

Прости C++,

но Наш Пакетный Менеджер в
Другом Замке

Тимофеев Александр

alexandr.p.timofeev@gmail.com

@Hipony - CppLang Slack / IncludeCPP Discord

Долгая История

- Создан в 1983 году (CFront)
- В 1985 было выпущено первое издание книги Язык Программирования C++
- В 1998 была выпущена первая ревизия стандарта: C++98
- Всего существует пять ревизий стандарта (C++98, C++03, C++11, C++14, C++17)
- Скоро будет выпущен C++20 (По вкладу сравним с C++11)
- Размер текущего драфта 1803 страницы

Hello, World!

```
#include <iostream>

int main() {
    std::cout << "Hello, World!\n";
}
```

```
Hello, World!
```

Немного сложнее

```
-0.123456 -0.987000  
0.123450 0.987654
```

"Chevron Hell"

```
#include <iostream>
#include <iomanip>

int main() {
    std::cout << std::fixed << std::setprecision(6) << std::setfill(' ');
    std::cout << std::setw(9) << -0.123456 << ' ';
    std::cout << std::setw(9) << -0.987 << '\n';
    std::cout << std::setw(9) << 0.12345 << ' ';
    std::cout << std::setw(9) << 0.987654321 << '\n';
}
```

printf?

<https://github.com/fmtlib/fmt>

{fmt}

```
#include <fmt/core.h>

int main()
{
    fmt::print(
        "{: 9.6f} {: 9.6f}\n"
        "{: 9.6f} {: 9.6f}\n",
        -0.123456, -0.987,
        0.12345, 0.987654321
    );
}
```

```
-0.123456 -0.987000
0.123450 0.987654
```

VS

```
#include <fmt/core.h>

int main()
{
    fmt::print(
        "{: 9.6f} {: 9.6f}\n"
        "{: 9.6f} {: 9.6f}\n",
        -0.123456, -0.987,
        0.12345, 0.987654321
    );
}
```

```
#include <iostream>
#include <iomanip>

int main() {
    std::cout << std::fixed << std::setprecision(6) << std::setfill(' ');
    std::cout << std::setw(9) << -0.123456 << ' ';
    std::cout << std::setw(9) << -0.987 << '\n';
    std::cout << std::setw(9) << 0.12345 << ' ';
    std::cout << std::setw(9) << 0.987654321 << '\n';
}
```







App + Fmt?

Прости C++,
но Наш Пакетный Менеджер в
Другом Замке



Варианты

- Сделать частью сборки

Сделать Частью Сборки

Скопировать

- Версии
- Транзитивные зависимости
- Разрешение зависимостей
- Другие системы сборки

Сделать Частью Сборки

Использовать Систему Сборки

- Версии
- Транзитивные зависимости
- Разрешение зависимостей
- Другие системы сборки!

Сделать Частью Сборки

Не Всегда Удобно



Варианты

- Сделать частью сборки
- **Системный пакетный менеджер**

Системный Пакетный Менеджер

Плюсы

- Удобно, уже есть в системе!
- Есть у всех?
- Стабильный, значит надежный

Минусы

- Какой есть, тем и пользуешься
- У всех разный
- Иногда слишком стабильный

Системный Пакетный Менеджер

Не Всегда Подходит для Разработки



Варианты

- Сделать частью сборки
- Системный пакетный менеджер
- **Решения из других языков?**

Инструменты из Других Языков

Gradle+Maven? Pip? NuGet? ...?

Инструменты из Других Языков

Поверхностная Поддержка C++



Варианты

- Сделать частью сборки
- Системный пакетный менеджер
- Решения из других языков?
- **Пакетные менеджеры C++**

Пакетный Менеджер C++



Пакетный Менеджер C++

Проблемы

- **Время компиляции**

THE #1 PROGRAMMER EXCUSE
FOR LEGITIMATELY SLACKING OFF:

"MY CODE'S COMPILING."

HEY! GET BACK
TO WORK!

COMPILING!

OH. CARRY ON.



