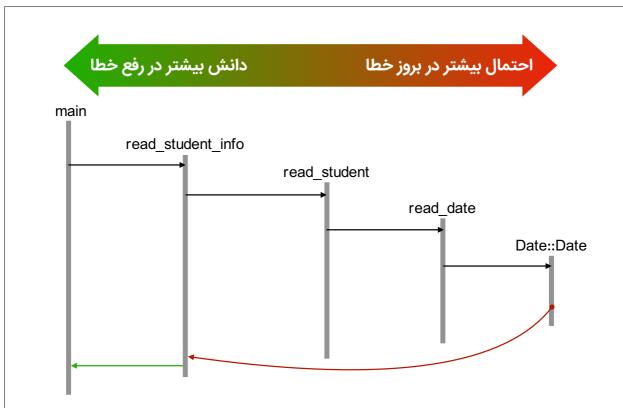


رسیدگی به خطاها — ۴

رسیدگی به خطاها با «استثناءها»

بهار ۹۹

برنامهسازی پیشرفته — رامتین خسروی



نیاز به سازوکاری برای انتقال حالت خطا از محل بروز خطا به محل رسیدگی به خطا

به سازوکاری برای رسیدگی به خطا نیاز داریم که:



- خطا را به طور خودکار تا نقطه رسیدگی منتشر کند
 - ما را مجبور به اضافه کردن کدهای بیمورد نکند

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++) {</pre>
                                             read student
      Student s = read_student(input);
                                           ➤ Student read_student(ifstream& input)
                                                                                        read date
      v.push_back(s);
    } catch(runtime_error& ex) {
                                               string name;
                                                                                       Date read_date(ifstream& input)
      input.clear();
                                               input >> name;
      string to_be_ignored;
                                               Date bdate = read_date(input);
                                                                                          int d, m, y;
      getline(input, to_be_ignored);
                                               return Student(name, bdate);
                                                                                          char ch;
                                                                                          input >> d;
                                                                                          input >> ch;
  input.close();
                                                                                          if (ch != '/')
  throw runtime_error("...");
                                                                                          input >> m;
                                                                                          input >> ch;
                                                                                          if (ch != '/')
                                                                                            throw runtime_error("...");
                                                                                          input >> y;
                                                                                          return Date(d, m, y);
```

https://github.com/ramtung/apnotes/tree/master/10_Exceptions/Import.3-ErrorHandlingByException

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

```
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++) {</pre>
                                     read student
     Student s = read Student read_student(ifstream& input) {
                                                                                   ate
     v.push_back(s);
                           string name;
   } catch(runtime_er
                                                                                   date(ifstream& input)
                           input >> name;
     input.clear();
     string to_be_ign
                           Date bdate = read_date(input);
                                                                                   n, y;
     getline(input, t
                           return Student(name, bdate);
                                                                                    > d;
                                                                                    > ch;
                     }
 input.close();
                                                                                   != '/')
                                                                               tnrow runtime_error("...");
                                                                             input >> m;
                                                                             input >> ch;
                                                                             if (ch != '/')
                                                                               throw runtime_error("...");
                                                                             input >> y;
                                                                             return Date(d, m, y);
```

```
read_student_info
void read_student_info(char* filename,
       Date read_date(ifstream& input) {
 ifstrea
            int d, m, y;
 int cou
 input :
            char ch;
 for (ir
            input >> d;
  try {
    Stu
            input >> ch;
    v.;
            if (ch != '/')
  } cat
    inp
                 throw runtime error("Slash separator expected");
    sti
            input >> m;
    get
  }
            input >> ch;
 }
            if (ch != '/')
 input.c
}
                 throw runtime_error("Slash separator expected");
                                                                               ...");
            input >> y;
            return Date(d, m, y);
                                                                               ...");
       }
```

```
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++) {</pre>
                                            read student
   Student s = read_student(input);
                                          Student read_student(ifstream& input)
      v.push_back(s);
    } catch(runtime_error& ex) {
                                              string name;
                                                                                       read date
     input.clear();
                                              input >> name;
                                              Date bdate = read_date(input);
     string to_be_ignored;
                                                                                       Date read_date(ifstream& input)

