

Лекция 1

# **Знакомство с языком C++**

ЗАДАЧА 1. Заданы длины двух катетов  
прямоугольного треугольника  $a$ ,  $b$ . Вычислить  
длину гипотенузы  $c$  и величины двух его углов.

Значения  $a$ ,  $b$ ,  $c$  ввести с клавиатуры .

основные формулы :

Гипотенуза  $c$  вычисляется по формуле  $c = \sqrt{a^2 + b^2}$ ,

углы треугольника  $\alpha$  и  $\beta$  :  $\alpha = \arctg(a/b)$  ,  $\beta = \pi/2 - \alpha$  .

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
double a,b,c,alf,bet,s;
cout<<"A=";
cin>>a;
cout<<"B=";
cin>>b;
s=a*b/2;
c=pow(a*a+b*b,0.5);
alf=atan(a/b);
bet=3.14159/2-alf;
cout<<"\n A="<<a<<"\t B="<<b<<"\t C="<<c<<"\n";
cout<<"\nS="<<s<<"\t alf="<<alf*180/3.14159;
cout<<"\t bet="<<bet*180/3.14159<<endl;
}
```

# Общая структура любой программы

Директивы препроцессора

Объявление глобальных переменных

Тип\_результата main(Список\_переменных)

{

Операторы

}

Тип\_результата f1(Список\_переменных)

{

Операторы

}

Тип\_результата f2(Список\_переменных)

{

Операторы

}

...

Тип\_результата fn(Список\_переменных)

{

Операторы

}

# Структура простейшей программы

```
int main()  
{  
  Операторы  
}
```

# Трансляторы



**интерпретаторы** - ,  
трансляторы которые  
переводят каждый  
оператор программы в  
машинный код и по  
мере перевода  
операторы  
выполняются  
процессором , ;

**компиляторы**  
переводят всю  
программу целиком и  
если перевод всей  
программы , прошел  
без ошибок то  
полученный двоичный  
код можно запускать  
на выполнение

# Алфавит языка

Программа на языке C++ может содержать следующие символы:

- прописные строчные латинские буквы , A, B, C, x, y, z ; и знак подчеркивания
- арабские цифры от 0 до 9;
- специальные знаки : { } , | , [ ] ( ) + / % \* . \ : ? < = > ! & # ~ ; ^
- символы пробела табуляции и перехода на новую строку

# Типы данных

К основным типам данных языка относят:

- char - символьный
- int - целый
- float - с плавающей точкой
- double - двойной точности
- bool - логический

# Составные типы данных

В C++ определены четыре спецификатора типов данных:

- `short` - короткий
- `long` - длинный
- `signed` - знаковый
- `unsigned` - беззнаковый

# СИМВОЛЬНЫЕ ТИПЫ

<b>Тип</b>	<b>Диапазон</b>	<b>Размер</b>
char	-128..127	1 байт
unsigned char	0..255	1 байт
signed char	-128..127	1 байт

# Целые типы

Тип	Диапазон	Размер
int	-32767 .. 32767	4 байта
unsigned int	0 .. 65535	4 байта
signed int	-32767 .. 32767	4 байта
short int	-32767 .. 32767	2 байта
long int	-2147483647 .. 2147483647	4 байта
unsigned short int	0 .. 65535	2 байта
signed short int	-32767 .. 32767	2 байта
long long int	$-(2^{63}-1) .. (2^{63}-1)$	8 байт
signed long int	-2147483647 .. 2147483647	4 байта
unsigned long int	0 .. 4294967295	4 байта
unsigned long long int	$0 .. 2^{64}-1$	8 байт

# Вещественные типы

Тип	Диапазон	Размер
float	3.4E-38 .. 3.4E+38	4 байта
double	1.7E-308 .. 1.7E+308	8 байт
long double	3.4E-4932 .. 3.4E+4932	8 байт

$mE\pm p$ ,

где  $m$  мантисса (целое или дробное число с десятичной точкой ),  
 $p$  порядок (целое число).

Например,

$-6.42E+2 = -6.42 \cdot 10^2 = -642$ ,

$3.2E-6 = 3.2 \cdot 10^{-6} = 0.0000032$

# Логический тип **bool**

true ( ) истина или false ( ) ложь

## Тип **void**

Множество значений этого типа пусто.

# Переменные в языке C++

**тип** список\_переменных ;

Например,

