Практическое занятие 9

Тема: Типы данных, операции и выражения в С++

Типы данных

Под типом данных (data type) понимают множество допустимых значений этих данных и множество разрешенных операций над ними. Вместе тип данных определяет и размер памяти, занимаемый переменными и константами данного типа. Память выделяется не для типа данных, а выделяется для размещения переменной или константы заданного типа.
В языке С++ выделяют встроенные и пользовательские типы данных.
Встроенные типы - это типы, которые поддерживает язык программирования.
Встроенные типы бывают простыми (базовыми) и производными, образующихся от базовых.
К простым типам относятся следующие:

- bool
- int
- char
- float
- double

К производным типам относятся следующие

- массивы (int a [5])
- указатели (int * a)

Существует также пустой тип void (не имеет значения).

Простые типы имеют набор значений и представлений, привязанных к архитектуре копьютера, на котором работает компиллятор.
В С++ простые типы могут быть модифицированы с помощью ключевых слов:

- short
- long
- signed
- unsigned

Кроме встроенных типов в языке С++ существуют пользовательские типы, которые требуют предварительного объявления. Как правило, такие типы являются составными

- структура (struct)
- объединения (union)
- перечисления (enum)
- классы (class)

Перечислим простые типы данных (от короткого до длинного по количеству информации). Основные характеристики типов данных языка С приведены в таблице 2.
bool, char, signed char, unsigned char, short, unsigned short, int, unsigned int, long, unsigned long, float, double, long double.

Таблица 2. Характеристики основных типов данных языкаС++

Имя типа	Размер памяти, в байтах	Діапазон значень для 16-розрядної архітектури	
[signed] char	1	-128	127
unsigned char	1	0	255

[signed] short [int]	2	-32 768	32 767
unsigned short [int]	2	0	65 535
[signed] int	Машинное слово	-32 768	32 767
unsigned int	Машинное слово	0	65 535
[signed] long [int]	4	-2 147 483 648	2 147 483 647
[unsigned] long [int]	4	0	4 294 967 295
float	4	3.4e-38	3.4e38
double	8	1.7e-308	1.7e308
long double	10	3.4e-4932	3.4e4932

Переменные и константы целых типов также могут объявляться с помощью модификаторов signed и unsigned (знаковое, беззнаковое).
При использовании модификаторов short и long разрешается опускать имя типа int.
Типы с плавающей точкой или действительные типы представлены тремя модификациями, характеризующие точность представления вещественных чисел:

float - единичной точности;
double - двойной точности;
long double - расширенной точности.

Для определения размера объекта программы в определенных единицах (для простых типов - в байтах), используется оператор sizeof:

```
cout << sizeof(int);
```

Диапазоны всех типов (максимальные и минимальные значения) можно получить из заголовочный файла climits (в более ранних реализациях — limits.h).

Операции и выражения в C++

Выражение - это последовательность знаков операций, операндов и круглых скобок, которая задает вычислительный процесс нахождения результата определенного типа. Простейшим выражением является константа, переменная или вызов функции.
Операции и выражения языка C++ позволяют задать определенную последовательность действий. Простые выражения содержат символ операции и операнды. Примером простого выражения с бинарной операцией является вычитание:

3.4 - x.

Операнд представляет собой элемент-участник операции. Операндами могут быть константы, переменные, вызовы функций и выражения.

В языке C++ определены следующие основные арифметические операции.
, -, *, / - это знаки бинарных операций сложения, вычитания, умножения и деления соответственно.

Существует два вида операций деления: целочисленное деление (деление с остатком) и деления без остатка (результатом такой операции является действительное число). Тип использующейся операции делений определяется типом операндов.

Для целых чисел определена операция % - деление по модулю (нахождение остатка от деления). Например, 5 % 2 = 1.

Примером унарной арифметической операции является операция изменения знака числа.
Компилятор по имени операции и типам операндов определяет возможность выполнения операции и операции, которые должны быть выполнены. Выполнение операции позволяет получить результат определенного типа и зависит от типов задействованных операндов.
Вызов функции - это использование имени функции, за которым в круглых скобках содержится список аргументов (возможно пустой). Примеры выражений с вызовом функции:

```
a*sin(x+k)+b*cos(x-k)
pow(2, n+1)-1
```

Символ = означает бинарную операцию простого присваивания, в результате выполнения которой значение правого операнда присваивается левому операнду.
Порядок вычисления выражения определяется расположением знаков операций, круглых скобок и приоритетами выполнения операций. Выражения с наивысшим приоритетом вычисляются первыми. Приоритеты операций представлены в табл. 1.

