
Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО
ОСНОВЫ
ПРОГРАММИРОВАНИЯ
2 ЧАСТЬ

УНАРНЫЕ ОПЕРАЦИИ

Операция	Наименование	Примеры применения
&	получения адреса операнда	int *p = &a;
*	обращение по адресу	int a = 2; int *p = &a;
+	унарный плюс	a=b
-	унарный минус	a=-b

Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Слайд 27

Тема 2. Операции и стандартные библиотечные функции языка

Одни и те же знаки операций могут использоваться в различных выражениях и по-разному интерпретируются в зависимости от контекста.

Знаки операций делятся на унарные и бинарные. Унарными называются такие операции, в которых участвует один операнд. Операндом может быть переменная, константа, функция.

В таблице приведены унарные операции с примерами их применения.

Операция получения адреса операнда позволяет ввести переменную целого типа, в которой запоминается адрес операнда. Эта переменная называется переменная-указатель.

Операция обращения по адресу - доступ по адресу к значению того объекта, на который указывает операнд, дает доступ к информации как через имя переменной, так непосредственно по адресу, где расположена данная информация.

УНАРНЫЕ ОПЕРАЦИИ

Операция	Наименование	Примеры применения
~	побитовое отрицание (НЕ)	10000101->11111010
!	логическое отрицание	!1 = 0, ! -5 = 0 , !0 = 1
++	инкремент	i++; (i=i+1)
- -	декримент	i--; (i=i-1)

a=2; c=3;
y=a+c++;
Результат: y=5, c=4
	a=2; c=3;
	y=a+(++c);
	Результат: y=6, c=4

Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Слайд 28

Операция побитовое отрицание означает поразрядное инвертирования внутреннего двоичного кода целочисленного аргумента.

Операция логического отрицания только в одном случае дает значение единица, что означает - истина, если аргумент равен нулю, во всех остальных случаях результат ноль, что означает - ложь.

Инкремент и декремент – это операции увеличение на единицу и уменьшение на единицу применяются только для левостоящих операндов. Операндом может быть только переменная.

Эти операции бывают:

- префиксные - увеличение или уменьшение значения операнда до его первого использования;
- постфиксные - увеличение или уменьшение значения операнда после его использование.

На слайде приведены примеры операции инкремент. В зависимости от записи этой операции, результаты вычислений различны. Во втором примере эта операция указана в скобках. Правила записи арифметических операций не позволяет в выражениях ставить подряд два знака. По этой причине знак декремент, увеличение на единицу, указан в скобках.

УНАРНЫЕ ОПЕРАЦИИ

Операция	Наименование	Примеры
sizeof	a) sizof (унарное выражение) b) sizof (тип)	sizeof (int) = 4 sizeof (float) = 4 sizeof (short) = 2 sizeof (double) = 8 sizeof (long) = 4 sizeof (char) = 1

Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Слайд 29

На данном слайде - продолжение таблицы унарных операций.

Функцию **sizeof** моно использовать в двух вариантах, когда аргументом может быть любое арифметическое выражение, или стандартные типы данных.

С помощью операции **sizeof** можно определить объем памяти любого выражения или объем памяти, отводимых под переменные определенного типа. Аргументом функции является обозначения стандартных типов данных. Это очень удобно, так как в разных системах объем памяти, предусмотренный для хранения переменных разных типов данных, могут быть различными.

Применение операции **sizeof** к массиву дает количество байтов, занимаемых массивом, а не количество его элементов и не размер в байтах каждого из них.

Применение операции **sizeof** к указателю дает размер самого указателя, а не объекта, на который он указывает

БИНАРНЫЕ ОПЕРАЦИИ

Операция	Наименование	Примеры применения
+	бинарный плюс	c=a+b
-	бинарный минус	c=a-b
*	умножение операндов	c=a*b
a=2, b=4, c=5	y=a+b*c, y=22	
	y=(a+b)*c, y=30	

Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Слайд 30

«Арифметические выражения в любом языке программирования записываются в строчку. Они могут содержать константы, то есть постоянные значения, имена переменных, знаки арифметических операций, круглые скобки, которые нужны для изменения порядка действий, и вызовы функций. При определении порядка действий используется приоритет, так называемое старшинство операций. Если в выражение входят переменные разных типов, в некоторых случаях происходит автоматическое приведение типа к более широкому. Например, результат умножения целого числа на вещественное – это вещественное число. Результат деления целого числа на целое – это всегда целое число, остаток при делении отбрасывается. Когда нужно получить вещественный результат деления, одно из чисел, делимое или делитель, нужно преобразовать к вещественному типу: для числа поставить десятичную точку, а для переменной или выражения использовать явное приведение типа.»