```
int a, bc, f;
```

```
float g, u, h12;
```

← локальные

↓ глобальные

↘ формальные

# Локальные переменные

```
int main()  
{  
float s; //В функции main определена вещественная переменная s,  
s=4.5; //и ей присвоено значение 4.5.  
}  
int f1()  
{ int s; //В функции f1 описана другая переменная s (типа int),  
s=6; //ей присвоено значение 6.  
}  
int f2()  
{  
long int s; //В функции f2 определена еще одна переменная s  
s=25; //(типа long int) и ей присвоено значение 25.  
}
```

# Глобальные переменные

```
float s; //Определена глобальная переменная s (типа float).
int main()
{
s=4.5; //В функции main переменной s присваивается
значение 4.5.
}
int f1()
{
s=6; //В функции f1 переменной s присваивается значение 6.
}
int f2()
{
s=25; //В функции f2 переменной s присваивается значение
25.
}
```

# Константы в языке C++

Описание константы имеет вид:

**const** тип имя=значение;

Пример:

```
const double pi=3.141592653589793
```

# Основные операции языка C++

Операция	Описание
Унарные операции	
++	увеличение значения на единицу
--	уменьшение значения на единицу
~	поразрядное отрицание
!	логическое отрицание
-	арифметическое отрицание (унарный минус)
+	унарный плюс
&	взятие адреса
*	разадресация
(type)	преобразование типа

Бинарные операции	
+	сложение
-	вычитание
*	умножение
/	деление
%	остаток от деления
<<	сдвиг влево
>>	сдвиг вправо
<	меньше
>	больше
<=	меньше или равно
>=	больше или равно
==	равно
!=	не равно
&	поразрядная конъюнкция (И)
^	поразрядное исключающее ИЛИ
	поразрядная дизъюнкция (ИЛИ)
&&	логическое И
	логическое ИЛИ
=	присваивание
*=	умножение с присваиванием
/=	деление с присваиванием
+=	сложение с присваиванием
-=	вычитание с присваиванием
%=	остаток от деления с присваиванием
<<=	сдвиг влево с присваиванием
>>=	сдвиг вправо с присваиванием
&=	поразрядная конъюнкция с присваиванием
=	поразрядная дизъюнкция с присваиванием
^=	поразрядное исключающее ИЛИ с присваиванием

# Операции присваивания

Общий вид:

имя\_переменной=значение;

операция *множественным присваиванием* :

имя\_переменной1= имя\_переменной2=..=  
имя\_переменнойN=значение;

Пример a=b=c=3.14159/6;

# *Составное присваивание*

Операции  $+=$ ,  $=$ ,  $*=$ ,  $/=$  называют ***составным присваиванием***

пример:

$x+=p$ ; //Увеличение  $x$  на  $p$ , то же что и  $x=x+p$ .

$x-=p$ ; //Уменьшения  $x$  на  $p$ , то же что и  $x=x-p$ .

$x*=p$ ; //Умножение  $x$  на  $p$ , то же что и  $x=x*p$ .

$x/=p$ ; //Деление  $x$  на  $p$ , то же что и  $x=x/p$ .

# Стандартные математические функции

Обозначение	Действие
abs(x)	Модуль целого числа
fabs(x)	Модуль вещественного числа
sin(x)	Функция синус
cos(x)	Функция косинус
tan(x)	Функция тангенс
atan(x)	Арктангенс $x$ в диапазоне $-\pi/2$ до $\pi/2$
exp(x)	Экспонента, $e^x$
log(x)	Натуральный логарифм ( $x > 0$ )
log10(x)	Десятичный логарифм ( $x > 0$ )
sqrt(x)	Корень квадратный ( $x \geq 0$ )
pow(x,y)	$x$ в степени $y$

# Структура программы

