**Глава 5. Операторы ввода и вывода данных**

**5.1 Потоки ввода и вывода данных в С++**

Поток – это абстрактное понятие. Оно относится к любому переносу данных от источника к приёмнику. Кроме того, поток – это последовательность байтов, независимых от конкретного устройства, с которого производится считывание данных.

Потоки ввода и вывода данных в С++ обеспечивают надёжную работу со стандартными и пользовательскими типами данных. Потоки имеют единообразный синтаксис. Обмен с потоком производится через специальную область памяти компьютера – буфер. Передача данных выполняется при выводе (если буфер заполнен) и при вводе (если буфер исчерпан).

*Извлечение* – это процесс вывода данных из потока.

*Помещение (включение*) – это процесс ввода данных в поток.

Чтобы вывести на экран набранное сообщение или, наоборот, ввести текст с клавиатуры, используется специальный библиотечный файл iostream. Библиотека iostream является частью стандартной библиотеки C++, которая работает с базовым вводом и выводом. Для подключения такого файла нужно в самом начале программы написать #include<iostream>. Использование функций представленной библиотеки нужно для ввода данных с клавиатуры и вывода данных в консоль.

Библиотека iostream определяет два стандартных потока: cin и cout. cin – это стандартный входной поток. Он связывается с клавиатурой. Туда пользователь записывает данные. cout – это стандартный выходной поток. Он связывается с экраном. Данные выводятся из потока в консоль. Так как оба потока находятся в пространстве имен std, его следует подключить. Для этого необходимо прописать отдельную строку «using namespace std;».

Библиотека iostream предусматривает два основных класса для ввода и вывода данных: istream и ostream. Класс istream использует для ввода операцию извлечения из потока (знак >>). Класс ostream использует для вывода операцию помещения в поток (знака <<). Операции извлечения из потока и помещения данных в поток определены перегрузкой операции сдвига. Чтобы перейти на следующую строку, нужно использовать манипулятор endl. Он работает со стандартным потоком cout. Кроме этого, можно использовать служебный символ '\n'.

*Перегрузка операторов* – это изменение смысла оператора. При этом возможно одновременное существование нескольких вариантов применения оператора. Операторы потокового ввода или вывода данных уже являются перегруженными: при использовании с объектами cin и cout происходит извлечение или помещение данных в поток, а с объектами типа int – сдвиг битов вправо или влево.

**5.2 Оператор битового сдвига**

В C++ реализованы операторы *побитового сдвига* – оператор сдвига влево (<<) и вправо (>>). Данные операторы заставляют биты левого операнда сдвинуться влево или вправо на то количество позиций, которое указано во втором операнде. Каждое целое число в памяти представлено в виде определенного количества разрядов. Операции сдвига применяются только к целочисленным операндам [7].

Синтаксис: сначала записывается имя переменной, к которой применяется сдвиг, затем один из двух знаков оператора сдвига (<< или >>), и количество бит, на которое число сдвигается (рисунок 5.1).

