

Лексемы языка
Операторы языка
Программа на языке Си



Лексемы языка Си

```
#include <stdio.h>
void main()
{
    int a, b, i, x;
    printf("\n Введи числа:");
    scanf ("%d %d", &a, &b);
    if (a > b) {x = a; a = b; b = x;}
    for ( i = a; i > 1; i --)
        if (( a%i ==0) && ( b%i ==0)) break;
    printf ("\n НОД = %d", i );
}
```

- Идентификаторы
- Ключевые слова
- Константы
- Строковые константы
- Знаки операций
- Знаки пунктуации

Лексемы языка Си

```
#include <stdio.h>
void main()
{
    int a, b, i, x;
    printf("\n Введи числа:");
    scanf ("%d %d", &a, &b);
    if (a > b) {x = a; a = b; b = x;}
    for ( i = a; i > 1; i--)
        if (( a%i == 0) && ( b%i == 0)) break;
    printf ("\n НОД = %d", i );
}
```

Идентификаторы – последовательность из букв латинского алфавита, десятичных цифр и символа подчеркивания, начинающаяся не с цифры: `a`, `b`, `i`, `x`, `printf`, `scanf`

Лексемы языка Си

```
#include <stdio.h>
void main()
{
    int a, b, i, x;
    printf("\n Введи числа:");
    scanf ("%d %d", &a, &b);
    if (a > b) {x = a; a = b; b = x;}
    for ( i = a; i > 1; i--)
        if (( a%i == 0) && ( b%i == 0)) break;
    printf ("\n НОД = %d", i );
}
```

Ключевые (служебные) слова – это идентификаторы, зарезервированные для специального использования: `include`, `void`, `main`, `int`, `if`, `for`, `break`

Лексемы языка Си

```
#include <stdio.h>
void main()
{
    int a, b, i, x;
    printf("\n Введи числа:");
    scanf ("%d %d", &a, &b);
    if (a > b) {x = a; a = b; b = x;}
    for ( i = a; i > 1; i--)
        if (( a%i == 0) && ( b%i == 0)) break;
    printf ("\n НОД = %d", i );
}
```

Знаки пунктуации, разделители:

() { } , ; = [] * : ...

Лексемы языка Си

```
#include <stdio.h>
void main()
{
    char s[]="Ploshad kruga:",
        dl[]="Dlina okruznosti: ";
    float pi=3.1415926, radius;
    printf("\n Enter radius: ");
    scanf("%f",&radius);
    printf("\n %s %f", dl, (2*pi*radius));
    printf("\n %s %f", s, (pi*radius*radius));
}
```

Константа – это лексема, представляющая изображение фиксированного числового, символьного или строкового значения: **3.1415926** (вещественная), **"Ploshad kruga: "** (строковая)



Лексемы языка Си

Константы:

325 (десятичная)

015 (восьмеричная)

0x13A6 (шестнадцатеричная)

2.71 (вещественная)

'z', '\n' (символьные)

«String 1» (строковая)

Перечислимые константы:

```
enum {one = 1, two = 2, three = 3};
```



Лексемы языка Си

Унарные операции:

& получение адреса операнда

***** разыменование указателя

- арифметическое отрицание

! логическое отрицание

++ инкремент

-- декремент

~ побитовое отрицание

sizeof (операнд) определение размера операнда в байтах

Знаки операций – обеспечивают формирование и последующее вычисление выражений



Лексемы языка Си

Бинарные операции:

- Арифметические
- Логические
- Отношения
- Присваивания
- Выбора компонента структурного типа
- Операция «запятая»
- Скобки в качестве операций



Лексемы языка Си

Бинарные *арифметические* операции:

- + сложение
- вычитание
- * умножение
- / деление
- % получение остатка от целочисленного деления

Бинарные *логические* операции:

- && конъюнкция (И)
- || дизъюнкция (ИЛИ)



Лексемы языка Си

Бинарные операции *отношения*:

- < меньше, чем
- > больше, чем
- < = меньше или равно
- > = больше или равно
- = = равно
- ! = не равно

Знаки операций – обеспечивают формирование и последующее вычисление выражений



Лексемы языка Си

Бинарные операции *присваивания*:

= присвоить значение выражения-операнда из правой части операнду левой части

Присвоить операнду левой части:

***=** произведение значений обоих операндов

/= частное от деления значения левого операнда на значение правого

+= сумму значений обоих операндов

-= разность значений левого и правого операндов

%= остаток от деления целочисленного значения левого операнда на значение правого

Лексемы языка Си

```
#include <stdio.h>
void main()
{
    int a, b, i, x;
    printf("\n Введи числа:");
    scanf ("%d %d", &a, &b);
    if (a > b) {x = a; a = b; b = x;}
    for ( i = a; i > 1; i--)
        if (( a%i == 0) && ( b%i == 0)) break;
    printf ("\n НОД = %d", i );
}
```

Унарные операции: **&a**, **&b**, **i--**

Бинарные: **x = a**, **i = a** (присваивания),

a > b, **a%i == 0** (сравнения),

a % i (взятие остатка от деления)



Лексемы языка Си

Бинарные операции выбора *компонентов структурного типа*:

- прямой выбор компонента структуры (объединения)
- > косвенный выбор компонента структуры (объединения), адресуемого указателем

***Запятая* в качестве бинарной операции:**

, несколько выражений, разделенных запятыми, вычисляются последовательно слева направо

***Скобки* в качестве бинарных операций:**

- () обязательны при вызове функций
- [] индексирование элементов массива

Лексемы языка Си

Тернарная (условная с тремя операндами) **операция:**

выражение_1 ? выражение_2 : выражение_3

Вычисляется значение выражения_1.