```
тип_результата имя_функции (параметры)
{
оператор1;
оператор2;
..
операторN;
}
```

# Ввод и вывод данных



С ПОМОЩЬЮ  
*функций ввода-  
вывода в стиле C*

С ИСПОЛЬЗОВАНИЕМ  
*библиотеки  
классов C++*

# Функции ввода- вывода

**printf**(строка форматов, список выводимых переменных)

функция выполняет форматированный *вывод* переменных, указанных в списке, в соответствии со строкой форматов.

**scanf**(строка форматов, список адресов вводимых переменных)

функция выполняет *ввод* переменных, адреса которых указаны в списке, в соответствии со строкой форматов.

# *Спецификации*

**% флаг ширина . точность модификатор  
ТИП**

# Символы управления

Параметр	Назначение
Флаги	
-	Выравнивание числа влево. Правая сторона дополняется пробелами. По умолчанию выравнивание вправо.
+	Перед числом выводится знак «+» или «-»
Пробел	Перед положительным числом выводится пробел, перед отрицательным «-»
#	Выводится код системы счисления: 0 - перед восьмеричным числом, 0x (0X) перед шестнадцатеричным числом.
Ширина	
n	Ширина поля вывода. Если n позиций недостаточно, то поле вывода расширяется до минимально необходимого. Незаполненные позиции заполняются пробелами.
0n	То же, что и n, но незаполненные позиции заполняются нулями.

Параметр	Назначение
Точность	
ничего	Точность по умолчанию
n	Для типов e, E, f выводить n знаков после десятичной точки
Модификатор	
h	Для d, i, o, u, x, X тип short int.
l	Для d, i, o, u, x, X тип long int.
Тип	
c	При вводе символьный тип char, при выводе один байт.
d	Десятичное int со знаком.
i	Десятичное int со знаком.
o	Восьмеричное int unsigned.
u	Десятичное int unsigned.
x, X	Шестнадцатеричное int unsigned, при x используются символы a-f, при X - A-F.
f	Значение со знаком вида [-]dddd.dddd.
e	Значение со знаком вида [-]d.dddde[+ -]ddd.
E	Значение со знаком вида [-]d.ddddE[+ -]ddd.
g	Значение со знаком типа e или f в зависимости от значения и точности.
G	Значение со знаком типа e или F в зависимости от значения и точности.
s	Строка символов.

# Специальные символы

<b>Символ</b>	<b>Назначение</b>
<code>\b</code>	Сдвиг текущей позиции влево.
<code>\n</code>	Перевод строки.
<code>\r</code>	Перевод в начало строки, не переходя на новую строку.
<code>\t</code>	Горизонтальная табуляция.
<code>\'</code>	Символ одинарной кавычки.
<code>\''</code>	Символ двойной кавычки.
<code>\?</code>	Символ ?.

```
#include <stdio.h>
```

Заголовочный файл `stdio.h` содержит описание функций ввода-вывода.

Задача. Зная  $a$ ,  $b$ ,  $c$  длины сторон треугольника ,  
вычислить площадь  $S$  и периметр  $P$  этого  
треугольника.

Входные данные:  $a$ ,  $b$ ,  $c$ . Выходные данные:  $S$ ,  $P$ .  
Для вычисления площади применим формулу  
Герона.

```
#include <stdio.h>
#include <math.h>
int main()
{
float a,b,c,S,r; //Описание переменных.
//В операторах scanf для вычисления адреса  переменной применяется
операция &.
printf("a="); //Вывод на экран символов a=
scanf("%f",&a); //Запись в переменную a значения введенного с клавиатуры.
printf("b="); //Вывод на экран символов b=.
scanf("%f",&b); //Запись в переменную b значения введенного с клавиатуры.
printf("c="); //Вывод на экран символов c=
scanf("%f",&c); //Запись в переменную c значения введенного с клавиатуры.
r=(a+b+c)/2; //Вычисление полупериметра.
S=sqrt(r*(r-a)*(r-b)*(r-c)); //Вычисление площади треугольника.
//Вывод символов S=, значения S и символа табуляции \t.
//Спецификация %5.2f означает, что будет выведено вещественное число,
//под которое отводится 5 знаков, причем 2 из них после точки.
printf("S=%5.2f \t",S);
//Вывод символов p=, значения выражения 2*r
//и символа окончания строки \n.
printf("p=%5.2f \n",2*r);
//Оператор printf("S=%5.2f \t p=%5.2f \n",S,2*r);
//выдаст тот же результат.
return 0;}
```

# Вариант второй