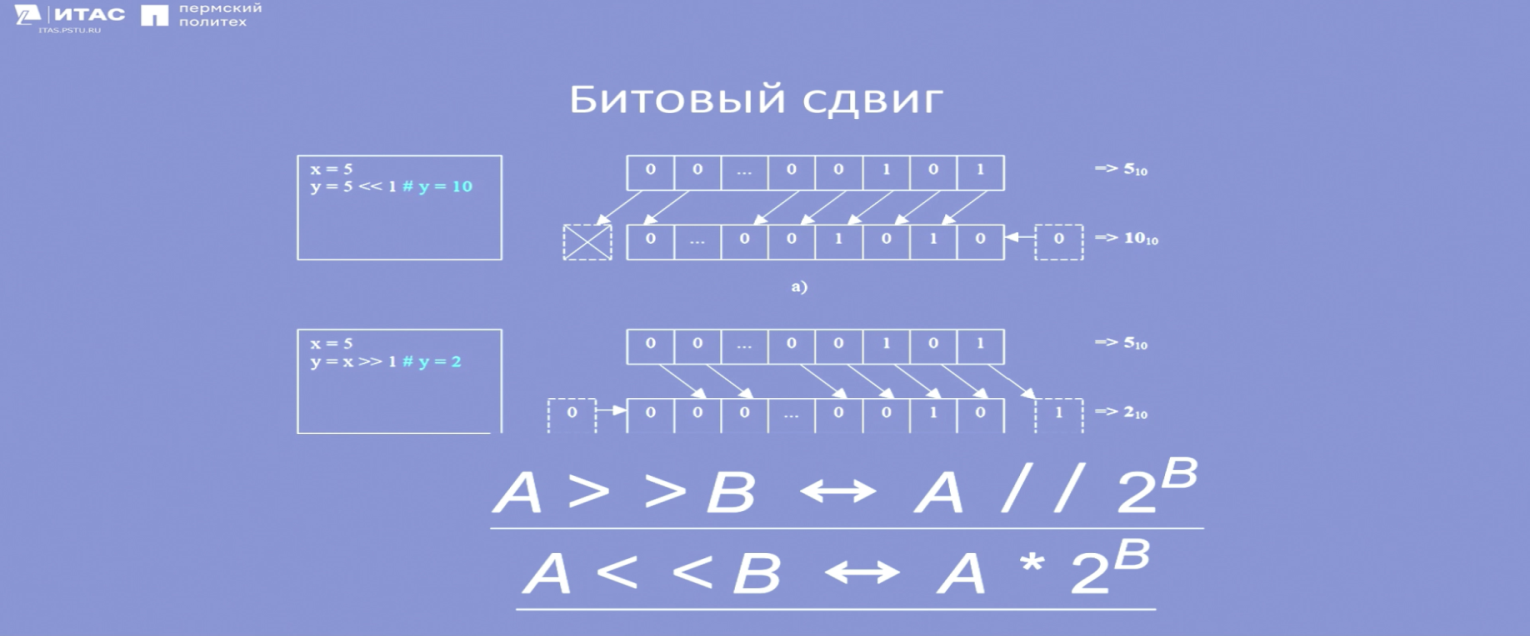


Рисунок 5.1 – Операция бинарного сдвига

Пример операции битового сдвига представлен на рисунке 5.2. При сдвиге числа 12 с битовым представлением 1100 влево на 4 бита получается число с битовым представлением 11000000, равное 192 в десятичной системе счисления. При битовом сдвиге числа 15, представленного в виде 1111, вправо на 1 получается число с битовым представлением 111 – это 7.

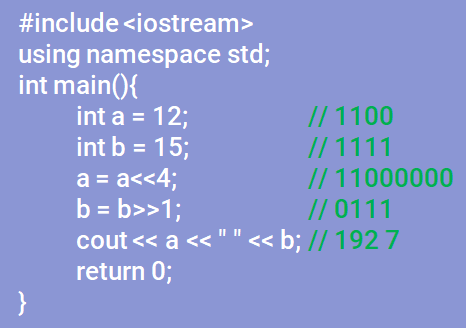


Рисунок 5.2 – Пример бинарного сдвига

**5.3 Практическая работа с потоком**

Более подробно работа с потоком представлена на рисунке 5.3. Так, при помещении чисел 24 и 48 в поток cout они выводятся слитно в консоли. Чтобы выводить числа, строки или информацию в целом в разных строках, нужно поместить в поток управляющие символы – ‘\n’ или слово "endl".

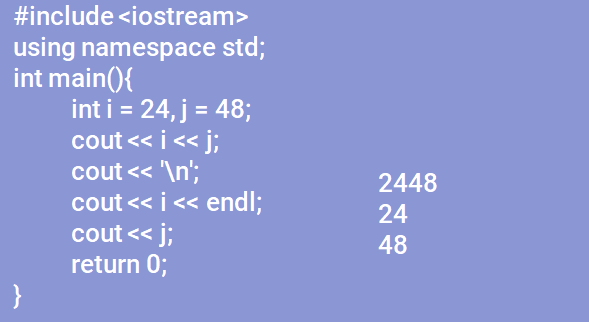


Рисунок 5.3 – Пример работы с потоком вывода cout

cin – это поток, куда пользователь записывает данные. На рисунке 5.4 показано, что данные извлекаются из потока cin и попадают в поток cout. Так, пользователь напечатал число 6, оно извлеклось, переменная теперь равна 6, и в поток вывода записалась строка с сообщением и значением переменной [16].

