



Лекція №2

Потокові класи мови C++

■ Поточкові класи мови C++

■ 2.1 Мета роботи

■ 2.2 Вказівки щодо організації самостійної роботи студентів

- Якщо формат виводу, використовуваний за умовчанням, не влаштовує програміста, він може скоригувати його за допомогою методів класів вводу / виводу, прапорів форматування і так званих маніпуляторів. Про це розповідається далі.

- **Форматування даних**
- У потокових класах форматування виконується трьома способами - за допомогою прапорів, маніпуляторів і форматуючих методів.
- **Прапори і форматуючі методи**
- *Прапори* являють собою окремі біти, об'єднані в поле *x_flags* типу `long` класу `ios`. Прапори перераховані в табл. 10.1.
- **Таблиця 10.1.** Прапори форматування

Флаг	Положение	Умолчание	Описание действия при установленном бите
skipws	0x0001	+	При извлечении пробельные символы игнорируются
left	0x0002		Выравнивание по левому краю поля
right	0x0004	+	Выравнивание по правому краю поля
internal	0x0008		Знак числа выводится по левому краю, число — по правому. Промежуток заполняется символами x_fill по умолчанию пробелами
dec	0x0010	+	Десятичная система счисления

oct	0x0020	Восьмеричная система счисления
hex	0x0040	Шестнадцатеричная система счисления
showbase	0x0080	Выводится основание системы счисления (0x для шестнадцатеричных чисел и 0 для восьмеричных)
showpoint	0x0100	При выводе вещественных чисел печатать десятичную точку и дробную часть
uppercase	0x0200	При выводе использовать символы верхнего регистра
showpos	0x0400	Печатать знак при выводе положительных чисел

scientific	0x0800		Печатать вещественные числа в форме мантиссы с порядком
fixed	0x1000		Печатать вещественные числа в форме с фиксированной точкой (точность определяется полем x precision, см. ниже)
unitbuf	0x2000		Выгружать буферы всех потоков после каждого вывода
stdio	0x4000		Выгружать буферы потоков stdout и stderr после каждого вывода

- ПРИМІТКА
- Прапори (left, right і internal), (dec, oct і hex), а також (scientific і fixed) взаємно виключають один одного, тобто в кожен момент може бути встановлений тільки один прапор з кожної групи.
- Для управління прапорами в класі ios є методи flags, setf і unsetf:

<code>long ios::flags();</code>	— возвращает текущие флаги потока;
<code>long ios::flags (long);</code>	— присваивает флагам значение параметра;
<code>long ios::setf (long, long);</code>	— присваивает флагам, биты которых установлены в первом параметре, значение соответствующих битов второго параметра;
<code>long ios::setf(long);</code>	— устанавливает флаги, биты которых установлены в параметре;
<code>long ios::unsetf(long);</code>	— сбрасывает флаги, биты которых установлены в параметре.

Всі функції повертають колишні прапори потоку.
Наприклад, якщо спочатку задати вирівнювання по правому полю:

```
cout.setf(ios:: right);
```

- а потім необхідно вивести інформацію вирівняну по лівому полю, то спочатку необхідно скинути прапор вирівнювання потоку виводу по правому полю:
`cout.unsetf(ios::right);`
- а потім встановити прапор вирівнювання по лівому полю
- `cout.setf(ios::left);`
- Крім прапорів, для форматування використовуються наступні поля класу `ios`:

<code>int x_width</code>	— минимальная ширина поля вывода;
<code>int x_precision</code>	— количество цифр в дробной части при выводе вещественных чисел с фиксированной точкой или общее количество значащих цифр при выводе в форме с мантиссой и порядком;
<code>int x_fill</code>	— символ заполнения поля вывода.

Для управления этими полями используются методы `width`, `precision` и `fill`:

<code>int ios::width()</code>	— возвращает значение ширины поля вывода;
<code>int ios::width (int)</code>	— устанавливает ширину поля вывода в соответствии со значением параметра;
<code>int ios::precision()</code>	— возвращает значение точности представления при выводе вещественных чисел;
<code>int ios::precision (int)</code>	— устанавливает значение точности представления при выводе вещественных чисел, возвращает старое значение точности;

<code>char fill()</code>	— возвращает текущий символ заполнения;
<code>char fill (char)</code>	— устанавливает значение текущего символа заполнения, возвращает старое значение символа.

Перед установкою деяких прапорів потрібно скинути прапори, які не можуть бути встановлені одночасно з ними. Для цього зручно використовувати другим параметром методу `setf` перераховані нижче статичні константи класу `ios`:

- `adjustfield (left | right | internal)`
- `basefield (dec | oct | hex)`
- `floatfield (scientific | fixed)`

- Приклад форматування при виводі за допомогою прапорів і методів:
- Нехай у масиві структур зберігатиметься інформація про успішність студентів, яку необхідно вивести на екран у зручному для користувача вигляді.
- `#include <iostream>`
- `using namespace std;`
- `const int n=2;`
- `//Створимо структуру для зберігання інформації про студентів з`
- `//двома полями ПІБ та середній бал успішності`
- `struct Tstudent`
- `{`

- `char FIO[20];`
- `float bal;`
- `}StudentnPotok[n]={"Petrov",3.5,"Sidorov",5.0/3};`
- `int main (int argc, char * const argv[])`
- `{`
- `// Наступний фрагмент коду ілюструє форматування потоку виводу`
- `for(int i=0;i<n;i++)`
- `{`
- `//Форматування за допомогою методів потоку`
- `cout.unsetf(ios::right);//скидаємо прапор потоку вирівнювання`

- по правому
- //полю (це необхідно зробити обов'язково якщо до е
- // того прапор був встановлений)
- cout.setf(ios::left); //встановити прапор вирівнювання по лівому полю
- cout.fill('.'); // задаємо символ – заповнювач
- cout.setf(ios::fixed); //встановити прапор виведення числа з фіксованою точкою
- // наступні два рядки встановлюють формат виведення чисел у вигляді
- // мантиси з порядком

- `// cout.setf(ios::scientific);`//встановити прапор виведення числа у форматі
- `//` мантиси з порядком
- `// cout.unsetf(ios::fixed);`//скидає прапор виведення числа з фіксованою
- `//`точкою (це необхідно зробити обов'язково якщо до цього прапор був
- `//`встановлений)
- `cout.precision(3);`//задаємо точність (кількість чисел після
- `//`коми або кількість значущих чисел)
- `cout.width(15);` // задати поле виведення під поле 20 символів.
- `//`параметр який необхідно вказувати кожен раз для виведення
- `cout <<StudentnPotok[i].FIO;`//виводимо прізвище
- `cout.setf(ios::right);`//задаємо виведення по правому

- //параметр який необхідно вказувати кожен раз для виведення
- `cout << StudetnPotok[i].FIO;`//виводимо прізвище
- `cout.setf(ios::right);`//задаємо виведення по правому полю
- `cout.width(10);`//задаємо кількість символів під виведення числа
- `cout.fill('_');`//змінюємо символ заповнювач
- `cout<<StudetnPotok[i].bal;`

■ Виведення

- `cout << StudetnPotok[i].FIO;`// виводимо прізвище
- `cout.setf(ios::right);`//задаємо виведення по правому полю
- `cout.width(10);`//задаємо кількість символів під виведення числа
- `cout.fill('_');`//мінємо символ заповнювач
- `cout<<StudetnPotok[i].bal;`

- `cout<<"\n";`
- `}`
- `cin.get();`
- `return 1;`
- `}`
- //Цей фрагмент виведе на екран наступний результат:
- Petrov....._____3.500
- Sidorov....._____1.667