



رسیدگی به خطاها — ۷

الگوهای کنترل در رسیدگی به استثناءها

بهار ۹۹

برنامه‌سازی پیشرفته — رامتین خسروی

read_student_info

```
void read_student_info(char* filename,
                      vector<Student>& v)
{
    ifstream input(filename);
    int count;
    input >> count;
    for (int i = 0; i < count; i++) {
        try {
            Student s = read_student(input);
            v.push_back(s);
        } catch(runtime_error& ex) {
            input.clear();
            string to_be_ignored;
            getline(input, to_be_ignored);
        }
    }
    input.close();
}
```

read_student

```
Student read_student(ifstream& input)
{
    string name;
    input >> name;
    Date bdate = read_date(input);
    return Student(name, bdate);
}
```

read_date

```
Date read_date(ifstream& input)
{
    int d, m, y;
    char ch;
    input >> d;
    input >> ch;
    if (ch != '/')
        throw runtime_error("...");
    input >> m;
    input >> ch;
    if (ch != '/')
        throw runtime_error("...");
    input >> y;
    return Date(d, m, y);
}
```

مسئله: می‌خواهیم پیامی را روی شبکه ارسال نماییم و در صورت بروز خطا،
آنقدر این کار را تکرار کنیم تا بالاخره پیام ارسال شود.

تعاریف پایه

```
class NetworkError {  
public:  
    NetworkError(string err) : error(err) {}  
    string get_error() { return error; }  
private:  
    string error;  
    // other fields describing the error  
};  
  
void send_bytes(const char* bytes, int len) {  
    // try to send s over the network ...  
    if (TIMEOUT)  
        throw NetworkError("Connection time-out");  
    // ...  
}
```

تابع ارسال پیام

```
void send_message(string msg) {
    open_connection();

    bool not_sent = true;
    while (not_sent)
        try {
            send_bytes(msg.c_str(), msg.length());
            not_sent = false;
        } catch(NetworkError& ex) {
            cerr << "could not send the message, retrying...\n";
        }

    close_connection();
}
```

آزمون برنامه

```
#define TIMEOUT (rand() % 10 != 0)
```

```
int main() {
    long seed = time(0);
    cout << "random seed = " << seed << endl;
    srand(seed);

    send_message("Hello");
}
```

اجرای برنامه

```
random seed = 1586062809  
could not send the message, retrying...
```

```
random seed = 1586062878  
could not send the message, retrying...  
could not send the message, retrying...  
could not send the message, retrying...  
could not send the message, retrying...  
could not send the message, retrying...  
could not send the message, retrying...  
could not send the message, retrying...
```

مسئله: می‌خواهیم پیامی را روی شبکه ارسال نماییم و در صورت بروز خطا، حداکثر n بار این کار را تکرار کنیم تا بالاخره پیام ارسال شود.

```

void send_message(string msg) {
    open_connection();
    int attempts = 0;
    while (attempts < MAX_ATTEMPTS)
        try {
            send_bytes(msg.c_str(), msg.length());
            cout << "Message sent successfully\n";
            break;
        } catch(NetworkError& ex) {
            cerr << "Could not send the message";
            attempts++;
            if (attempts < MAX_ATTEMPTS)
                cerr << ", retrying...";
            else {
                close_connection();
                throw;
            }
            cerr << endl;
        }
    close_connection();
}

```

```

random seed = 1586063706
could not send the message, retrying...
could not send the message, retrying...
could not send the message,
libc++abi.dylib: terminating with uncaught exception of type NetworkError

```

در مثال قبل فرض کنید تابع به نام `send_bytes_alt` داریم که ارسال داده‌ها را از یک پروتکل دیگر انجام می‌دهد. تابع `send_message` مثال قبل را طور زیر تغییر دهید تا سه بار به تناوب `send_bytes` و `send_bytes_alt` فراخوانی شوند. اگر هر یک از فراخوانی‌ها با موفقیت به پایان

```
void send_message(string msg) {
    open_connection();
    int attempts = 0;
    while (attempts < MAX_ATTEMPTS)
        try {
            send_bytes(msg.c_str(), msg.length());
            cout << "Message sent successfully\n";
            break;
        } catch(NetworkError& ex) {
            cerr << "Could not send the message";
            attempts++;
            if (attempts < MAX_ATTEMPTS)
                cerr << ", retrying...";
            else {
                close_connection();
                throw;
            }
            cerr << endl;
        }
    close_connection();
}
```

رسید، فراخوانی‌ها را بعد از انجام
نمی‌شود. اگر تمام فراخوانی‌ها با
استثنا مواجه شدند، پس از بستن
اتصال شبکه استثنا را بازپرتاب کنید.