Лекция 9. Потоки и обработка файлов

Казахский национальный университет имени аль-Фараби

Кафедра компьютерных наук

Объектно-ориентированное программирование

Лектор: старший преподаватель, PhD, Кенжебек Ержан Ғалымжанұлы

Цель лекции

Цель лекции

Изучить потоковые классы, их иерархию и средства ввода-вывода, а также научиться записывать данные в файл и читать их из файла.

Ожидаемые результаты

1. Описывать иерархию потоковых классов
2. Объяснять преимущества потокового ввода-вывода
3. Записывать данные в файл с помощью ofstream
4. Считывать данные из файла с помощью ifstream

План лекции

- Потоковые классы
- Иерархия потоковых классов
- Запись в файл
- Чтение данных

Потоковые классы

- Поток — это общее название, как ни странно, потока данных. В C++ поток представляет собой объект некоторого класса. Именно поэтому вы могли встретить в листингах потоковые объекты cin и cout. Разные потоки предназначены для представления разных видов данных. Например, класс ifstream олицетворяет собой поток данных от входного дискового файл.

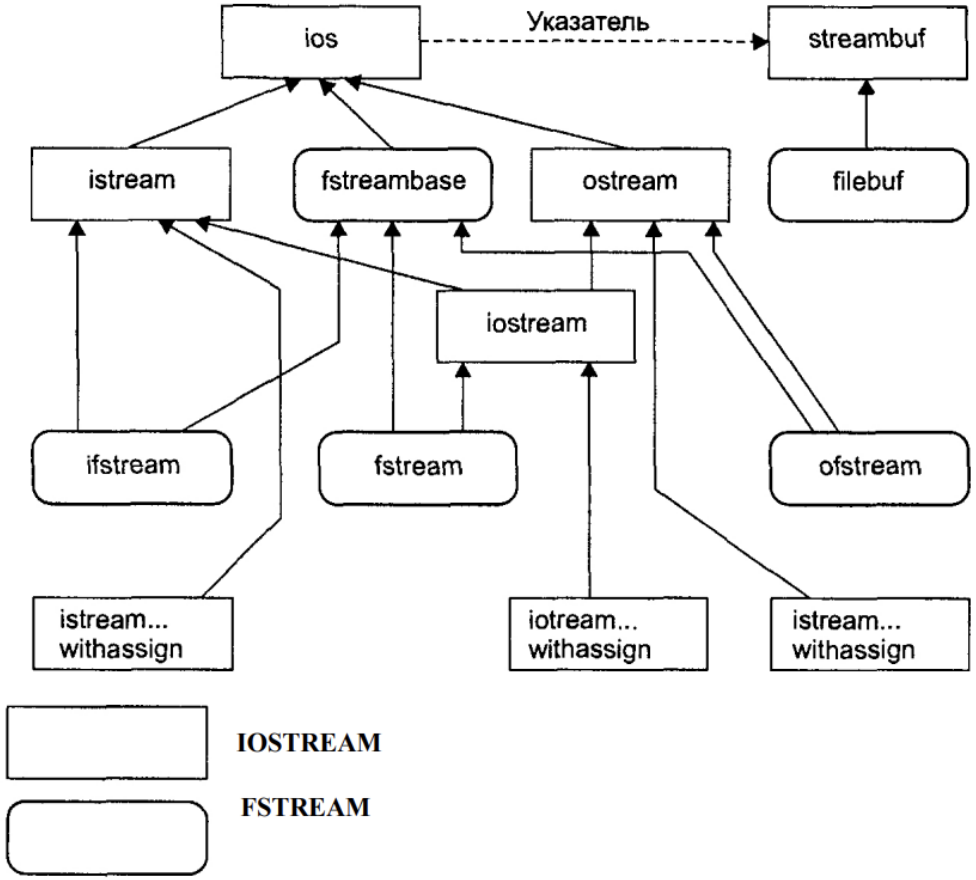
Преимущества потоков

- Программисты на C могут удивиться, какие же преимущества дает использование потоковых классов для ввода/вывода вместо традиционных функций C printf(), scanf(), для файлов — fprintf(), fscanf()...
- Одним из аргументов в пользу потоков является простота использования. Если вам приходилось когда-нибудь использовать символ управления форматом %d при форматировании вывода с помощью %f в printf(), вы оцените это. Ничего подобного в потоках вы не встретите, ибо каждый объект сам знает, как он должен выглядеть на экране. Это избавляет программиста от одного из основных источников ошибок.
- Другой причиной является то, что можно перегружать стандартные операторы и функции вставки (<>) для работы с создаваемыми классами. Это позволяет работать с собственными классами как со стандартными типами, что, опять же, делает программирование проще и избавляет от множества ошибок, не говоря уж об эстетическом удовлетворении

Иерархия потоковых классов

- Потоковые классы имеют довольно сложную иерархическую структуру. На рисунке показана организация важнейших из них. Мы уже встречались с использованием некоторых потоковых классов. Операция извлечения >> является методом класса istream, операция вставки << — методом класса ostream. Оба этих класса являются наследниками ios. Объект cout, представляющий собой стандартный выходной поток, который обычно выводит на экран данные, является предопределенным объектом класса ostream_ withassign, являющегося наследником класса ostream. Аналогично, cin — объект istream_withassign, наследника istream.

