

Программирование линейных алгоритмов

Комментарии

// это однострочный комментарий

// каждый раз надо ставить символ //

// в начале строки

/* а это многострочный комментарий,
можно писать сколько угодно строчек
текста, главное не забыть закрыть
его с помощью символов */

Составной оператор

```
{  
...  
{  
//это составной оператор  
int a = 3; //локальная переменная "a"  
int b = a + 2;  
}  
float a = 5.5; // это уже другая  
//переменная "a"  
...  
}
```

Алфавит C++

- A..Z, a..z, _
- 0..9
- " { } , | [] () + - / % * . \ ' : ? < = > ! &
~ ; ^
- Пробел, табуляция, переход на новую строку

Объект

Объект - место в памяти, имеющее тип, определяющий вид информации, разрешенной для хранения.

Переменная - именованный объект.



Идентификатор

Идентификатор — это имя программного объекта.

Правила задания идентификаторов:

- используются латинские буквы, цифры и знак подчеркивания;
- прописные и строчные буквы различаются;
- первым символом может быть буква или знак подчеркивания, но не цифра;
- пробелы внутри имен не допускаются;
- не должен совпадать с ключевыми словами языка.

Определение

Инструкция, вводящая новое имя в программе и выделяющая память для переменной, называется *определением*.



Типы данных

int	2 байта 4 байта	$-2^{15}..2^{15} - 1$ $-2^{31}..2^{31} - 1$
float	4 байта	$3.4 \cdot 10^{-38}..3.4 \cdot 10^{38}$
double	8 байт	$1.7 \cdot 10^{-308}..1.7 \cdot 10^{308}$
bool	1 байт	true или false
char	1 байт	$0..2^8 - 1$
void	2 или 4 байта	-
string	Определяется количеством СИМВОЛОВ	-

Модификаторы типа

- short;
- long;
- signed;
- unsigned.

Определение переменной

```
int Name;
```

```
double object_name;
```

Name

unsigned int



object_name

double



Значения

Данные, записанные в переменные, называют *значениями*.

	Name
unsigned int	15
	object_name
double	3.5

Присваивание

Инициализация – присваивание
переменной ее начального значения

```
unsigned int Name = 15;
```

```
double object_name = 3.5;
```

```
Name = Name + 3;
```

Цепочка операторов присваивания

```
int j;
```

```
int i = j = 0;
```

```
float x, y = 0;
```

Константы

- Символьные: 'g', 'к', '=';
- Строковые: "Это строковая константа";
- Целые: 123, 2345;
- вещественные: 1.23, .2345;
- Именованные:
const тип имя=значение;

Например:

```
const int some_const = 123;
```

Явное преобразование типов

- `StrToInt()`;
- `IntToStr()`;
- `StrToFloat()`;
- `FloatToStr()`;
- Новый тип переменной записывается в круглых скобках перед ее именем (язык C).

Примеры

```
int a = StrToInt(Edit1->Text);  
Memo1->Lines->Add("Вы ввели a=");  
Memo1->Lines->Add(IntToStr(a));  
float b = StrToFloat(Edit2->Text);  
Edit3->Text=FloatToStr(b);  
int c = (int)b;  
Memo1->Lines->Add("целочисленная  
c=");  
Memo1->Lines->Add(IntToStr(c));
```

[Sample](#)

Безопасные неявные преобразования:

bool в char

bool в int

bool в double

char в int

char в double

int в double

Небезопасные неявные преобразования:

double в int

double в char

double в bool

int в char

int в bool

char в bool

Примеры неявного преобразования типов

```
char c = 'x';
```

```
int i = c; // i = 120
```

```
int j = 'x'; // j = 120
```

```
int a = 300;
```

```
char b = a;
```

```
/* b=',', - СИМВОЛ С КОДОМ 44 */
```

```
double x = 1.5;
```

```
a = x;
```

Выражение

Выражение – последовательность операторов, операндов и знаков пунктуации, воспринимаемую компилятором как руководство к определенному действию над данными.

