

Лабораторно упражнение № 3:

Въведение в езика C.

времетраене: 2 уч. часа

3.1 Програмен език C – Основни елементи.

- **азбука** - Това е набора от разрешените за езика символи (крайно, предварително фиксирано множество), чрез които се изграждат отделните смислови единици в него;
- **лексема (дума)** - Най-малката смислова единица в езика;
- **лексика** - Тя включва множеството от правила за изграждане на лексемите на езика;
- **граматика** - Дефинира правилата, по които се изграждат синтактичните и семантичните (смисловите) единици на езика;

табл. 1

auto	asm	break	case	char
const	continue	default	do	double
else	enum	extern	float	for
goto	if	Int	long	register
return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned
void	volatile	while		

- **ключови думи**

- **идентификатори**

- **разделители**

- **коментари**

- **понятие за програмен обект**

Всяка програма, написана на процедурен език, може да се разглежда като формално описание на начина, по който се обработва входната информация (множество на входните данни). В резултат от тази обработка се получава определена съвкупност от резултати (множество на изходните данни). В този смисъл всяка програма може да се разглежда като функционален обект, който за краен брой стъпки преработва входното множество до изходното множество от данни

програмен обект  (идентификатор, тип, стойност)

където:

идентификатор -> име (идентификатор) на обекта;
тип -> тип на обекта (начин на представяне);
стойност -> величина (количествена характеристика).

Може да се счита, че **името, типът и стойността** са трите характеристики, свойства на програмния обект, които го дефинират еднозначно в програмната единица, в която е описан.

Идентификаторът на обекта е характеристиката, която осигурява възможностите за достъп и опериране с него в програмната единица - в **ЕВН достъпът до данните се осъществява чрез имената на обектите, които ги описват.**

Типът е свързан с машинното представяне на данните в оперативната памет (ОП) на компютъра - той е характеристика, която "приобщава" програмния обект към определен клас (по-скоро начин) на представяне. Най-общо е свързан с броя байтове от ОП, в които се съхранява стойността на обекта.

Стойността на обекта може да се определи като вид количествена характеристика на обекта. Като пример може да се разгледа дефиницията:

address	b7	b6	b5	b4	b3	b2	b1	b0
...	x	x	x	x	x	x	x	x
α	x	x	x	x	x	x	x	x
$\alpha+1$	x	x	x	x	x	x	x	x
$\alpha+2$	0	0	0	0	0	0	0	0
$\alpha+3$	0	0	1	1	0	1	1	0
$\alpha+4$	x	x	x	x	x	x	x	x

operand1
↑
operand2

фиг.1

int operand1, operand2=54;

Видове обекти - променливи и константи

така наречените оператори за дефиниране на програмни обекти. Този подход задължително се прилага в С за дефиниране на обекти - променливи.

Константите в С, съгласно ANSI стандарта се наричат **литерали**. В контекста определението за програмен обект, трябва да се посочи, че за константите липсва атрибута идентификатор. Той не е необходим, тъй като всички останали атрибути на обекта се съдържат в самия запис на обекта (т.е. чрез записа на конкретната му стойност).

3.2 Типове данни в езика С.

Класификация:



3.3 Бройни системи

Видове бройни системи:

- **непозиционни бройни системи**

Непозиционната бройна система е тази, при която стойността на цифрата не зависи от нейното място в поредицата от цифри с които е записано числото.

- **позиционни бройни системи**

В общия случай за числата в бройна система с основа Q :

$$(a_n a_{n-1} \dots a_1 a_0, c_1 c_2 c_3 \dots)_Q = \sum_{k=0}^n a_k Q^k + \sum_{k=1}^{\infty} c_k Q^{-k}$$

където Q^k и Q^{-k} са теглата на съответните цифри, а k — тяхната позиция в записа на числото.

