Программирование на C++

Минцифры России

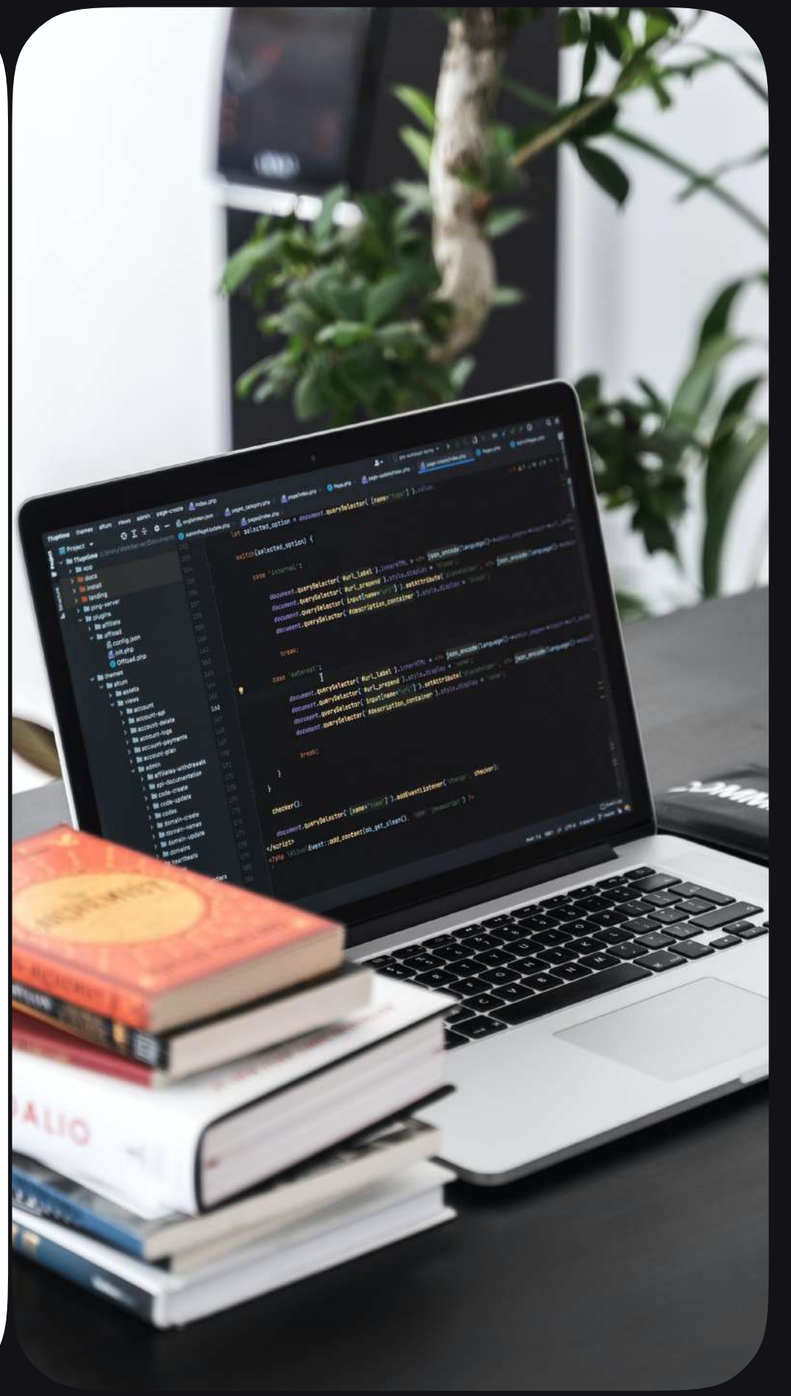
UCHi DOMA

20.35 УНИВЕРСИТЕТ

Урок 4 Модуль 2

Функции работы с файлами

Полезные материалы



Цели урока

- изучить работу с файлами
- отработать на практике составление алгоритмов, считывающих данные из файла и записывающих информацию в файл на Си

Файл

Файл — именованный набор байтов, который может быть сохранен на некотором накопителе.

