

Знакомство с C++

Общие понятия, структура программы, Hello World

Михаил Марков
C++ - Разработчик



Проверка связи



Поставьте “+”, если меня видно и слышно



Если у вас нет звука:

- убедитесь, что на вашем устройстве и на колонках включен звук
- обновите страницу вебинара (или закройте страницу и заново присоединитесь к вебинару)
- откройте вебинар в другом браузере
- перезагрузите компьютер (ноутбук) и заново попытайтесь зайти

Михаил Марков

О спикере:

C++-разработчик, фрилансер

- Разработка алгоритма для релевантной выдачи объявлений.
- Разработка эмуляторов оборудования



Знакомы ли вы с программированием?

Напишите, пожалуйста, в чат:

1

если ничего
пока не знаете

5

если уже немного
разбираетесь в
теме

8

если вы уже в
работаете IT

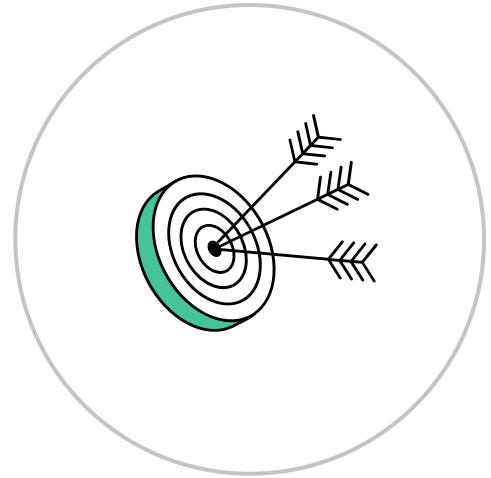
Какие у вас цели?

сменить профессию / хобби /
повышение на работе и тд

Напишите в чат

Цели занятия

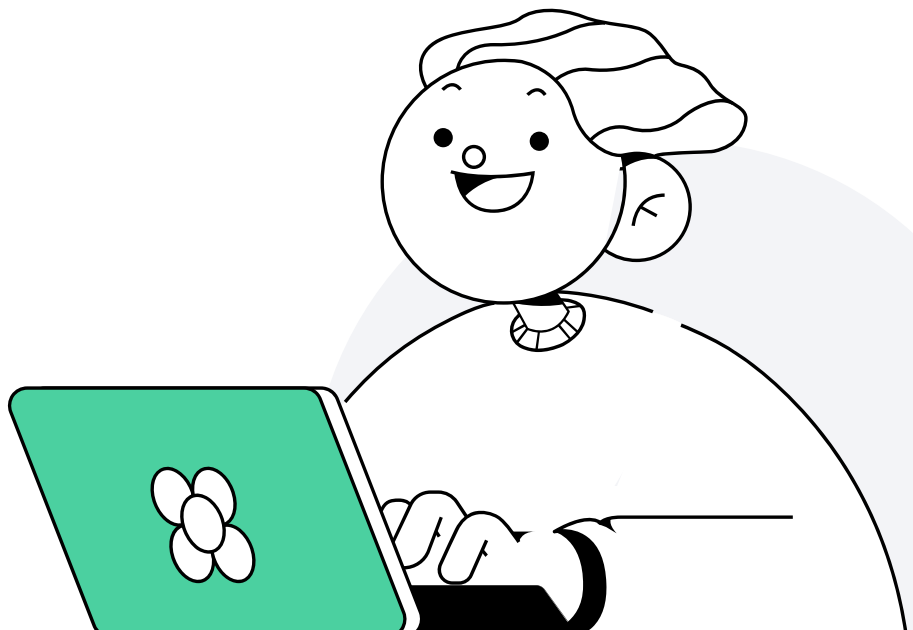
- Узнаем, зачем нужно программирование и что такое программы
- Немного поговорим об устройстве компьютера
- Выясним, что такое машинный код, ассемблер и ЯВУ
- Познакомимся с языком C++
- Напишем первую программу на C++



План занятия

- 1 Зачем нужны программы?
- 2 Как устроен компьютер?
- 3 Машинный код и ассемблер
- 4 Высокоуровневые языки программирования
- 5 Языки C/C++ и история их возникновения
- 6 Что нужно для написания программы
- 7 Первая программа на C++ в Repl.it
- 8 Итоги
- 9 Домашнее задание

*Нажми на нужный раздел для перехода



Зачем нужны программы?