- **смесени бройни системи**

Римска бройна система

I или **i** за едно,

V или **v** за пет,

X или **x** за десет,

L или **l** за петдесет,
C или **c** за сто (лат. *centum*),
D или **d** за петстотин
M или **m** за хиляда (лат. *mille*),

числата от 1 до 100 с римски цифри

	1	2	3	4	5	6	7	8	9	
	I	II	III	IV	V	VI	VII	VIII	IX	
10	X	XI	XII	XIII	XIV	XV	XVI	XVII	XVIII	XIX
20	XX	XXI	XXII	XXIII	XXIV	XXV	XXVI	XXVII	XXVIII	XXIX
30	XXX	XXXI	XXXII	XXXIII	XXXIV	XXXV	XXXVI	XXXVII	XXXVIII	XXXIX
40	XL	XLI	XLII	XLIII	XLIV	XLV	XLVI	XLVII	XLVIII	XLIX
50	L	LI	LII	LIII	LIV	LV	LVI	LVII	LVIII	LIX
60	LX	LXI	LXII	LXIII	LXIV	LXV	LXVI	LXVII	LXVIII	LXIX
70	LXX	LXXI	LXXII	LXXIII	LXXIV	LXXV	LXXVI	LXXVII	LXXVIII	LXXIX
80	LXXX	LXXXI	LXXXII	LXXXIII	LXXXIV	LXXXV	LXXXVI	LXXXVII	LXXXVIII	LXXXIX
90	XC	XCI	XCII	XCIII	XCIV	XCV	XCVI	XCVII	XCVIII	XCIX

- **Преминаване от една позиционна бройна система в друга**

преобразуване от произволна позиционна система в десетична

преобразуване от десетична в произволна позиционна бройна система

преобразуване от двоична бройна система в 16-тична и обратно

3.4 Вътрешно представяне на аритметичните данни:

- тип **int** – целочислени (в оперативната памет се представят в 2 (4) съседни байта – за различните компилатори като се запазва следната зависимост:

char ≤ short ≤ int ≤ long ≤ float ≤ double

- тип **char** – към **int** - 1 байт

Type	Size
char, unsigned char, signed char	1 byte
short, unsigned short	2 bytes
int, unsigned int	4 bytes

long, unsigned long	4 bytes
Float	4 bytes
Double	8 bytes
long double¹	8 bytes

3.5 Запис на int константи в езика C.

- десетични (**DEC**) – символи (цифри) (**0,1,2,3,4,5,6, 7,8,9**);
- осмични (**OCT**) – започват задължително със символа **0** последван от символите (**0-7**);
- шеснадесетични (**HEX**) – започват със символите **0x**, последвани от символите (**0...9, A, B, C, D, E, F**). Символите (**10**), **B(11)**, **C(12)**, **D(13)**, **E(14)** и **F(15)** са общоприетите знаци (букви) на цифрите за запис във шеснадесетична бройна сиситема.

3.6 Запис на char константи в езика C.

- Прости знакови литерали (един символ)
- Символни низове.

Например:

‘C’ - където: C е произволен символ от **ASCII** таблицата за стандартно кодиране на символите съгласно стандарта **ANSI**. Изключение правят управляващите символи започващи с ‘\’ последван от един символ, които служат за управление на периферните устройства.

- управляващи ASCII символи

литерал	интерпретация	ASCII означение
'\n'	Нов ред	new line
'\b'	Позиция назад	backspace
'\r'	връщане на каретката в началото на реда	carriage return
'\f'	нова страница	form feed
'\a'	звънец	bell
'\\'	обратна наклонена черта	backslash
'\''	апостров	apostrophe
'\0'	ASCII код 0	NULL
'\t'	хоризонтална табулация	HT
'\v'	вертикална табулация	VT

3.7 Изброими типове данни в езика C.

формат: `enum Name_enumeration { name_val1=const1, name_val2=const2, };`

примери:

`enum BOOLEAN { false, true};`

/* дефинира се изброим тип с име **BOOLEAN**, състоящ се от множеството от два елемента :

`false` – със стойност **0** и `true` със стойност **1**

`enum ANYDAY { Monday=1, Tuesday, Wednesday, Thursday,`

`Friday,Saturday,Sunday } week, work_week ;`

/* дефинира се изброим тип със **7** елемента с име **ANYDAY**, като са дефинирани две променливи `week` и `work_week`, елемента `Monday` се инициализира със стойност **1** , `Tuesday` със стойност **2** и т.н.

задание за самостоятелна работа:

Да се състави блок алгоритъм и C-програма за пресмятане на квадратно уравнение от вида:

$ax^2 + bx + c = 0$, където **a,b,c** са коефициенти и се въвеждат от клавиатурата. В алгоритъма да се предвиди анализ на въвежданите коефициенти.