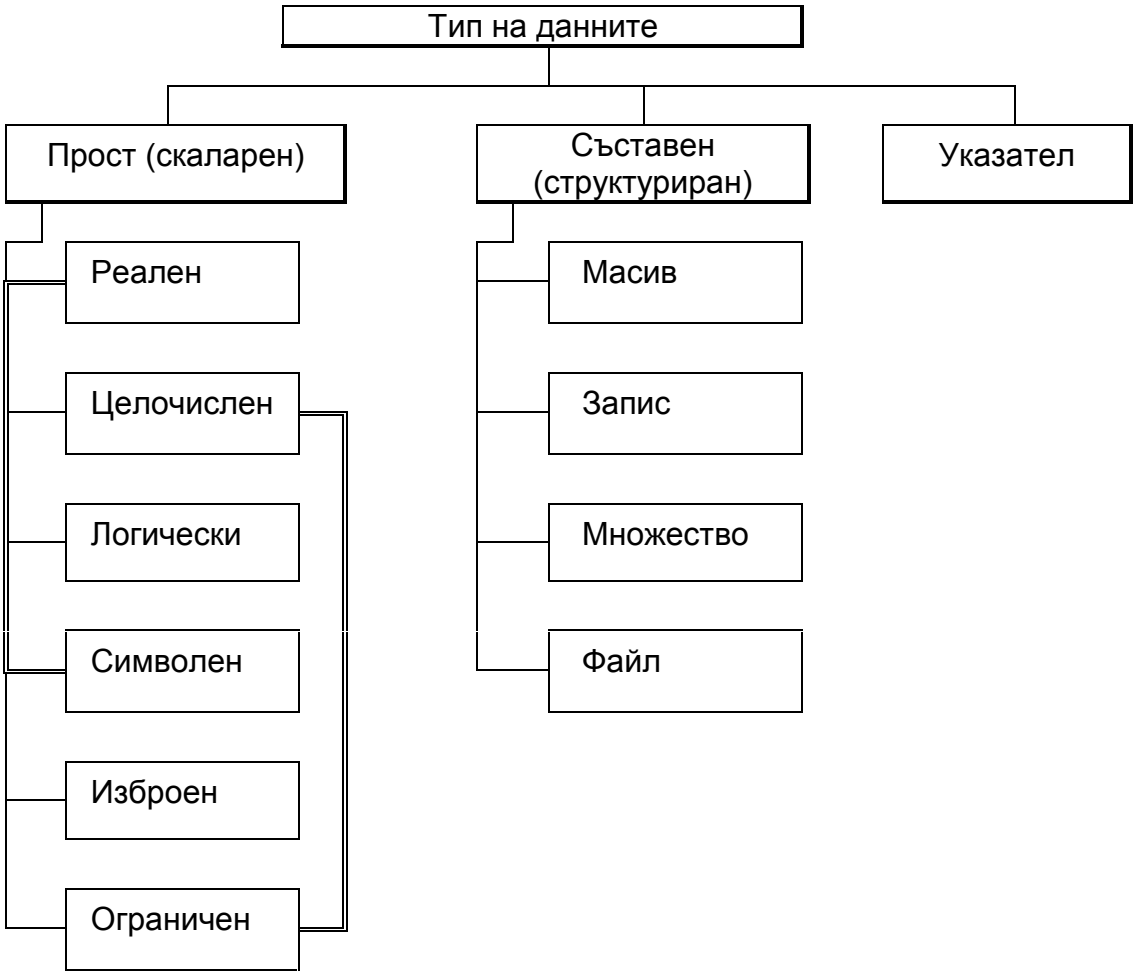
Съществува възможност на константа да се съпостави име. Такава константа е *именувана константа* и по-нататък в програмата може да се използва името вместо самата константа. В много случаи това създава редица удобства за програмиста. Паскал обаче позволява в програмата на името на такава константа да се съпоставят нови стойности, но при следващо стартиране на програмата, константата отново е с първоначалната си стойност. Ето защо на използването на именовани константи се гледа и като на задаване на начални стойности.

