Лекція №11. Файли. Строки. Обробка виключень

ПРОГРАМУВАННЯ ТА ПРИКЛАДНІ ІНФОРМАЦІЙНІ СИСТЕМИ

Визначення

Файлом називають спосіб зберігання інформації на фізичному пристрої. **Файл** - це поняття, яке може бути застосовано до всього - від файлу на диску до терміналу.

У C ++ відсутні оператори для роботи з файлами. Всі необхідні дії **виконуються за допомогою функцій, включених в стандартну бібліотеку**. Вони дозволяють працювати з різними пристроями, такими, як диски, принтер, комунікаційні канали і т.д. Ці пристрої сильно відрізняються один від одного. Однак файлова система перетворює їх в єдиний абстрактний логічний пристрій, який називається **потоком**.

Текстовий потік - це послідовність символів. При передачі символів з потоку на екран, частина з них не виводиться (наприклад, символ повернення каретки, переведення рядка).

Двійковий потік - це послідовність байтів, які однозначно відповідають тому, що знаходиться на зовнішньому пристрої.

Файловий ввід-вивід з використанням потоків

Бібліотека потокового введення-виведення

```
#include <fstream>
```

Зв'язок файлу з потоком виведення

```
ofstream <ім'я логічного файлу>;
```

Зв'язок файлу з потоком введення

```
ifstream <ім'я логічного файлу>;
```

Відкриття файлу

```
<ім'я логічного файла>.open(<ім'я фізичної файлу>);
```

Закриття файлу

```
<ім'я логічного файла>.close();
```

Приклад 1. Заповнити файл значеннями функції y = x * cos x.

```
#include <fstream>
void main(){
    double a, b, h, x; char s[20];
    cout << "Enter the beginning and end of the segment, step: ";
    cin >> a >> b >> h;
    cout << "File name? "; cin >> s;
    ofstream f;
    f.open(s);
    for (x = a; x <= b; x += h){
        f << x;
        f <<" "<< fun(x) << endl;
    }
    f.close();
    system("PAUSE");
}
double fun(double x){
    return x*cos(x);
}
```

Режими відкриття файлів

Константа	Опис
ios_base::in	відкрити файл для читання
ios_base::out	відкрити файл для запису
ios_base::ate	при відкритті перемістити покажчик в кінець файлу
ios_base::app	відкрити файл для запису в кінець файлу
ios_base::trunc	видалити вміст файлу, якщо він існує
ios_base::binary	відкриття файлу в двійковому режимі

```
ofstream fout("cppstudio.txt", ios_base::app);
// відкриваємо файл для додавання інформації до кінця файлу
fout.open("cppstudio.txt", ios_base::app);
// відкриваємо файл для додавання інформації до кінця файлу
```

Основні методи ifstream

Метод	Опис
open	Відкриває файл для читання
get	Читає один або більше символів з файлу
getline	Читає символьний рядок з текстового файлу або дані з бінарного файлу до певного обмежувача
read	Зчитує задане число байт з файлу в пам'ять
eof	Повертає нульове значення (true), коли покажчик потоку досягає кінця файлу
peek	Видає черговий символ потоку, але не вибирає його
seekg	Переміщує покажчик позиціонування файлу в задане положення
tellg	Повертає поточне значення покажчика позиціонування файлу
close	закриває файл

Основні методи ofstream

Метод	Опис
open	Відкриває файл для запису
put	Записує одиночний символ в файл
write	Записує задане число байт з пам'яті в файл
seekp	Переміщує покажчик позиціонування в зазначене положення
tellp	Повертає поточне значення покажчика позиціонування файлу
close	Закриває файл

Приклад 2. У заданому файлі цілих чисел підрахувати кількість компонент, кратних 3.

```
void main(){
    int r, ch;
    ifstream f;
    f.open("CH_Z.TXT");
    ch = 0;
    for (; f.peek() != EOF;){
        f >> r;
        cout << r << " ";
        if (r % 3 == 0) ch++;
    }
    f.close();
    cout << endl << "Answer: " << ch;
    system("PAUSE");
}
```

cstdio

Доступ до файлів	
fclose	Від'єднати потік і закрити файл
fopen	Відкрити файл
freopen	Перенаправлення потоків введення / виведення
Символи введення / виведення	
fgetc	Повертає символ на який посилається внутрішній індикатор позиції файлу зазначеного потоку
fgets	Зчитує символи з потоку і зберігає їх у вигляді рядка
fputc	Записує символ в потік і переміщує позицію індикатора положення
fputs	Записує рядок, зазначену в параметрі в потік
Читання / запис потоків	
fread	Зчитати блок даних з файлу
fwrite	Записати в файл блок даних
Позиціонування по файлу	
fgetpos	Отримати значення поточного положення в файлі
fseek	Зміна позиції внутрішнього покажчика положення в файлі, щодо деякого положення.
fsetpos	Зміна позиції внутрішнього покажчика положення в файлі.
ftell	Отримати значення покажчика поточного положення потоку.
Обробка помилок	
feof	Функція-індикатор кінця файлу, визначає кінець файлу.
ferror	Функція-індикатор помилок, відловлює помилки, пов'язані з обробкою потоків.