В бинарных операциях участвует два операнда. Бинарные операции делятся на аддитивные и мультипликативные.

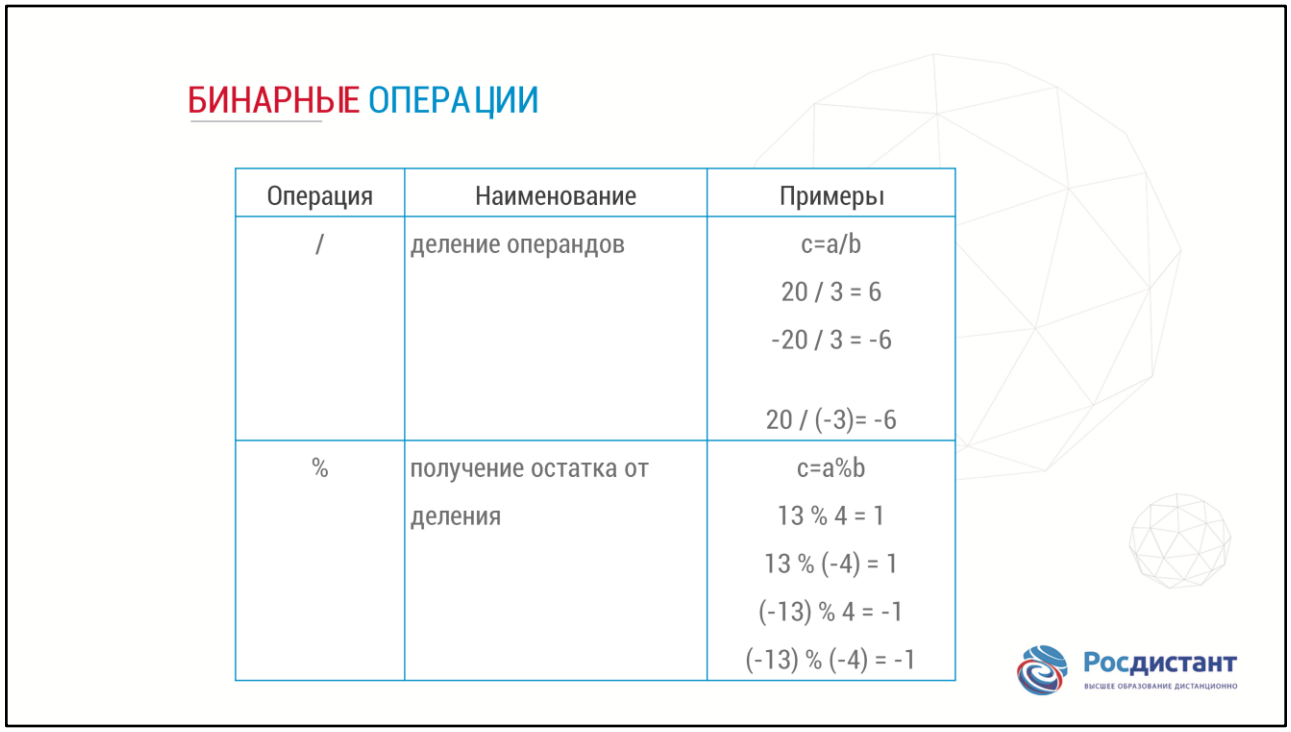
К аддитивным операциям относятся операции бинарный плюс, бинарный минус и умножение операндов.

При выполнении аддитивных операций выполняется основное правило – приоритет выполнения операций. На слайде приведены выражения, в которых

используются аддитивные операции. Изменить приоритет выполнения операций можно с помощью скобок. Если в выражении встречаются операции, имеющие одинаковый приоритет, то в силу вступает правило их выполнения с лева на право.