Напишите в чат



1

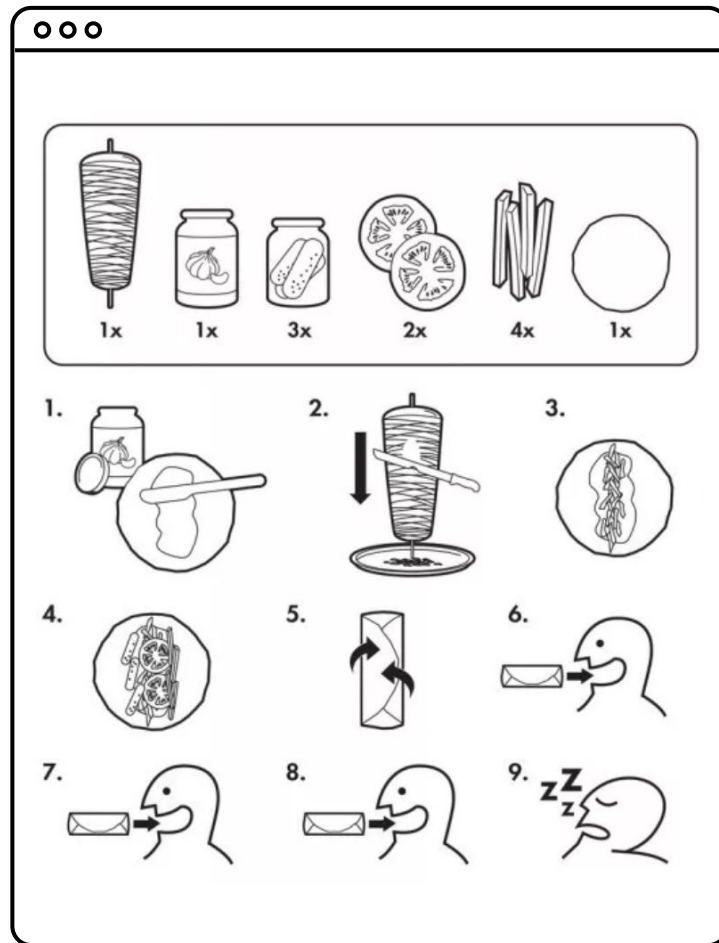
Что такое программа?

Программа — это один из способов автоматизации деятельности человека



Зачем нужны программы?

Программа — это набор инструкций, выполняемый компьютером, для достижения определенной цели.



Как устроен компьютер?



2

Упрощённое устройство компьютера

Процессор (ядро)	Процессор — умеет исполнять команды. Исполнение команд \approx возможности выполнять программы. Современные процессоры многоядерны: каждое ядро может исполнять команды. Данные берутся либо из регистров (быстрая память на самом процессоре), либо из ОЗУ
Память (ОЗУ)	Быстрая память для хранения оперативных данных. Чтобы данные попали в процессор, нужно сначала, чтобы они попали в ОЗУ. Очищается после выключения питания
Хранилища данных (HDD/SSD)	Долговременная память для хранения данных даже после выключения питания (на самом деле — нет, всё зависит от срока без питания и воздействия внешних ЭМ полей)
Сеть (сетевая карта)	Возможность обмена информацией (передачи байт) за пределы устройства

[Как работает процессор и языки программирования](#)

Машинный код и ассемблер



3

Машинный код

Это последовательность команд и их операндов, выполняемая процессором.

Процессор умеет исполнять команды, закодированные в специальном виде: **машинном коде**.