Данные, содержащиеся в файлах, имеют самый разнообразный характер:

- программы
- тексты
- графические изображения
- исходные данные для работы программ или результаты выполнения программ
- аудио
- видео данные и т. п.

Каталог (папка, директория) — именованная совокупность байтов на носителе информации, содержащая название подкаталогов и файлов, используется в файловой системе для упрощения организации файлов.

Работа с файлами

1. Открыть файл, для того, чтобы к нему можно было обращаться. Соответственно, открывать можно для чтения, записи, чтения и записи, переписывания или записи в конец файла и т.п.
2. Непосредственно работа с файлом — запись и чтение. Работа производится не с памятью с произвольным доступом, а с буферизированным потоком.
3. Закрыть файл.

Работа с файлами

Для программиста открытый файл представляется как последовательность считываемых или записываемых данных. При открытии файла с ним связывается **поток ввода-вывода.** Выводимая информация записывается в поток, вводимая информация считывается из потока.

Когда поток открывается для ввода-вывода, он связывается со стандартной структурой типа **FILE**, которая определена в **stdio.h.** Структура **FILE** содержит необходимую информацию о файле.

Открытие файла

Для открытия файла используется функция **fopen()**, которая возвращает указатель на структуру типа **FILE**, который можно использовать для последующих операций с файлом.

```
FILE *fopen(name, type);
```

`name`	имя открываемого файла (включая путь)
`type`	указатель на строку символов, определяющих способ доступа к файлу:

`"r"`	открыть файл для чтения (файл должен существовать)
`"w"`	открыть пустой файл для записи; если файл существует, то его содержимое теряется
`"a"`	открыть файл для записи в конец (для добавления); файл создается, если он не существует
`"r+"`	открыть файл для чтения и записи (файл должен существовать)
`"w+"`	открыть пустой файл для чтения и записи; если файл существует, то его содержимое теряется
`"a+"`	открыть файл для чтения и дополнения, если файл не существует, то он создаётся

Возвращаемое значение — указатель на открытый поток. Если обнаружена ошибка, то возвращается значение NULL.

Закрытие файла

Функция **fclose()** закрывает поток или потоки, связанные с открытыми при помощи функции **fopen()** файлами. Закрываемый поток определяется аргументом функции **fclose().**

Возвращаемое значение

- значение 0, если поток успешно закрыт
- константа **EOF**, если произошла ошибка

Пример

```c
#include <stdio.h>
int main()
{
  FILE *fp;
  char name[] = "name.txt";
  if ((fp = fopen(name, "r")) == NULL)
{
  printf("Не удалось открыть файл");
  return 0;
}
// открыть файл удалось
... // требуемые действия над данными
fclose(fp);
return 0;
}
```

Чтение символа из файла

```
fgetc(поток);
```

Аргументом функции является указатель на поток типа **FILE**. Функция возвращает код считанного символа. Если достигнут конец файла или возникла ошибка, возвращается константа **EOF**.

Запись символа в файл

```
fputc(символ,поток);
```

Аргументами функции являются символ и указатель на поток типа **FILE.**

Чтение строки из файла

```
fgets(УказательНаСтроку, КоличествоСимволов, поток);
```

Символы читаются из потока до тех пор, пока не будет прочитан символ новой строки «**\n**», который включается в строку, или пока не наступит конец потока **EOF** или не будет прочитано максимальное количество символов. Результат помещается в указатель на строку и заканчивается нуль-символом «**\0**». Функция возвращает адрес строки.

Чтение строки из файла

```
fputs(УказательНаСтроку,поток);
```

Копирует строку в поток с текущей позиции.
Завершающий нуль-символ не копируется.

