

Основни елементи на C++

Азбука

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	^	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

Источник: wikipedia.org

Синтаксис

- Правила за построяване на текст
- **Иван чете интересна книга.**
- **Студентът пише програма.**
- **книга. чете Иван? интерес на**
- $\langle \text{изречение} \rangle ::= \langle \text{подлог} \rangle \langle \text{сказуемо} \rangle [\langle \text{определение} \rangle] \langle \text{допълнение} \rangle .$
- $\langle \text{подлог} \rangle ::= \langle \text{собствено_съществително} \rangle | \langle \text{нарицателно_съществително} \rangle \langle \text{пълен_член} \rangle$
- $\langle \text{пълен_член} \rangle ::= \text{ът} | \text{ят} | \text{та} | \text{то}$

Синтаксис

- **Иван чете интересна книга.**
- **Студентът пише програма.**
- $\langle \text{изречение} \rangle ::= \langle \text{подлог} \rangle \langle \text{сказуемо} \rangle [\langle \text{определение} \rangle] \langle \text{допълнение} \rangle .$
- $\langle \text{подлог} \rangle ::= \langle \text{собствено_съществително} \rangle | \langle \text{нарицателно_съществително} \rangle \langle \text{пълен_член} \rangle$
- $\langle \text{пълен_член} \rangle ::= \text{ът} | \text{ят} | \text{та} | \text{то}$
- $\langle \text{сказуемо} \rangle ::= \langle \text{глагол} \rangle$
- $\langle \text{определение} \rangle ::= \langle \text{прилагателно} \rangle$
- $\langle \text{допълнение} \rangle ::= \langle \text{собствено_съществително} \rangle | \langle \text{нарицателно_съществително} \rangle$

Синтаксис

- <изречение>
- <подлог> <сказуемо> [<определение>] <допълнение>.
- <собствено_съществително> <сказуемо> <определение> <допълнение>.
- **Иван** <глагол> <определение> <допълнение>.
- **Иван чете** <определение> <нарицателно_съществително>.
- **Иван чете** <прилагателно> **книга**.
- **Иван чете интересна книга**.

Синтаксис

- <изречение>
- <подлог> <сказуемо> [<определение>] <допълнение>.
- <нарицателно_съществително><пълен_член> <сказуемо> <допълнение>.
- **Студент**<пълен_член> <глагол> <допълнение>.
- **Студентът** <глагол> <нарицателно_съществително>.
- **Студентът** <глагол> **програма**.
- **Студентът** **пише** **програма**.

Синтаксис

- <изречение>
- <подлог> <сказуемо> [<определение>] <допълнение>.
- <нарицателно_съществително><пълен_член> <сказуемо>
<собствено_съществително>.
- **Програма**<пълен_член> <глагол> **Иван**.
- **Програмата гледа Иван**.
- ???
- Освен да е построено правилно,
изречението трябва да има смисъл
- **Семантика: смисъл, значение на текст**



Мета-език на Бекус-Наур

- $\langle \text{цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$
- $\langle \text{цяло_число_без_знак} \rangle ::= \langle \text{цифра} \rangle \{ \langle \text{цифра} \rangle \}$
- $\langle \text{цяло_число} \rangle ::= [+ | -] \langle \text{цяло_число_без_знак} \rangle$
- $-15, 2, +412$
- $\langle \text{латинска_буква} \rangle ::= A \mid B \mid \dots \mid Y \mid Z \mid a \mid b \mid \dots \mid y \mid z$
- $\langle \text{идентификатор} \rangle ::= _ \mid \langle \text{латинска_буква} \rangle \{ \langle \text{латинска_буква} \rangle \mid \langle \text{цифра} \rangle \mid _ \}$
- $a, name, X1, _Data15$

Основни думи (tokens)

- `<идентификатор> ::= _ | <латинска_буква> {<латинска_буква> | <цифра> | _}`
- запазени думи
- стандартни идентификатори
- литерали
 - числови (1, -5, +2.34, 1e-02, 012, 0x123)
 - символни ('a', '\t')
 - низови ("hello", "yes!")
- операции (+, -, *, /)
- разделители (: ; , () [] { } < >)

Коментари

- `<коментар> ::= //<текст_без_нов_ред>`
`| /* <текст> */`
- Компиляторът игнорира:
 - коментари
 - празни символи (интервал, табулация, нов ред)

Основни думи (tokens)

- $\langle \text{идентификатор} \rangle ::= \underline{_} \mid \langle \text{латинска_буква} \rangle$
 $\{ \langle \text{латинска_буква} \rangle \mid \langle \text{цифра} \rangle \mid \underline{_} \}$
- запазени думи
- стандартни идентификатори
- литерали
 - числови (1, -5, +2.34, 1e-02, 012, 0x123)
 - символни ('a', '\t')
 - низови ("hello", "yes!")
- операции (+, -, *, /)
- разделители (: ; , () [] { } < >)