Иерархия потоковых классов



Классы, используемые для вывода данных на экран и ввода с клавиатуры, описаны в заголовочном файле IOSTREAM, который мы всегда подключали в наших примерах из предыдущих глав. Классы, используемые для ввода/вывода файлов, объявлены в файле FSTREAM

Класс istream содержит такие функции, как get(), getline(), read() и перегружаемую операцию извлечения (>>), тогда как класс ostream содержит функции put(), write() и перегружаемую операцию вставки (<<)

Класс iostream — наследник одновременно классов istream и ostream (пример множественного наследования).

Класс ios

- ios является дедушкой всех потоковых классов и обладает большинством особенностей, без которых работа с потоками была бы невозможна. Три его главных направления — это флаги форматирования, флаги ошибок и режим работы с файлами.

Классы iostream и _withassign

- Существуют три класса вида _withassign:
  - istream_withassign — наследник istream;
  - ostream_withassign — наследник ostream;
  - iostream_withassign — наследник iostream.

- cin, объект istream_withassign, используется для операций ввода с клавиатуры;
- cout, объект ostream_withassign, используется для операций вывода на экран монитор

Потоковый Ввод в файл

```
#include <fstream> // для файлового ввода/вывода
#include <iostream>
#include <string>
using namespace std;
int main()
{
 char ch = 'x';
 int j = 77;
 double d = 6.02;
 string str1 = "Kafka"; // строки без
 string str2 = "Proust"; // пробелов
 ofstream outfile("fdata.txt"); // создать объект ofstream
 outfile << ch // вставить (записать) данные
 << j
 << ' ' // пробелы между числами
 << d
 << str1
 << ' ' // пробелы между строками
 << str2;
 cout << "Файл записан\n";
 return 0;
}
```

Инициализация резервирует для дискового файла с данным именем различные ресурсы и получает доступ (или открывает файл) к нему.

Если файл не существует, он создается.

Если файл уже существует, он переписывается — новые данные в нем заменяют старые.

Объект outfile ведет себя подобно cout из предыдущих программ, поэтому можно использовать операцию вставки (<

Чтение данных

```
#include <fstream> // для файлового ввода/вывода
#include <iostream>
#include <string>
using namespace std;
int main()
{
 char ch;
 int j;
 double d;
 string str1;
 string str2;
 ifstream infile("fdata.txt"); // создать объект ifstream
 // извлечь (прочесть) из него данные
 infile >> ch >> j >> d >> str1 >> str2;
 cout << ch << endl // вывести данные
 << j << endl
 << d << endl
 << str1 << endl
 << str2 << endl;
 return 0;
}
```

Режимы открытия файла

Флаг	Назначение
`ios::in`	открыть для чтения
`ios::out`	открыть для записи (прежнее содержимое удаляется)
`ios::app`	дописывать данные в конец файла
`ios::ate`	после открытия перейти в конец файла
`ios::trunc`	удалить содержимое существующего файла
`ios::binary`	двоичный режим ввода-вывода

Флаги комбинируются операцией |, например ios::out | ios::app. Успешность открытия проверяют условием if (!file) или методом file.is_open().

Выводы

1. Поток — абстракция потока данных; ввод-вывод в C++ реализован потоковыми классами.
2. Потоки просты в использовании и допускают перегрузку операторов << и >> для собственных классов.
3. Иерархия потоковых классов построена на классе ios; iostream наследует istream и ostream (множественное наследование).
4. Для работы с файлами используются ofstream (запись) и ifstream (чтение); при открытии несуществующего файла для записи он создаётся.
5. Файл следует закрывать после работы, а успешность открытия — проверять.

Контрольные вопросы

1. Что такое поток? Какие преимущества даёт потоковый ввод-вывод?
2. Опишите иерархию потоковых классов. Какой класс является базовым?
3. Чем отличаются классы istream, ostream и iostream?
4. Как записать данные в файл? Что произойдёт, если файл уже существует?
5. Как прочитать данные из файла до его конца?
6. Какие режимы открытия файла существуют?

Список литературы

Основная литература

1. Лафоре Р. Объектно-ориентированное программирование в C++. 4-е изд. — СПб.: Питер, 2004.
2. Страуструп Б. Язык программирования C++. Специальное издание. — М.: Бином, 2011.
3. Джосаттис Н. Стандартная библиотека C++: справочное руководство. 2-е изд. — М.: Вильямс, 2014.

Дополнительная литература

4. Шилдт Г. Полный справочник по C++. 4-е изд. — М.: Вильямс, 2006.
5. Липпман С., Лажойе Ж., Му Б. Язык программирования C++. Базовый курс. 5-е изд. — М.: Вильямс, 2014.
6. Stroustrup B. A Tour of C++. 3rd ed. — Addison-Wesley, 2022.

Электронные ресурсы

7. Справочник по C++ и стандартной библиотеке: en.cppreference.com
8. Учебник по C++ (learncpp): www.learncpp.com

- Спасибо за внимание!