

Лабораторная работа 1.3(3) Новое в C++

МГТУ им. Н.Э. Баумана

February 20, 2016

Ввод и вывод

C:

```
1 #include <stdio.h>
2 // ...
3 char c = 'a';
4 int n = 10;
5 double d = 30.;
6 // ...
7 scanf("%c", &c);
8 scanf("%d", &n);
9 // ...
10 printf("%c %d %lf\n", c, n, d);
11 //
```

C++:

```
1 #include <iostream>
2 // ...
3 char c = 'a';
4 int n = 10;
5 double d = 30.;
6 // ...
7 std::cin >> c;
8 std::cin >> n;
9 // ...
10 std::cout << c << " " << n << " "
    << d << std::endl;
```

TODO:

- Вывести "Hello, world"
- Получить введенное пользователем число N
- Вывести на двух строчках (числа от 1 до N) и (числа от N до 1)

Explore: #include <iomanip>

```
1 std::cout << std::setw(4) << 0 << 1;
2 std::cout << std::setw(4) << std::setfill('0') << 1;
3 std::cout << std::hex << 10;
```

Выделение динамической памяти

Указатель на одно значение

C:

```
1 int* a=malloc(sizeof(int));  
2 *a=0  
3 // ...  
4 free(a);
```

C++:

```
1 int* a=new int(0); //initialized  
2 // ...  
3 delete a;
```

Указатель на массив

C:

```
1 int* a=malloc(10*sizeof(int));  
2 // ...  
3 free(a);
```

C++:

```
1 int* a=new int[10];  
2 // ...  
3 delete[] a;
```

Перегрузка функций

Нет ошибок компиляции:

```
1 bool ge(int i1, int i2) {  
2     return i1 >= i2;  
3 }  
4 bool ge(double d1, int d2) { // same name  
5     // params type differs  
6     return d1 >= d2;  
7 }  
8 bool ge(double d1, int d2, int upper_bound) { // same name  
9     // params count differs  
10    return d1 >= d2;  
11 }
```

Ошибка компиляции:

```
1 bool ge(int i1, int i2) {  
2     return i1 >= i2;  
3 }  
4 char ge(int i1, int i2) { // ERROR  
5     // return type difference is not enough  
6     return i1 >= i2;  
7 }  
8 bool ge(int i1, int i2, int upper_bound = 10) { // ERROR  
9     // Do I call bool ge(int i1, int i2) or bool ge(int i1, int i2, int  
10    upper_bound = 10)?
```

Ссылка

- как указатель
- не требуется * при доступе
- ссылка не может указывать на другой объект после инициализации

C:

```
1 int a = 0;  
2 int *b = &a; //pointer  
3 *b = 1  
4 assert(a == 1);
```

C++:

```
1 int a = 0;  
2 int& b = a; // reference  
3 b = 1  
4 assert(a == 1);
```

TODO:

- Реализовать функцию swar для целых чисел без указателей
- Реализовать функцию swar для вещественных чисел без указателей
- Проверить, что для каждого типа зовется своя функция

Range-based for

C:

```
1 int a[10] = {0};  
2 for (int i = 0; i < sizeof(a)/sizeof(a[0]); ++i)  
3     a[i] += 1;
```

C++:

```
1 int a[10] = {0};  
2 for (int& val : a) // Note the reference  
3     v += 1;
```

TODO:

- Получить введенное пользователем число N
- Создать и заполнить числами целочисленный массив размера N
- Создать целочисленный массив и записать в него квадраты элементов первого массива
- Вывести второй массив

Ввод и вывод в файлы

C:

```
1 #include <stdio.h>
2 // ...
3 char c = 'a';
4 int n = 10;
5 double d = 30.;
6
7 FILE* fp = fopen("1.txt", "w");
8 fprintf(fp, "%c %d %lf\n", c, n,
9         d);
10 fclose(fp);
11
12 fp = fopen("1.txt", "r");
13 fscanf(fp, "%c", &c);
14 fscanf(fp, "%d", &n);
15 fclose(fp);
16 //
```

C++:

```
1 #include <fstream>
2 // ...
3 char c = 'a';
4 int n = 10;
5 double d = 30.;
6 {
7     std::fstream fout("1.txt", std
8                       ::fstream::out);
9     std::fout << c << " " << n << "
10    " << d << std::endl;
11 } //closed on destruction
12 {
13     std::fstream fin("1.txt", std::
14                      fstream::in);
15     std::fin >> c;
16     std::fin >> n;
17 } //closed on destruction
```

- Создать массив со случайными целыми числами размера, указанного пользователем
- Записать массив в файл.
- Считать массив из файла.
- Для всех элементов массива выполнить преобразование $x^{**31} \bmod 1023$, результат записать в новый массив.
- Отсортировать оба массива.
- Выполнить слияние массивов в один с сохранением упорядоченности
- Записать результат в файл
- Почистить память

Промежуточные данные о выполнении действий выводите в поток вывода