

ЛАБОРАТОРНАЯ РАБОТА № 4.

БИБЛИОТЕКИ ВВОДА-ВЫВОДА.

ЦЕЛЬ РАБОТЫ: Изучение реализации в языке C++ возможностей ввода-вывода файлов.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ РАБОТЫ

Потоки ввода/вывода

В C++ ввод-вывод данных производится потоками байтов. Поток – это последовательность байтов. При операциях ввода байты направляются от устройства (клавиатуры, дисковод, сетевой платы) в ОП. Прикладные программы связывают с этими байтами некоторый смысл – графические изображения, мелодия, цифровое видео.

Большинство программ C++ включает файл заголовка `<iostream>`, который содержит основную информацию, необходимую операциям потокового ввода-вывода. В частности, определяются объекты `cin`, `cout`, `cerr`, `clog`, которые связываются со стандартным потоком ввода, вывода, вывода ошибок (небуферизированным и буферизированным).

Заголовок `<iomanip>` содержит информацию, необходимую для управления форматированным вводом-выводом с помощью параметризованных манипуляторов потока.

Функции библиотеки `iostream`

В библиотеке **`iostream`** для проверки данных, вводимых пользователем, используются специальные методы класса **`ios`** и двух его наследников **`istream`** и **`ostream`**. Всего в этих классах находится порядка 25 методов, позволяющих получить информацию о состоянии объектов и управлять их поведением. Вот некоторые из них.

имя функции	действие
<code>int good ()</code>	возвращает 1, если ошибок не обнаружено
<code>int eof ()</code>	возвращает 1, если поток находится в состоянии "конец файла"
<code>int fail ()</code>	возвращает 1, если обнаружена восстановимая ошибка ввода-вывода (обычно, ошибка преобразования данных)
<code>int bad ()</code>	возвращает 1, если обнаружена невосстановимая ошибка ввода-вывода
<code>int clear ()</code>	сбрасывает состояние ошибки ввода-вывода
<code>int precision (int i)</code>	устанавливает точность вывода чисел с плавающей точкой
<code>int width (int i)</code>	устанавливает ширину поля вывода

Файловый ввод-вывод

Файловый ввод-вывод обрабатывается с помощью библиотеки `fstream.h`. Для управления файловыми потоками используются потоки `ifstream` для входного файлового потока и `ofstream` для выходного.

Объявление и открытие файла можно выполнить следующим образом:

```
ifstream f_in("c:\\test.txt");
```

Аналогично может быть открыт выходной поток.

Проверка того, что файл открыт успешно, производится следующим образом:

```
if (!f_in) cout<<"файл невозможно открыть";
```

т.е. после попытки открытия файловая переменная содержит информацию о том, успешна или нет операция открытия файла.

Пример. В файл `f` записать сначала элементы файла `g`, затем элементы файла `h`. Файлы текстовые.

```
#include <iostream>
#include <conio.h>
#include <fstream.h>
int main() {
    const int size=20;
    char buffer[size];

    ifstream f_in("d:\\g.txt");
    if (!f_in) cout<<"Файл g невозможно открыть";
    f_in.get(buffer, size); // чтение из файла g в буфер
    f_in.close(); // закрыть файл g

    ofstream f_out("d:\\f.txt");
    if (!f_out) cout<<"Файл f невозможно открыть";
    f_out.write(buffer, f_in.gcount()); // запись из буфера в файл f

    f_in.open("d:\\h.txt");
    if (!f_in) cout<<"Файл h невозможно открыть";
    f_in.get(buffer, size);
    f_in.close();

    f_out.write(buffer, f_in.gcount());
    f_out.close();
}
```

```
getch();  
return 0;  
}
```

Примечание. При выполнении заданий к лабораторной работе необходимо осуществлять контроль вводимых данных.

СОДЕРЖАНИЕ ОТЧЕТА:

1. Название работы.
2. Цель работы.
3. Задание на работу. Иерархия и назначение классов C++, обеспечивающих ввод-вывод.
4. Листинг программы.
5. Распечатка результатов работы программы.
6. Выводы.

