constexpr关键字同时也可以修饰函数,如果当参数都是常量时,则进行在编译期求值

```
template<typename...Args>
constexpr decltype(auto) foo(Args &&...args)
{
//    C++14起
//    1. 函数允许声明变量
//    2. 函数允许出现if和switch语句，不能使用go语句。
//    3. 函数允许所有的循环语句，包括for、while、do-while。
//    4. 函数可以修改生命周期和常量表达式相同的对象。
//    5. 函数的返回值可以声明为void。
// 也就是说在C++11时上述行为都不允许.
    int sum{};
    ((sum += args), ...);
    return sum;
}
```

Demo

```
int main()
{
    std::cout << foo(1, 2, 3) << std::endl;//6 OK
    int x = 1, y = 1, z = 1;
    std::cout << foo(x, y, z) << std::endl;//非常量OK,退化为普通函数

    constexpr int t1 = foo(1, 2, 3);//OK
    //constexpr int t2 = foo(x, y, z);//foo(x,y,z)非常量

    std::array<int, foo(1, 2, 3)> t2{};//OK
    //std::array<int,foo(x,y,z)>t3{}; 报错非常量
}
```