

Введение в C++

Чуть-чуть познакомлюсь с вами

- Меня зовут Данил
- Закончил ТФТЛ в 2018-м года
- Учусь в НИУ ВШЭ на Факультете Компьютерных Наук на 3-м курсе
- Прошлым летом работал в Яндексе



Почему я думаю, что хоть чему-то смогу вас научить?

Олимпиада		Код подтверждения (номер электронного диплома)	Имя на дипломе	Класс
№13. "Всесибирская открытая олимпиада школьников" ("информатика"), 2 уровень. Диплом 3 степени.	чёрно-белая версия (~120Кб) цветная версия (~1.2Мб)	117 0423-31562	Мячин Данил Александрович, 29.04.2000	11
№27. "Межрегиональная олимпиада школьников «Высшая проба»" ("информатика"), 2 уровень. Диплом 1 степени.	чёрно-белая версия (~120Кб) цветная версия (~1.2Мб)	117 2169-09612	Мячин Данил Александрович, 29.04.2000	11
№32. "Межрегиональная олимпиада школьников по информатике и компьютерной безопасности" ("компьютерная безопасность"), 2 уровень. Диплом 3 степени.	чёрно-белая версия (~120Кб) цветная версия (~1.2Мб)	119 0209-25770	Мячин Данил Александрович, 29.04.2000	11

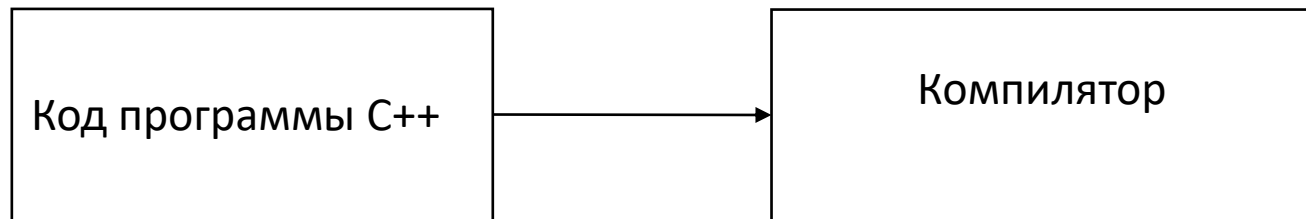
Я работал учебным ассистентом по курсу «Алгоритмы и структуры данных-1» и сейчас работаю учебным ассистентом по курсу «Алгоритмы и структуры данных-2» на нашем направлении

Как исполняются программы

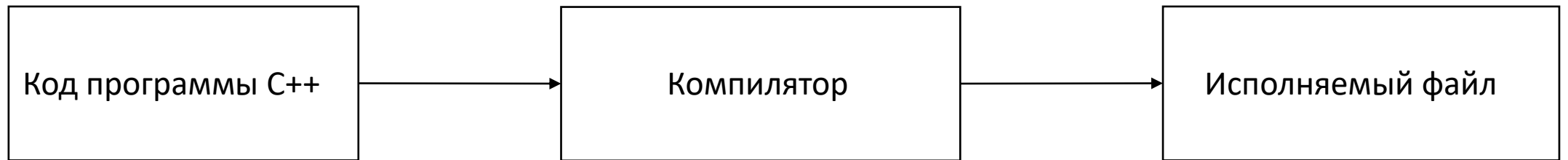
Как исполняются программы

Код программы C++

Как исполняются программы



Как исполняются программы



Какие есть стандартные типы данных?

Какие есть стандартные типы данных?

- `int` – целое 4 байта (не всегда)
- `char` – «символ» 1 байт
- `long long` – целое 8 байт
- Слово `unsigned` – неотрицательные значения
пример: `unsigned int`
- `float` – вещественное 4 байта
- `double` – вещественное 8 байт

Условный оператор

Условный оператор

Дано: на вход программе задаётся отрезок двумя числами – левым и правым концом. После отрезка на вход программе подаётся точка на координатной прямой
Что нужно сделать: узнать то, лежит ли точка на заданном отрезке.

Пример входных данных 1:

Input:

1 10 7

Output:

dot in section

Пример входных данных 2:

Input:

1 10 15

Output:

dot not in section

Условный оператор

Дано: на вход программе задаётся отрезок двумя числами – левым и правым концом. После отрезка на вход программе подаётся точка на координатной прямой
Что нужно сделать: узнать то, лежит ли точка на заданном отрезке.