Время Компиляции

```
#include <iostream>

int main() {
    std::cout << "Hello, World!\n";
}
```

Время Компиляции

main.cpp

Время Компиляции



```
#include <...>
```

```
#include <iostream>
```

source.cpp

```
#include <...>
```

```
#include <iostream>
```

main.cpp

```
#include <...>
```

```
#include <iostream>
```

source.cpp



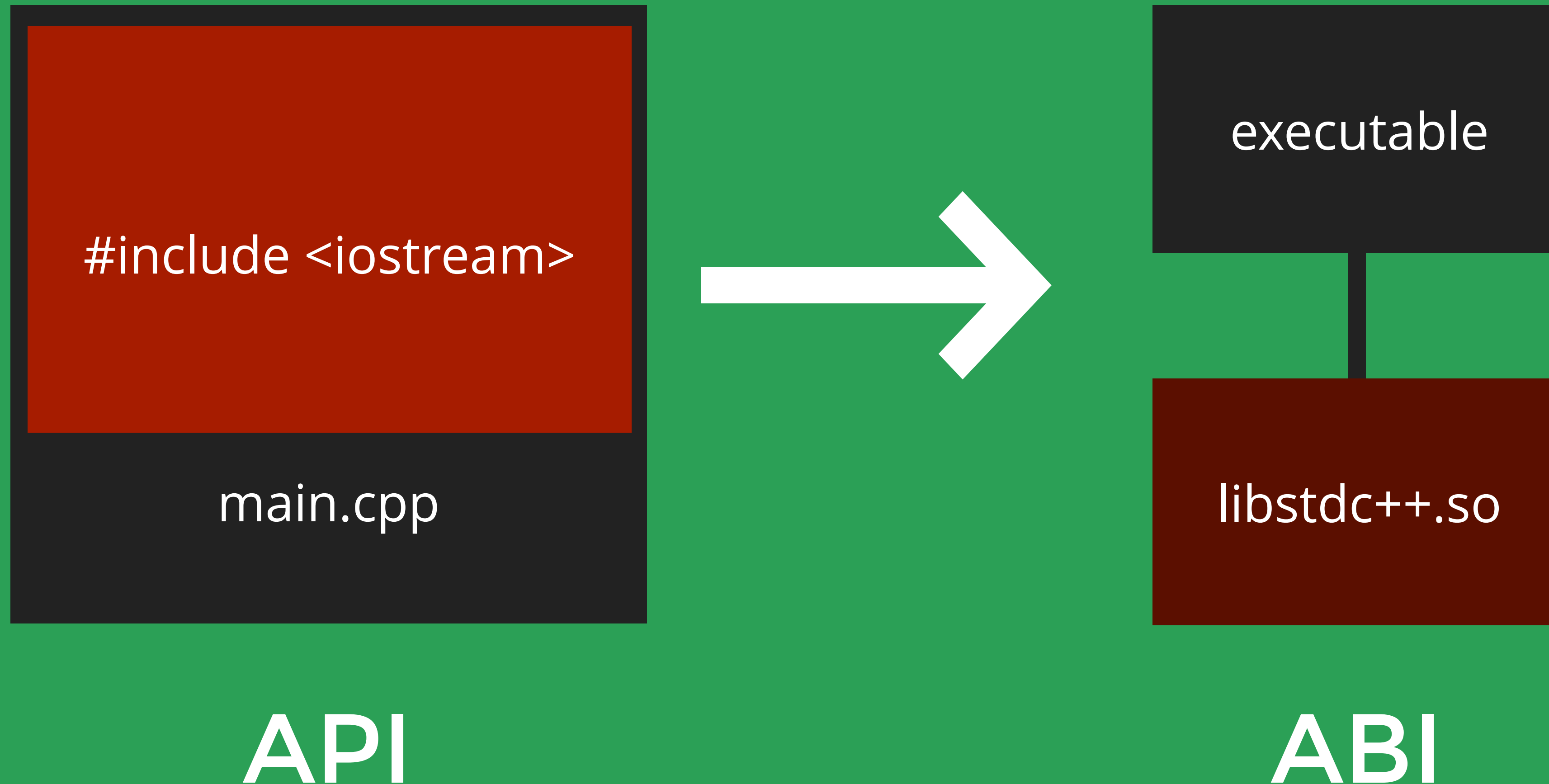
Пакетный Менеджер C++

