

Семинар 1.12 «Программирование времени компиляции»

Метапрограммирование - программирование программ, шаблоны C++ - рекурсивный Тьюринг-полный (?) язык программирования. Компилятор способен проводить сложные вычисления при работе с кодом при условии, что все входные данные известны.

Доступные средства:

- застижная специализация
 - рекурсивное инстанцирование
 - SFINAE и type traits
 - constexpr и compile-time if
- } часть вычислений - на этапе компиляции

Метапрограммирование { значений / типов } → гибридное

Пример (live) вычисление факториала при компиляции

Пример (live) проверка числа на простоту (enum/static)

SFINAE (повторение)

Пример (live) длина массива или контейнера, выбор функции

Пример (live) enable_if + сравнение чисел float/integer

constexpr (константы и функции)

int x₁ = f(5); // может быть

int x₂ = f(n); // только run-time

constexpr int x₃ = f(42); // обязательно

constexpr int x₄ = f(n); // error

Пример (live) факториал числа C++ 11 (см. isprime11.hpp)

Пример (live) факториал числа C++ 14 (см. isprime14.hpp)

compile-time if

```
template < typename T, typename ... Types >
```

```
void print (const T& first, const Types &... args)
```

```
{    std::cout << first << std::endl;
```

```
    if constexpr (sizeof... (args) > 0)
```

```
    {    print (args...); // не инстанцируется, когда не надо
```

```
}
```


Примечание: выполняет только 1-ый этап трансляции

```
if constexpr (std::numeric_limits<char>::is_signed)
{
    f(42);
}
else
{
    undeclared(42); // error
    static_assert(false, "..."); // ВСЕГДА
    static_assert(<condition>, "..."); // РАБОТИТ ПО СИТУАЦИИ
}
```

ГИБРИДНОЕ МЕТАПРОГРАММИРОВАНИЕ

- сочетание действий compile-time и run-time

Пример ratio.hpp - ДРОБЬ

Пример ratio_add.hpp - сложение дробей

Пример duration.hpp - ИНТЕРВАЛ

Пример duration_add.hpp - сложение интервалов

} все вместе