高级程序设计 2020 春

实验报告

指导老师：陈家骏 黄书剑 姓名：王晨渊, 学号：181220057

1 概念题

1.1 请简述 C++ 中异常处理的两种策略

就地处理：调用 exit 或 abort 异地处理：发现异常后在程序的其他地方进行处理。

1.2 C++ 异常处理机制中 try，throw 和 catch 语句的作用分别是什么？

try 启动异常处理机制。可以在 throw 异常对象之后，停止执行接下来的语句块。throw 可以抛掷异常对象。catch 捕获并处理异常对象

1.3 请简述 C++ 中断言（assertion）的概念和作用

断言是一个逻辑表达式，描述程序执行到断言处应满足的条件。用于发现、定位错误。

2 编程题

2.1 ExceptionTest

```
#include <iostream>
#include <cstdlib>
#include <cmath>
#include <fstream>
#include <stdexcept>
using namespace std;
class ExceptionTest
{
private:
    int prime[100];     //存前100个素数（质数）
public:
    //求分数，分子分母为a和b；分母为零异常
    double fraction(double a,double b);
```

```
14          //求底数为10的对数，真数为a；真数为负异常
15          double logarithm(double a);
16          //求算出前100个素数，放在prime中，并写入文件；文件打开失败异常
17          void calPrime(const char* address);
18          //从prime中获取第i个素数；数组下标越界异常
19          int getPrime(int i);
20      };
21
22      double ExceptionTest::fraction(double a, double b)
23      {
24          try
25          {
26              if(b==0) throw runtime_error("ZeroDivision!");
27              return a/b;
28          }
29          catch(runtime_error err)
30          {
31              cerr <<err.what()<<endl;
32              abort();
33          }
34      }
35
36      double ExceptionTest::logarithm(double a)
37      {
38          try
39          {
40              if(a<=0) throw range_error("Negative log!");
41              return log(a);
42          }
43          catch(range_error& e)
44          {
45              std::cerr << e.what() << '\n';
46              abort();
47          }
48
49      }
50
51      void ExceptionTest::calPrime(const char* address)
52      {
```

```
53          int prime_num=0;
54          int cur_num = 2;
55          while(prime_num!=100)
56          {
57              bool is_prime = true;
58              for(int i=2;i<cur_num;i++)
59              {
60                  if(cur_num%i==0)
61                  {
62                      is_prime = false;
63                      break;
64                  }
65              }
66              if(is_prime)
67              {
68                  prime[prime_num]=cur_num;
69                  prime_num++;
70              }
71              cur_num++;
72          }
73          try
74          {
75              fstream file(address,ios::app|ios::in);
76              if(file.fail()) throw runtime_error("Can't open file!");
77              for(int i=0;i<100;i++)
78              {
79                  file<<prime[i]<<",";
80                  if(i%10==9) file<<endl;
81              }
82              file.close();
83          }
84          catch(runtime_error& e)
85          {
86              cerr << e.what() << '\n';
87              abort();
88          }
89
90      }
91
```

```
92  int ExceptionTest::getPrime(int i)
93  {
94      try
95      {
96          if(!(0<=i&&i<=99)) throw invalid_argument("out of range!");
97          return prime[i];
98      }
99      catch(invalid_argument& e)
100     {
101         cerr << e.what() << '\n';
102         abort();
103     }
104 }
105
106 int main()
107 {
108     ExceptionTest a;
109     a.fraction(1,0);
110     a.calPrime("primes.txt");
111 }
```

2.2 注册

```
1  #include <string>
2  #include <iostream>
3  #include <fstream>
4  using namespace std;
5  class Web
6  {
7  public:
8      void inputName();
9      void inputAge();
10     void inputPhone();
11     void uploadFile();
12     void enroll();
13 private:
14     string name;
15     int age;
16     string phone;
```

```
17          string address;
18      };
19
20      void Web::enroll()
21      {
22          inputName();
23          inputAge();
24          L1:
25          try
26          {
27              inputPhone();
28          }
29          catch(const std::exception& e)
30          {
31              std::cerr << e.what() << '\n';
32              goto L1;
33          }
34          L2:
35          try
36          {
37              uploadFile();
38          }
39          catch(const std::exception& e)
40          {
41              std::cerr << e.what() << '\n';
42              goto L2;
43          }
44
45      }
46
47      void Web::inputName()
48      {
49          cout<<"Input your name. End with ENTER"<<endl;
50          getline(cin,name);
51      }
52
53      void Web::inputAge()
54      {
55          cout<<"Input your Age. End with ENTER"<<endl;
```

```
56          string age_buf;
57          getline(cin,age_buf);
58          age = stoi(age_buf);
59          if(age<11 || age>18)
60          {
61              cerr<<"Inappropriate age!"<<endl;
62              abort();
63          }
64      }
65
66      void Web::inputPhone()
67      {
68          cout<<"Input your Phone Number. End with ENTER"<<endl;
69          getline(cin,phone);
70          for(char s:phone)
71          {
72              if(!((s>='0'&&s<='9')||s=='-'))
73                  throw runtime_error("Inappropriate phone!");
74          }
75      }
76
77       void Web::uploadFile()
78       {
79          cout<<"Input your Article Address. End with ENTER"<<endl;
80          getline(cin,address);
81          fstream file(address,ios::in);
82          if(file.fail()) throw runtime_error("Something wrong with your address!");
83          //pretends to have some operation
84          //as no format is given, I can't do anything.
85          file.close();
86       }
87
88      int main()
89      {
90          Web myweb;
91          myweb.enroll();
92      }
```

2.3 memcpy

```
#include <iostream>
using namespace std;
void *memcpy(void *dst, void *src, unsigned count);
int main()
{
    int arr[]={1,2,3,4,5,6,7,8,9,10};

    for(int i=0;i<10;i++)
    {
        cout<<arr[i]<<",";
    }
    cout<<endl;
    try
    {
//          memcpy(NULL, arr+3,4);
//          memcpy(arr+4,NULL,4);
        memcpy(arr+3,arr+4,16);
    }
    catch(const std::exception& e)
    {
        std::cerr << e.what() << '\n';
    }
    for(int i=0;i<10;i++)
    {
        cout<<arr[i]<<",";
    }

}

void *memcpy(void *dst, void *src, unsigned count)
{

    if(!dst || !src) throw runtime_error("NULL pointer!");
    if(!((char*)dst>=(char*)src+count||(char*)dst+count<=(char*)src))
    {
        throw out_of_range("Intersected!");
    }
    for(unsigned i=0;i<count;i++)
```

```
39          {
40              *((char*)dst+i)=*((char*)src+i);
41          }
42          return dst;
```

2.4 书上的小问题

```
1       #include <iostream>
2       using namespace std;
3       int main()
4       {
5           int n,*p;//p173
6           cin>>n;
7           L1:
8           try
9           {
10              p = new int[n];
11              if(!p) throw runtime_error("Malloc Failed!");
12          }
13          catch(const std::exception& e)
14          {
15              std::cerr << e.what() << '\n';
16              goto L1;
17          }
18          delete[] p;
19          L2:
20          try
21          {
22              max(NULL,10);
23          }
24          catch(const std::exception& e)
25          {
26              std::cerr << e.what() << '\n';
27          }
28      }
29
30      int max(int x[],int num)//p131
31      {
32          int max_index =0;
```

```
33          if(!x) throw runtime_error("NULL point!");
34          for(int i=1;i<num;i++)
35          {
36              if(x[i]>x[max_index]) max_index = i;
37          }
38          return max_index;
39      }
```