```
#include <stdio.h>
#include <math.h>
int main()
{
float a,b,c,S,r;
printf("Vvedite a, b, c \n"); //Вывод на экран строки символов.
scanf("%f%f%f",&a,&b,&c); //Ввод значений.
r=(a+b+c)/2;
S=sqrt(r*(r-a)*(r-b)*(r-c));
printf("S=%5.2f \t p=%5.2f \n",S,2*r); //Вывод на экран
результатов.
return 0;
}
```

# Объектно-ориентированные средства ввода-вывода

заголовочный файл `iostream.h`.

директивы `#include <iostream.h>` в

программе автоматически создаются объекты-потоки `cin` для ввода с клавиатуры и `cout` для

вывода на экран, а так же операции помещения в поток `<<` и чтения из потока `>>`.

Итак, с помощью объекта `cin` и операции `>>` можно присвоить значение любой переменной

Например, если переменная `i` описана как целочисленная, то команда `cin>> i;` означает, что в

переменную `i` будет записано некое целое число, введенное с клавиатуры. Если нужно ввести

несколько переменных, следует написать `cin>>x>>y>>z;`.

Объект `cout` и операция `<<` позволяют вывести на экран значение любой переменной или текст. Текст необходимо заключать в двойные кавычки, кроме того, допустимо применение

специальных символов `\t` и `\n`. Запись `cout<<i;` означает вывод на экран значения переменной

`i`. А команда `cout<<x<<"\t"<<y;` выведет на экран значения переменных `x` и `y` разделенные символом табуляции.

Известны плотность  $\rho$ , высота  $h$  и радиус основания  $R$  цилиндрического слитка, полученного в металлургической лаборатории.

Найти объем  $V$ , массу  $m$  и площадь  $S$  основания слитка.

Входные данные:  $\rho$ ,  $h$ ,  $R$ . Выходные данные:  $S$ ,  $V$ ,  $m$ .

Учитывая, что  $S=2\pi R$ ,  $V=\pi R^2 h$  и  $m=\rho V$

```
#include "stdafx.h"
#include <iostream>
using namespace std;
#define pi 3.14159 //Определение константы.
int main()
{double R,h,r,S,V,m; //Описание переменных.
cout<<"R="; //Вывод на экран символов R=.
cin>>R; //Ввод значения переменной R.
cout<<"h="; //Вывод на экран символов h=.
cin>>h; //Ввод значения переменной h.
cout<<"r="; //Вывод на экран символов r=.
cin>>r; //Ввод значения переменной r.
S=2*pi*R; //Расчет площади.
V=pi*R*R*h; //Вычисление объема.
m=r*V; //Определение массы.
cout<<"S="<<S; //Вывод на экран символов S= и значения переменной S.
cout<<"\n V="<<V; //Вывод на экран с новой строки V= и значения V.
cout<<"\n m="<<m; //Вывод на экран с новой строки m= и значения m.
return 0;
}
```

# Структура программы в C++

## 1 Блок заголовков программы.

С помощью инструкции `#include` подключаются внешние модули

## 2 Блок с объявлением классов

(базовых и производных), прототипами и объявлениями функции.

**3 Главный метод программы:** каждая программа имеет такой метод. У метода стандартное название `main ()`

## 4 Блок с описанием функций

(прототип которых указан во втором блоке)

# Пример 1 Программа для отображения приветствия

```
#include <iostream>
using namespace std;
int main()
{
    // Выводится сообщение
    cout << "Hello \n";
    return 0;
}
```

```
#include <имя файла> или
#include <имя файла>
```

```
(#include "имя файла" или
#include <имя файла>)
```

# Использование переменных

Под переменной подразумевают именованную область памяти, к которой можно обратиться в процессе выполнения программы

Пример 2. Пользователь на запрос программы вводит свой возраст, после чего программ выводит сообщение о возрасте пользователя.