Таблица 1. Приоритеты, порядок выполнения и назначения операций в языке**C++**

При-оритет	Знаки операцій	Назви операцій	Порядок виконання
1.	. ->[] () ++ --	вибір елемента за іменем вибір елемента за вказівником вибір елемента за індексом виклик функції або конструювання значення постфіксний інкремент постфіксний декремент	зліва-направо
2.	sizeof ++ -- ~ ! + - & new delete (ім'я_типу)	розмір операнда в байтах префіксний інкремент префіксний декремент інверсія (порозрядне заперечення) логічне заперечення унарний плюс унарний мінус адреса виділення пам'яті або створення звільнення пам'яті або знищення перетворення типу	справа-наліво
3.	.* ->*	вибір елемента по імені через вказівник вибір елементу по вказівнику через вказівник	зліва-направо
4.	* / %	множення ділення остача від ділення цілих (ділення по модулю)	зліва-направо
5.	+ -	додавання віднімання	зліва-направо
6.	<< >>	зсув вліво зсув вправо	зліва-направо

7.	< > <= >=	менше більше менше або дорівнює більше або дорівнює	зліва-направо
8.	== !=	дорівнює не рівне	зліва-направо
9.	&	порозрядне І	зліва-направо
10.	^	порозрядне виключаюче АБО	зліва-направо
11.	|	порозрядне АБО	зліва-направо
12.	&&	логічне І	зліва-направо
13.	||	логічне АБО	зліва-направо
14.	?:	умовна операція	справа-наліво
15.	= *=, /=, %=, +=, -=, &=, ^=, |=	присвоювання (просте і складене)	справа-наліво
16.	*throw*	генерація виключення	справа-наліво
17.	,	послідовність виразів	зліва-направо

Если в выражении содержится несколько операций одного приоритета на том же уровне, то их обработка осуществляется в соответствии с порядком выполнения: слева направо или справа налево.

В языке С++ поддерживаются также две унарные операции инкремента и декремента ++ и -- для увеличения и уменьшения на единицу значение операнда соответственно. Их можно указывать как перед операндом, так и после него. В первом случае (++x или --x) значение операнда x меняется перед его использованием в выражении, а во втором (x++ или x--) - после его использования.

Примеры использования операций:

b=b1=3; (результат b=3, b1=3)

c=c1=5; (результат c=5, c1=5)

a=b+c++; (результат a=3+5=8, c=5+1=6)

a1=b1+(++c1); (результат b1=3; c1=5+l=6 ; a1=3+6=9)

<, >, ==, >=, <=, **!=** - Это символы операций отношения (меньше, больше, равно, больше или равно, меньше или равно, не равно), которые используются в логических выражениях;

&&, ||, ! - Знаки логических операций «И», «ИЛИ» и «НЕ»;

&, |, ^, <<, >>, ~ - Знаки поразрядных логических операций, выполняемых над одноименными битами машинного слова: «И», «ИЛИ», исключающее «ИЛИ», сдвиг влево, сдвиг вправо, инверсия;

?, : - Знаки, которые используются в условной тернарной операции. Например, результатом выражения (a>0) ? a : 0 будет значение a, если a>0 и нулевое значение - в противном случае.

*, & - Знаки адресных операций.

[i] - квадратные скобки используются при работе с массивами для задания размеров массива и доступа к элементу массива;

( ) - скобки используются для изменения приоритета выполнения операций в выражениях и при указании аргументов функций;

Удобно использовать также следующие операции присваивания:

*****=, /=, **%**=, +=, -=, **&**=, ^=, |=

Они применяются для компактноq записи операторов присваивания. Например, при выполнении двух операторов **a=a+b** и **a+=b** будет получен одинаковый результат.
Операторы заканчиваются точкой с запятой. Простые операторы можно объединить в составной или блок с помощью фигурных скобок, по которым точку с запятой можно не ставить.

Практическое задание

Задание 1. По приведенному ниже примеру создайте программу, в которой описаны несколько переменных разного типа, введите в них значения и выведите на экран. Измените значения переменных и проанализируйте результаты выполнения программы.

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
int chislo1;
float chislo2;
char const* simvol="d";
string stroka;
bool logika;

    cout << endl << "OUTPUT" << endl;
   chislo1=12;
   cout << endl << "chislo 1 = " << chislo1;
   chislo2=chislo1*2;
   cout << endl << "chislo1 * 2 = " << chislo2;
   chislo2=chislo2/5;
   cout << endl << "chislo 2 = " << chislo2;
   cout << endl << "symbol = " << simvol;
   stroka="Hello, World!";
   cout << endl << "stroka = " << stroka;
   logika= chislo1 > chislo2;
   cout << endl << "logika = " << logika << endl;

    cout << "Press the enter key to continue ...";
    cin.get();
    return EXIT_SUCCESS;
}
```

Задание 2. Напишите программу, которая демонстрирует приоритеты операций. Дайте пояснения к ней.
Задание 3. Напишите программу которая выводит размеры объектов различного типа. Дайте пояснения к ней.
Задание 4. Составьте отчет, в который включите результаты по пунктам 1-3. Приведите в отчет скрин-шоты результатов работы программ.