Это набор байт, который умеет декодировать процессор и выполнять операции:

- загрузка из ОЗУ данных в процессор
- запись данных из процессора в ОЗУ
- сложение, вычитание и т.д.
- другие операции

```
ooo

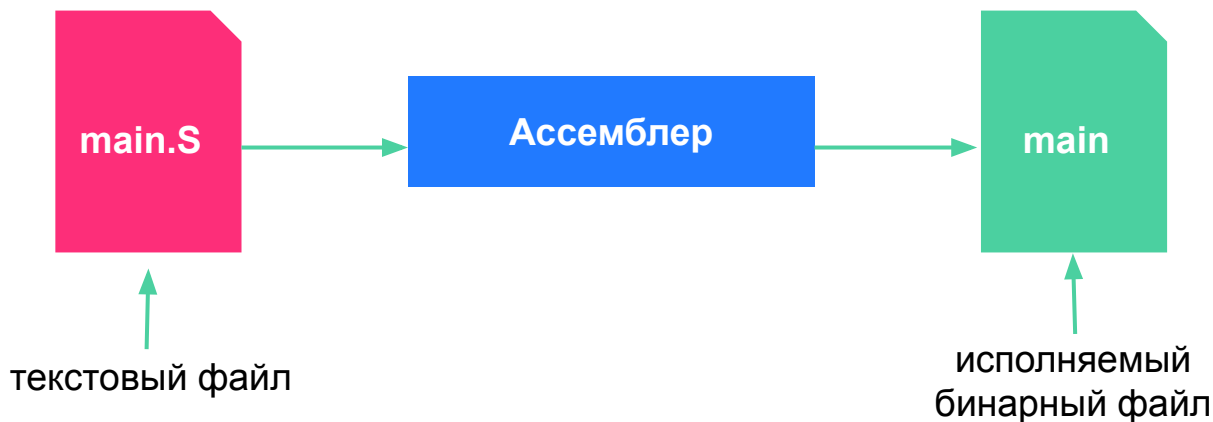
48 c7 c0 01 00 00 00
48 c7 c7 01 00 00 00
48 c7 c6 00 20 40 00
48 c7 c2 0d 00 00 00
0f 05
48 c7 c0 3c 00 00 00
48 c7 c7 00 00 00 00
0f 05
```

Что такое ассемблер

Кодировать «вручную» биты и байты достаточно сложно (мозг тяжело работает с числами)

Поэтому придумали специальные мнемоники, которые позволяют записывать программу в виде текста и затем «переводить» её в машинный код с помощью специальной программы — **ассемблера**

А соответствующие языки называют языками ассемблера



Так выглядит работа ассемблера

```
ooo
48 c7 c0 01 00 00 00
48 c7 c7 01 00 00 00
48 c7 c6 00 20 40 00
48 c7 c2 0d 00 00 00
0f 05
48 c7 c0 3c 00 00 00
48 c7 c7 00 00 00 00
0f 05
```



```
ooo
mov    $0x1,%rax
mov    $0x1,%rdi
mov    $0x402000,%rsi
mov    $0xd,%rdx
syscall
mov    $0x3c,%rax
mov    $0x0,%rdi
syscall
```


Высокоуровневые языки программирования



4

Языки высокого уровня (ЯВУ)

Писать на ассемблере большие программы достаточно проблематично

Поэтому появились высокоуровневые языки, позволяющие не думать о регистрах процессора и системных вызовах

Пример на C++:

```
int i1, i2;  
bool res;  
i1 = 1 + 2 + 3;
```

В этом нам помогает компилятор —————→

Компилятор

Компиляция (compile) – процесс перевода исходного текста программы, написанной программистом, в машинный код. Или, другими словами, – создание из программы исполняемого файла

Для того, чтобы компилятор правильно и однозначно понимал то, что пишет программист, в тексте программы используются специальные выражения:

Например:

- зарезервированные слова `for`, `if`, `switch`, `int` и т.д.
- операции `+`, `-`, `*`, `/`, `<<`, `>>`, и т.д.
- операторы `sizeof`, `<<`, `>>`, и т.д.
- модификаторы `unsigned`, и т.д.