Инструкции – выражения, завершающиеся точкой с запятой (;)

Пример:

```
//a = b ++ b; // синтаксическая ошибка
```

```
//1
```

```
a = b++; b;
```

```
//2
```

```
a = b; ++b;
```

Арифметические операции

Сложение	+
Вычитание	-
Умножение	*
Деление	/
Остаток от деления	%

Примеры

```
int a = 5%3; //a=2
```

```
int b = 5%2; //b=1
```

```
int c = 5/2; //c=2
```

```
float d = 5/2.; //d=2.5
```

```
float e = 5./a; //e=2.5
```

Инкремент и декремент

`a++`; `++a`; `a--`; `--a`;

Пример:

```
int a = 2, b=4;
```

```
int c = b*a++; //c=8, a=3;
```

```
int d = b*++a; //d=16, a=4;
```

Модификации оператора присваивания

операция	пример	аналог записи
+=	a+=b;	a=a+b
-=	a-=b;	a=a-b
=	a=b;	a=a*b
/=	a/=b;	a=a/b
%=	a%=b;	a=a%b

Операторы сравнения

оператор	пример
<	bool c=a<b;
<=	bool c=a<=b;
>	bool c=a>b;
>=	bool c=a>=b;
==	bool c=a==b;
!=	bool c=a!=b;

Приоритет операций

оператор	описание	приоритет	направл.	оператор	описание	приоритет	направл.
.	Выбор члена	16	->	-	Унарный минус	14	<-
++	Постфиксный инкремент	15	<-	+	Унарный плюс	14	<-
--	Постфиксный декремент	15	<-	*	Умножение	13	->
++	Префиксный инкремент	14	<-	/	Деление	13	->
--	Префиксный декремент	14	<-	%	Остаток от деления	13	->
(тип)	Преобразование Типа	14		+	Сложение	12	->
!	Логическое НЕ	14	<-	-	Вычитание	12	->

Приоритет операций

оператор	описание	приоритет направл.		оператор	описание	приоритет направл.	
<	Меньше	10	->		Логическое ИЛИ	4	->
<=	Меньше или равно	10	->	=	Присваивание	2	<-
>	Больше	10	->	*=	Умножение с присваив.	2	<-
>=	Больше или равно	10	->	/=	Деление с присваив.	2	<-
==	Равно	9	->	%=	Остаток от деления с присваив.	2	<-
!=	Не равно	9	->	+=	Сложение с присваив.	2	<-
&&	Логическое И	5	->	-=	Вычитание с присваив.	2	<-

Математические функции

```
#include <math.h>
```

```
double pow (double x, double y);
```

```
double fmod(double x, double y);
```

```
int abs (int);
```

```
double fabs (double);
```

```
long labs (long);
```

```
float fabsf (float);
```

```
long double fabsl (long double);
```

Математические функции

`double log (double);`

`double log10 (double);`

`double exp(double x);`

`double sqrt (double);`

`float logf (float);`

`long double logl (long
double);`

`float log10f (float);`

`long double log10l (long
double);`

`long double expl (long
double);`

`float sqrtf (float);`

`long double sqrtl (long
double);`

Пример

Определить остаток от деления целого числа a на целое число b .

```
double f = fmod(a, b);
```

Математические константы

- M_{PI}
- M_E

Пример

1. Вычислить: $rez = \sqrt{|x^3 + \ln y - \lg z|}$

x^3 `pow(x, 3)`

$x^3 + \ln y$ `pow(x, 3) + log(y)`

$x^3 + \ln y - \lg z$ `pow(x, 3) + log(y)`
 `- log10(z)`

$|x^3 + \ln y - \lg z|$ `fabs(pow(x, 3) + log(y)`
 `- log10(z))`

$\sqrt{|x^3 + \ln y - \lg z|}$ `sqrt(fabs(pow(x, 3) + log(y)`
 `- log10(z)))`

```
double rez=sqrt(fabs(pow(x, 3)  
                                 +log(y) -log10(z))) ;
```


Тригонометрические функции

`double cos(double);`

`double cosh(double);`

`double sin(double);`

`double sinh(double);`

`double tan(double);`

`double tanh(double);`

`double acos(double);`

`double asin(double);`

`double atan(double);`

`float cosf(float);`

`float coshf(float);`

`float sinf(float);`

`float sinhf(float);`

`float tanf(float);`

`float tanhf(float);`

`float acosf(float);`

`float asinf(float);`

`float atanf(float);`

Тригонометрические функции

```
double atan2(double x, double y);
```

```
float atan2f(float, float);
```

Вычислить: $\frac{\sin x + \cos y}{\operatorname{arctg} z}$

```
double rez = (sin(x) + cos(y)) / atan(z);
```

Функции округления

`double floor (double x);`

`long double floorl (long double x);`

`double ceil (double x);`

`long double ceill (long double x);`

Пример:

```
double a = 7.765432;
```

```
double f = floor(a); // f=7.0
```

```
double c = ceil(a); // c=8.0
```

Функция modf

```
double modf(double, double*);
```

дробная часть числа =

modf (число, &целая часть числа)

```
double number, int_part, frac_part;
```


```
number = FloatToStr(Edit1->Text);
```

```
frac_part = modf(number, &int_part);
```