При использовании аддитивных операций операндами могут быть любые выражения.

Эти операции являются операциями присваивания, поэтому слева от знака равенства допустимы только имена переменных, а справа – любое арифметическое выражение.



Слайд 31

К мультипликативным операциям относятся деление операндов и получение остатка от деления целочисленных операндов.

«Операции умножения и деления выполняются над целыми и вещественными операндами. Типы первого и второго операндов могут отличаться, при этом выполняются преобразования операндов по умолчанию. Типом результата является тип операндов после преобразования.

В процессе выполнения мультипликативных операций ситуация переполнения или потери значимости не контролируется. Если результат мультипликативной операции не может быть представлен типом операндов после преобразования, то информация теряется.

Операция деления выполняет деление первого своего операнда на второй. Если оба операнда являются целыми значениями не делятся нацело, то результат округляется в сторону нуля. Деление на ноль дает ошибку во время выполнения.»

При делении целочисленных операндов результат округляется до целого, знак результата будет отрицательным, если отрицательным является или делимое или делитель.

При выполнении операции остаток от деления целочисленных операндов знак результата зависит только от знака делимого.

Особое внимание следует уделить результатам выполнения этих операций при отрицательных операндах. На слайде приведены примеры, которые демонстрируют указанные выше правила.

БИНАРНЫЕ ОПЕРАЦИИ СДВИГА

Операция	Наименование	Примеры применения
<<	сдвиг влево	4<<2=16 $4_{(10)}$ --> $100_{(2)}$ после сдвига: $10\ 000_{(2)}$ --> $16_{(10)}$
>>	сдвиг вправо	5>>2=1 $5_{(10)}$-->$101_{(2)}$ после сдвига: $10_{(2)}$-->$1_{(10)}$

char x;
x=7;
x=x<<1;
x=x>>2
	
	0000 0111
	0000 1110
	0000 0011
		
		7
		14
		3

Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Слайд 32

«В далеком прошлом компьютерной памяти было очень мало и ею сильно дорожили. Это было стимулом максимально разумно использовать каждый доступный бит. Используя побитовые операторы, можно создавать функции, которые позволят уместить восемь значений логического типа в переменную размером один байт, что значительно сэкономит потребление памяти. В прошлом такой трюк был очень популярен.

Теперь памяти стало существенно больше и программисты обнаружили, что лучше писать код так, чтобы было проще и понятнее его поддерживать, нежели усложнять его ради незначительной экономии памяти. Поэтому спрос на использование побитовых операторов несколько уменьшился, за исключением случаев, когда необходима уж максимальная оптимизация.»

Рассмотрим операции сдвига. Операция сдвиг влево битового представление значение левого целочисленного операнда на количество разрядов правого целочисленного операнда на **n** позиции соответствует увеличению первого операнда в **2n** раз**.**

«Операция сдвиг вправо битового представления значения левого целочисленного операнда на количество разрядов правого целочисленного операнда соответствует уменьшению первого операнда в **2n** раз с отбрасыванием дробной части результата.

Операции битового сдвига могут быть полезны при декодировании информации от внешних устройств и для чтения информации о статусе. Операторы битового сдвига могут также использоваться для выполнения быстрого умножения и деления целых чисел. Сдвиг влево равносилен умножению на 2, а сдвиг вправо - делению на 2, как показано в примере в таблице.»

Приведенные примеры демонстрируют работу операций сдвига с указанием двоичного представления чисел и результатов выполнения этих операций.

ПОРАЗРЯДНЫЕ ЛОГИЧЕСКИЕ ОПЕРАЦИИ

Операция	Наименование	Примеры применения
&	конъюнкция	110 6 & 5 = 4 101 100 --> $4_{(10)}$
|	дизъюнкция	110 6 | 5 = 7 101 111 -->$7_{(10)}$
^	поразрядная логическая операция исключающего «ИЛИ»	110 6 ^ 5 = 3 101 011 --> $3_{(10)}$

Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Слайд 33

На данном слайде приведены поразрядные логические операции.

Конъюнкция – это поразрядная логическая операция И, умножение, сравнивается каждый бит первого операнда с соответствующим битом второго операнда. Если оба бита равны единице, то результат равен единице, иначе равен нулю.