и множество других, все их мы разберём позже

Языки C/C++ и история их возникновения



5

Языки C/C++

Ключевые идеи:

- Максимальная производительность и минимальная ресурсоёмкость при достаточно высоком уровне абстракции
- Возможность низкоуровневого доступа и выполнения потенциально небезопасных операций (за счёт практически полной свободы действий разработчику)

Откуда такое название – С ?

Факты:

- С - третья буква латинского алфавита
- Принято произносить [si:] / [си], в соответствии с английской транскрипцией
- Постфикс ++ есть ничто иное, как отсылка к популярному оператору языка, указывающему на приращение к базовому функционалу С++ относительно С

Немного истории

- В 1966 году в Кембриджском университете Мартин Ричардсом был предложен язык программирования BCPL (Basic Combined Programming Language)
- В 1969 году на его основе австралийской компанией AT&T Bell Telephone Laboratories создаётся упрощённая версия языка, получившая название **B**, его авторами принято считать Кена Томпсона при содействии Денниса Ритчи
- В 1969-1973 годах Деннис Ритчи продолжил работу над усовершенствованием языка B, своё новое творение он назвал **C**

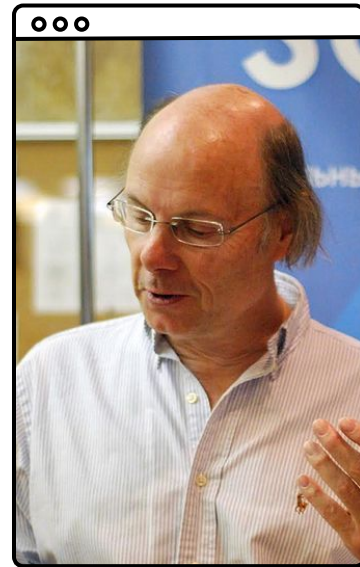
Какие плюсы у C++ ?

Нет предела совершенству

- В начале 1980-х годов талантливый сотрудник Bell Laboratories **Бьёрн Страуструп** придумал ряд усовершенствований к языку C под собственные нужды
- Уже в 1983 году им был предложен новый расширенный стандарт языка, получивший впоследствии название **C++**, хотя поначалу он сам скромно называл его языком “C с классами”
- С тех пор язык претерпел множество трансформаций и расширений, в нём появились принципиально иные возможности и современные средства разработки
- В последнее время практически ежегодно обновляется стандарт языка C++, что говорит о его актуальности и востребованности

И что же дальше?

- В 2006 году появился язык программирования D, он является потомком языка C++, но не имеет с ним обратной совместимости, из-за чего не завоевал популярности



Области применения языков C/C++

- 1 Бизнес-приложения (например, Интернет-Банкинг)
- 2 Сторонние библиотеки для платформы (чужой код)
- 3 «Платформа»: JVM/.Net/PHP/Python
- 4 Системные библиотеки
- 5 C/C++ Library & System Calls
- 6 Ядро ОС
- 7 «Железо», микропрограммы

C/C++, ассемблерные вставки
или ассемблер

Что нужно для написания программы?



6

Что нужно для написания программ на C++ ?

Для написания программы на любом языке (в том числе C++) требуется следующий минимальный набор инструментов:

- **текстовый редактор**

для ввода и редактирования ключевых слов, операторов и иных конструкций языка

- **компилятор**

для перевода программы с языка C++ в машинный код

Что нужно для написания программ на C++ ?

Для дальнейшей отладки и сопровождения написанной программы могут также понадобиться:

- **отладчик**
для пошагового выполнения программы и выявления ошибочного поведения алгоритма
- **система управления версиями**
обеспечивает контроль за вносимыми в код правками, особенно важно при работе команды над большим проектом

Что такое среда разработки ?