Блок-схемы линейных алгоритмов

 начало – конец

 ввод-вывод

 процесс

 Комментарий

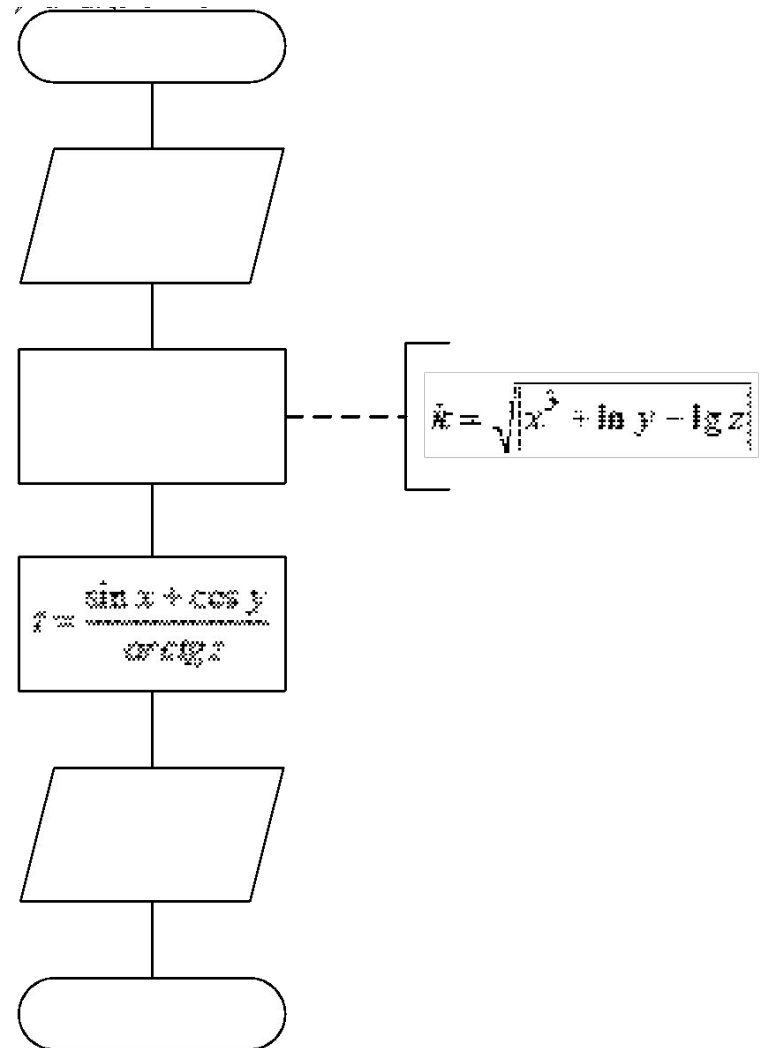
Блок-схемы линейных алгоритмов

Даны действительные
 x, y, z .

Вычислить:

1. $k = \sqrt{|x^3 + \ln y - \lg z|}$

2. $t = \frac{\sin x + \cos y}{\operatorname{arctg} z}$



Пример 1

Вычислить выражение $\operatorname{tg}^2 x^3 - \frac{\ln y}{x+1}$

где x и y вводятся с клавиатуры.

```
#include <math.h>
```

```
...
```

```
{
```

```
float x=StrToFloat (Edit1->Text) ;
```

```
float y=StrToFloat (Edit2->Text) ;
```

```
float
```

```
z=pow (tan (pow (x, 3) ) , 2) -log (y) / (x+1  
);
```

```
Label3->Caption=FloatToStr (z) ;
```

```
}
```

[Sample](#)

Пример 2

Известны стороны a и b прямоугольника.
Найти площадь прямоугольника.

```
{  
float a=StrToFloat(Edit1->Text);  
float b=StrToFloat(Edit2->Text);  
float s=a*b;  
Label3->Caption=FloatToStr(s);  
}
```

[Sample](#)

Пример 3

Известен радиус окружности r . Найти площадь окружности.

```
#include <math.h>
...
{
float r=StrToFloat(Edit1->Text);
double s=M_PI*pow(r,2);
Label3->Caption=FloatToStr(s);
}
```

[Sample](#)

Пример 4

Дано трехзначное число n . Найти сумму цифр
данного числа

```
unsigned short n=StrToInt (Edit1->Text) ;  
unsigned short n1=n/100;  
unsigned short n2=n/10%10;  
unsigned short n3=n%10;  
Label2->Caption=IntToStr (n1+n2+n3) ;
```

[Sample](#)

Пример 5

Дано трехзначное число n . Первую и третью цифру числа поменять местами.

$$n1 \cdot 10^2 + n2 \cdot 10 + n3$$

```
Label2->Caption=
```

```
    IntToStr (n3*100+n2*10+n1) ;
```

[Sample](#)