

# Объектно-ориентированное программирование. Лабораторная работа №3. ПОТОКОВЫЙ ВВОД/ВЫВОД

Макаров П. А.

7 октября 2022 г.

## Содержание

<a href="#">1</a>	<a href="#">Задания для самостоятельной работы</a>	<a href="#">1</a>
<a href="#">2</a>	<a href="#">Список источников</a>	<a href="#">5</a>

## 1 Задания для самостоятельной работы

1. Изучите приведённую в списке источников документацию, посвящённую потоковому вводу/выводу, а также строкам в языках С и С++.
2. Используя Листинг 1 получите начальный опыт работы с объектами и манипуляторами объектно-ориентированной библиотеки ввода/вывода.

---

```
1 #include <iostream>
2 #include <iomanip>
3
4 int main () {
5     double pi = 3.14159;
6     double x;
7     std::cout << "Input x: ";
8     std::cin >> x;
9     std::cout << std::setprecision(5) << pi*x << '\n';
10    std::cout << std::setprecision(9) << pi*x << '\n';
11    std::cout << std::fixed;
12    std::cout << std::setprecision(5) << pi*x << '\n';
13    std::cout << std::setprecision(9) << pi*x << '\n';
14    return 0;
15 }
```

---

## Листинг 1: Содержимое файла `precision.cpp`

3. Внимательно исследуйте пример, приведённый в Листинге 2. Действительно ли использование возможностей библиотеки потокового ввода/вывода `iostream` удобнее “классических средств” языка C (библиотеки стандартного ввода/вывода `cstdio`)?

---

```
1 #include <cstdio>
2 #include <iostream>
3
4 using namespace std;
5
6 int main () {
7     int a = 060;
8     int b = 60;
9     int c = 0x60;
10
11     printf("C style\n");
12     printf("symbol\toctal\tdecimal\thex\n");
13     printf("%6c\t%5o\t%7d\t%3x\n", a, a, a, a);
14     printf("%6c\t%5o\t%+7d\t%3x\n", b, b, b, b);
15     printf("%-6c\t%-5o\t%-7d\t%-3x\n", c, c, c, c);
16
17     cout << "\nC++ style" << endl;
18     cout << "symbol\toctal\tdecimal\thex" << endl;
19     cout.width(6);    cout << (char)a;
20     cout.width(7);    cout << oct << a;
21     cout.width(10);   cout << dec << a;
22     cout.width(4);    cout << hex << a << endl;
23
24     cout << showpos;
25     cout.width(6);    cout << (char)b;
26     cout.width(7);    cout << oct << b;
27     cout.width(10);   cout << dec << b;
28     cout.width(4);    cout << hex << b << endl;
29
30     cout << noshowpos << left;
31     cout.width(8);    cout << (char)c;
32     cout.width(8);    cout << oct << c;
33     cout.width(8);    cout << dec << c;
34     cout.width(3);    cout << hex << c << endl;
35
36     return 0;
37 }
```

---

## Листинг 2: Содержимое файла `bases.cpp`

4. Разберитесь на примере Листинга 3 в отличиях между строками в C и C++.

---

```
1 #include <iostream>
2 #include <string>
```

```

3 #include <cstring>
4
5 using namespace std;
6
7 int main() {
8     string one = "Gibberish!";
9     char two[] = "Gibberish!";
10
11     cout << "one: " << one << endl;
12     cout << "two: " << two << endl;
13
14     cout << "\nModify both strings\n\n";
15
16     one[4] = '\0';
17     two[4] = '\0';
18
19     cout << "one: " << one << endl;
20     cout << "Length of one: " << one.size() << endl;
21     cout << "Sizeof(one): " << sizeof(one) << endl <<
        endl;
22
23     cout << "two: " << two << endl;
24     cout << "Length of two: " << strlen(two) << endl;
25     cout << "Sizeof(two): " << sizeof(two) << endl;
26
27     return 0;
28 }

```

---

Листинг 3: Содержимое файла `strings.cpp`

5. Изучите второй пример работы со строками в C++, приведённый в Листинге 4. Как выглядел бы аналогичный текст на языке C? В чём преимущество подхода, основанного на использовании класса `string` и функции `getline`?

---

```

1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 int main () {
7     string mystr;
8     cout << "What's your name? ";
9     getline(cin, mystr);
10    cout << "Hello " << mystr << ".\n";
11    cout << "What is your favorite team? ";
12    getline(cin, mystr);
13    cout << "I like " << mystr << " too!\n";
14    return 0;
15 }

```

---

Листинг 4: Содержимое файла `getline.cpp`

6. Третий пример работы со строками в C++, приведён в Листинге 5. Используя его, ознакомьтесь с основами работы с потоковым классом `stringstream`, предназначенным для манипулирования со строками.

---

```
1 #include <iostream>
2 #include <string>
3 #include <sstream>
4
5 using namespace std;
6
7 int main () {
8     string mystr;
9     double price = 0;
10    int quantity = 0;
11
12    cout << "Enter price: ";
13    getline(cin, mystr);
14    stringstream(mystr) >> price;
15    cout << "Enter quantity: ";
16    getline(cin, mystr);
17    stringstream(mystr) >> quantity;
18    cout << "Total price: " << price*quantity << endl;
19    return 0;
20 }
```

---

Листинг 5: Содержимое файла `stringstream.cpp`

7. Напишите программу, выполняющую операции сложения, умножения и транспонирования прямоугольных матриц. Ввод и вывод матриц должен осуществляться с помощью объектов `cin` и `cout`.
8. Организуйте ввод матрицы из текстового файла с помощью объекта класса `ifstream` и вывод транспонированной матрицы в текстовый файл с помощью объекта класса `ofstream`.
9. Дан текстовый файл, содержащий информацию о городах мира (название города, название страны, численность населения, площадь). Выполните следующие задания:
- Найти самый населенный город в заданной стране;
  - Найти город, который имеет наибольшую плотность населения;
  - Распечатать названия всех стран, которые имеют города с населением более 1 000 000 человек;
  - Найти количество городов, которые расположены в заданной стране.

## 2 СПИСОК ИСТОЧНИКОВ

1. <https://cplusplus.com/reference/cstdio/>.
2. <https://cplusplus.com/reference/iolib/>.
3. <https://cplusplus.com/reference/ios/>.
4. [https://cplusplus.com/reference/ios/ios\\_base/](https://cplusplus.com/reference/ios/ios_base/).
5. <https://cplusplus.com/reference/iostream/>.
6. <https://cplusplus.com/reference/fstream/>.
7. <https://cplusplus.com/reference/cstring/>.
8. <https://cplusplus.com/reference/string/>.