



C++. Уровень 3

Урок 6

Потоки ввода-вывода

Иерархия потоковых классов.
Организация ввода-вывода.
Средства форматирования вывода.
Манипуляторы.

План урока

- Потоки в C++.
- Функционал классов `istream` и `ostream`.
- Потокосы классы.
- Перегрузка операторов ввода и вывода.



A photograph of a light blue ceramic cup filled with coffee, sitting on a burlap surface surrounded by scattered dark brown coffee beans. The image is dimmed to serve as a background for the title.

Потоки в C++



Системы ввода-вывода

printf()

scanf()



cout <<

cin >>

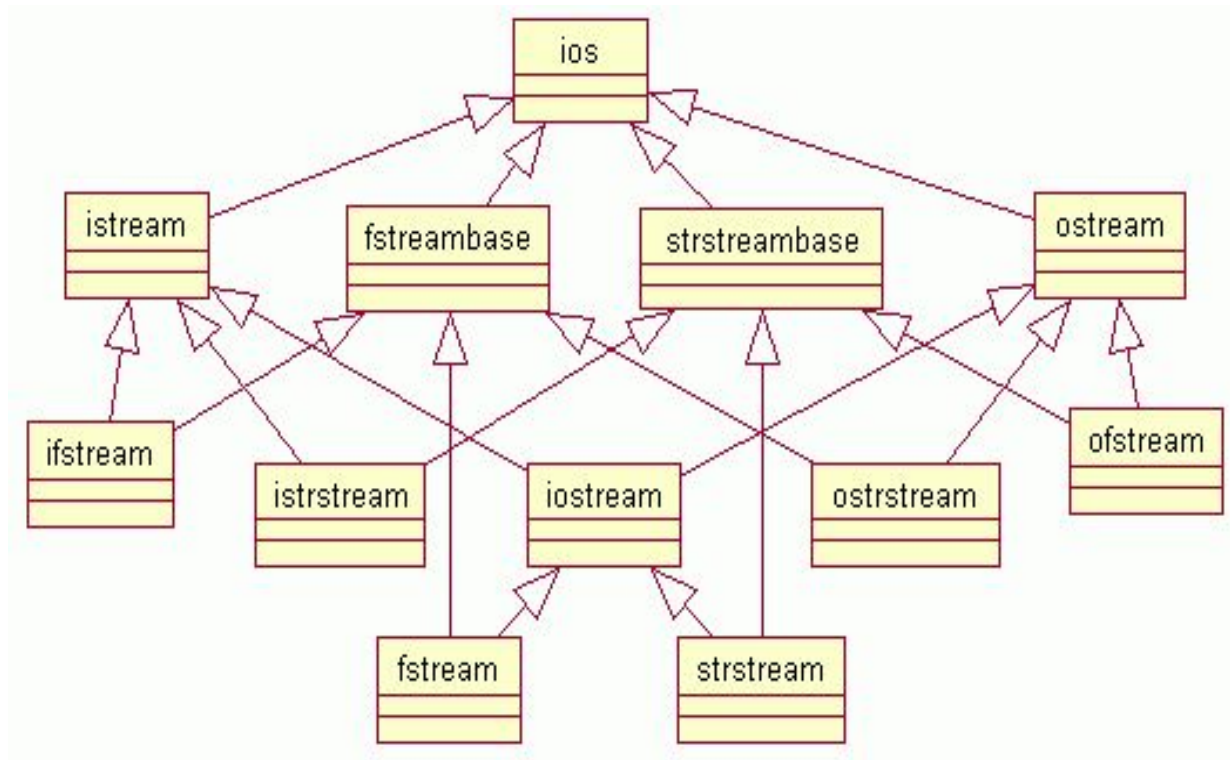


Потоки в C++

- Поток — это абстрактное понятие, относящееся к любому переносу данных от источника к приемнику.
- Потоки в C++, в отличие от функций ввода/вывода в стиле C, обеспечивают надежную работу как со стандартными, так и с определенными пользователем типами данных, а также единообразный и понятный синтаксис.



Потоковые классы



Операторы вставки и извлечения

`cout << variable`

Данные выводятся из
переменной на экран



`cin >> variable`

Данные поступают к
переменной из
клавиатуры



Предопределенные объекты-потоки

Имя	Тип класса	Тип класса
cin	istream_withassign	Стандартный ввод (клавиатура)
cout	ostream_withassign	Стандартный вывод (экран)
cerr	ostream_withassign	Стандартное устройство ошибок (экран) с небуферизованным выводом
clog	ostream_withassign	Стандартное устройство ошибок (экран) с буферизованным выводом



Функционал класса istream



Манипуляторы

Манипулятор	Назначение	Ввод/вывод
dec	Ввод/вывод данных в десятичной форме	ВВОД и ВЫВОД
endl	Вывод символа новой строки с передачей в поток всех данных из буфера	ВЫВОД
ends	Вывод нулевого символа	ВЫВОД
flush	Передача в поток содержимого буфера	ВЫВОД
hex	Ввод/вывод данных в шестнадцатеричной системе	ВВОД и ВЫВОД
oct	Ввод/вывод данных в восьмеричной форме	ВВОД и ВЫВОД
resetiosflags(long f)	Сбрасывает флаги, указанные в f	ВВОД и ВЫВОД
setbase(int base)	Устанавливает базу счисления равной параметру base	ВЫВОД
setfill(int ch)	Устанавливает символ заполнения равным ch	ВЫВОД
setiosflags(long f)	Устанавливает флаги, указанные в f	ВВОД и ВЫВОД
setprecision(int p)	Устанавливает число цифр после запятой	ВЫВОД
setw(int w)	Устанавливает ширину поля равной w	ВЫВОД
ws	Пропускает начальный символ-разделитель	ВВОД