1.4. Тип на данните

Типът на данните (константи и променливи) е важна тяхна характеристика. Той определя диапазона от стойности, които данните от този тип могат да приемат, и операциите (действията), които могат да се извършват с тях. Всяка величина, използвана в програмата може да бъде от един единствен тип.

Типовете на данните, използвани в езика Паскал, се делят на три групи: ***прости (скаларни), съставни (структурирани) и указатели*** (Фиг. 1.1).

Данните от **скаларните** типове могат да имат само една стойност - едно число, един символ и др. Скаларните типове включват четири стандартни типа (***целочислен, реален, символен и логически)*** и два нестандартни типа (***изброен и ограничен***). От тях само ***реалният тип не е дискретен тип, останалите са дискретни типове***. Стойности на данните от реален тип могат да бъдат реалните числа. Стойностите на данните от дискретен тип могат да бъдат от предварително определено наредено множество от стойности. Стандартните скаларни типове са предварително дефинирани, т.е. те имат служебни имена и диапазон от стойности, докато нестандартни скаларни типове се описват от програмиста по определени правила, т.е. програмистът им определя имената и диапазона от допустимите стойности.



Фиг. 1.1

Данните от **съставните** (структурираните) типове представляват даннови структури, обединяващи множество компоненти. Съставните типове се изграждат и описват от програмиста от стандартни типове и/или предварително изградени и описани други типове данни и указатели. В Паскал могат да се дефинират четири типа съставни данни: ***масиви, множества, записи и файлове***.

Указателите са тип данни, които указват мястото в оперативната памет, където са разположени специални структури от данни, наречени динамични структури, които програмата създава и унищожава докато се изпълнява.

В следващите раздели ще бъдат разгледани начините за дефиниране и използване на всички допустими в Паскал типове данни, без типа указател.

1.5. Обща структура на програмата

Общата структура на програма на Паскал може да се види от следния пример:

```
Program Sum;                    Заглавие на програмата
Var
  Num1,Num2,Res : integer;      Описателна част
Begin
  Readln(Num1,Num2);
  Res := Num1 + Num2;           Изпълнима част
  Writeln(Res)
End.
```

Както се вижда, програмата на Паскал се състои от заглавие и две части:

- декларативна или описателна част, в която се описват обектите, участващи в програмата;
- изпълнима част, в която се описват операциите, които трябва да изпълни компютърът.

Заглавието съдържа служебната дума **Program** и името на програмата, което се избира от програмиста. В последните версии на Турбо Паскал не е задължително програмата да има заглавие.

Описателната част може да съдържа следните пет раздела:

- раздел за дефиниране на етикети, започва със служебната дума **Label**;
- раздел за дефиниране на константи, започва със служебната дума **Const**;
- раздел за дефиниране на нестандартни типове данни, започва със служебната дума **Type**;
- раздел за дефиниране на променливи, започва със служебната дума **Var**;
- раздел подпрограми, в който се разполагат подпрограмите по определен ред.

Присъствието на всеки раздел в описателната част се определя от неговата потребност. Например, ако в програмата не се използват етикети, разделът **Label** отсъства. Желателно е посоченият по-горе ред да бъде спазван, но когато се налага може и да се наруши - например, ако трябва да се опише някаква съставна константа, първо в раздел **Type** трябва да се опише нейният тип и едва след това в раздел **Const** да се зададе самата константа. Допуска се даден раздел да се появи няколко пъти, например **Const, Type, Const** ... При дефиниране на обект не се допуска да се използват недефинирани обекти. В тази и следващите глави предназначението на всеки от разделите ще бъде разгледано по-подробно.

Изпълнимата част се състои от операторите, описващи действията, които трябва да изпълни компютърът съгласно алгоритъма. Тя започва със служебната дума **begin** и завършва с **end** с точка накрая. С цел да се постигне по-голяма прегледност, операторите се записват на отделни редове, но това не е задължително. Дългите оператори могат да се разположат на няколко реда, а късите - да се записват по няколко на един ред. Всеки оператор завършва със знака точка и запетая (;), който го отделя от следващия оператор. Този знак може да се пропусне, ако след оператора следва някоя от запазените думи **end** или **until**.