С целью сделать процесс написания, отладки и сопровождения программ максимально удобным и эффективным, существует такое понятие - **среда разработки**

Или **IDE** - Integrated Development Environment

Среда разработки

Как правило, это программный (иногда программно-аппаратный) комплекс, включающий в себя:

- адаптированный под написание кода редактор со встроенной системой контекстных подсказок и подсветкой синтаксиса
- компилятор, который запускается нажатием одной кнопки
- интегрированный с редактором отладчик, позволяющий проследить ход работы алгоритма
- клиент системы контроля версий, прозрачно для разработчика загружающий из репозитория актуальную ревизию кода
- профайлер - инструмент, позволяющий оценить и оптимизировать ресурсозатраты: машинное время, используемую оперативную память, сетевой и дисковый трафик

Выбор среды разработки

Их существует огромное множество. По статистике, самыми популярными являются:

- **Microsoft Visual Studio**

платный, доступен только для Windows

- **Visual Studio Code**

бесплатный, доступен для Windows, MacOS, Linux, не является полноценной средой - это редактор с элементами IDE

- **Qt Creator**

кроссплатформенный, есть бесплатная версия, снискала популярность из-за качественной и подробной документации к библиотекам

- **Xcode**

полноценная IDE, созданная компанией Apple для написания ПО под macOS, iOS, WatchOS и tvOS

Выбор среды разработки

Однако, на начальном этапе можно обойтись и без установки какой либо IDE

Для этого существует масса бесплатных Web-ресурсов, позволяющих писать небольшие программы. Их иногда называют “песочницы”.

На курсе мы будем работать в repl.it

Первая программа на C++ в Repl.it



7

Структура программы

```
int main() {  
  
}
```

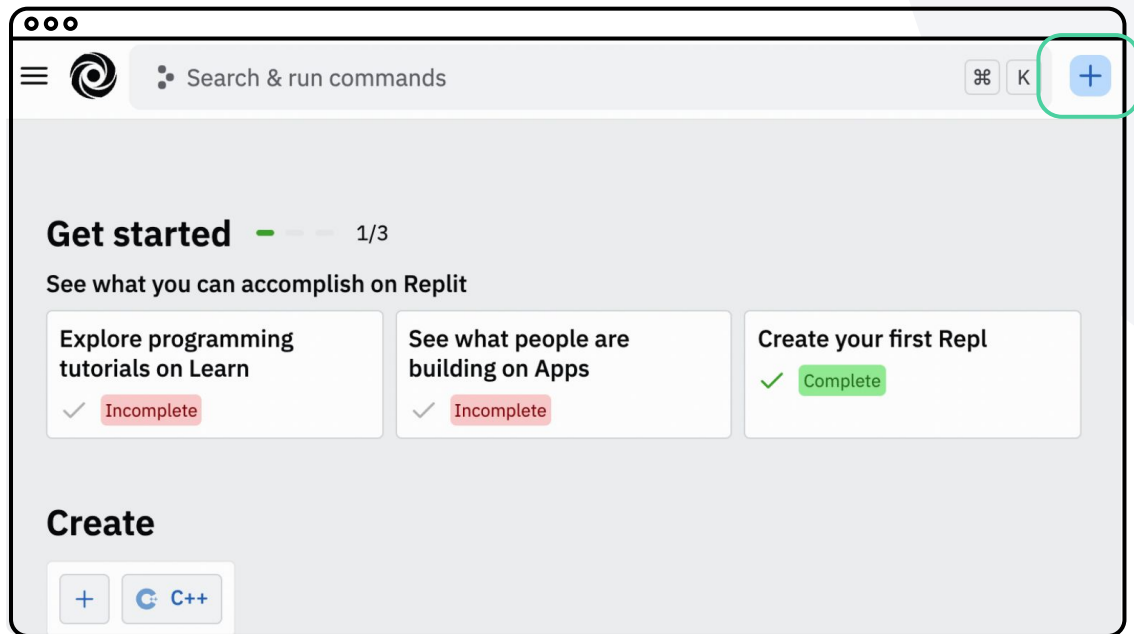
Это самая простая и самая маленькая программа на C++ (она не делает абсолютно ничего и поэтому совершенно бесполезна ;-)