Проблемы

- Время компиляции
- **Application Binary Interface (ABI)**

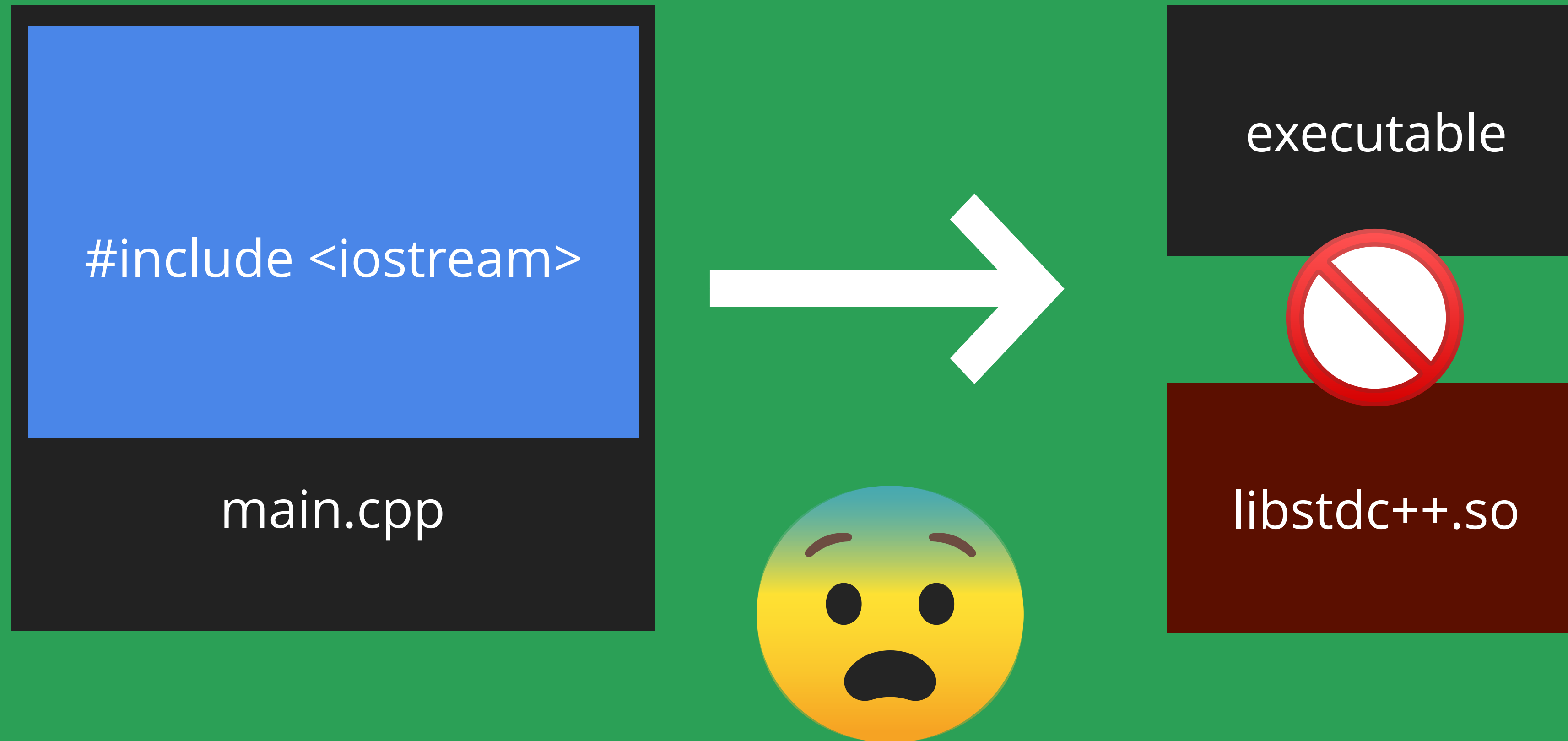
Application Binary Interface

Бинарный Интерфейс Приложения



Application Binary Interface

Несовместимость



Application Binary Interface

ABI библиотеки

ABI компилятора

- Компилятор
- Флаги*

+

API библиотеки

#include

Application Binary Interface

ABI компилятора

Стандарт Не Описывает
ABI!

