Variadic templates [<https://youtu.be/I5SUIRMzTgw>]

В теле объявления функции мы раскрываем пакет типов и получаем пакет аргументов:

template<typename … Types> void f(Types … args);

template<typename … Types> void g(Types … args) {

f(args …); // 🡪 f(x,y);

f(&args …); // 🡪 f(&x, &y);

f(h(args) …); // 🡪 f(h(x), h(y));

f(const\_cast<const Types \*>(&args)…);

// 🡪 f(const\_cast<const int\*>(&x), const\_cast<const double\*>(&y));

}

g(1, 1.0); // 🡪 g(int x, double y);

В теле функции g() мы раскрываем пакет аргументов. В данном случае пакет аргументов просто отправляется в функцию f(). Пакет аргументов раскрывается в соответствии со своим паттерном раскрытия. Паттерн раскрытия – это максимальный правильный препроцессинговый токен слева от троеточия. Паттерн раскрытия берется и подставляется в каждый аргумент, который раскрывается, и далее из этих аргументов делается список.

Последний пример – т.н. паттерн совместного раскрытия, он делает то, что называется «inner join». Inner join – простыми словами, – взять два вектора, взять первый элемент с первым, второй со вторым, третий с третьим и т.д. Вот он это делает, первый аргумент x, который имеет тип int будет приведен к const int\*, второй аргумент, который имеет тип double, будет приведен к const double\* и т.д.

*Замечание*: если в примере 3 скобку закрыть после троеточия, то это будет соответствовать передача раскрытого пакета далее в функцию h(), т.е. h(x, y).

*Замечание*: нельзя пользоваться пакетом аргументов, не раскрыв его!

*Пример*: раскрыть пакет аргументов arg = x,y,z:

f(h(args…) + h(args)…); // 🡪 f( h(x,y,z) + h(x), h(x,y,z) + h(y), h(x,y,z) + h(z) );

Паттерн раскрытия выделен синим цветом.

*Пример*: раскрыть пакет аргументов arg = x,y,z:

f(h(args, args…)…); // 🡪 f( h(x, x, y, z), h(y, x, y, z), h(z, x, y, z));

Раскрытие может иметь место только в некоторых контекстах:

1. Применяется в списках аргументов функции или шаблона:

myfunc(args…); // раскрывается пакет аргументов

Myclass<Args…> c1; // раскрывается пакет параметров шаблона

1. Применяется в списках инициализации и инициализаторах:

Myclass c2(args…); // раскрывается пакет аргументов конструктора

int dummy[sizeof…(Args)] = {static\_cast<int>(args)…}; // раскрывается пакет аргументов инициализатора массива с паттерном, выделенным синим цветом.

1. Применяется в списках параметров функции или шаблона.

*Замечание*: более точное определение раскрытия: паттерн раскрытия – это то, что до некоторой степени стоит в порядке c-decl.

template<typename … Args, int… N> void g(Args (&…arr)[N]){}; // ссылка на массив из N-аргументов каждого следующего типа. Здесь совместно раскрываются пакеты параметров и аргументов.

1. Списки базовых классов и списки инициализации в конструкторах.
2. Некоторые другие места (списки захвата лямбда-выражений и др).