

## - Что такое метапрограммирование?

Вычисление выражений на стадии компиляции

## - Как связаны шаблоны классов с метапрограммированием?

Шаблоны классов используются для вычислений в compile time.

## - Какие ограничения C++ существуют для метафункций?

Могут быть вычислены только целочисленные константы и произвольные типы

## - Приведите примеры простейших метафункций?

```
template<unsigned index>
struct Fibonacci {
    static constexpr unsigned value = Fibonacci<index - 1>::value + Fibonacci<index - 2>::value;
};

template<>
struct Fibonacci<1> {
    static constexpr unsigned value = 1;
};

template<>
struct Fibonacci<0> {
    static constexpr unsigned value = 0;
};
```

## - Как используются метафункции при вычислении типа значения итератора?

С помощью них можно узнать необходимый тип. И в последствие создать экземпляр этого типа на этапе компиляции

## - Какие проблемы возникают при использовании функции `accumulate` из стандартной библиотеки и как ее можно решить с помощью метафункций?

Не всегда можно определить тип возвращаемого значения. С помощью метафункций можно узнать необходимый тип, создать его экземпляр и вернуть его.