Дизъюнкция – это поразрядная логическая операция ИЛИ, сложение, сравнивается каждый бит первого операнда с соответствующим битом второго операнда. Если хотя бы один или оба бита равны единице, то результат равен единице, иначе равен нулю.

При выполнении поразрядной логической операция исключающего ИЛИ **с**равнивается каждый бит первого операнда с соответствующим битом второго. Если один бит нулю, а второй единице, результат устанавливается единица, если оба бита равны нулю или оба равны единице, то результат будет равен нулю.

Приведенные примеры демонстрируют работу поразрядных операций.

ЛОГИЧЕСКИЕ ОПЕРАЦИИ ОТНОШЕНИЙ

Операция	Наименование	Примеры применения
< , <= > , >=	операции отношения арифметических операндов	a+b > = x+y+z
= =	равно	a+b = = c+d
!=	не равно	a+b != c+d

Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Слайд 34

В программах на языке **C++** операции отношений используются для сравнения двух величин между собой. Этими величинами могут быть числа, переменные, константы, результаты вычислений выражений. Операции отношений возвращают одно из двух возможных значений:

- **true ,** если результат операции отношения исполняется, то есть истинно;
- **false ,** если результат операции отношения не исполняется, то есть ложно.

Операции отношений являются бинарными. Они требуют двух операндов.

В операциях отношения операнды должны быть арифметического типа, результат всегда целочисленный.

Обратите внимание на операцию равно, это два рядом стоящих знака равенства, и на знак не равно – это восклицательный знак и знак равенства.

Операции отношений могут использоваться:

- в циклах, где есть условие выполнения цикла;
- в операторах присваивания, содержащих логические выражения, если нужно определить результат сложного логического выражения;
- в операторах условного перехода **if .**

Если проверяется одновременно несколько условий, то они должны быть

связаны операциями отношений.

ЛОГИЧЕСКИЕ ОПЕРАЦИИ ОТНОШЕНИЙ

Операция	Наименование	Примеры применения
&&	логическая операция «И» арифметических операндов	3 + 4>5 && 4 + 5>3 Результат «ложь»
||	логическая операция «ИЛИ» арифметических операндов	3 ! = 5 || 3 = = 5 Результат «истина»

Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Слайд 35

В логических операциях, операнды должны быть арифметического типа, результат всегда целочисленный.

В логических операциях результатом может быть одно из двух значений:

- ноль, что означает - ложь;
- единица, что означает - истина.

Логические операции И и ИЛИ используются как логические связки, если одновременно проверяется несколько условий в операторах проверки условий.

Результатом операция И - будет истина, если истинны все входящие в выражение операнды.

Результатом операция ИЛИ - будет истина, если истинным является хотя бы один из операндов, входящих в выражение.

На слайде приведены примеры логических операций.

ОПЕРАЦИИ ПРИСВАИВАНИЯ

Операция	На C++	Идентичная запись
+=	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
&=	a&=b	a=a&b
| =	a |= b	a = a | b
^ =	a ^ = b	a = a ^ b

1. int n; float y,b; y=b/(float)n;	2. int a=b=2;

Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Слайд 36

«Присваивание значений переменных делается командой присваивания. На языке C++ имеются помимо привычных и сокращенные формы записи арифметических операций.»

На слайде во втором столбце приведены примеры сокращенной формы записи арифметических операций присваивания на **C++** и в третьем столбце - привычная запись.

В операциях присваивания участвует левый операнд и выражение, стоящее справа от знака равенства.

На данном слайде показаны два варианта использования операций присваивания. Однако, считается хорошим тоном использовать на языке **C++** запись операций присваивания, как показано во второй колонке данной таблицы.

В выражении, стоящим справа от знака равенства, участвует операнд, стоящий слева от знака равенства.

При выполнении арифметических операций соблюдается иерархия выполнения арифметических операций.

Изменить приоритет выполнения арифметических операций можно с помощью скобок. Если в выражении участвуют операции, имеющие одинаковый

приоритет, в силу вступает правило их выполнения слева на право.

«Обратите внимание на первый пример. Допустимы смешанные арифметические выражения, то есть в одном арифметическом выражении присутствуют операнды различных типов.