Application Binary Interface

API библиотеки



Application Binary Interface

API библиотеки - SemVer

1.1.0

1.1.1

1.2.0

2.0.0

API

#include 1.1

#include 1.1

#include 1.2

#include 2.0

ABI

Binary 1.1

Binary 1.1

Binary 1.2

Binary 2.0

Application Binary Interface

Внешний Фактор

Препроцессор

-DOPTION, #define OPTION

#ifdef OPTION

```
#include 3.0
```



```
#include 3.0
```

Компиляция || Кэширование

Пакетный Менеджер C++

Проблемы

- Время компиляции
- Application Binary Interface (ABI)
- **One Definition Rule (ODR)**

One Definition Rule

Правило Одного Определения

- **Объявление** - говорит что сущность с определенным именем существует.
- **Определение** - объявление, полностью описывающее сущность.

One Definition Rule

Правило Одного Определения

Каждая программа должна содержать одно и точно одно определение каждой "используемой" сущности не объявленной `inline`, не считая отброшенных выражений (statements). Не требует диагностики.



One Definition Rule

Пример

```
#include <ivan_smirnov.hpp>

#include <iostream>

int main() {
    std::cout << ivan::favorite_number();
}
```

One Definition Rule

Пример

```
// ivan_smirnov.hpp
#pragma once

namespace ivan {

int favorite_number();

}

// ivan_smirnov.cpp
#include <ivan_smirnov.hpp>

int ivan::favorite_number() {
    return 15;
}
```

15

One Definition Rule

Пример

```
// ivan_ivanov.hpp
#pragma once

namespace ivan {

int favorite_number();

}

// ivan_ivanov.cpp
#include <ivan_ivanov.hpp>

int ivan::favorite_number() {
    return 27;
}
```

27

One Definition Rule

Пример

```
// ivan_smirnov.hpp
#pragma once

namespace ivan {

int favorite_number();

}

// ivan_smirnov.cpp
#include <ivan_smirnov.hpp>

int ivan::favorite_number() {
    return 15;
}
```

```
#include <ivan_smirnov.hpp>

#include <iostream>

int main() {
    std::cout << ivan::favorite_number();
}
```

???

```
// ivan_ivanov.hpp
#pragma once

namespace ivan {

int favorite_number();

}

// ivan_ivanov.cpp
#include <ivan_ivanov.hpp>

int ivan::favorite_number() {
    return 27;
}
```

One Definition Rule

Пример

```
#include <ivan_smirnov.hpp>

#include <iostream>

int main() {
    std::cout << ivan::favorite_number();
}
```

One Definition Rule

Пример

```
namespace ivan {  
  
int favorite_number();  
  
}  
  
#include <iostream>  
  
int main() {  
    std::cout << ivan::favorite_number();  
}
```

One Definition Rule

Затрагивает Все Зависимости!

```
// ivan_smirnov.hpp
#pragma once

namespace ivan {

int favorite_number();

}

// ivan_smirnov.cpp
#include <ivan_smirnov.hpp>

int ivan::favorite_number() {
    return 15;
}
```

```
#include <ivan_smirnov.hpp>

#include <iostream>

int main() {
    std::cout << ivan::favorite_number();
}
```

```
// ivan_ivanov.hpp
#pragma once

namespace ivan {

int favorite_number();

}

// ivan_ivanov.cpp
#include <ivan_ivanov.hpp>

int ivan::favorite_number() {
    return 27;
}
```

One Definition Rule

Затрагивает Все Зависимости!

```
#include <ivan_smirnov.hpp>
#include <iostream>

int main() {
    std::cout << ivan::favorite_number();
}
```



```
// ivan_smirnov.hpp
#pragma once

namespace ivan {

int favorite_number();

}

// ivan_smirnov.cpp
#include <ivan_smirnov.hpp>

int ivan::favorite_number() {
    return 15;
}
```

```
// ivan_ivanov.hpp
#pragma once

namespace ivan {

int favorite_number();

}

// ivan_ivanov.cpp
#include <ivan_ivanov.hpp>

int ivan::favorite_number() {
    return 27;
}
```

Пакетный Менеджер C++

Проблемы

- Время компиляции
- Application Binary Interface (ABI)
- One Definition Rule (ODR)
- Существующие проекты

Существующие Проекты

- Огромное количество
- Очень востребованы
- Уже написаны!

