

## Лабораторная работа № 2

**Тема:** Изучение отличий языка C++ от языка C

**Цель:** Изучить основные отличия языка C++ от языка C: библиотека ввода/вывода, объявление структур, объединений и перечислений, работа с динамической памятью, перегрузка имен функций и передача параметров по ссылке.

### Краткая теория

В языке C++ ввод и вывод осуществляется также как и языке C через потоки. Но в языке C++ используется библиотека `iostream`, в которой описаны классы `istream` и `ostream`. Класс `istream` предназначен для организации ввода данных, у этого класса перегружена операция `>>`, с помощью которой осуществляется ввод данных в переменные. Класс `ostream` предназначен для организации вывода данных, у этого класса перегружена операция `<<`, с помощью которой осуществляется вывод значений переменных. В любой консольной программе автоматически доступны объекты `cin` (класса `istream`) – стандартный поток ввода, `cout` (класс `ostream`) – стандартный поток вывода. Так как классы `istream` и `ostream`, а также объекты `cin` и `cout` описаны в пространстве имен `std`, то при написании программы целесообразно включить в текст программы `using namespace std`. Пример:

```
#include <iostream>
using namespace std;

int main(int argc, char* argv[])
{
    int x = 0, y = 0;
    cout << "Введите X: ";
    cin >> x;
    cout << "Введите Y: ";
    cin >> y;
    cout << "Сумма " << x << '+' << y << " = " << x+y << '\n';
    return 0;
}
```

Второе отличие языка C++ от языка C заключается в том, что в языке C++ не нужно использовать оператор `typedef` для создания новых типов структур, объединений или перечислений. Например:

```
#include <iostream>
using namespace std;
int main(int argc, char* argv[])
{
    struct Student{
        char fio[3][16];
        int kurs;
        float rate;
    };

    Student st;
    cout << "Введите в формате \"Имя Фамилия Отчество\
        курс успеваемость\"\n";
    cin >> st.fio[0] >> st.fio[1] >> st.fio[2] >> st.kurs >> st.rate;
    cout << "Имя: " << st.fio[0] << ' ' << st.fio[1] << ' '
        << st.fio[2] << '\n';
    cout << "Курс: " << st.kurs << '\n';
    cout << "Успеваемость: " << st.rate << '\n';
}
```

```

    return 0;
}

```

В языке C++ при объявлении функции без параметров допускается оставлять пустые скобки, не указывая ключевое слово `void`, как это делается в языке C.

```

#include <iostream>
using namespace std;

void Func();
int main(int argc, char* argv[])
{
    Func();
    return 0;
}
void Func()
{
    cout << "Функция без параметров!\n";
}

```

В языке C++ реализован механизм передачи параметров по ссылке. Для этого используется описание параметра в виде:

тип &имя

Пример реализации функции возведения целочисленного значения в квадрат:

```

#include <iostream>
using namespace std;

void Sqr(int &v) { v *= v;}

int main(int argc, char* argv[])
{
    int z = 4;
    cout << "Перед Sqr: " << z << '\n';
    Sqr(z);
    cout << "После Sqr: " << z << '\n';
    return 0;
}

```

При передаче в параметрах функции большого значения (структура) целесообразно использовать передачу константной ссылки:

```

#include <iostream>
using namespace std;

struct Student{
    char fio[3][16];
    int kurs;
    float rate;
};

void PrintStudent(const Student &s)
{
    cout << "Имя: " << s.fio[0] << ' ' << s.fio[1] << ' '
        << s.fio[2] << '\n';
}

```

```

    cout << "Курс: " << s.kurs << "\nУспеваемость: "
        << s.rate << '\n';
}

int main(int argc, char* argv[])
{
    Student st = {"Иванов", "Иван", "Иванович"}, 3, 6.5};
    PrintStudent(st);
    return 0;
}

```

Для работы с динамической памятью в языке C++ можно использовать функции из библиотеки `stdlib.h` (`calloc`, `malloc`, `realloc`, `free`), но целесообразнее использовать операторы `new` и `delete` (особенно при создании объектов).

Выделение памяти:                      указатель = **new** тип;

Освобождение памяти:                **delete** указатель;

Примеры:

```

int *ptr = new int;
cin >> *ptr;
cout << "Value: " << *ptr << endl;
delete ptr;

ptr = new int (10);
cout << "Value: " << *ptr << endl;
delete ptr;

```

Выделение памяти под массив:        указатель = **new** тип[размер];

Освобождение памяти:                **delete []** указатель;

Пример:

```

int *ptr = new int [10];
for(int i=0;i<10;i++) cin >> ptr[i];
for(int i=0;i<10;i++) cout << ptr[i] << ' ';
cout << endl;
delete [] ptr;

```

В языке C++ можно описывать функции с параметрами по умолчанию. Если при вызове функции значение данного параметра не указано, то используется значение по умолчанию.