Наличие хотя бы одного вещественного операнда приводит к получению результата вещественного. При делении, если в операции участвуют два целых числа, то результат деления будет округляться до целого числа, даже если результат присваивается переменной вещественной переменной. Чтобы результат представлял число с плавающей точкой, один из операндов также должен представлять число с плавающей точкой, то есть применить явное преобразование типов.»

В **C++** можно совмещать операции присваивания. Обратите внимание на второй пример.

СТАНДАРТНЫЕ ФУНКЦИИ

<math.h>			
Функция	C++	Тип результата	Тип аргумента
arcsin x	asin(x)	double	double
arccos x	acos(x)	double	double
arctg x	atan(x)	double	double
sin x	sin(x)	double	double
cos x	cos(x)	double	double
tg x	tan(x)	double	double
e^x	exp(x)	double	double
$x^{рад} = x^0 \cdot 180 / \pi$			(1)

Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Слайд 37

Программа может вызывать множество функций из реализации стандартной библиотеки **C++ .** Данные функции выполняют важные задачи, например занимаются вводом и выводом, а также предоставляют эффективные реализации часто используемых операций.

На слайде представлена таблица стандартных математических функций. Для использования стандартных функций необходимо подключить библиотеку стандартных функций, заголовочный файл математика. В первой строке таблицы показано название стандартной библиотеки, которую необходимо подключить к программе для использования этих функций.

В таблице указаны тип аргумента, для которого можно использовать функции, и тип получаемого результата, то есть эти функции используются для вещественных аргументов.

Не забывайте об этом условии использования стандартных библиотечных функций, иначе могут возникнуть сообщения об ошибках при компиляции программы.

Аргумент любой функции указывается в скобках и может быть любым арифметическим выражением. Не забывайте, что тип арифметического выражения зависит от типов входящих в выражение операндов. Аргументы тригонометрических функций должны быть указаны в радианах. Если значение

аргумента функции имеет значение в градусах, его необходимо перевести в радианы. На слайде приведена формула перевода из градусов в радианы – формула 1.

СТАНДАРТНЫЕ ФУНКЦИИ

<math.h>			
Фун-ия	C++	Тип результата	Тип аргумента
$\sqrt{x}$	sgrt(x)	double	double
ln x	log(x)	double	double
$\log_{10} x$	log10(x)	double	double
x^y	pow(x,y)	double	double
P^{10}	pow10(p)	double	int
|x|	abs(x) fabs(x)	double	int double

Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Слайд 38

На языке **C++** нет ключевых слов, обеспечивающих ввод-вывод, обрабатывающих строки, выполняющих различные математические вычисления или какие-нибудь другие полезные процедуры. Все эти операции выполняются за счет использования набора библиотечных функций, поддерживаемых компилятором. Существует два основных вида библиотек: библиотека **C** , которая поддерживается всеми компиляторами и библиотека классов **C++ ,** которая годится только для языка **C++ .** Прежде чем программа сможет использовать какую-нибудь библиотеку функций, она должна включить соответствующий заголовок. Под заголовками понимают заголовочные файлы.

На слайде приведены стандартные функции, использование которых требует подключения стандартной библиотеки математика.

Рассмотрим функцию вычисления модуля числа. В зависимости от типа аргумента, целого типа или вещественного типа, используют разные функции.

Аргументом этих функций могут быть любые арифметические выражения. Аргумент указывается справа от имени функции в скобках.

При выполнении арифметических операций, у стандартных функций наивысший приоритет выполнения перед арифметическими операциями.

При использовании этих функций, необходимо следить за типом аргументов перечисленных функций. В таблице указаны типы аргументов и тип результата.

ПРИОРИТЕТ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ

Приоритет выполнения арифметических операций	
++a, -- a	постфиксный инкремент и декремент
( )	стандартные функции
a++, a - -	префиксный инкремент и декремент
*	разыменование
&	адрес
!	отрицание
+, -	унарный плюс, унарный минус
*, /, %	умножение, деление, остаток
<< , >>	побитовые сдвиги влево и вправо

Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Слайд 39

При вычислении арифметических операций соблюдаются следующие правила:

- соблюдается приоритет выполнения арифметических операций;
- два знака не могут следовать один за другим;
- если в выражении есть знаки одинакового приоритета в силу вступает правило их выполнения слева на право;
- изменить приоритет выполнения арифметических операций можно с помощью скобок.

