

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА
Институт Информационных Технологий
Кафедра Промышленной Информатики



ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ

Тема практических занятий «Циклические алгоритмы. Работа с файлами в C++»

Лектор **Каширская Елизавета Натановна** (к.т.н., доцент, ФГБОУ ВО "МИРЭА - Российский технологический университет") e-mail: liza.kashirskaya@gmail.com

Домашнее задание № 3



1. «Заем».

Рассчитать месячную выплату m по займу в S рублей на n лет под процент p .

Вычисляется по формуле: $m = \frac{Sr(1+r)^n}{12((1+r)^n - 1)}$, где $r = \frac{p}{100}$ Дано: S, p, n . Найти: m .

(повторение материала: конструкции ветвления)

2. «Ссуда».

Рассчитать под какой процент p выдана ссуда величиной S рублей, которая гасится месячными выплатами величиной m в течение n лет. Дано: S, m, n . Найти: p . Формула приведена в предыдущей задаче. P.S. Решать задачу нужно методом перебора значений через цикл. (повторение материала: конструкции ветвления, циклы)

3. «Копирование файла».

Создать на диске текстовый файл в программе и вывести содержимое файла на экран. P.S. Сохранять файл можно в любом месте (диск/папка) (новый материал: работа с файлами)

4. «Фильтр».

Вывести на экран только числа из созданного Вами на диске текстового файла, содержащего буквы и цифры. P.S. Фильтрация осуществляется при помощи кодировки ASCII (новый материал: работа с файлами)

5. «Сортировка букв».

Задать строку из 30 букв и расставить их в алфавитном порядке. P.S. Можно использовать любой алгоритм сортировки. (повторение материала: методы сортировки)



Как вы уже знаете, в программировании применяются циклы трех видов с параметром, с предусловием и с постусловием, причем два последних принципиально не отличаются (смотрите лекцию 4).



Усеченный конус как физическое тело должен существовать!

Не при всех сочетаниях размеров его можно построить. Обратите на это внимание!



Контрольные значения:

$$S = 100 \text{ руб.}$$

$$n = 1 \text{ год.}$$

$$p = 20 \%$$

$$m = 10 \text{ руб.}$$



В цикле от $p = 1\%$ до $p = 100\%$ методом последовательных приближений получать на каждом шаге значение m и сравнивать его с исходным m_0 .

Процесс вести до тех пор, пока не выполнится условие

$$|m - m_0| < \epsilon ps,$$

где ϵps – допустимая погрешность, например, 1 рубль или 1 копейка.



Для работы с файлами необходимо подключить заголовочный файл `<fstream>`. (библиотека: `#include <fstream>`) В `<fstream>` определены несколько классов и подключены заголовочные файлы `<ifstream>` - файловый ввод и `<ofstream>` - файловый вывод.

Файловый ввод/вывод аналогичен стандартному вводу/выводу, единственное отличие - это то, что ввод/вывод выполнятся не на экран, а в файл. Если ввод/вывод на стандартные устройства выполняется с помощью объектов `cin` и `cout`, то для организации файлового ввода/вывода достаточно создать собственные объекты, которые можно использовать аналогично операторам `cin` и `cout`.

Чтобы создать текстовый файл и записать в него строку необходимо проделать следующие шаги: 1. создать объект класса `ofstream`; 2. связать объект класса с файлом, в который будет производиться запись; 3. записать строку в файл; 4. закрыть файл.



Почему необходимо создавать объект класса ofstream, а не класса ifstream?

Потому, что нужно сделать запись в файл, а если бы нужно было считать данные из файла, то создавался бы объект класса ifstream.

Создаём объект для записи в файл, назовём объект - fout.

```
ofstream fout; // объект класса ofstream
```

Для чего нам объект? Объект необходим, чтобы можно было выполнять запись в файл. Уже объект создан, но не связан с файлом, в который нужно записать строку.

```
fout.open("cppstudio.txt"); // связываем объект с файлом
```

Через операцию «точка» получаем доступ к методу класса open(), в круглых скобках которого указываем имя файла. Указанный файл будет создан в текущей директории с программой. Если файл с таким именем существует, то существующий файл будет заменен новым. Итак, файл открыт, осталось записать в него нужную строку.

```
fout << "Работа с файлами в C++"; // запись строки в файл
```

Используя операцию передачи в поток совместно с объектом fout строка «Работа с файлами в C++» записывается в файл. Так как больше нет необходимости изменять содержимое файла, его нужно закрыть, то есть отделить объект от файла.

```
fout.close(); // закрываем файл
```