Променливи

- Място в паметта
- Име (идентификатор)
- Тип
- Стойност

fn		c		pi	
...	81111	...	'F'	...	3.14159
<i>int</i>		<i>char</i>		<i>double</i>	

Дефиниция и присвояване

- $\langle \text{дефиниция} \rangle ::= \langle \text{тип} \rangle$
 $\langle \text{идентификатор} \rangle [= \langle \text{израз} \rangle] \{,$
 $\langle \text{идентификатор} \rangle [= \langle \text{израз} \rangle] \};$
- $\langle \text{присвояване} \rangle ::= \langle \text{идентификатор} \rangle =$
 $\langle \text{израз} \rangle ;$
- `int a, b = 15;`
- `a = b + 5;`

Исход на екрана

- `cout << <израз> { << <израз> };`
- `cout << a << b << c;`
- `((cout << a) << b) << c;`
- `cout << "a + b = " << a + b << endl;`
- ~~`cout << "a = " 2;`~~

Вход от клавиатурата

- `cin >> <идентификатор> { >> <идентификатор> };`
- `cin >> a >> b >> c;`
- `((cin >> a) >> b) >> c;`
- ~~`cin >> a + b;`~~
- ~~`cin >> 15;`~~

Константи

- **const** <тип> <идентификатор> = <израз>;
- СТОЙНОСТТА
 - трябва да бъде зададена при дефиниране
 - не може да се променя след това
- `const int FINGERS = 10;`
- ~~`FINGERS = FINGERS + 2;`~~

Типове

- Класификация на видовете данни
- Носят **семантична** информация
- Помагат за проверка на коректност
- Множество от допустими стойности
- Операции
- Вградени функции

Видове типове

- Скаларни (атомарни)

- интегрални

- булев (bool)
 - целочислен (int)
 - символен (char)
 - изброен (enum)

- други

- числа с плаваща запетая (float, double)
 - указател (*)
 - псевдоним (&)

Видове типове

- Съставни
 - масив ([])
 - низ (char[])
 - структура (struct)
 - клас (class)
 - обединение (union)

Логически тип (bool)

- МС: { false, true }
- <булева_константа> ::= true | false
- логически операции

&&	false	true
false	false	false
true	false	true

	false	true
false	false	true
true	true	true

!	
false	true
true	false

Символен тип (char)

- MS: [-128; 127] (signed char)
 - unsigned char: [0; 255]
- <символен_литерал> ::= '<символ>' |
'\<контролен_символ>'

Целочислен тип (int)

- MS: [-2^{31} ; $2^{31}-1$]
- модификатори:
 - short [-2^{15} ; $2^{15}-1$]
 - long [-2^{63} ; $2^{63}-1$]
 - unsigned [0 ; 2^x-1] ($x = 16, 32, 64$)

Целочислен тип (int)

- аритметични операции
 - унарни операции за знак (+, -)
 - бинарни аритметични оператори
 - $a + b$ (събиране)
 - $a - b$ (изваждане)
 - $a * b$ (умножение)
 - a / b (частно)
 - $a \% b$ (остатък)

Целочислен тип (int)

- операции за сравнение (предикати)
 - $a == b$ (равно)
 - $a != b$ (различно)
 - $a < b$ (по-малко)
 - $a > b$ (по-голямо)
 - $a \leq b$ (по-малко или равно)
 - $a \geq b$ (по-голямо или равно)

Числа с плаваща запетая

- **Внимание: това не са реални числа!**
- Мантиса и експонента
- Машинна нула
- Типове
 - float (4 байта)
 - double (8 байта)
 - long double (16 байта)

Числа с плаваща запетая

- $\langle \text{число_с_плаваща_запетая} \rangle ::=$
 $\langle \text{цяло_число} \rangle . [\langle \text{цяло_число_без_знак} \rangle]$
 $[(E|e) \langle \text{цяло_число} \rangle]$
- Примери: 1, 2.34, 12e-2, 10.14E+03, .23
- Операции: всички за целочислен тип **без %**
 - **внимание:** / е деление, а не частно!
 - **внимание:** сравненията == и != са ненадеждни!

Математически функции

```
#include <cmath>
```

- `abs(x)`, `fabs(x)`
- `sin(x)`, `cos(x)`, `tan(x)`, `asin(x)`, `acos(x)`, `atan(x)`
- `exp(x)`, `log(x)`, `log10(x)`
- `ceil(x)`, `floor(x)`
- `sqrt(x)`, `pow(x, n)`

Преобразуване на типове

- bool → char → short → int → long → float → double
- unsigned char → unsigned short → unsigned → unsigned long
- обратната посока може да доведе до загуба на данни!
- експлицитно преобразуване на типове:
<преобразуване> ::= (<тип>)<израз>

Приоритет на операциите

1. Вградени функции
2. Скоби
3. $!$, $+$, $-$ (унарни)
4. $*$, $/$, $\%$
5. $+$, $-$ (бинарни)
6. \ll , \gg
7. $<$, \leq , $>$, \geq
8. $==$, $!=$
9. $\&\&$
10. $\|\|$