# Лабораторная работа № 6 Тема. Потоки, обработка исключительных ситуаций в C++

**Теоретическое введение.** В С++ ввод-вывод осуществляется через потоки. Потоки являются объектами соответствующих классов. При запуске программы автоматически открываются стандартные потоки ***cin, cout, cerr, clog.*** Последние два потока используются для вывода сообщений об ошибках. В файле ***iostream.h*** определены классы: ввода – ***istream***, вывода – ***ostream***, ввода-вывода – ***iostream****.*

Для реализации файлового ввода-вывода небходимо включить файл ***fstream.h***, содержащий производные от ***istream*** и ***ostream*** классы ***ifstream,******ofstream*** и ***fstream***,и объявить соответствующие объекты. Например:

**ifstream in;//ввод**

**ofstream out;//вывод**

**fstream io;//ввод-вывод**

После объявления потоков производится открытие файла, связывающее его с потоком с помощью функции ***open()*** или с помощью конструктора. Прототип функции ***open():***

**void open (char \*filename,int mode,int access);**

Здесь **filename** – имя файла, включающее путь; ***mode*** – режим открытия файла (**ios::in** – открытие файла для чтения, **ios::out** – открытие для записи, **ios::binary** – открытие файла в двоичном режиме, по умолчанию в текстовом); **access:** 0 – файл со свободным доступом, 1 – только для чтения, 8 – архивный файл. Файл закрывается с помощью функции ***close().***

Для чтения-записи здесь можно использовать перегружаемые оператор-функции >> и << или использовать методы классов. Для ввода-вывода одного символа используются функции:

**istream &get(char &ch); ostream &put(char ch);**

Для записи и считывания блоков двоичных данных используются функции считывания-записи *n* байт в буфер или из буфера:

**istream &read(unsigned char \*buf, int n);**

**ostream &write(const unsigned char \*buf, int n);**

*Обработка исключительных ситуаций.* В программах на С++ следует использовать механизм обработки исключительных ситуаций. Операторы программы при обработке исключительных ситуаций располагаются в блоке *try*. Перехватывается и обрабатывается исключительная ситуация в блоке *catch*. Форма операторов *try-catch* следующая:

**try {/\*блок try\*/ }**

**catch(type1 arg){/\*блок catch\*/}**

С блоком ***try*** можетсвязываться несколько блоков ***catch***. Выполняется тот блок ***catch***, для которого тип аргумента соответствует типу возникшей исключительной ситуации. При этом ее значение присваивается аргументу ***catch***. Если ошибка имеет место внутри блока ***try***, она может генерироваться с помощью ***throw***, после чего управление передано блоку ***catch.*** В случае необходимости перехвата всех исключительных ситуаций независимо от типа используется

**catch(…)**

**{/\*тело\*/}**

Для функций, вызываемых из блока ***try***, можно указать число типов исключительных ситуаций, которые будет генерировать функция:

**тип имя (список аргументов) throw(список типов)**

**{/\*тело\*/**}

**Пример.** Из текстового потока ввести информацию о студентах и записать ее в виде дерева. Отсортировать или по фамилии, или по баллу (вывести тех студентов, у которых балл выше среднего). Использовать потоки ввода-вывода и шаблоны.

#include <iostream.h>

#include <conio.h>

#include <fstream.h>

struct Student{

int num;

char surname[10];

int group;

int balls;

friend ostream &operator<< (ostream &stream, Student stud){

stream << " " << stud.num << " " << stud.surname << " " << stud.group

<< " " << stud.balls;

return stream;

}

friend istream &operator>> (istream &stream, Student &stud){

stream >> stud.num >> stud.surname >> stud.group >> stud.balls;

return stream;

}

};

struct node{

Student info;

node \*nextl, \*nextr;

node (){

info.num = info.group = info.balls=0;

nextl = nextr = 0;

}

node (Student newinfo){

info = newinfo;

nextl = nextr = 0;

}

};

template <class T, class T1> class tree{

public:

T \*root;

tree() { root = 0; }

void push (T \*& wer, T1 dat, int n){

if (wer == 0){

try{

wer = new T;

if(!wer) throw 1;

wer->nextl = 0; wer->nextr = 0; wer->info = dat;

