



**KEEP
CALM
AND
LOVE
C++**

Why C++?

- Why not?
- Это модно
- Избавляет от прокрастинации



Why C++? [olymp]

- Pascal устарел
- Скорость. C++ - скорость
- Есть реализация базовых структур данных

Why C++? [production]

- Занимает 4 место согласно индексу [TIOBE](#)
- После C++ легче осваиваются C# / Java

Картинка для энтузиастов

Дни 1 - 10
Изучи переменные, константы, массивы, строки, выражения, операторы, функции...



Дни 11 - 21
Изучи структуру программы, указатели, ссылки, классы, объекты, наследование, полиморфизм...



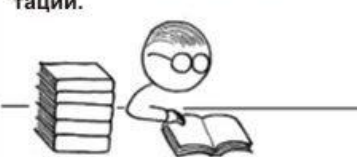
Дни 22 - 697
Развлекайся программированием всяких забавных штук, лови кайф от наступания на грабли.



Дни 698 - 3648
Взаимодействуй с другими программистами, работай над проектами вместе, учись от них.



Дни 3649 - 7781
Изучи теоретическую физику и разработай последовательную теорию квантовой гравитации.



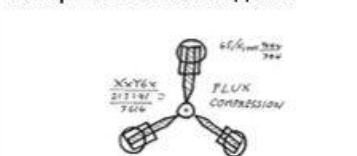
Дни 7782 - 14611
Изучи биохимию, молекулярную биологию, генетику...



День 14611
Используй знание биологии для создания лекарства омоложения.



День 14611
Используй знание физики для построения машины времени и вернись на 21-ый день.



День 21
Замени себя более молодым.



Насколько мне известно, это самый простой способ «Изучить C++ за 21 день».

Структура программы

```
// include section  
#include <iostream>  
  
// main body  
  
int main()  
{  
    std::cout << "Be cool while code";  
  
    return 0;  
}
```



#include directive

- Синтаксис: *#include <file_name>*
- Заменяется на код указанного файла при компиляции
- Возникнет ошибка компиляции, если файл не найден
- stdafx.h – зло, с точки зрения олимпиад

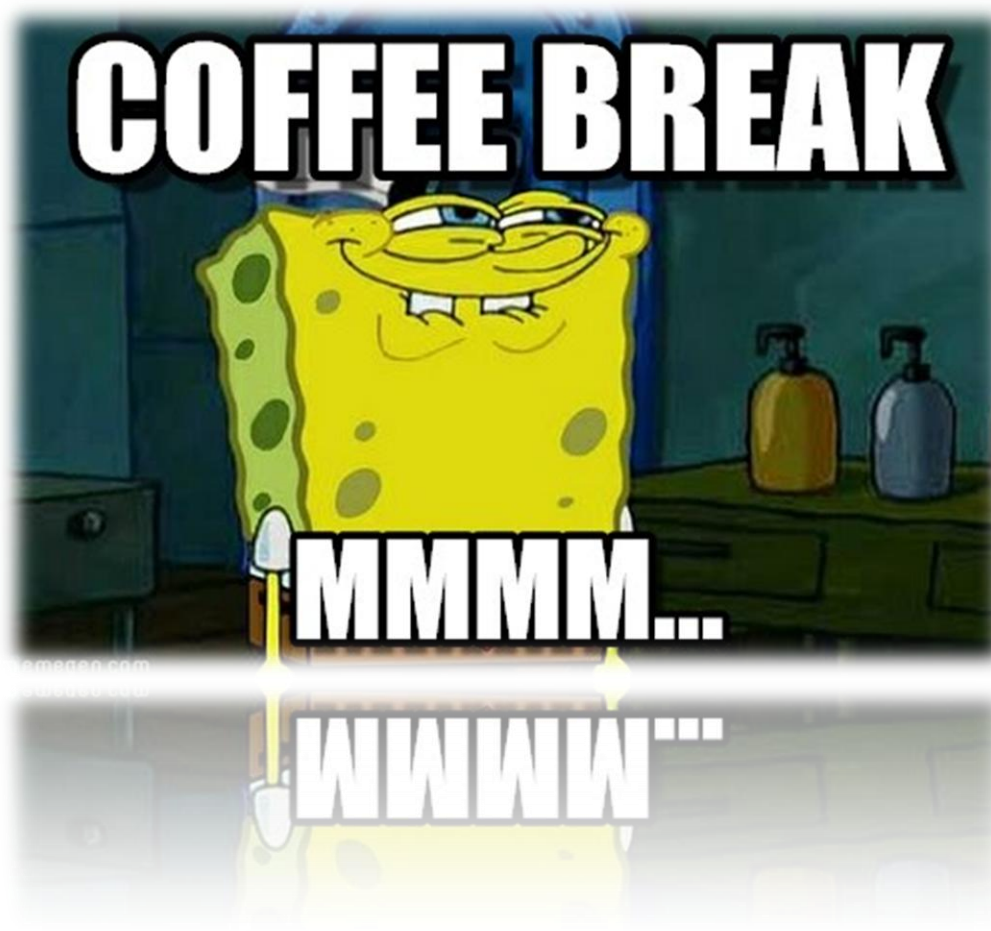
#include <iostream>

- Такие кошерные вещи, как:
 - std::cin
 - std::cout
 - std::endl
- Пожалуй, все

main body

- Сигнатура: `int main()`
- `return 0` не обязателен, но дает +2 в карму

Coffee break



std::cin / std::cout [usage example]

```
int numerator;  
int denominator;  
  
std::cin >> numerator >> denominator;  
  
std::cout << "Div: " << (numerator / denominator) << ", "  
          << "mod: " << (numerator % denominator) << std::endl;
```

std::cin / std::cout

- Настолько умны, что сами определяют тип переменной (в отличие от C)
- Крайне неудобный синтаксис
- Крайне скудные опции форматирования

Variable declaration

- Можно объявить в любом месте
- Пожалуй, самое значимое отличие от Pascal



if statement

```
if(парень_в_горах_не_ах)
{
    if(сразу_раскис                && вниз &&
       шаг_ступил_на_ледник && сник &&
       оступился                && в_крик)
    {
        // рядом с тобой - чужой

        finish_him();
    }
}
```

for loop statement

```
for(int i = 10; i > 1; i--)  
{  
    std::cout << "I have " << i << " lollypops" << std::endl;  
    std::cout << "Eating one of them..." << std::endl;  
}  
  
std::cout << "I have only one lollypop :(" << std::endl;
```

Problems

- [Timus #2012](#) (Про Гришу Н.)
- [Timus #1876](#) (Утро сороконожки)
- [Timus #1873](#) (Летопись GOV)

Useful links

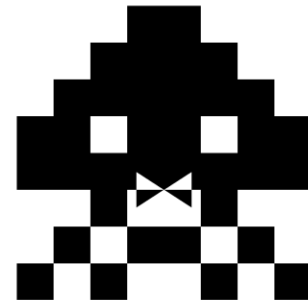
- cppreference.com [справочник по языку]
- cyberforum.ru [студенческие задачи]
- acm.timus.ru [олимпиадные задачи]
- codeforces.com [олимпиадные задачи]



NATIONAL RESEARCH
UNIVERSITY

About me

- Author: Andrew Kuchev
- e-mail: kuchevad@gmail.com
- Special thanks to NRU HSE Perm



see you
later,
alligator.