Максималната дължина на един програмен ред е 255 символа.

1.5.1. Съставен оператор

В програмите на Паскал често се налага определена последователност от оператори да се представи като един оператор, наречен **съставен оператор**. Това се постига като се постави пред първия от тези оператори служебната дума **begin**, а след последния - **end**, т.е. съставният оператор има вида:

begin *оператор1; оператор2; . . ., операторN* **end***;*

1.5.2. Коментари

Програмата на Паскал може да съдържа пояснителни текстове. Такъв текст се нарича коментарен и се огражда с големи скоби **{ }** или двойките символи **(*** и ***)**. Например:

{ Това е коментарен текст }

(* Това също е коментар *)

Коментарните текстове се игнорират от транслатора, т.е. не намират място в изпълнимата програма. Те служат само за подобряване четливостта на текста на програмата.

1.6. Стандартни типове данни

1.6.1. Реален тип

Данните, чиито стойности са познатите от математиката реални числа, са от реален тип. Турбо Паскал поддържа пет подтипа на реалния тип. Техните характеристики са дадени по-долу в таблицата.

Както се вижда от таблицата, всеки подтип се характеризира със служебно име, интервал на допустимите стойности и точност. Точността е указана с броя на значещите (верните) цифри, с които се представя числото в компютъра.

Име на подтипа	Интервал на допустимите стойности	Точност - брой значещи цифри	Необходима памет
Real	$\pm 2{,}9.10^{-39}$ . . $\pm 1{,}7.10^{+38}$	12	6 байта
Single	$\pm 1{,}5.10^{-45}$ . . $\pm 3{,}4.10^{+38}$	8	4 байта
Double	$\pm 5{,}0.10^{-324}$ . . $\pm 1{,}7.10^{+308}$	16	8 байта
Extended	$\pm 3{,}4.10^{-4951}$ . . $\pm 11.10^{+4932}$	20	10 байта
Comp	$-9{,}2.10^{+18}$ . . $9{,}2.10^{+18}$	20	8 байта

Знакът две последователни точки (**. .**) в Паскал се използва при указване на интервал.

Реалните константи представляват реални числа и се записват както е прието в математиката, но вместо запетая, се използва десетична точка (.). Реалните константи се представят по два начина:

- с **фиксирана точка** - състои се от цифри и десетична точка и може да започва със знак + или -, както е показано със следните примери: 4.3472, 0.0, -0.5, -0.00021, 34562.045 и -12.0

- с **плаваща точка** - състои се от мантиса (цяло число или реално число с фиксирана точка), буквата E и цяло число с или без знак. Стойността на

константата е равна на произведението от мантисата и десет на степен, указана с числото след буквата E, както е показано със следните примери: 0.43472E1, 0E0, -5E-1, -0.21E-03, 3.4562045E+04 и -0.12E2

Няколко примери за неправилни реални константи:

0.08.08 - съдържа две точки;
1.2E-3.5 - числото след буквата E не е цяло;
E5 - липсва число пред E.

Именувани константи се задават в раздел **Const**, например

```
Const
 E=2.71828182845 90 45;
```

Подтипът на реалната константа се определя автоматично от броя на значещите цифри.

1.6.2. Целочислен тип

Данните, чиито стойности са познатите от математиката цели числа, са от целочислен тип. Турбо Паскал поддържа пет подтипа на целочисления тип. Техните характеристики са дадени по-долу в таблицата.

Както се вижда от таблицата, всеки подтип се характеризира със стандартно (служебно) име и интервал. Подредбата на стойностите от множеството на допустимите стойности съответства на подредбата на целите числа в математиката.

Име на подтипа	Интервал на допустимите стойности	Формат на представяне
ShortInt	-128 . . 127	1 байт. Първият бит е знаков.
Integer	-32768 . . 32767	2 байта. Вторият бит е знаков.
LongInt	-2147483648 . . 2147483647	4 байта. Вторият бит е знаков.
Byte	0 . . 255	1 байт. Без знаков бит.
Word	0 . . 65535	2 байта. Без знаков бит.

