36. Маніпулятори. Побудова та використання маніпуляторів без параметрів.

Все функции-манипуляторы вывода без параметров имеют следующий формат записи:  
  
ostream &имя\_манипулятора (ostream &поток)// кодreturn поток;}

Здесь имя\_манипулятора представляет собой имя манипулятора. Важно отметить, что несмотря на то, что манипулятор имеет один аргумент-указатель на поток, при использовании этого ма­нипулятора в операции вывода в его записи нет аргументов.

В следующей программе создается манипулятор setup(), устанавливающий флаг левого вырав­нивания, задающий ширину поля равной 10 и устанавливающий знак доллара в качестве символа заполнения.  
  
#include <iostream.h>  
#include <iomanip.h>  
ostream &setup(ostream &stream)  
{  
stream.setf(ios::Ieft);  
stream << setw(10) << setfill ( ' $ ' );  
return stream;  
}  
int main()  
{  
cout << 10 << " " << setup << 10;  
return 0;  
}

Имеются две причины для того, чтобы приспосабливать манипуляторы под потребности пользо­вателя. Первая заключается в том, что может понадобиться выполнять операции ввода-вывода на устройство, для которого не существует предопределенных манипуляторов. Примером служит плоттер. В таком случае создание собственного манипулятора делает вывод данных на такое устройство более удобным. Вторая причина заключается в том, что однажды можно обнаружить повторение одной и той же последовательности операций во многих случаях. Можно собрать эти операции в одном едином манипуляторе, как это проиллюстрировано в предыдущей программе.

Общая форма записи функций-манипуляторов ввода без параметров имеет следующий вид:  
  
istream &имя\_манипулятора(istream &поток)// кодreturn поток;}

Например, следующая программа создает манипулятор prompt() для вывода на экран подсказки и переключения ввода на шестнадцатиричный:  
  
#include <iostream.h>  
#include <iomanip.h>  
istream &prompt(istream &stream)  
{  
cin >> hex;  
cout << "Enter number using hex format: ";  
return stream;  
}  
int main()  
{  
int i;  
cin >> prompt >> i;  
cout << i;  
return 0;  
}

Принципиально важно то, что манипулятор возвращает поток. Если этого не сделать, то тогда манипулятор не может быть использован в больших выражениях ввода/вывода.