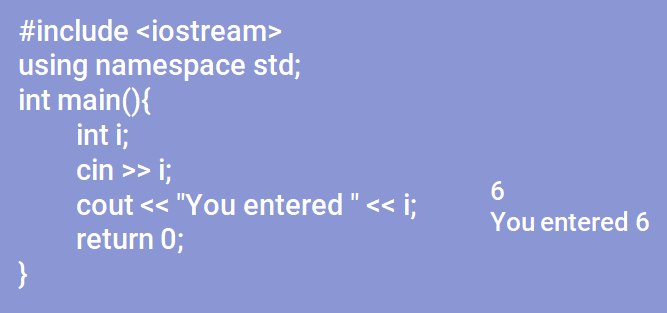


Рисунок 5.4 – Пример работы потоков ввода и вывода данных

В С++ есть возможность извлечь из одного потока не одну переменную, а сразу несколько (Рисунок 5.5). Для этого пользователю необходимо вводить переменные либо через пробел, либо через ввод, отделяя каждую переменную нажатием клавиши enter. От выбранного способа ввода будет зависеть вид записи в консоли (Рисунок 5.6) [19]. В остальном представленные способы похожи, и поэтому не важно, каким образом пользователь будет вводить данные.

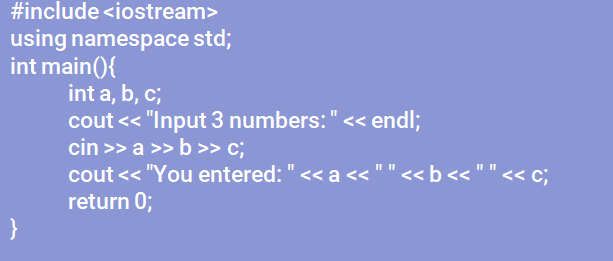


Рисунок 5.5 – Основные способы ввода данных с клавиатуры

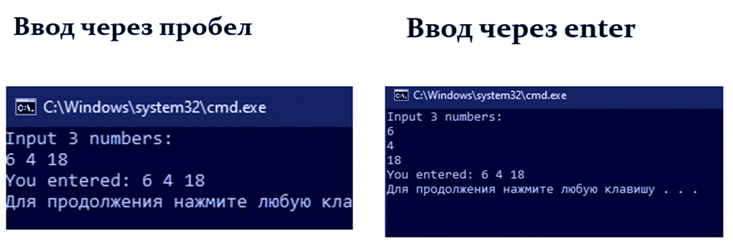


Рисунок 5.6 – Отображение данных, введенных разными способами

В C++ операторы имеют различные приоритеты. Если в выражении используется несколько операторов, то операции с высоким приоритетом выполняются раньше операций с низким приоритетом. Оператор помещения в поток имеет приоритет ниже, чем оператор сложения, однако оператор помещения в поток имеет приоритет выше, чем оператор сравнения и битового сдвига (рисунок 5.7).