Пример входных данных 1:

Input:

1 10 7

Output:

dot in section

Пример входных данных 2:

Input:

1 10 15

Output:

dot not in section

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int l, r, d;
6      cin >> l >> r >> d;
7
8      if (l <= d && d <= r) {
9          cout << "dot in section";
10     } else {
11         cout << "dot not in section";
12     }
13
14     return 0;
15 }
```

Цикл while

Цикл while

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5
6      while (/*условие здесь истинно*/) {
7          /*
8           .
9           .
10          .
11
12          Делай это всё, пожалуйста
13
14          .
15          .
16          .
17          */
18      }
19      return 0;
20 }
```

Цикл while

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5
6      while (/*условие здесь истинно*/) {
7          /*
8           .
9           .
10          .
11
12          Делай это всё, пожалуйста
13
14          .
15          .
16          .
17          */
18      }
19      return 0;
20 }
```

```
1.  #include <iostream>
2.  using namespace std;
3.
4.  int main() {
5.      int n = 10;
6.
7.      while (n > 0) {
8.          cout << n << ' ';
9.          n--;
10.     }
11.
12.
13.     return 0;
14. }
```

⚙️ stdout

10 9 8 7 6 5 4 3 2 1

Цикл for

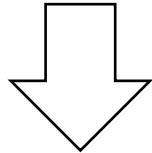
Здесь всё чуть посложнее, обо всё по порядку

Цикл for

```
for (int i = 0; i < 30; ++i) {  
    cout << i << ' ' ;  
}
```

Цикл for

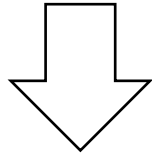
Выполнится один раз, перед
началом исполнения цикла



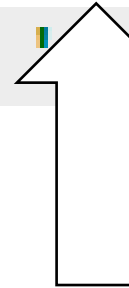
```
for (int i = 0; i < 30; ++i) {  
    cout << i << ' ' ;  
}
```

Цикл for

Выполнится один раз, перед
началом исполнения цикла



```
for (int i = 0; i < 30; ++i) {  
    cout << i << ' ';  
}
```

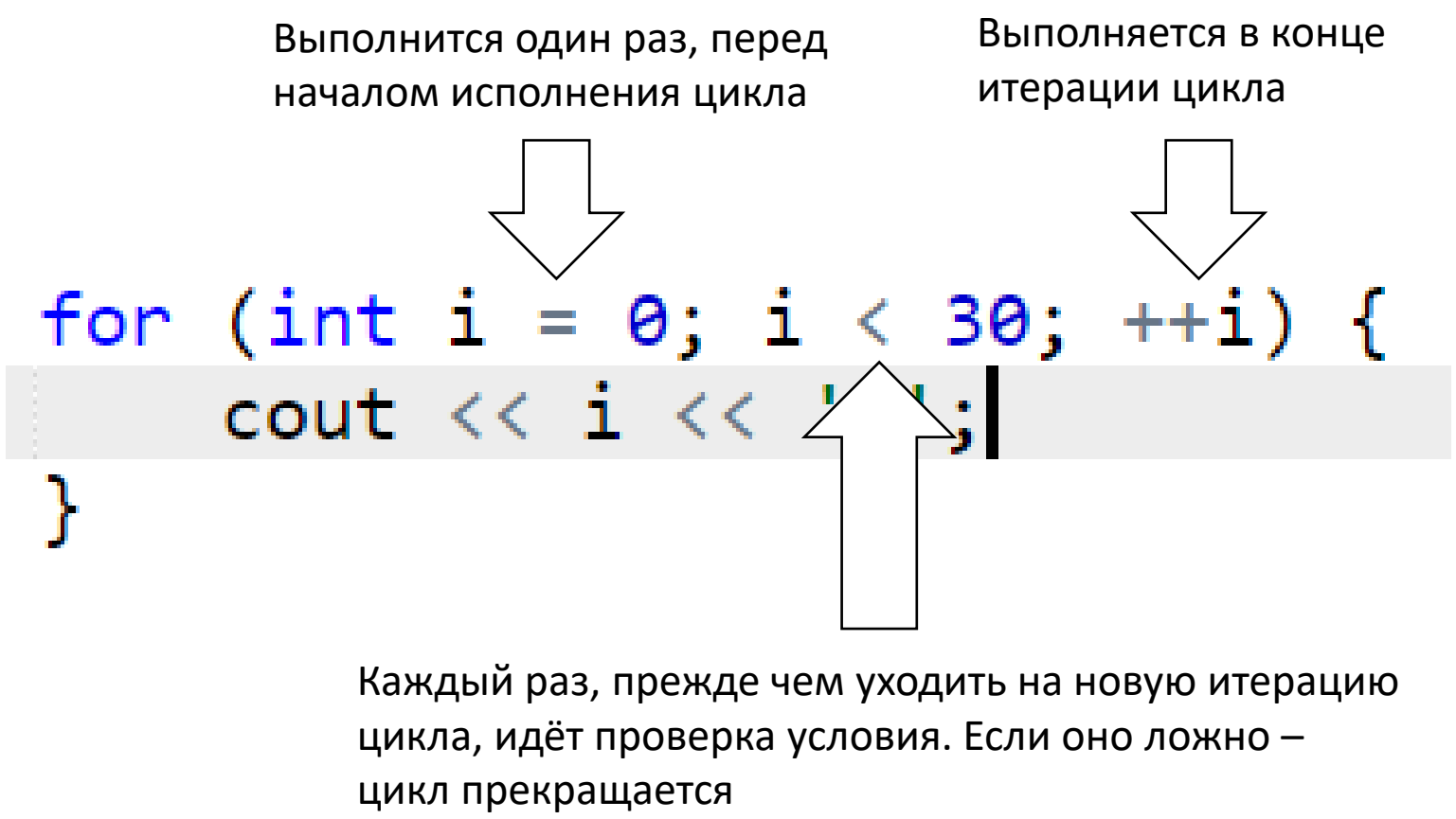


Каждый раз, прежде чем уходить на новую итерацию
цикла, идёт проверка условия. Если оно ложно –
цикл прекращается

Цикл for

Выполнится один раз, перед
началом исполнения цикла

Выполняется в конце
итерации цикла