Шаги 1 и 2 можно объединить, то есть в одной строке создать объект и связать его с файлом.

```
ofstream fout("cppstudio.txt"); // создаём объект класса ofstream и связываем его с файлом cppstudio.txt
```

Также мы можем указать директорию, где хотим создать файл или прочитать его.

```
ofstream fout("d:\\program\\cppstudio.txt");
```

```
#include <iostream>
```

```
#include <fstream>
```

```
using namespace std;
```

```
int main () {
```

```
ofstream fout("cppstudio.txt");
```

```
fout << "Работа с файлами в C++";
```

```
fout.close();
```

```
return 0;
```

```
}
```



Осталось проверить правильность работы программы, а для этого открываем файл `cppstudio.txt` и смотрим его содержимое, должно быть: «Работа с файлами в C++.»

Для того чтобы прочитать файл понадобится выполнить те же шаги, что и при записи в файл с небольшими изменениями:

1. создать объект класса `ifstream` и связать его с файлом, из которого будет производиться считывание;
2. прочитать файл;
3. закрыть файл;
4. вывести считанную информацию на экран.

```
#include <iostream>
```

```
#include <fstream>
```

```
using namespace std;
```

```
int main() {
```

```
setlocale(0, ""); // корректное отображение Кириллицы
```

```
char buff[20]; // буфер промежуточного хранения для текста из файла
```

```
ifstream fin("cppstudio.txt"); // открыли файл для чтения
```

```
fin >> buff; // считали первое слово из файла
```

```
cout << buff << endl; // напечатали слово
```

```
fin.getline(buff, 20); // считали строку из файла
```

```
fin.close(); // закрываем файл
```

```
cout << buff << endl; // вывод строки
```

```
return 0;
```

```
}
```



В программе показаны два способа чтения из файла, первый – используя операцию передачи в поток, второй - используя функцию `getline()`. В первом случае считывается только первое слово, а во втором случае считывается строка, длиной 20 символов. Но так как в файле осталось меньше 20 символов, то считываются символы включительно до последнего. Обратите внимание на то, что считывание во второй раз продолжилось, после первого слова, а не с начала, так как первое слово было прочитано ранее.

Программа сработала правильно, но не всегда так бывает, даже в том случае, если с кодом всё в порядке. Например, в программу передано имя несуществующего файла или в имени допущена ошибка. Что тогда? В этом случае ничего не произойдёт вообще. Файл не будет найден, а значит и прочитать его не возможно. Поэтому компилятор проигнорирует строки, где выполняется работа с файлом. В результате корректно завершится работа программы, но ничего, на экране показано не будет.

Казалось бы это вполне нормальная реакции на такую ситуацию. Но простому пользователю не будет понятно, в чём дело и почему на экране не появилась строка из файла. Так вот, чтобы всё было предельно понятно в C++ предусмотрена такая функция - `is_open()`, которая возвращает целые значения: 1 - если файл был успешно открыт, 0 - если файл открыт не был.



```
#include <iostream>  
#include <fstream>  
using namespace std;  
  
int main() {  
setlocale(0, "r");  
char buff[40];  
ifstream fin("cppstudio.doc"); // (ВВЕЛИ НЕ КОРРЕКТНОЕ ИМЯ ФАЙЛА)  
if (!fin.is open()) // если файл не открыт  
cout << "Файл не может быть открыт!\n"; // сообщить об этом  
else  
{  
fin >> buff;  
cout << buff << endl;  
fin.getline(buff, 40);  
fin.close();  
cout << buff << endl;  
}  
return 0;  
}
```



1. Процедурное программирование на языках СИ и С++ : учебно-методическое пособие / Л. А. Скворцова [и др.]. — М.: РТУ МИРЭА, 2018. — 238 с.
[Электронный ресурс]. Режим доступа: <https://library.mirea.ru/books/53585>
2. Трофимов В.В., Павловская Т.А. Алгоритмизация и программирование: учебник для академического бакалавриата. М.: Издательство Юрайт, 2017.
[Электронный ресурс]. Режим доступа:
<https://www.intuit.ru/studies/courses/16740/1301/info>
3. Уроки С++ с нуля. [Электронный ресурс]. Режим доступа: <https://code-live.ru/tag/cpp-manual>
4. Введение в языки программирования Си С++. [Электронный ресурс]. Режим доступа: <https://www.intuit.ru/studies/courses/1039/231/info>