Целочислените константи представляват цели числа и в програмата се записват като цели числа с или без знак + или -. Знакът е задължителен само, когато константата е отрицателно число. Подтипът на цялата константа се определя автоматично по броя на цифрите в нея и по това дали има знак + или -.

Примери за правилни целочислени константи са:

-73, 258, 0, 10345 и -9300

Следните целочислени константи са записани неправилно:

-+34 - съдържа два знака един след друг;
6.0 - съдържа символ, различен от цифра и знак.

Програмистът може да декларира и използва именувани целочислени константи. Те се декларират в раздела **Const**, например

```
Const
  BrStudenti=100; BrGrupi=10; BrRedove=30; Brkoloni=20; Mmax=50;
```

Съществуват и стандартни целочислени константи, например MaxInt, която е равна на 32767.

1.6.3. Логически тип

Служебното име на този стандартен тип е **boolean**.

Логическите константи са само две: **True** (истина) и **False** (лъжа).

1.6.4. Символен тип

Всяка изчислителна система използва предварително определена съвкупност от символи. Съществуват няколко стандарта за такива съвкупности, които се различават по броя на символите и подреждането им. Най-често се използва стандартът ASCII. Той съдържа 256 символа, всеки от които има номер (код) - едно от числата от 0 до 255. Тези 256 символи образуват подредено множество от стойности, които могат да приемат данните от символен тип. Подредбата на символите съответства на реда на номерата им. Служебното име на този стандартен тип е **char**.

Символната константа представлява ASCII - символ, заграден с апострофи. Примери за правилни символни константи:

'A' '>' '6' 's' '@'

1.7. Тип символен низ

Типът символен низ не съществува във всички реализации на Паскал. В Турбо Паскал такъв тип съществува и тук ние ще имаме предвид него.

1.7.1. Константи от тип символен низ

Те представляват произволна поредица от символи, оградени с апострофи (' '). Допустими са всички символи от ASCII - стандарта. Максималната дължина на константата е 255 символа. Когато константата съдържа символа апостроф, той се представя с два последователни апострофа, за да се разграничи от апострофа, указващ края на низа.

Съществува константата празен символен низ. Тя има вида ''.

Примери за константи от тип символен низ:

'Това е пример за константа от тип символен низ'
'Множимо:'
'гр. Русе, ул. “Студентска”, 8'
'Олимпиада''98'.

Могат да се декларират и именувани константи от тип символен низ:

```
Const
  S = 'Който се учи, той ще сполучи!';
```

В глава 8 ще разгледаме операциите със символни низове в Паскал.

1.8. Дефиниране на променливи

Променливите – *тези данни, чиито стойности се променят по време на изпълнение на програмата* – винаги се означават с имена. Дефинирането на променливите представлява указание на името и типа на всяка променливата. Общият вид на една дефиниция е следния:

Var *списък от имена : тип на променливите;*

В списъка имената на променливите се разделят с запетая.

Примери:

```
Var
    XCoor,YCoor : real;
```

```
FMin,FMax,FMid : single;
P,Q: double;
U,VW:extended;
Mstart, MStop : shortint;
Nmin, Nmax : integer;
Nmid : integer;
K, L: longint;
A, B, C: byte;
X,Y: word.
Log1, Log2 : boolean;
SimProm1, SimProm 2 : char;
Str1, Str2: string[40];
StrA,StrB : string[10];
StrX : string;
```

String е служебна дума за указване на типа символен низ. Числото в скоби указва максимално допустимата дължина на символния низ в брой символи и не може да надвишава 255. Тук стойностите на Str1 и Str2 могат да бъдат символни низове не по-дълги от 40 символа, а стойността на StrA и StrB - символни низове не по-дълги от 10 символа. Когато не е указана максимално допустимата дължина, както е при StrX, се подразбира, че тя е 255 символа.