Запись в файл и чтение из файла

Функции **fscanf()** и **fprintf()** аналогичны функциям **scanf()** и **printf()**, но работают с файлами данных, и имеют первый аргумент — указатель на файл.

```
fscanf(поток, "ФорматВвода", аргументы);
```

```
fprintf(поток, "ФорматВывода", аргументы);
```

Пример

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
int main()
{
  FILE* S1, * S2;
  int x, y;
  system("chcp 1251");
  system("cls");
  printf("Введите число : ");
  scanf("%d", &x);
  S1 = fopen("S1.txt", "w");
  fprintf(S1, "%d", x);
  fclose(S1);
  S1 = fopen("S1.txt", "r");
  S2 = fopen("S2.txt", "w");
  fscanf(S1, "%d", &y);
  y *= 3;
  fclose(S1);
  fprintf(S2, "%d\n", y);
  fclose(S2);
  return 0;
}
```

S1 — Блокнот

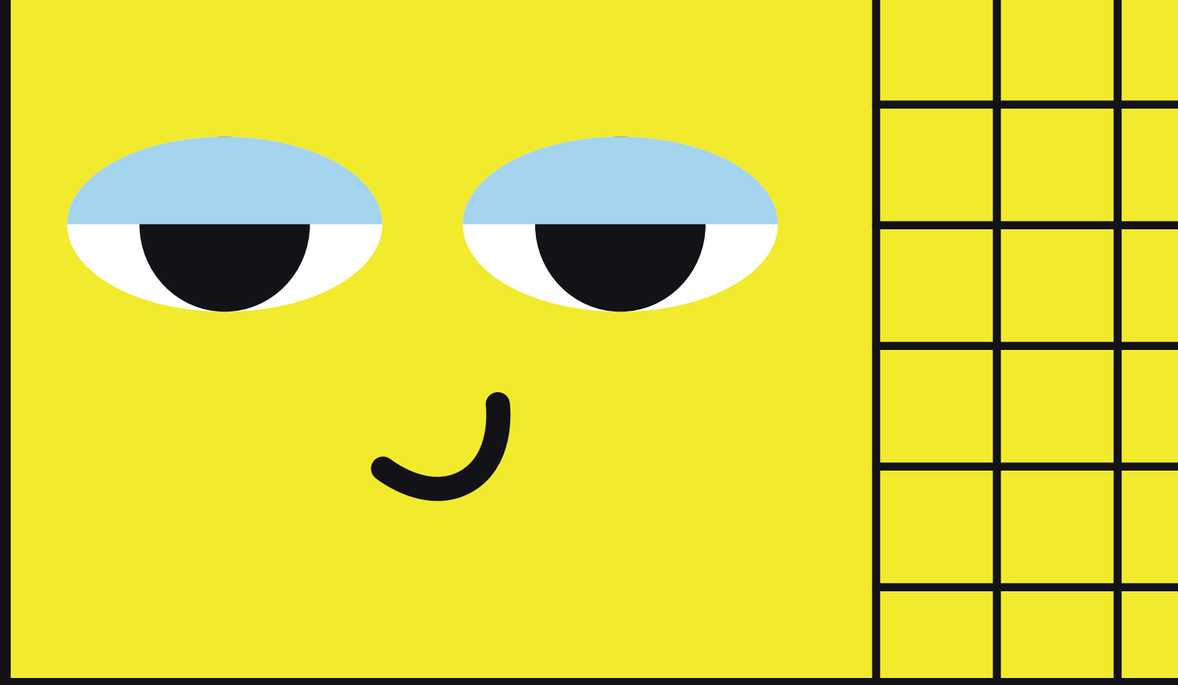
Файл Правка Формат Вид Справка

44

S2 — Блокнот

Файл Правка Формат Вид Справка

132



Практика

Задачи

1	Откройте файл «f1.txt», считайте данные из файла (целое число), возведите считанное значение во 2 степень и выведите на экран.
2	В файл записаны целые числа, по одному числу в строке. Считайте из файла «f1.txt» построчно все данные. Выведите на экран сумму считанных чисел.
3	В файл записаны целые числа, по одному числу в строке. Считайте из файла «f1.txt» построчно все данные. Удвойте значение каждого считанного значения и запишите в обратном порядке в файл «f2.txt».
4	В файле «f1.txt» записан текст. Считать текст, определить сколько раз в считаном тексте встречается символ «s», вывести значение на экран.
5	Дан файл «f.txt». Создать два файла «f1.txt» и «f2.txt», записав в первый из них все четные числа, а во второй — все нечетные (каждое число в отдельной строке).
6	В файле «f.txt» записан текст. Перезаписать текст в файл «f1.txt» разделяя на строки, переходить на новую строку после каждого символа «s».

Задача 1

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
int main()
{
  FILE* input=NULL;
  int y;

  input = fopen("f1.txt", "r");
  if (input == NULL)
{
  printf("Error opening file");
  exit(0);
}
fscanf(input, "%d", &y);
y *= y;
fclose(input);
printf("%d\n", y);
return 0;
}
```