Пример:

```

double Volume(double l, double =1.0, double =1.0);
int main(int argc, char* argv[])
{
    cout << "Объем: " << Volume(2,3,4) << endl;    //вывод: 24
    cout << "Объем: " << Volume(2,3) << endl;      //вывод: 6
    cout << "Объем: " << Volume(2) << endl;        //вывод: 2
    return 0;
}

double Volume(double l, double w, double h)
{
    return l*w*h;
}

```

В языке C++ допускается перегрузка функций – возможность использования одного и того же идентификатора для именования нескольких функций. Перегруженные функции должны различаться по количеству, порядку и типу формальных параметров. Пример:

```
double Square(double) ;  
double Square(double,double) ;  
  
int main(int argc, char* argv[])  
{  
    cout << "Square of circle: " << Square(5) << endl;  
    cout << "Square of rectangle: " << Square(2,6) << endl;  
    return 0;  
}  
double Square(double r) { return 3.1415*r*r; }  
double Square(double a, double b) { return a*b; }
```

### Ход работы

При выполнении задания необходимо реализовать дружественный интерфейс: при вводе (выводе) данных выводится приглашение, которое содержит описание вводимой (выводимой) величины (назначение и тип).

В данной лабораторной работе предполагается, что все значения могут вводиться некорректно. Поэтому необходимо осуществлять проверку на корректность ввода с использованием операторов управления.

#### Задание 1

Во всех заданиях необходимо разработать программу, в которой выполняется ввод списка записей определенного типа, а затем – обработка списка. Сначала в программе должен вводиться размер списка (целое число), а сам список создается в виде массива структур в динамической памяти. Для выделения и последующего освобождения динамической памяти необходимо использовать операторы **new** и **delete** языка C++.

Ввод одной записи осуществляется по полям. При вводе каждого поля должно выводиться приглашение к вводу этого поля. Вывод одной записи осуществляется в одну строку согласно формату из варианта задания. Для ввода и вывода записей необходимо реализовать функции, в качестве параметра в которые передается ссылка в стиле языка C++ на вводимую или выводимую запись. При этом в функцию вывода должна передаваться константная ссылка. Ввод и вывод реализовывать с использованием возможностей библиотеки **iostream**.

После ввода осуществляется сортировка списка по определенным полям согласно варианту задания. Для сортировки списка необходимо реализовать функцию сортировки, в которую передается указатель на список, количество элементов в списке, а также направления сортировки по первому и второму полям сортировки. При этом функцию сортировки необходимо реализовать так, чтобы направления сортировки были параметрами со значением по умолчанию. В самой программе необходимо продемонстрировать использование этой возможности языка C++.

После сортировки списка осуществляется его вывод на экран. При этом у пользователя необходимо запросить: хочет ли он выполнять дополнительную обработку. Если пользователь ответит «да», то у пользователя запрашивается ввод условия вывода и выполняется вывод списка с дополнительной обработкой. Если же пользователь ответит «нет», то осуществляется вывод всего списка. Для вывода списка необходимо реализовать две функции: одна с обработкой, другая без обработки. При реализации этих функций необходимо использовать перегрузку имен функций. В этих функции должны передаваться: указатель на список, количество элементов в списке, а в функцию с обработкой дополнительно должно передаваться условие обработки.

При реализации структур по возможности необходимо использовать битовые поля, объединения и перечисления. Необходимо учитывать, что названия книг, наименования товаров, тексты сообщений, названия сеансов могут состоять из нескольких слов и заключаются в двойные кавычки. Направления всех сортировок указывает пользователь. Варианты заданий:

№	Структура	Сортировка	Обработка
1	СТУДЕНТ	Сортировать по курсу, а внутри курса по ФИО	Вывести всех записи о студентах определенной специальности.
2	КНИГА	Упорядочить по автору, а внутри автора по количеству страниц.	Вывести на экран записи о книгах, изданных за определенный период.
3	МАШИНА	Упорядочить список по марке, а внутри марки по году выпуска.	Вывести на экран записи о машинах определенного цвета.
4	ФАЙЛ	Упорядочить файлы по дате создания, а внутри даты по имени.	Вывести на экран все файлы, размер которых лежит в заданном интервале.
5	ПРОЦЕССОР	Упорядочить записи по производителю, а внутри – по частоте.	Вывести на экран все записи, определенной модели.
6	САМОЛЕТ	Упорядочить по марке-модели, а внутри по количеству часов налета.	Вывести все записи о самолетах определенного типа.
7	БИЛЕТ	Упорядочить билеты по дате, а внутри даты – по времени сеанса.	Вывести на экран все записи о билетах, номер ряда которых принадлежит заданному набору.
8	СООБЩЕНИЕ	Упорядочить по номеру устройства, а внутри – по тексту сообщения.	Вывести на экран сообщения, полученные за определенный временной период.
9	НАКЛАДНАЯ	Упорядочить по дате, а внутри даты по сумме накладной.	Вывести на экран все накладные с определенным ответственным лицом.
10	ФИРМА	Упорядочить по ФИО владельца, а внутри по дате основания.	Вывести все записи, в названии которых встречается определенное слово.
11	МАРШРУТ	Упорядочить по станции отправления, а внутри по станции прибытия.	Вывести все маршруты, с временем следования больше определенного количества минут.
12	ТЕЛЕФОН	Упорядочить по тарифу, а внутри тарифа по фамилии владельца.	Вывести на экран все записи с определенной комбинацией из двух цифр.
13	ВИДЕОДИСК	Упорядочить по кинокомпании, а внутри – по продолжительности.	Вывести на экран все записи о фильмах, определенного жанра.
14	ПОЕЗД	Упорядочить по станции отправления, а внутри по номеру поезда.	Вывести на экран все записи, количество вагонов в которых лежит в заданном диапазоне.
15	УСТРОЙСТВО	Упорядочить по типу устройства, а внутри по фирме производителю.	Вывести на экран все записи об устройствах, выпущенных в определенный период.
16	СТУДЕНТ	Сортировать по специальности, а внутри специальности по дате рождения	Вывести записи о студентах определенного курса.

17	КНИГА	Упорядочить по году издания, а внутри года издания по автору.	Вывести на экран записи о книгах, в названии которых встречается определенное слово.
18	МАШИНА	Упорядочить список по цвету, а внутри цвета по модели.	Вывести на экран записи о машинах, выпущенных за определенный период.
19	ФАЙЛ	Упорядочить файлы по расширению, а внутри по размеру.	Вывести на экран все файлы, созданные в определенный интервал времени.
20	ПРОЦЕССОР	Упорядочить записи по имени модели, а внутри – по сумме размеров КЭШ-памяти.	Вывести на экран все записи о процессорах, частота которых лежит в определенном диапазоне.
21	САМОЛЕТ	Упорядочить по году выпуска, а внутри по марке-модели.	Вывести все записи о самолетах, налетавших определенное количество часов.
22	БИЛЕТ	Упорядочить билеты по названию, а внутри – по номеру ряда.	Вывести на экран все записи о билетах, на определенную дату.
23	СООБЩЕНИЕ	Упорядочить по дате получения, а внутри – по времени получения.	Вывести на экран сообщения, полученные от определенного устройства.
24	НАКЛАДНАЯ	Упорядочить по ФИО отв. лица, а внутри по номеру.	Вывести на экран все накладные за определенный период.
25	ФИРМА	Упорядочить по дате основания, а внутри по названию.	Вывести все записи о фирме, с определенным владельцем.
26	МАРШРУТ	Упорядочить по времени отправления, а внутри по номеру маршрута.	Вывести все маршруты, которые отправляются или прибывают с определенной станции.
27	ТЕЛЕФОН	Упорядочить по дате подключения, а внутри по номеру.	Вывести на экран все записи, для определенного владельца.
28	ВИДЕОДИСК	Упорядочить по жанру, а внутри – по названию.	Вывести на экран все записи о фильмах, определенной кинокомпании.
29	ПОЕЗД	Упорядочить по количеству вагонов, а внутри по названию.	Вывести на экран все записи, в которых фигурирует определенная станция.
30	УСТРОЙСТВО	Упорядочить по типу фирме, а внутри по дате выпуска.	Вывести на экран все записи об устройствах, определенного типа.

СТУДЕНТ:

- фамилия, имя, отчество (строки по 15 символов);
- специальность (строка из двух символов);
- курс, на котором учится студент (целое число от 1 до 5);
- дата рождения (в формате дд.мм.гггг).

Формат ввода: *Фамилия Имя Отчество [Специальность, Курс], Дата рождения*  
КНИГА:

- код ISBN (строка 15 символов)
- фамилия и инициалы автора (строка 15 символов);
- название книги (строка 20 символов);
- год издания (целое четырехзначное число);

- количество страниц (целое четырехзначное число).

Формат ввода: *код ISBN: Фамилия И. О. «Название», год - количество страниц*  
**МАШИНА:**

- марка (строка 15 символов);
- модель (строка 5 символов);
- серийный номер (целое семизначное число);
- год выпуска (целое четырехзначное число);
- цвет (трехзначный целочисленный код)

Формат ввода: *Марка Модель [Серийный номер], Год выпуска, Цвет*  
**ФАЙЛ:**

- имя файла (строка 30 символов, уникальное поле);
- размер файла (целое число);
- дата создания (в формате дд.мм.гггг);
- время создания (в формате чч:мм).