```
#include <iostream>
using namespace std;
int main() {
    int age;          //Переменная для записи возраста
    cout << “Сколько Вам лет? \n”; //Вывод сообщения
    cin >>age;       //Ввод возраста в переменную
    cout <<“Вам” << age << “лет \n”; //Вывод результата
    return 0;
}
```

# Команды

`cin` (console input) – оператор ввода

`cout` (console output) – оператор вывода

# Объявление и инициализация переменных

//Объявление 3-х целочисленных переменных

```
int m, n, k;
```

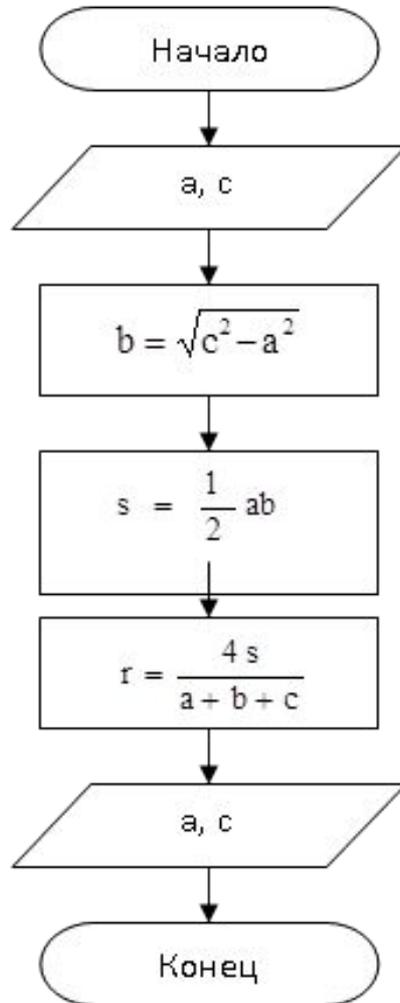
//Объявление с одновременной инициализацией нескольких переменных

```
int one=1, two=2, three, four=4, five;
```

### Пример 3. Рассчитать высоту, на которую за указанное время поднимается брошенное вверх тело.

```
#include <iostream>
using namespace std;
int main()    {
    double v;      //Объявление переменной – скорость тела
    double t=1.2; //Время полета
    double g=9.8; //Ускорение свободного падения
    v=12.3;       //инициализация переменной – скорость
    Double s=v*t-g*t*t/2; //Высота – динамическая инициализация переменной
    cout <<“Высота=” << s << “\n”; //Вывод результата
    return 0;
}
```

**Задача 1.** Дана гипотенуза и катет  
прямоугольного треугольника. Найти второй  
катет и радиус вписанной в этот треугольник  
окружности.



Имя переменной	Тип переменной	Класс	Описание
a	действительная	входная	Первый катет треугольника
b	действительная	выходная	Второй катет треугольника
c	действительная	входная	Гипотенуза треугольника
s	действительная	рабочая	Площадь треугольника
r	действительная	выходная	Радиус вписанной окружности

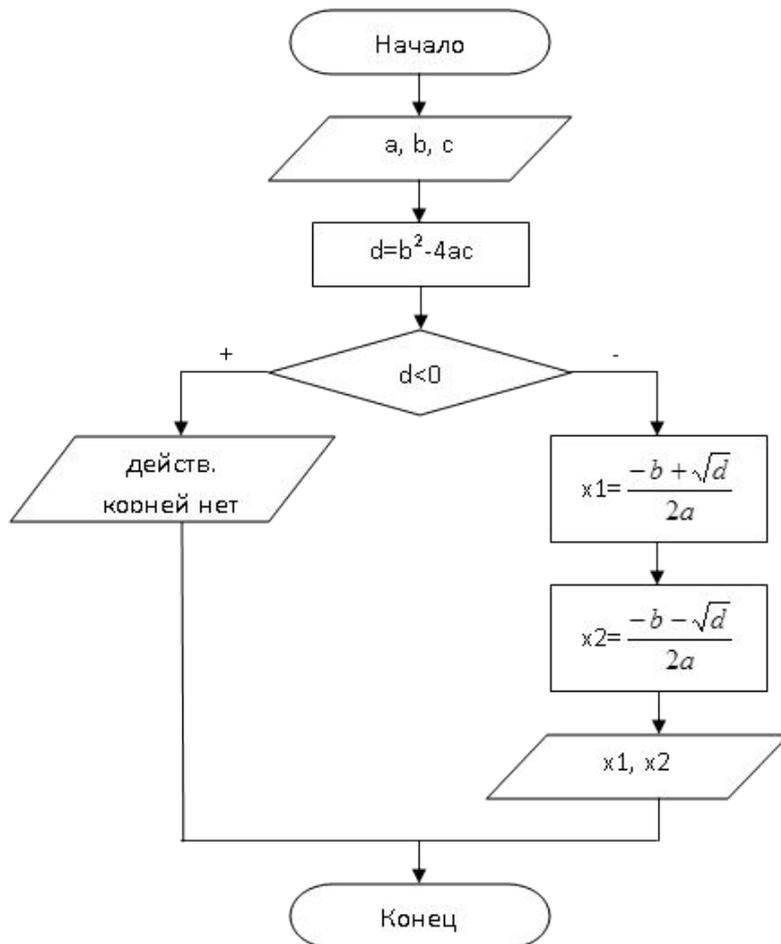
## Код программы на языке Pascal

