

# Модули

4 марта 2026 г.

# Декларирование модуля

Модуль состоит из одного или более module units. Модуль начинается с декларирования

```
[export] module module_name [ :partition_name ];
```

- Primary module interface unit
- Module implementation unit
- Module partition interface unit
- Internal module partition unit

# Primary module interface unit

```
export module module_name;
```

Модуль должен иметь только один основной интерфейсный блок.

Декларирование module implementation unit

```
module module_name;
```

В одном модуле можно объявить несколько module implementation units.

Декларирование module partition interface unit

```
export module module_name:partition_name;
```

partition\_name должно быть уникальным в данном модуле.

Декларирование

```
module module_name:partition_name;
```

partition\_name должно быть уникальным в данном модуле.

**Built Module Interface** (или BMI) — это результат предварительной компиляции импортируемого module unit.

**Global Module Fragment (GMF)** — это код, расположенный между `module`; и объявлением модуля внутри `module units`.

```
module;  
#include <iostream>  
export module Hello;  
export void hello() {  
    std::cout << "Hello World!\n";  
}
```

```
import Hello;
```

```
int main()  
{  
    hello();  
}
```

```
clang++ -std=c++20 hello.cppm --precompile \  
-o hello.pcm  
clang++ -std=c++20 use.cpp -fprebuilt-module-path=. \  
-o hello.out
```

# Как включить C++ модули

Стандартные модули C++ включаются автоматически, если режим стандарта языка установлен на `-std=c++20` или более позднюю версию.

Для генерации BMI импортируемого module unit используйте один из параметров командной строки:

- `--precompile`,
- `--precompile--reduced--bmi`
- или `--fmodule--output`.

Опция `—precompile` генерирует ВМІ в качестве результата компиляции, при этом путь к выходным файлам указывается с помощью опции `-o`.

Опция `—fmodule—output` генерирует ВМІ в качестве побочного продукта компиляции.

Если указан параметр `—fmodule—output=`, ВМІ будут отправлены в указанное место.

Если указаны параметры `—fmodule—output` и `—с`, ВМІ будет выведен в каталог выходного файла с именем входного файла и расширением `.pcm`.

# Требования к именам файлов

По общепринятой практике, файлы модулей, предназначенные для импорта, должны иметь расширение `.hpp` (или `.hst`, `.sxxh` или `.c++h`). Файлы модулей реализации должны иметь расширение `.cpp` (или `.cc`, `.cxx` или `.c++`).

# Требования к именам файлов

В качестве расширения файла VMI следует использовать `.pvm`. Имя файла VMI для основного модуля интерфейса должно быть `module_name.pvm`. Имя файла VMI для модуля раздела должно быть `module_name-partition_name.pvm`.

# Specifying BMI dependencies

- `-fprebuilt-module-path=<path/to/directory>`
- `-fmodule-file=<module-name>=<path/to/BMI>`

Параметр `-fprebuilt-module-path` указывает путь, по которому следует искать зависимости BMI. Можно указать несколько путей. При импорте модуля `M` компилятор ищет файл `M.pcm` в каталогах, указанных параметром `-fprebuilt-module-path`. Аналогичным образом, при импорте модуля раздела `M:P` компилятор ищет файл `M-P.pcm` в каталогах, указанных параметром `-fprebuilt-module-path`.

При компиляции module implementation unit необходимо указать BMI соответствующего primary module interface unit, поскольку module implementation unit неявно импортирует primary module interface unit.

```
sudo apt install clang-20  
sudo apt install libc++-20-dev
```

```
clang++-20 -std=c++23 -stdlib=libc++ \  
-Wno-reserved-module-identifier \  
-precompile -o std.pcm \  
/usr/lib/llvm-20/share/libc++/v1/std.cppm
```