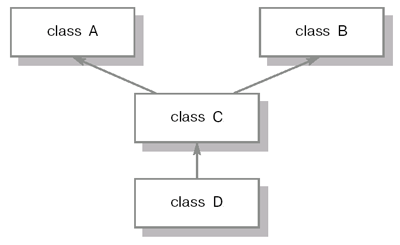
**6.3. Множественное наследование**

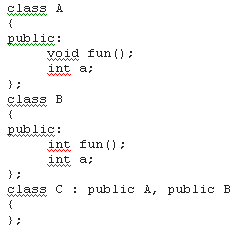
В языке С++ имеется возможность в качестве *базовых* задать несколько классов. В таком случае *производный класс* наследует методы и атрибуты всех его родителей. Пример иерархии классов в случае *множественного наследования* приведен на следующем рисунке.

В данном случае класс C наследует двум классам, A и B.

*Множественное наследование* – мощное средство языка.

Например, предположим, имеющуюся библиотечную систему решено установить в университете и интегрировать с другой системой учета преподавателей и студентов. В библиотечной системе имеются классы, описывающие читателей и работников библиотеки. В системе учета кадров существуют классы, хранящие информацию о преподавателях и студентах. Используя *множественное наследование*, можно создать классы студентов-читателей и преподавателей-читателей.

При применении *множественного наследования* возникает ряд проблем. Первая из них – возможный конфликт имен методов или атрибутов нескольких *базовых классов*.

При записи

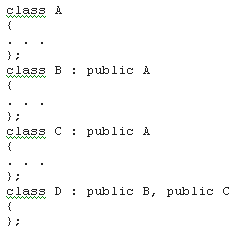
C\* cp = new C;

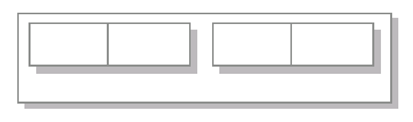
cp->fun();

невозможно определить, к какому из двух методов fun происходит обращение. Ситуация называется неоднозначной, и компилятор выдаст ошибку. Заметим, что ошибка выдается не при определении класса C, в котором заложена возможность возникновения неоднозначной ситуации, а лишь при попытке вызова метода fun.

Неоднозначность можно разрешить, явно указав, к которому из *базовых классов* происходит обращение:

cp->A::fun();

Вторая проблема заключается в возможности многократного включения *базового класса*, когда базовые классы для одного класса в свою очередь наследуют от одного и того же класса.

При таком описании классов объект класса D будет содержать объект базового класса A дважды.

A B A C

D

Кроме того, что подобная ситуация отражает нерациональное использование памяти, никаких неудобств в данном случае она не вызывает.