Если оно истинно, то вычисляется значение выражения_2 , которое становится результатом.

Иначе в качестве результата берется значение выражения_3.

min = (a < b) ? a : b;

x < 0 ? -x : x;

a > 60 ? printf ("Зачёт") : printf ("Незачёт");

isalnum(ch) ? printf ("Элемент слова") : printf ("Разделитель");



Операторы языка Си

Пять видов операторов:

- описания
- присваивания
- передачи управления
- вызова функции
- пустой оператор

Примеры:

```
int a, b, i, x;
```

```
x = a;
```

```
if (a > b)... else ...
```

```
printf("Введи x:");
```

```
for (;;) 
```


Операторы языка Си

Передачи управления:

- условный
- переключатель
- цикла с предусловием
- цикла с постусловием
- цикла с параметром
- возврата из функции
- выхода из цикла
- перехода к следующей итерации цикла

Примеры:

```
if (a > b) min = b; else...
```

```
switch(ch) { case '0' :...}
```

```
while(k-- > 0) f = f*k;
```

```
do{f=f*k;k--;} while(k>0);
```

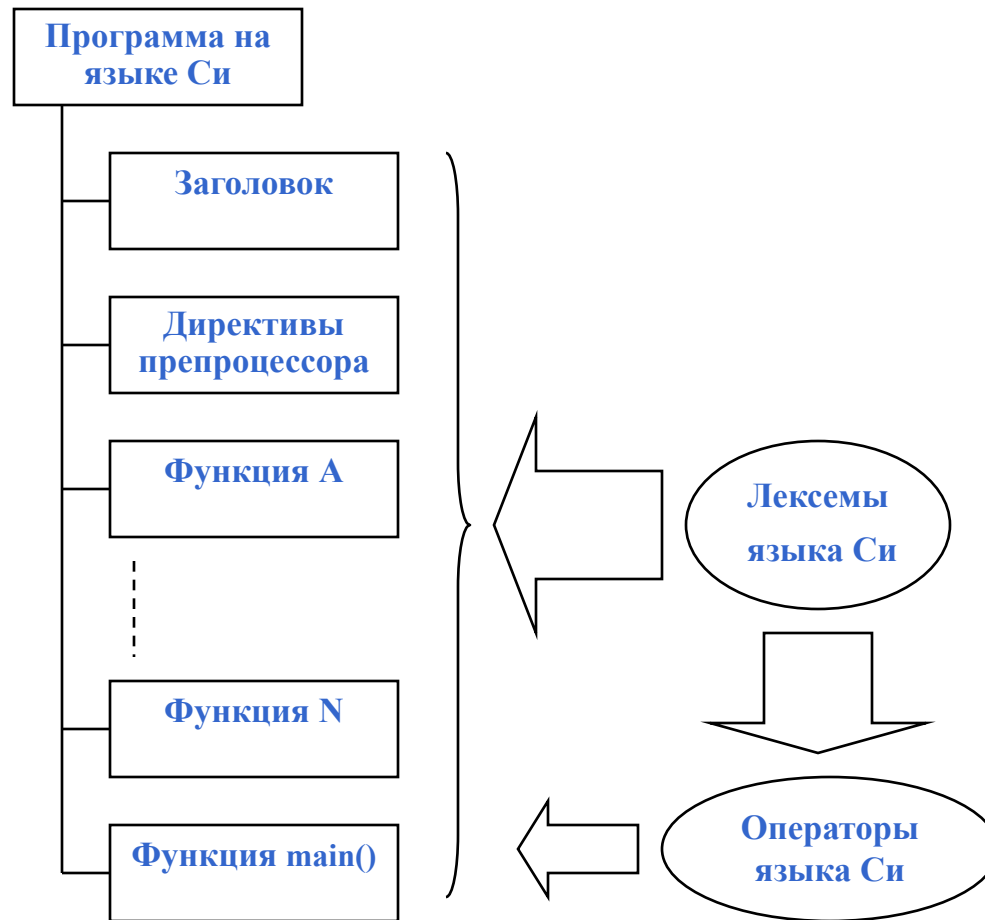
```
for(f=1; k>0; f=f*k,k--);
```

```
return;
```

```
break;
```

```
continue;
```

Структура программы на языке Си



Структура программы на языке Си

```
//Программа вычисляет факториал числа // Заголовок
#include<stdio.h> // Директива препроцессора
long fact(int n) // функция fact
{
long int f=1, k=n; // Оператор описания
while(k-->0) f*=k; // Цикл с предусловием
return f; // Возврат из функции
}
void main() // функция main
{
int num; // Оператор описания
printf(«Введи число: »); // Вызов библиотечных
scanf(«%d», &num); // функций
printf(«Факториал = %ld», fact(num)); // Вызов функции
} // пользователя
```