Но, пока мы не познакомимся с принципами объектно-ориентированного программирования (ООП), необходимо начинать именно с этой конструкции

Создание программы в Repl.it

Шаг 1

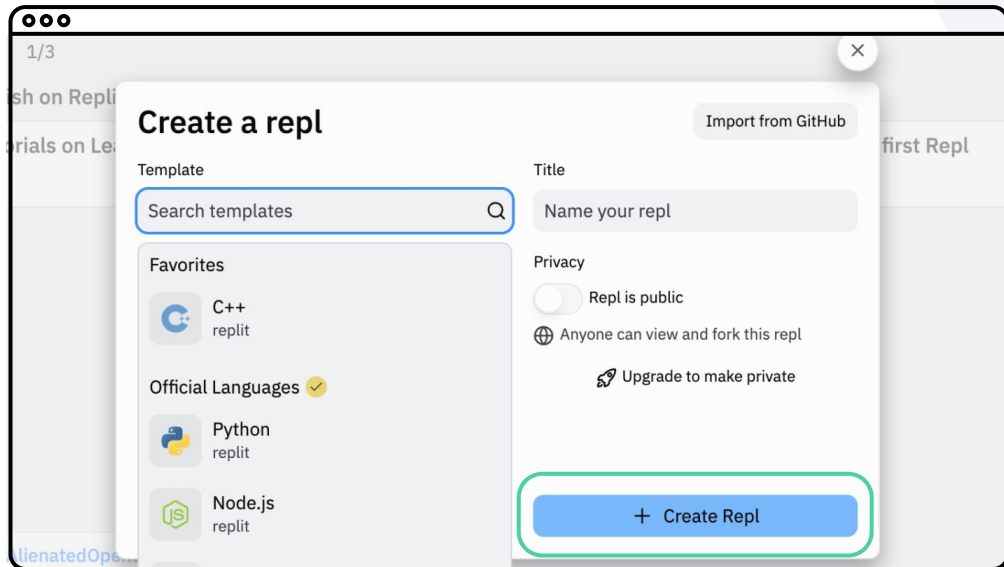
Открываем в браузере сайт repl.it и нажимаем кнопку New Repl (плюсик в правом верхнем углу)



Создание программы в Repl.it

Шаг 2

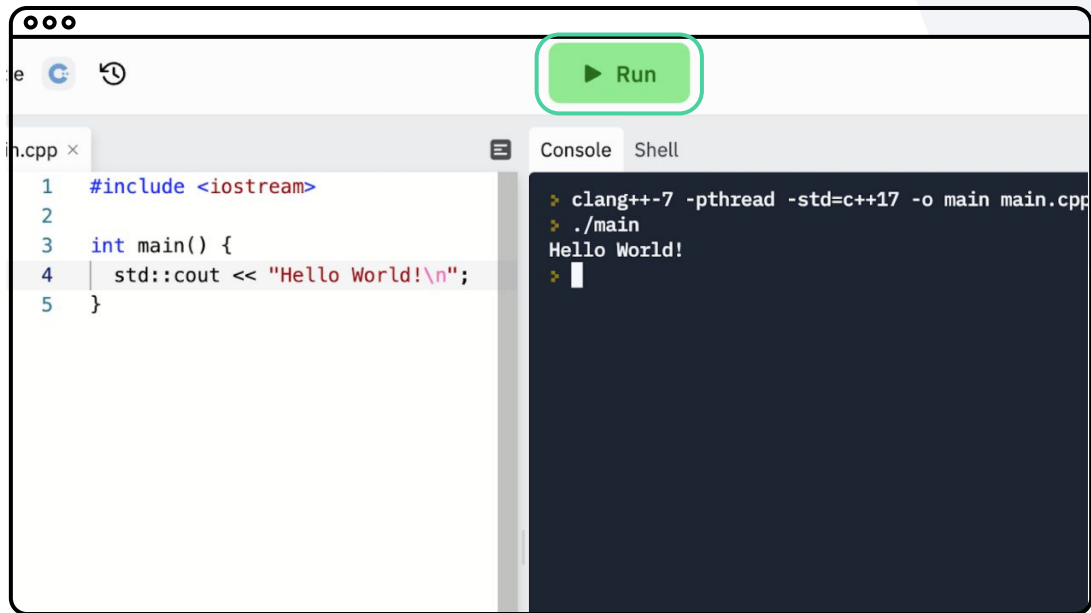
- В открывшемся окне выбираем язык C++
- Указываем название
- Нажимаем кнопку Create Repl