Варианты заданий к лабораторной работе № 4 по курсу "ООП"

Простые (8 баллов)

1. Ввести массив чисел, записать его в текстовый файл, прочитать его оттуда, увеличить каждое число вдвое и вывести в другой файл.
2. Переписать из файла d в файл s все летние даты, а в файл w—все зимние даты.
3. Дан файл с непустой последовательностью слов, содержащих от 1 до 8 букв. Напечатать все слова наименьшей длины.
4. Определить, имеет ли файл нечетную длину, и, если имеет, вывести средний элемент этого файла.
5. Дан текстовый файл. Удвоить в нем все цифры "0" и "1" и записать в другой файл.
6. Дан файл с текстом. Определить число слов, которые начинаются и заканчиваются на одну и ту же букву.
7. Дан файл с текстом. Вывести текст, пронумеровав слова. Записать результат в другой файл.
8. Дан файл с текстом. Определить, сколько слов состоит из одинаковых букв.
9. Дан файл с текстом. Записать в файл a0.txt слова, которые содержат цифру 0, а в файл a1.txt - все остальные слова.
10. Переписать из файла a0.txt в файл a1.txt все числа, большие среднего арифметического, а в файл a2.txt – все остальные числа.
11. Файл содержит значения времени. Переписать из него в файл a1.txt все значения, соответствующие утреннему времени, в файл a2.txt – вечернему времени, в файл a3.txt - все остальные значения.
12. Определить количество пустых строк в текстовом файле. Найти максимальную длину строки текстового файла.
13. Определить количество строк текстового файла, которые начинаются и заканчиваются на один и тот же символ.

14. Считая, что текстовый файл разбит на строки длиной не больше 80, сформировать другой файл, дополнив короткие строки файла до 80 символов заполнителем.

15. Дан файл с последовательностью слов, содержащих от 1 до 12 букв. Напечатать все слова наибольшей длины.

Сложные (10 баллов)

16. Дан файл с текстом. Проверить, сколько раз в тексте встречается какой-либо отрезок ряда 0123456789 длины не меньшей, чем 2. Записать все найденные отрезки в другой файл.

17. Имеется текстовый файл следующей структуры:

N
K
 $a_1 a_2 \dots a_k$
...

Здесь N – число блоков данных, K – количество элементов данных в блоке, a_i – значение элемента данных.

a_i представляют собой целые числа в пределах от 0 до 99.

Найти среднее арифметическое всех данных, максимальный элемент данных и его позицию.

18. Имеется текстовый файл следующей структуры:

N
K
 $a_1 a_2 \dots a_k$
...

Здесь N – число блоков данных, K – количество элементов данных в блоке, a_i – значение элемента данных.

a_i представляют собой целые числа в пределах от 0 до 99.

Поменяйте блоки данных местами таким образом, чтобы они располагались по возрастанию максимального элемента данных блока. Сформируйте выходной файл.

19. Имеется текстовый файл следующей структуры:

N
K
 $a_1 a_2 \dots a_k$
...

Здесь N – число блоков данных, K – количество элементов данных в блоке, a_i – значение элемента данных.

a_i представляют собой слова.

Определите, сколько в каждом блоке данных слов-палиндромов.

20. Имеется текстовый файл следующей структуры:

N

K

$a_1 a_2 \dots a_k$

...

Здесь N – число блоков данных, K – количество элементов данных в блоке, a_i – значение элемента данных.

a_i представляют собой слова.

Определите в каждом блоке слово максимальной длины и переставьте блоки данных по убыванию слов максимальной длины. Сформируйте выходной файл.

21. Имеется текстовый файл следующей структуры:

N

$a_1 a_2 b_1 b_2 c_1 c_2 \dots c_1 c_2$

Здесь N – число точек на плоскости, $(a_1 a_2)$ – координаты первой точки, $(b_1 b_2)$ – координаты второй точки, и т.д. Координаты представляют собой целые числа.

Расположите координатные пары по возрастанию расстояния от начала координат. Сформируйте выходной файл.

22. Имеется текстовый файл следующей структуры:

N

K

$a_1 a_2 \dots a_k$

...

Здесь N – число блоков данных, K – количество элементов данных в блоке, a_i – значение элемента данных.

a_i представляют собой слова, состоящие из букв и цифр.

Определите количество блоков, удовлетворяющих следующему условию: текст блока является десятичной записью целого числа, кратного 2.

Переставьте блоки данных так, чтобы найденные блоки были первыми.

Сформируйте выходной файл.

23. Имеется текстовый файл следующей структуры:

```
N
K
a1 a2 ... ak
...
```

Здесь N – число блоков данных, K – количество элементов данных в блоке, a_i – значение элемента данных.

a_i представляют собой слова.

Определите количество блоков, удовлетворяющих следующему условию. Блок содержит только одну цифру, причем ее числовое значение равно длине блока.

Переставьте блоки данных так, чтобы найденные блоки были первыми. Сформируйте выходной файл.