Тип Доступу

- "r" відкрити файл для читання.(Цей файл повинен існувати).
- "w" відкрити порожній файл для; якщо цей файл раніше існував, його зміст видаляється
- "a" відкрити файл для запису (додавання) в кінець. Якщо даного файлу не існує, він спочатку створюється.
- "r+" відкрити файл одночасно для читання та запису. Файл повинен існувати.
- "w+" відкрити порожній файл для читання та запису. Якщо цей файл раніше існував, його зміст видаляється
- "a+" відкрити файл для читання та додавання. Якщо даного файлу не існує, він спочатку створюється.

Спосіб перетворення символу нової строчки

”t” відкрити в текстовому (перетворюючому) режимі; при введені комбінація “Повернення каретки - переведення строки" перетворюється до єдиного символу "переведення строки". При виводі символ переведення строки перетворюється в комбінацію ВК-ПС.

“в” відкрити в двоїчному (не перетворюючому) режимі; вище згадані перетворення не здійснюються.

Приклад 3. Скопіювати дані з одного файлу в інший

```
void main() {
    FILE *in, *out;
    char f1[] = "INPUT.TXT";
    char f2[] = "OUTPUT.TXT";
    in = fopen(f1, "rt");
    out = fopen(f2, "wt");

    while (!feof(in))
        fputc(fgetc(in), out);
    fclose(in);
    fclose(out);
}
```

Обробка винятків в C ++

- **try** (намагатися) - початок блоку винятків;
- **catch** (зловити) - початок блоку, "ловить" виключення;
- **throw** (кинути) - ключове слово, що "створює" ("збуджує") виняток.

Приклад 4. Простий приклад обробки винятків

```
void func(){
    try{
        throw 1;
    }
    catch (int a){
        cout << "Caught exception number:  " << a << endl;
        return;
    }
    cout << "No exception detected!" << endl;
    return;
}
```

Приклад 5. Скопіювати дані з одного файлу в інший, якщо файл існує

```
FILE *safe_fopen(char const *path, char const *mode){
    FILE *f = fopen(path, mode);
    if (f == NULL)
        throw "file not found";
    return f;
}
```

```
int main() {
    FILE *in, *out;
    char f1[] = "INPUT2.TXT";
    char f2[] = "OUTPUT.TXT";
    try {
        in = safe_fopen(f1, "rt");
    }
    catch(char *a){
        cout << a; return 0;
    }
    try {
        out = safe_fopen(f2, "wt");
    }
    catch(char *a){
        cout << a; return 0;
    }
    while (!feof(in))
        fputc(fgetc(in), out);
    fclose(in);
    fclose(out);
    system("PAUSE");
return 0;
}
```

Стандартні функції опрацювання масивів символів (String.h)

strlen(<рядок>) - визначає фактичну кількість символів у рядку, застосовується у виразах;

strcat(r1,r2) - команда з'єднання рядків r1, r2 в один рядок, результат присвоює змінній r1;

strncat(r1, r2, n) - до змінної r1 додає перших n символів рядка r2;

strcpy(r1, r2) - копіює символи з рядка r2 в рядок r1;

strncpy(r1, r2, n) - копіює перших n символів рядка r2 в рядок r1;

strchr(r1, <символ>) - визначає перше входження деякого символу у рядок r1 так: повертає рядок, який починається від першого входження заданого символу до кінця рядка r1, застосовується у виразах;

strrchr(r1, <символ>) - визначає останнє входження заданого символу у рядок, застосовується у виразах;

Стандартні функції опрацювання масивів символів (String.h)

strspn(r1, r2) - визначає номер першого символу, який входить у рядок r1, але не входить у рядок r2, застосовується у виразах;

strstr(r1, r2) - визначає в рядку r1 підрядок, що починається з першого входження рядка r2 у рядок r1, застосовується у виразах;

strlok(r1,r2) - визначає частину рядка r1, яка закінчується перед першим однаковим символом рядків r1 та r2;

strnset(r1, <символ>, n) - вставляє n разів заданий символ перед рядком r1, застосовується у виразах;

strupr(r1) - перетворює усі малі літери рядка у великі;

strlwr(r1) - перетворює усі великі літери рядка у малі;

strrev(r1) - записує рядок у зворотному порядку.