Манипуляторы

Манипулятор	Аргумент	Назначение
<code>setw()</code>	ширина поля (<code>int</code>)	Устанавливает ширину поля для вывода данных
<code>setfill()</code>	символ заполнения (<code>int</code>)	Устанавливает символ заполнения (по умолчанию, пробел)
<code>setprecision()</code>	точность (<code>int</code>)	Устанавливает точность (число выводимых знаков)
<code>setiosflags()</code>	Флаги форматирования (<code>long</code>)	Устанавливает указанные флаги форматирования
<code>resetiosflags()</code>	Флаги форматирования (<code>long</code>)	Сбрасывает указанные флаги форматирования



Функции класса istream

Метод	Описание
ignore()	Метод извлекает один символ из потока istream и игнорирует его.
ignore(streamsize count)	Метод извлекает count-символ из потока istream и игнорирует их.
ignore(streamsize count, char delim)	Метод извлекает count-символ из потока istream и игнорирует их. Метод прекращает работу, если в потоке встречается символ delim.
get(char c)	Метод извлекает из потока один символ, не считывает символ новой строки.
getline(cin, string s)	Метод считывает символы из потока, при этом также считывает символ новой строки.
gcount()	Метод, возвращающий количество символов, извлеченных последним getline().
peek()	Метод возвращает следующий символ потока без реального чтения его из потока istream.
unget()	Метод помещает в поток последний считанный символ таким образом, что его можно прочитать следующей операцией чтения.



Функционал класса ostream



Класс ostream

Есть два способа управления параметрами форматирования вывода:

- **флаги** — это логические переменные, которые можно *включать/выключать*;
- **манипуляторы** — это объекты, которые помещаются в поток и влияют на способ ввода/вывода данных.

Группа форматирования — это группа флагов, которые выполняют аналогичные (иногда взаимоисключающие) параметры форматирования вывода.



Группы форматирования

Флаги группы форматирования floatfield:

fixed — используется десятичная запись чисел типа с плавающей запятой;

scientific — используется экспоненциальная запись чисел типа с плавающей запятой;

showpoint — всегда отображается десятичная точка и конечные нули для чисел типа с плавающей запятой.

Манипуляторы:

fixed — используется десятичная запись значений;

scientific — используется экспоненциальная запись значений;

showpoint — отображается десятичная точка и конечные нули чисел типа с плавающей запятой;

noshowpoint — не отображаются десятичная точка и конечные нули чисел типа с плавающей запятой;

setprecision(int) — задаем точность для чисел типа с плавающей запятой.



Флаги группы форматирования basefield:

oct («**octal**» = «восьмеричный») — восьмеричная система счисления;

dec («**decimal**» = «десятичный») — десятичная система счисления;

hex («**hexadecimal**» = «шестнадцатеричный») — шестнадцатеричная система счисления.

Флаги группы форматирования adjustfield:

internal — знак значения выравнивается по левому краю, а само значение — по правому краю;

left — значение и его знак выравниваются по левому краю;

right — значение и его знак выравниваются по правому краю.

Манипуляторы:

internal — знак значения выравнивается по левому краю, а само значение — по правому краю;

left — значение и его знак выравниваются по левому краю;

right — значение и его знак выравниваются по правому краю;

setfill(char) — задаем символ-заполнитель;

setw(int) — задаем ширину поля.



Перегрузка операторов ВВОДА-ВЫВОДА



Перегрузка оператора вывода

```
class Date
{
private:
    int m_day, m_month, m_year;
public:
    Date(int d=1, int m=1, int y=2019): m_day(d), m_month(m), m_year(y) { }
    friend ostream& operator<< (ostream &out, const Date &date);
};

ostream& operator<< (ostream &out, const Date &date)
{
    out << "Date: " << date.m_day << "." << date.m_month << "." << date.m_year << "\n";
    return out;
}

int main()
{
    Date date(5, 4, 2019);
    cout << date;
    return 0;
}
```



Перегрузка оператора ввода

```
istream& operator>> (istream &in, Date &date)
{
    // обратите внимание, параметр date (объект класса Date) должен быть не
    // константным, чтобы мы имели возможность изменить члены класса
    in >> date.m_day;
    in >> date.m_month;
    in >> date.m_year;

    return in;
}
```



Решите задачи



Что выведет следующая программа, если ввести строку:
1234

1

```
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    char firstNumber[3];
    char lastNumber[3];
    cin >> setw(3) >> firstNumber;
    cin >> setw(3) >> lastNumber;
    cout << firstNumber << endl;
    cout << lastNumber << endl;
    return 0;
}
```



Что произойдет, если пользователь захочет ввести:

***Hello
world!***

2

```
#include <iostream>
using namespace std;

int main()
{
    char text[12];
    cin.get(text, 12);
    cout << text << endl;
    return 0;
}
```



3

Что будет выведено на экран?

```
#include <iostream>
using namespace std;

int main()
{
    cout.setf(ios::showpos); // включаем флаг
    showpos
    cout << 30 << endl;
    cout.setf(ios::hex);
    cout << 32 << endl;
    cout.unsetf(ios::hex);
    cout << 32 << endl;
    return 0;
}
```



Вопросы участников