Создание программы в Repl.it

Шаг 3

Мы создали самую простую программу для вывода сообщений. Для запуска программы нужно нажать кнопку **Run**. После запуска увидим сообщение в чёрном окне (терминале) «**Hello world**».



The screenshot shows the Repl.it web interface. On the left, a code editor displays a C++ program in a file named `main.cpp`. The code consists of five lines: a preprocessor directive to include `<iostream>`, an opening brace for the `main` function, a `cout` statement to print "Hello World!\n", and a closing brace. On the right, there are two tabs: "Console" and "Shell". The "Console" tab is active, showing the compilation and execution process. It displays the command `clang++-7 -pthread -std=c++17 -o main main.cpp`, followed by `./main`, and the output `Hello World!`. A green rectangular box highlights the "Run" button, which is a green circle with a white play icon and the text "Run".

```
1  #include <iostream>
2
3  int main() {
4      std::cout << "Hello World!\n";
5  }
```

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
Hello World!
```

Пишем Hello World в repl.it

```
const log = require('log');
let embed;

function transform($, transform) {
  // Promise.resolve to protect against null
  return transform($);
}

function removeLinkHeader($, prev) {
  return prev.then(() => {
    $(':header').each((i, header) => {
      const children = $(header).children();
      if (children.length) {
        $(header).text(children);
        $(children).remove();
      }
    });
    return header;
  });
}
```

Как же наша программа работает?

```
int main()
{
    // здесь нужно писать команды языка (инструкции)
    // логика инструкции всегда завершается точкой с запятой ;

    // кстати, это комментарий —
    // то, что начинается с // и до конца строки

    /*
    ещё бывают многострочные комментарии
    как сейчас
    всё, что находится между символами /* и */,
    также является комментарием
    */

    // комментарии не являются логикой программы
    // они полностью игнорируются при выполнении
    // и нужны для пометок и пояснения частей кода
    // если их удалить, то программа не сломается
}
```

Итоги

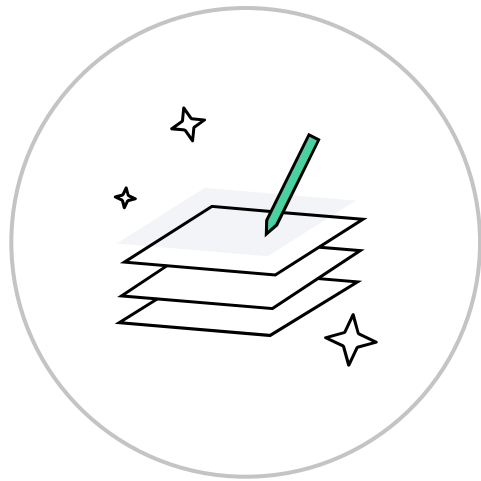
- 1 Выяснили, как работает компьютер и что такое машинный код
- 2 Узнали, что такое ассемблер и языки программирования высокого уровня
- 3 Познакомились с историей возникновения языков C/C++
- 3 Научились создавать простые программы на C++



Домашнее задание

Давайте посмотрим ваше домашнее задание

- 1 Вопросы по домашней работе задавайте в чате группы
- 2 Задачи можно сдавать по частям
- 3 Зачёт по домашней работе ставят после того, как приняты все задачи



Задавайте вопросы и пишите отзыв о лекции

Михаил Марков
C++ - разработчик