```
var a, b, c, s, r:real;
begin
write('Введите длину катета: ');
readln(a);
write('Введите длину гипотенузы: ');
readln(c);
b:=Sqrt(c*c-a*a);
s:=0.5*a*b;
r:=(4*s)/(a+b+c);
writeln('Второй катет треугольника
равен ', b:5:2);
writeln('Радиус вписанной в
треугольник окружности равен ',
r:5:2);
  readln;
end.
```

## Код программы на языке C++

```
#include <stdio.h>
#include <math.h>
float a, b, c, s, r;
void main(){
printf("Введите длину катета: ");
scanf("%f",&a);
printf("Введите длину гипотенузы: ");
scanf("%f",&c);
b=sqrt(c*c-a*a);
s=0.5*a*b;
r=(4*s)/(a+b+c);
printf("Второй катет треугольника
равен %.2f\n", b);
printf("Радиус вписанной в
треугольник окружности равен
%.2f\n", r);
getchar();
}
```

# Задача 2. Составить программу решения квадратного уравнения $ax^2+bx+c=0$ , заданного своими коэффициентами ( $a \neq 0$ ).



Имя переменной	Тип переменной	Класс	Описание
a	целая	входная	Коэффициент перед старшей степенью x
b	целая	входная	Коэффициент перед x
c	целая	входная	Свободный член уравнения
d	целая	рабочая	Дискриминант уравнения
x1	действительная	выходная	Первый корень уравнения
x2	действительная	выходная	Второй корень уравнения

## Код программы на языке Pascal

```
var a, b, c, d:integer;
    x1, x2:real;
begin
write('Введите коэффициент перед старшей
степенью x: ');
readln(a);
write('Введите коэффициент перед x: ');
readln(b);
write('Введите свободный член уравнения: ');
readln(c);
d:=sqr(b)-4*a*c;
if d<0 then
  writeln('Действительных корней нет')
else
  begin
    x1:=(-b+sqrt(d))/(2*a);
x2:=(-b-sqrt(d))/(2*a);
writeln('Первый корень равен ', x1:5:2);
writeln('Второй корень равен ', x2:5:2);
  end;
  readln;
end.
```

## Код программы на языке C++

```
#include <stdio.h>
#include <math.h>
int a, b, c, d;
float x1, x2;
void main(){
printf("Введите коэффициент перед старшей
степенью x: ");
scanf("%d",&a);
printf("Введите коэффициент перед x: ");
scanf("%d",&b);
printf("Введите свободный член уравнения: ");
scanf("%d",&c);
d=pow(b,2)-4*a*c;
if (d<0)
  printf("Действительных корней нет\n");
else
  {
    x1=(-b+sqrt(d))/(2*a);
x2=(-b-sqrt(d))/(2*a);
printf("Первый корень равен %.2f\n", x1);
printf("Второй корень равен %.2f\n", x2);
  }
  getchar();
}
```