```
for (int i = 0; i < 30; ++i) {  
    cout << i << ' ';  
}
```

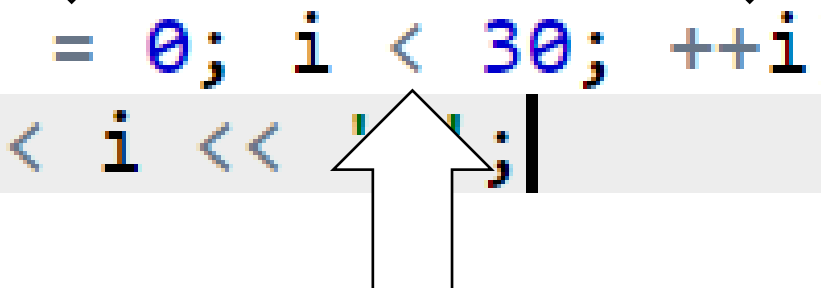
The diagram illustrates the execution of a C++ for loop. The code is shown with a light gray background. Three arrows indicate the flow: a downward arrow from the first text block points to the initialization 'int i = 0'; a second downward arrow from the second text block points to the increment '++i'; and a large upward arrow from the third text block points to the condition 'i < 30'.

Каждый раз, прежде чем уходить на новую итерацию
цикла, идёт проверка условия. Если оно ложно –
цикл прекращается

Цикл for

Выполнится один раз, перед
началом исполнения цикла

Выполняется в конце
итерации цикла



```
for (int i = 0; i < 30; ++i) {  
    cout << i << ' ';<br/>}
```

Каждый раз, прежде чем уходить на новую итерацию
цикла, идёт проверка условия. Если оно ложно –
цикл прекращается

```
1  #include <iostream>  
2  using namespace std;  
3  
4  int main() {  
5      {  
6          int i = 0;  
7          while (i < 30) {  
8              cout << i << ' ';<br/>9              ++i;<br/>10         }  
11     }  
12     return 0;  
13 }
```

vector<T>

vector<T>

Мы можем создать сразу много переменных и обращаться к ним по их порядковым номерам.

vector<T>

Мы можем создать сразу много переменных и обращаться к ним по их порядковым номерам.

Для тех, кто знаком с массивами – вектор это и есть массив.

vector<T>

Мы можем создать сразу много переменных и обращаться к ним по их порядковым номерам.

Для тех, кто знаком с массивами – вектор это и есть массив.

Для тех, кто знаком с динамическими массивами – вектор это динамический массив.

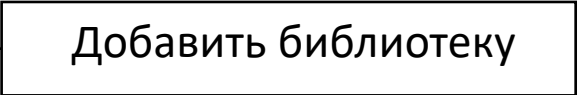
Как использовать vector