Задача 2

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
int main() {
  FILE* input=NULL;
  int c;
  int sum = 0;
  input = fopen("f1.txt", "r");
  if (input == NULL)
{
  printf("Error opening file");
  exit(0);
}
while (fscanf(input, "%d", &c) == 1)
  sum += c;
fclose(input);
printf("%d", sum);
return 0;
}
```

Задача 3

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
int main()
{
  FILE* input=NULL;
  FILE* output=NULL;
  int c;

  input = fopen("f1.txt", "r");
  output = fopen("f2.txt", "w");
  if (input == NULL)
{
  printf("Error opening file f1.txt");
  exit(0);
}
while (fscanf(input, "%d", &c) == 1)
{
  c *= 2;
  fprintf(output, "%dn\n", c);
}
fclose(input);
fclose(output);
return 0;
}
```

Задача 4

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
int main()
{
  FILE* input = NULL;
  int c;
  int count=0;

  input = fopen("f1.txt", "r");
  if (input == NULL)
{
  printf("Error opening file f1.txt");
  exit(0);
}
do
{
  c = fgetc(input);
  if (c == ‘s’) count++;
} while (c != EOF);
fclose(input);
printf("%d", count);
return 0;
}
```

Задача 5

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
int main()
{
  FILE* input = NULL;
  FILE* output1 = NULL;
  FILE* output2 = NULL;
  int c;
  input = fopen("f.txt", "r");
  output1 = fopen("f1.txt", "w");
  output2 = fopen("f2.txt", "w");

  if (input == NULL)
{
  printf("Error opening file f1.txt");
  exit(0);
}
while (fscanf(input, "%d", &c) == 1)
  if (c%2==0) fprintf(output1,"%d\n",c);
  else fprintf(output2, "%d\n", c);
fclose(input);
fclose(output1);
fclose(output2);
return 0;
}
```

Задача 6

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
int main()
{
  FILE* input = NULL;
  FILE* output = NULL;
  int c;
  input = fopen("f1.txt", "r");
  output = fopen("f2.txt", "w");
  if (input == NULL)
{
  printf("Error opening file f1.txt");
  exit(0);
}
do
{
  c = fgetc(input);
  fprintf(output,"%c",c);
  if (c == ‘s’) fprintf(output,"\n");
} while (c != EOF);
fclose(input);
fclose(output);
return 0;
}
```

Закрепление

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
int main()
{
  FILE* input = NULL;
  int c;
  int sum = 0;

  input = fopen("text.txt", "rt");
  if (input == NULL) {
    printf("Error opening file");
    exit(0);
}
while (fscanf(input, "%d", &c) == 1)
    sum += c;
fclose(input);
printf("%d", sum);
return 0;
}
```

Что будет выведено на экран в результате работы экрана?

text — Блокнот

Файл Правка Формат Вид Справка

1

2

3

Закрепление

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
int main()
{
  FILE* input = NULL;
  int c;
  int sum = 0;

  input = fopen("text.txt", "rt");
  if (input == NULL) {
    printf("Error opening file");
    exit(0);
}
while (fscanf(input, "%d", &c) == 1)
    sum += c;
fclose(input);
printf("%d", sum);
return 0;
}
```

Что будет выведено на экран в результате работы экрана?

Результат работы программы

6