Приоритетом называется очередность выполнения операций в выражении. Операция, находящаяся между двумя операциями с равными приоритетами, связывается с той операцией, которая находится слева. Выражение, заключенное в скобки, перед выполнением вычисляется как отдельный операнд.

Любое выражение языка состоит из операндов, переменных, констант, соединенных знаками операций. Операции выполняются в строгой последовательности. Величина, определяющая преимущественное право на выполнение той или иной операции, называется приоритетом. Порядок выполнения операций может регулироваться с помощью круглых скобок.

СТАНДАРТНЫЕ ФУНКЦИИ

<math.h>			
Функция	C++	Тип результата	Тип аргумента
Целая часть	floor(x)	double	floor (3, 7) = 3 floor (-3,7) = -4
Дробная часть	fmod(x)	double	fmod (10,5)=0 fmod (1,5)=1
Математическое выражение: $y = \frac{\sqrt{x+\sin(a+x)}}{a+e^{\lvert x\rvert}}$		Выражение на языке C++: y= sgrt(x+sin(a+x)) / (a+exp(abs(x)))	

Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Слайд 40

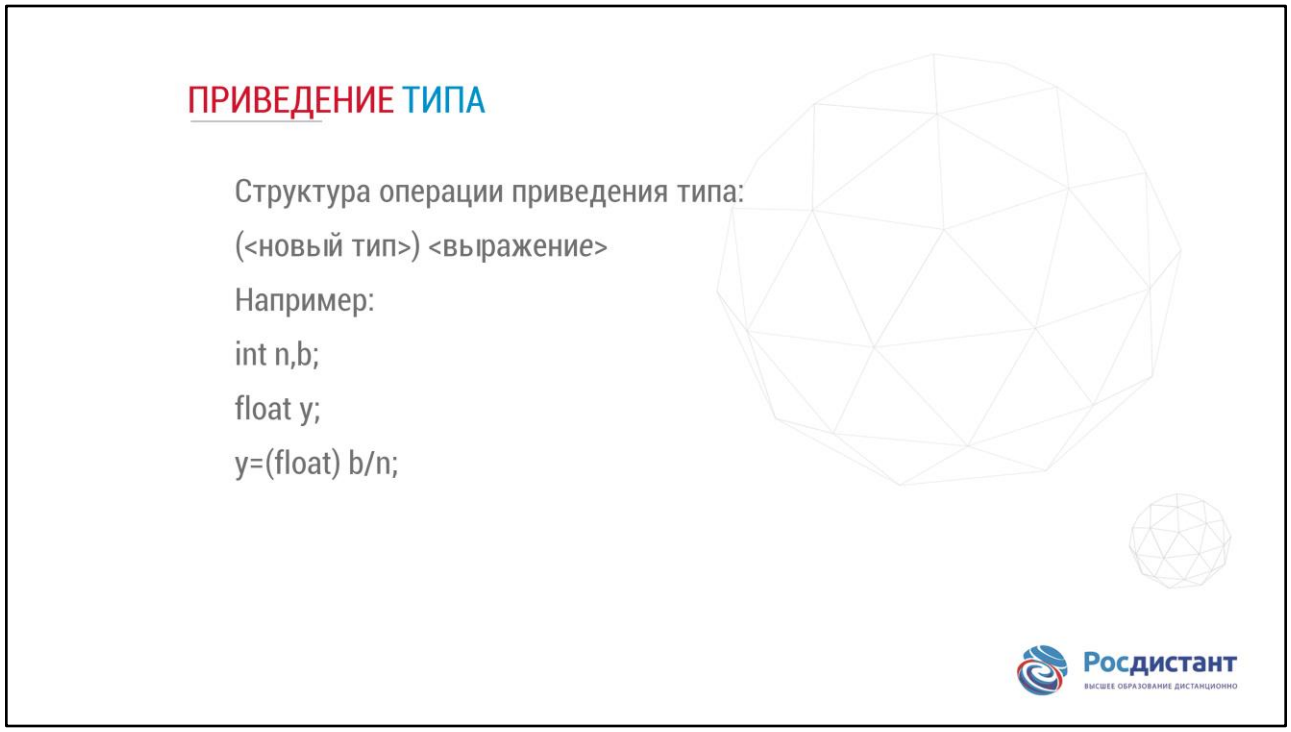
В верхней строке таблицы приведено название заголовочного файла, который необходимо подключить к программе при обращении к функциям, приведенным в таблице.