```
1.  #include <iostream>
2.  #include <vector>
3.  using namespace std;
4.
5.  int main() {
6.
7.      vector<int> vector_name(10, 1);
8.
9.      for (int i = 0; i < 10; ++i) {
10.         cout << vector_name[i] << ' ';
11.     }
12.
13.     return 0;
14 }
```

Как использовать vector

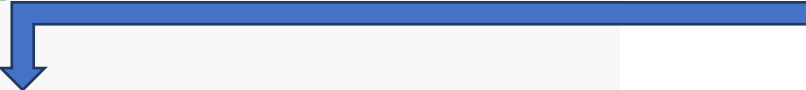
```
1. #include <iostream>
2. #include <vector>
3. using namespace std;
4.
5. int main() {
6.
7.     vector<int> vector_name(10, 1);
8.
9.     for (int i = 0; i < 10; ++i) {
10.         cout << vector_name[i] << ' ';
11.     }
12.
13.     return 0;
14. }
```

Добавить библиотеку



Как использовать vector

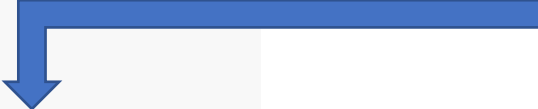
```
1.  #include <iostream>
2.  #include <vector>
3.  using namespace std;
4.
5.  int main() {
6.      vector<int> vector_name(10, 1);
7.
8.      for (int i = 0; i < 10; ++i) {
9.          cout << vector_name[i] << ' ';
10.     }
11.
12.     return 0;
13. }
14
```



Здесь указываем тип данных,
для которого мы хотим
создать вектор

Как использовать vector


```
1. #include <iostream>
2. #include <vector>
3. using namespace std;
4.
5. int main() {
6.
7.     vector<int> vector_name(10, 1);
8.
9.     for (int i = 0; i < 10; ++i) {
10.         cout << vector_name[i] << ' ';
11.     }
12.
13.     return 0;
14. }
```



Первым параметром
передаём кол-во элементов,
которые мы хотим создать

Как использовать vector


```
1.  #include <iostream>
2.  #include <vector>
3.  using namespace std;
4.
5.  int main() {
6.
7.      vector<int> vector_name(10, 1);
8.
9.      for (int i = 0; i < 10; ++i) {
10.         cout << vector_name[i] << ' ';
11.     }
12.
13.     return 0;
14 }
```



Вторым параметром передаём
то, какими значениями мы хотим
заполнить элементы

Как использовать vector

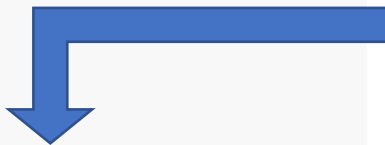
```
1.  #include <iostream>
2.  #include <vector>
3.  using namespace std;
4.
5.  int main() {
6.
7.      vector<int> vector_name(10, 1);
8.
9.      for (int i = 0; i < 10; ++i) {
10.         cout << vector_name[i] << ' ';
11.     }
12.
13.     return 0;
14. }
```



Нумерация элементов начинается с нуля
Вспомнив цикл for, мы понимаем, как нам
перебрать все значения в vector'e

Как использовать vector

```
1.  #include <iostream>
2.  #include <vector>
3.  using namespace std;
4.
5.  int main() {
6.
7.      vector<int> vector_name(10, 1);
8.
9.      for (int i = 0; i < vector_name.size(); ++i) {
10.         cout << vector_name[i] << ' ';
11.     }
12.
13.     return 0;
14. }
```



А ещё vector умеет сам
поддерживать свой
размер

Что делать, если мы хотим двумерный вектор?

Что делать, если мы хотим двумерный вектор?



Что делать, если мы хотим двумерный вектор?

```
1.  #include <iostream>
2.  #include <vector>
3.
4.  using namespace std;
5.
6.  int main() {
7.      int n = 30;
8.      vector<vector<int>> x(n, vector<int> (10, 5));
9.      return 0;
10. }
```


Строки

Строки

```
1.  #include <iostream>
2.  #include <string>
3.  using namespace std;
4.
5.  int main() {
6.      string name, nickname;
7.
8.      cin >> name >> nickname;
9.
10.     cout << name[0] << ' ' << nickname[0] << '\n';
11.
12.     for (int i = 0; i < nickname.size(); ++i) {
13.         if ('A' <= nickname[i] && nickname[i] <= 'Z') {
14.             nickname[i] = ('a' + (nickname[i] - 'A'));
15.         }
16.     }
17.
18.     cout << nickname << ' ' << name;
19.
20.     return 0;
21. }
```

 stdin

Danil MagicD

 stdout

D M

magiced Danil

Если успеем, разберём пару задачек:

- 324 с астр
- 44 с астр
- 22 с астр

Где можно будет найти всю инфу о факультативе?

- Телеграм-канал https://t.me/TFTL_informatics . В нём будут все новости и какие-то уведомления
- Репозиторий https://github.com/mdan2000/tftl_olymp

ДЗ

Задачи уже выложены на

https://github.com/mdan2000/tftl_olymp/intro_C++/I0.md

(презентацию, надеюсь, вы там тоже сможете найти)

Вы просто сдаёте задачи на тех сервисах, на которых предложено.

Пока функционал будет такой, что вы прикрепляете потом задачи в google-classroom'е и сдаёте.