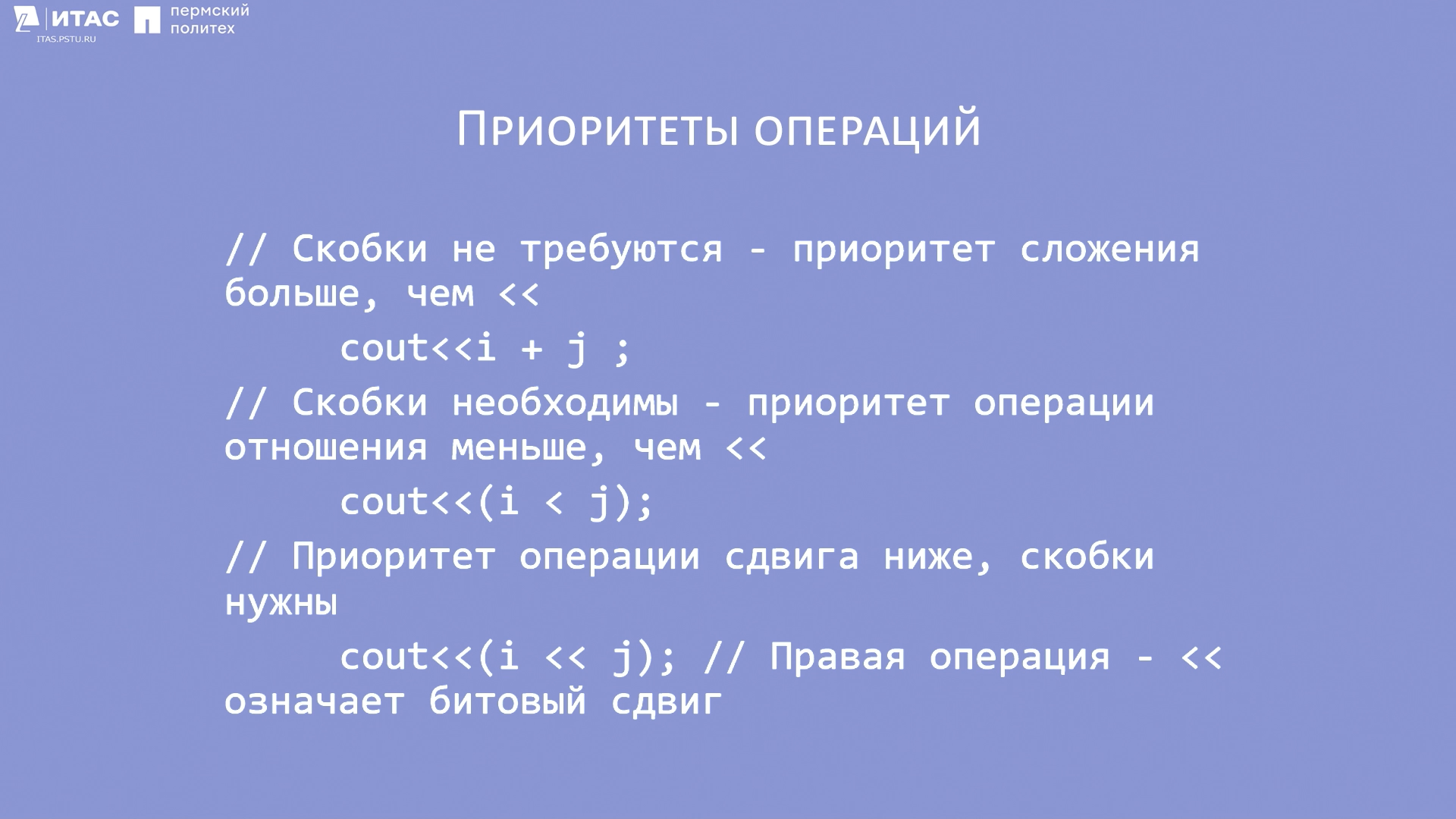


Рисунок 5.7 – Приоритеты операторов

**5.4 Локализация программы к кириллице**

Если программа при выводе на консоль использует кириллицу, то пользователь может столкнуться с ситуацией, когда вместо символов кириллицы будут отображаться другие знаки. Решение – использование функции setlocale(), унаследованной из библиотеки С. Для применения такой функции в ОС Windows в настройках языков и стандартов должна быть установлена кириллица. Либо можно открыть файл в кодировке windows1251, написав строку: system(“chcp1251>NULL”); и подключить к программе заголовочный файл <clocale>. Пример ввода кириллицы изображён на рисунке 5.8.

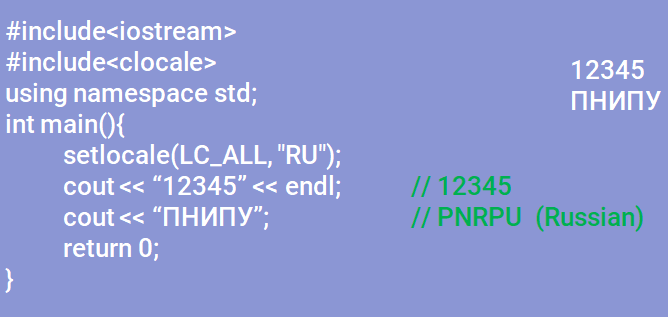


Рисунок 5.8 – Локализация программы

В представленном коде программы произведена локализация через функцию setlocale(), поэтому на экран вывелись буквы российского алфавита.

Для закрепления материала по операторам ввода и вывода нужно выполнить две практические задачи, представленные ниже. Представленные задачи связаны с вводом данных пользователем и выводом информации и значений переменных в консоль:

1. Написать программу для умножения двух чисел и вывода на экран результата. Числа задаёт пользователь.

2. Написать фрагмент программы, в которой вводятся пять символов, а затем они выводятся на экран в обратной последовательности.