Функция **floor** вычисляет наименьшее ближайшее целое число, не превышающее **x** . Обратите внимание на пример применения этой функции к отрицательным числам.

Следует обратить внимание на результат работы этой функции для отрицательных аргументов. В таблице приведены примеры.

Функция **fmod** вычисляет остаток от деления **x** на **y** . Результат применения этой функции имеет вещественный тип.

Разберем подробнее пример, приведенный в последней строке таблицы. Аргументом функции вычисления корня квадратного является выражение, состоящее из двух слагаемых, соответственно все выражение заключено в скобки. Знаменатель, состоящий из двух слагаемых, также заключается в скобки. Иначе выражение исказится.



Слайд 41

На языке С++ допустимы смешанные выражения, то есть выражения, в которые входят операнды разных типов.

«При выполнении бинарных операций производятся преобразования по умолчанию для приведения операндов к одному и тому же типу, который потом используется как тип результата.

- если один из операндов имеет тип double , другой тоже преобразуется в double ;
- если один операнд имеет тип float , то второй операнд преобразуется к типу float ;
- bool преобразуется в int.
- если один операнд имеет тип **long int ,** то второй операнд преобразуется к типу long int ;
- если один операнд имеет тип **unsigned int ,** то второй операнд преобразуется к типу **unsigned int .**

В языке С++ нет операций преобразования между символом и кодом символа, т.к. в оперативной памяти символ и так храниться в виде его кода. Поэтому можно к переменной, хранящей символ, прибавить единицу и получить следующий символ.

Операция приведения типа продемонстрирована на слайде. При преобразовании типов надо помнить, что при преобразовании между знаковыми, или беззнаковыми значениями и при преобразовании от типа с большей размерностью к типу с меньшей размерностью могут возникнуть ошибки.»

ПРИМЕР ИСПОЛЬЗОВАНИЯ ОПЕРАЦИЙ

Действия (при начальном p=5)	Пояснения
p << = 2	$5_{(10)}$ -- > $101_{(2)}$ сдвиг влево
p >> = 6 – 5	сдвиг вправо
p & = 9 + 4	конъюнкция
p | = 8 – 2	дизъюнкция
p ^ = 10	поразрядная логическая операция

Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Слайд 42

Для закрепления пройденного материала, на слайде приведены некоторые операции.

В таблице, поэтапно приведены операции и наименования операций.

При начальном значении переменной, равным пяти, выполните указанные операции.

При выполнении операций сдвига, конъюнкции и дизъюнкции необходимо использовать двоичное представление чисел. Сдвиг влево – соответствует увеличению левого операнда, справа доставляются нули в количестве, равном правому операнду в выражении. Сдвиг вправо – означает уменьшение левого операнда. Удаляются разряды справа на количество, равное правому операнду в выражении.

Если полученный вами результат равен четырем, значит, вы правильно выполнили указанные операции.

Введение в языки программирования C и C++ | Уроки C++ - Ravesli / https://ravesli.com/urok-2-vvedenie-v-yazyki-programmirovaniya-c-i-s/

Т, А. Павловская С/С++ Программирование на языке высокого

уровня/http://cph.phys.spbu.ru/documents/First/books/7.pdf

Введение в программирование | Уроки C++ - Ravesli/https://ravesli.com/urok-1-vvedenie-v-programmirovanie/

Технология программирования - Информатика, автоматизированные информационные технологии и системы/ https://studref.com/441961/informatika/tehnologiya_programmirovaniya

Бьерн Страуструп. Язык программирования C++ 11/ https://vk.com/doc-145125017_450056359

Язык СИ++ Учебное собие / http://5fan.ru/wievjob.php?id=4301

А.А. Шелупанов, В.Н. Кирнос ИНФОРМАТИКА. БАЗОВЫЙ КУРС Учебник в четырех частях. /https://edu.tusur.ru/publications/521/download