Модули в C++20

Введение

- **Зачем нужны заголовки**

  Большинство C++ проектов включают несколько единиц трансляции, которые используют общие объявления и определения. Для этого применяются заголовочные файлы.

- **Минусы использования заголовков**

  Заголовки могут вызывать проблемы с избыточной компиляцией и зависимостями между модулями, а также нет строгих правил относительно использования директивы `#include`. Она просто вставляет содержимое файла в место, где она расположена.

- **Зачем модули**

  Модули это языковая функциональность, позволяющая обмениваться объявлениями и определениями между единицами трансляции. Они являются альтернативой некоторым вариантам использования заголовочных файлов.

Пример использования заголовочных файлов

Файл function.hpp:

```
const char* function();
```

Файл function.cpp:

```
const char* function() {
    return "Hello";
}
```

Файл main.cpp:

```
#include <print>

#include "function.hpp"

int main() {
    std::println("{}", function());
    return 0;
}
```

команда компиляции:

```
g++ -std=c++23 main.cpp function.cpp
```

#include нельзя контролиповать

Можно делать странные вещи и это не вызовет ошибку:

Файл return.hpp:

```
return "Hello";
```

Файл function.hpp:

```
inline const char *function() {
#include return.hpp
}
```

Файл main.cpp:

```
#include <print>

#include "function.hpp"

int main() {
    std::println("{}", function());
    return 0;
}
```

команда компиляции:

```
g++ -std=c++23 main.cpp
```

Нельзя включать файлы два раза

Файл foo.hpp:

```
struct Foo {
    int bar;
    int bazz;
};
```

Файл main.cpp:

```
#include foo.hpp
#include foo.hpp // ошибка переопределения но не повторного включения

int main() {
}
```

Не пользуясь защитой заголовков этот пример вызовет ошибку переопределения(но не повторного включения файла заголовка так как это разрешено)

Файл может исползовать не определенные в нем вещи

Файл person.hpp:

```
class Person {
    std::string name; // std::string нет в фйле ошибка конечно же не возникает
};
```

Файл main.cpp:

```
#include <string>

#include "person.hpp"

int main() {
}
```

Пример использования модулей

Файл function.cpp:

```
export module my_module;

export const char *function() {
    return "Hello";
}
```

Файл main.cpp:

```
import <print>;

import my_module;

int main() {
    std::println("{}", function());
    return 0;
}
```

команда компиляции:

```
g++ -std=c++23 -fmodules-ts -xc++-system-header print
g++ -std=c++23 -fmodules-ts function.cpp main.cpp
```