Пакетный Менеджер C++

Требования

- Многоплатформенность
- Поддержка систем сборки
- Простое внедрение

- Vcpkg

- Conan

- ????

Vcpkg

- Microsoft
- Git/CMake/C++
- Просто начать
- Гарантирует совместимость всех зависимостей

Conan

- JFrog
- Python
- Большая Гибкость
- Авторы владеют релизами
- Интегрирован с **Artifactory**

Используйте Пакетные Менеджеры C++



Варианты

- Сделать частью сборки
- Системный пакетный менеджер
- Решения из других языков?
- Пакетные менеджеры C++
- **Добавить библиотеку в стандарт!**

Добавляем Библиотеку в Стандарт

*" ... Множество пользователей C++ не могут позволить себе добавить даже такие библиотеки как **Boost** в свои проекты. Для них использование публичных репозиториев запрещено, из-за юридических вопросов либо проблем безопасности. Добавление новых библиотек в C++ открывает их для большего числа пользователей, делая C++ более доступным для большего числа людей. "*

Стандартная Библиотека

- `<regex>`
- `<random>`
- `<locale>`
- `<iostream>`
- `<charconv>`

std::format

```
#include <fmt/core.h>

int main()
{
    fmt::print(
        "{: 9.6f} {: 9.6f}\n"
        "{: 9.6f} {: 9.6f}\n",
        -0.123456, -0.987,
        0.12345, 0.987654321
    );
}
```

std::format

```
#include <format>
#include <iostream>

int main()
{
    std::cout << std::format(
        "{: 9.6f} {: 9.6f}\n"
        "{: 9.6f} {: 9.6f}\n",
        -0.123456, -0.987,
        0.12345, 0.987654321
    );
}
```

std::format



2016 - 2020 - ...

Стандартная Библиотека

Нужно Больше Количество Пользователей
ДО стандартизации!



Варианты

- Сделать частью сборки
- Системный Пакетный Менеджер
- Решения из Других Языков?
- Пакетные Менеджеры C++
- Добавить Библиотеку в Стандарт!
- **Добавить Пакетный Менеджер в Стандарт?**

Стандартный Пакетный Менеджер



Стандартный Пакетный Менеджер

If you could wave a magic wand and change one thing about any part of C++, what would it be?

C++ Foundation Developer Survey 2018

Стандартный Пакетный Менеджер

Если бы вы могли махнуть волшебной палочкой и поменять одну любую часть C++, что бы вы выбрали?

C++ Foundation Developer Survey 2018

Reduce Unit Test Package Manager Msvc Syntax Faster Build Default

Better Support Language Pointer Library Rust Code Implicit

Compile Header Files Standard Metaclasses Template

Drop Remove Size Std Move Semantics Compatibility ABI

Стандартный Пакетный Менеджер

Проблемы

- Много Противоречащих Требований
- Стоимость Разработки
- Стоимость Ошибки
- Инерция
- Эксперты
- Что должно быть в стандарте C++?

Стандартный Пакетный Менеджер

Не реалистично



SG15 - Tooling

<https://wg21.link/p1767r0>

<https://wg21.link/p1689r0>

<https://wg21.link/p1688r0>

...

Выводы

- Управлять зависимостями в C++ - Непросто
- Универсального варианта - Нет
- Исправить ситуацию - Сложно
- Путь - Использовать пакетные менеджеры C++
- Лучше - Давать Отзывы
- Идеально - Участвовать в проектах!

Прости C++,

но Наш Пакетный Менеджер в
Другом Замке

Тимофеев Александр

alexandr.p.timofeev@gmail.com

@Hipony - CppLang Slack/IncludeCPP Discord