}

catch (int mthrow) {cout<<”No memory!”<<endl;return;}

}

else if (n == 1)

if (strcmp(dat.surname,wer->info.surname) < 0) push (wer->nextl, dat, 1);

else push (wer->nextr, dat, 1);

else

if (dat.balls > wer->info.balls) push (wer->nextl, dat, 2);

else push (wer->nextr, dat, 2);

}

void insert (T1 dat, int n){

if (root == 0) root = new T(dat); else push (root, dat, n);

}

void look (ostream &stream, T \*&wer){

if (wer != 0){

look (stream, wer->nextl);

stream << " " << wer->info << endl;

look (stream, wer->nextr);

}

}

friend ostream &operator<< (ostream &stream, tree ob)

{ ob.look (stream, ob.root); return stream; }

};

void main(){

int m;

do{

cout << "1. Sort with names\n";

cout << "2. Sort with balls\n";

cout << "3. Exit\n";

cin >> m;

switch (m){

case 1: {

tree<node, Student> q;

node \*n;

ifstream infile("stud.txt");

while(!infile.eof()){

Student c;

infile >> c;

q.insert(c, 1);

}

infile.close();

cout<<q;

break;

}

case 2: {

tree<node, Student> q;

node \*n;

ifstream infile("stud.txt");

Student \*c;

c = new Student;

int i = 1;

float s = 0;

while(!infile.eof()){

infile >> c[i];

s+=c[i].balls;

i++;

}

for (int j=1; j<=i; j++)

if (c[j].balls > s/i)

q.insert(c[j], 2);

infile.close();

cprintf(" Miide ball is %1.3f",s/i);

cout<<'\n' << q;

break;

}

case 3: {return;}

default: {cout<<"Error! Try again\n"; break;}

}

getch();

clrscr(); }

while(m != 3);

return;}

Задания для самостоятельного решения

При выполнении приводимых ниже заданий можно использовать классы, разработанные в лабораторных работах № 1–3. Осуществлять контроль состояния потоков. В случае возникновения ошибок потоков генерировать и обрабатывать исключительные ситуации. Для соответствующих классов перегрузить операции вставки в поток и извлечения из потока. При динамическом выделении памяти предусмотреть обработку исключения, возникающего при нехватке памяти.

I

а) Для класса **Student** (лаб. работа № 1) предусмотреть ввод данных из файла. Полученные при выполнении лаб. работы № 1 списки студентов вывести в файл.

То же задание для классов:

б) **Abiturient** (лаб. работа №1);

в) **Aeroflot** (лаб. работа № 1);

г) **Worker** (лаб. работа № 1);

д) **Train** (лаб. работа № 1);

е) **Product** (лаб. работа № 1);

ж) **Patient** (лаб. работа № 1);

з) **Bus** (лаб. работа № 1);

и) **Customer** (лаб. работа № 1);

к) **File** (лаб. работа № 1);

л) **Word** (лаб. работа № 1);

м) **House** (лаб. работа № 1);

н) **Phone** (лаб. работа № 1);

о) **Person** (лаб. работа № 1).

II

а) При выполнении задания № 1 лаб. работы № 2 (класс **Com­plex**) предусмотреть формирование массива объектов путем считывания комплексных чисел из файла. Результат также вывести в файл.

То же задание для классов:

б) **Fraction** (лаб. работа № 2);

в) **Vector** (лаб. работа № 2). Предусмотреть обработку исключения при динамическом выделении памяти;

г) **Matrix** (лаб. работа № 2);

д) **Polynom** (лаб. работа № 2);

е) **Stack** (лаб. работа № 2);

ж) **Строка** (лаб. работа № 2);

з) **Set** (лаб. работа № 2);

и) **«Массив строк»** (зад. № 10 лаб. работы № 2);

к) «**Булев вектор»** (лаб. работа № 2);

л) «**Троичный вектор»** (лаб. работа № 2);

м) «**Булева матрица»** (лаб. работа № 2).

III

Те же задания, что и в разделах I и II, но для классов, реализующих работу с динамическими структурами данных (см. лаб. работу № 3).