Формат ввода: *Имя файла, размер файла, дата и время создания*  
**ПРОЦЕССОР:**

- производитель (строка 10 символов);
- модель (строка 15 символов);
- тактовая частота в МHz (целое число меньше 10000);
- размеры КЭШ памяти для данных и команд в КВ (целые числа меньше 32000).

Формат ввода: *Производитель Модель (Тактовая частотаМHz, КЭШДанныхКВ, КЭШКомандКВ)*

**САМОЛЕТ:**

- серийный номер (целое восьмизначное число)
- марка – модель (строка 20 символов);
- год выпуска (целое четырехзначное число);
- признак (Р – пассажирский, С – грузовой);
- время полета (целое число, в часах, меньше 15000).

Формат ввода: *Серийный номер Марка – модель (Признак), Год выпуска, Время полета*  
**БИЛЕТ:**

- название сеанса (строка 30 символов);
- дата и время сеанса (в формате дд.мм.гггг чч:мм);
- номер ряда (целое число меньше 50);
- номер места (целое число меньше 50).

Формат ввода: *Номер «Название сеанса» Дата и время, Номер ряда, Номер места*  
**СООБЩЕНИЕ:**

- номер устройства (целое трехзначное число)
- ID сообщения (целое восьмизначное число в 16-ричной системе);
- текст (строка 30 символов);
- дата и время отправления (в формате дд.мм.гггг чч:мм).

Формат ввода: *Номер устройства: ID сообщения «Текст» Дата и время отправления*  
**НАКЛАДНАЯ:**

- номер накладной (целое число, уникальное поле);
- дата накладной (в формате дд.мм.гггг);
- общая сумма по накладной (вещественное число);
- ФИО ответственного лица (строка 20 символов).

Формат ввода: *Номер: «Дата», Сумма, ФИО ответственного лица*  
**ФИРМА:**

- название (строка 20 символов);

- УНН (целое десятизначное число, уникальное поле);
- ФИО владельца (строка 20 символов);
- дата основания (в формате дд.мм.гггг).

Формат ввода: *Название [УНН], ФИО владельца, Дата основания*

МАРШРУТ:

- номер маршрута (целое четырехзначное число);
- станция отправления (строка 15 символов);
- конечная станция (строка 15 символов);
- время отправления (в формате чч:мм);
- время прибытия (в формате чч:мм).

Формат ввода: *Номер маршрута Станция отправления (Время отправления) – Конечная станция (Время прибытия)*

ТЕЛЕФОН:

- номер (целое семизначное число, уникальное поле);
- ФИО владельца (строка 30 символов);
- дата подключения (в формате дд.мм.гггг);
- тарифный план (строка 15 символов).

Формат ввода: *Номер ФИО владельца Дата подключения «Тарифный план»*

ВИДЕОДИСК:

- серийный номер (целое десятизначное число)
- название (строка 20 символов);
- продолжительность (целое число, в минутах);
- кинокомпания (строка 15 символов);
- жанр (0 – боевик, 1 – комедия, 2 – драма и т.д.).

Формат ввода: *Серийный номер «Название фильма» Кинокомпания (Жанр), Продолжительность*

ПОЕЗД:

- номер (целое трехзначное число)
- название (строка 20 символов);
- станция отправления (строка 20 символов);
- станция назначения (строка 20 символов);
- количество вагонов (целое число меньше 30).

Формат ввода: *Номер «Название» (Станция отправления – Станция назначения), Количество вагонов*

УСТРОЙСТВО:

- тип устройства (строка 15 символов);
- модель (строка 10 символов);
- фирма производитель (строка 15 символов);
- серийный номер (цифро-символьный код длиной 20 знаков);
- дата изготовления (в формате дд.мм.гггг).

Формат ввода: *Тип устройства Модель (Фирма), Номер, Дата изготовления*

### Контрольные вопросы

1. Какая библиотека языка C++ используется для консольного ввода и вывода?
2. Какой объект ассоциирован со стандартным потоком ввода в языке C++?
3. Какой объект ассоциирован со стандартным потоком вывода в языке C++?
4. В чем заключается отличие в описании структур, объединений и перечислений в языке C++?
5. Опишите синтаксис передачи параметров по ссылке в языке C++.
6. С помощью какого оператора осуществляется выделение динамической памяти в C++?



7. С помощью какого оператора осуществляется освобождение динамической памяти в C++?
8. Как осуществляется выделение динамической памяти под массив в языке C++?
9. Как в C++ осуществляется освобождение динамической памяти, занимаемой массивом?
10. Как в языке C++ описываются параметры со значением по умолчанию?
11. Как в языке C++ могут вызываться функции, у которых есть параметры со значением по умолчанию?
12. Что такое перегрузка функций, как ее правильно применять?