

1. Когда удобно использовать вариативные шаблоны

Когда нужно применить функцию к неопределенному числу различных аргументов, число которых может также определяться в runtime.

2. Как можно по очереди обработать все аргументы из пакета аргументов функции

Рекурсивные алгоритмы, которые завершаются перегрузкой функции для 0 или 1 аргументов. Или также можно использовать завершение рекурсии в constexpr if

3. Как вычислить количество параметров в пакете параметров

```
template<typename T, typename ... Types>
void print(T arg, Types ... args)
{
    sizeof...(Types);
    ...
}
```

4. Какие существуют разновидности выражений свертки и когда они применяются

1. (... op args) = (...(((arg1 op arg2) op arg3) op arg4) op argN)
2. (args op ...) = arg1 op (....(arg(N-2) op (arg(N-1) op argN))...)
3. (init op ... op args) = (...((init op arg1) op arg2) op argN)
4. (args op ... op init) = (arg1 op (.... (argN op init)....))

Их все удобно применять для арифметических операций. 3 в особенности удобно для оператора ввода и вывода (<<, >>). Также, как мы разбирали на семинаре, удобно с помощью оператора ->* совершать обход дерева.

5. Динамический полиморфизм осуществляется посредством наследования, виртуальных функций. То есть создается некая иерархия классов, позволяющая компактно хранить различные данные.
- Статический полиморфизм основан на шаблонных базовых функциях. То есть для каждого класса, для каждого нового используемого типа просто инстанцируется новый шаблон.
- Статический полиморфизм позволяет увеличить скорость работы программы, так как не приходится пробегаться по таблице виртуальных функций и по указателям, а будут использоваться функции, созданные непосредственно для этого объекта. Но код при этом становится существенно больше, так как необходимо инстанцировать шаблон для каждого нового типа.