Тесты

1. Если имеется код char a[8]; cin>>a; и вводится текст “Hello World”, то что будет в массиве a?

*Варианты ответа:*

1) “Hello W”; 2) “Hello Wo”; \*3) “Hello”; 4) “Hello World”; 5) “lo World”.

2. Что будет выведено в результате

double x=12.4;

cout<<setw(5)<<x<<setw(3)<<setfill(‘\*’)<<””<<endl; ?

*Варианты ответа:*

1) 12.40\*\*\*; \*2) 12.4\*\*\*; 3) 12.4 \*\*\*; 4) 12.40; 5) .124e2\*\*\*.

3. Если имеется код int x; cin>>x; и вводится “1.2”, что будет в перемен­ной x?

*Варианты ответа:*

\*1) 1; 2) 2; 3) 1.2; 3) другое; 4) произойдет ошибка.

4. Какой из классов используется для вывода строк на экран?

*Варианты ответа:*

1) strstream; \*2) ostream; 3) ofstream; 4) istream; 5) ifstream.

5. Каким будет результат работы программы:

#include <iostream.h>

void main (){

char A[]=”ABC”;

char \*U=&A[2];

cout<<”\n”<<\*U--<<\*U--<<\*U<<endl; } ?

*Варианты ответа:*

1) BAA; \*2) CBA.

6. Если имеется код double x; cin>>x; и вводится “12-3”, то что бу­дет в пе­ре­менной x?

*Варианты ответа:*

1) 9.0; \*2) 12.0; 3) другое; 4) произойдет ошибка.

7. Для того, чтобы выполнить чтение из файла с произвольной по­зиции, надо использовать объект класса:

1) strstream; 2) ostream; 3) ofstream; 4) istream; \*5) ifstream.

8. Что будет выведено при выполнении оператора throw C, если заданы классы

class A {…};

class B: public A {…};

class C: public A {…};

а обработка исключительной ситуации записана

catch (B&b) {cout<<1;}

catch (C&c) {cout<<2;}

catch (A&a) {cout<<3;}

catch (…) {cout<<4;} ?

*Варианты ответа:*

1) 1; \*2) 2; 3) 3; 4) 4; 5) 34; 6) 234.

9. Если в конструкторе класса

class A {

char \*ptr;

public:

A () {ptr=new char [size]; Init ();}

~A () { if (ptr) delete [] ptr;}

};

произойдет исключительная ситуация, будет ли потеряна память при откате по стеку?

*Варианты ответа:*

1) да, будет во всех случаях; 2) будет, только если объект класса созда­вался с помощью new; \*3) будет, если создавалась автома­тическая пере­менная класса A; 4) нет, не будет.

10. Об ошибке в конструкторе класса может сигнализировать:

1) возвращаемое значение; \*2) исключительная ситуация; 3) вы­зов дест­руктора сразу в конструкторе.

11. Что будет выведено, если заданы классы

class A {…};

class B: public A {…};

class C: public A {…};

а операторы throw и catch записаны так:

throw A;

catch (B&b) {cout<<1;}

catch (C&c) {cout<<2;}

catch (A&a) {cout<<3;}

catch (…) {cout<<4;} ?

*Варианты ответа:*

1) 1; 2) 2; \*3) 3; 4) 4; 5) 34; 6) 234.

12. Оператор throw без аргументов

\*1) повторно вызывает обрабатываемую исключительную си­туацию;

2) вызывает исключительную ситуацию типа Exception.

13. Что будет выведено, если заданы классы

class A {…};

class B: public A {…};

class C: public B {…};

а операторы throw и catch записаны так:

throw C;

catch (B&b) {cout<<1;}

catch (C&c) {cout<<2;}

catch (A&a) {cout<<3;}

catch (…) {cout<<4;} ?

*Варианты ответа:*

\*1) 1; 2) 2; 3) 3; 4) 4; 5) 1234; 6) 234.

ЛИТЕРАТУРА

1. *Страуструп, Б.* Язык программирования С++/*Б. Страуструп.*  СПб.:БИНОМ, 1999.