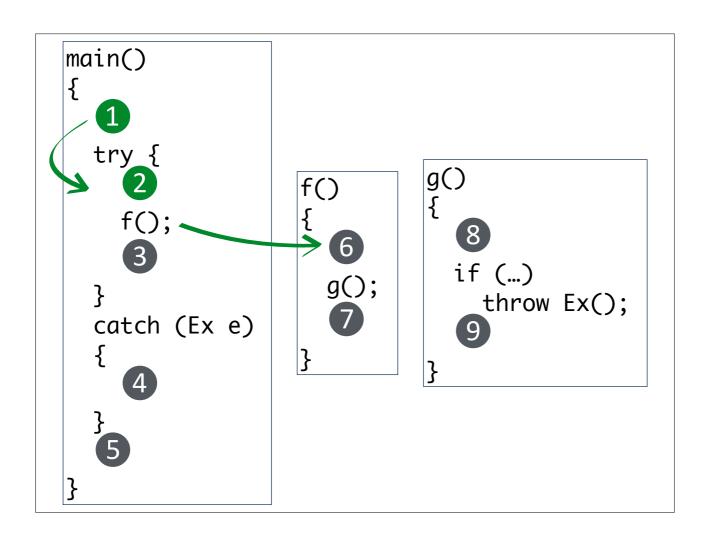
    getline(input, to_be_ignored);

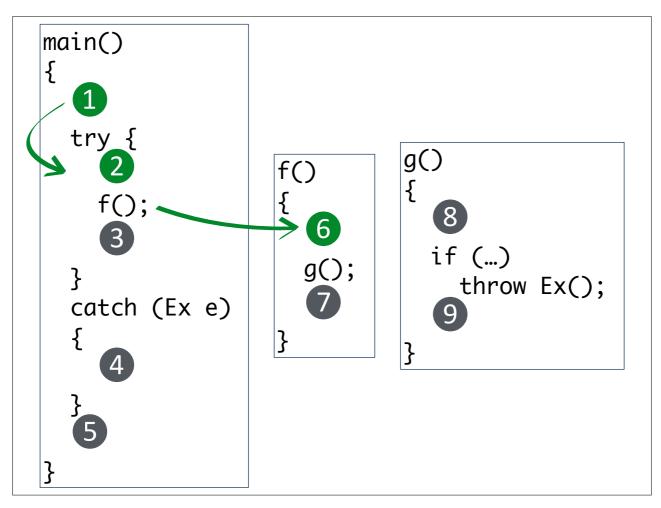
                                              return Student(name, bdate);
    }
                                            }
                                                                                          int d, m, y;
                                                                                          char ch;
  input.close();
                                                                                          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);
```

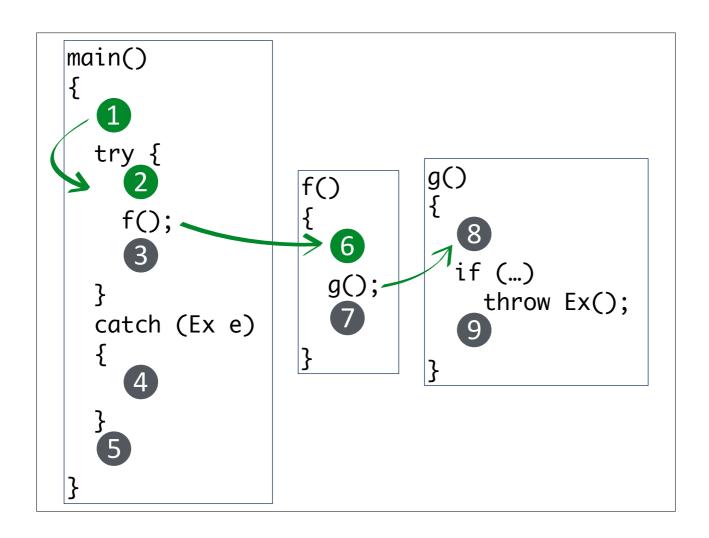
```
main()
{
   1
  try_{
                               g()
{
                    f()
                    {
    f();
                                  8
                       6
                                 if (...)
                      g();
                                    throw Ex();
  catch (Ex e)
                    }
                               }
}
```

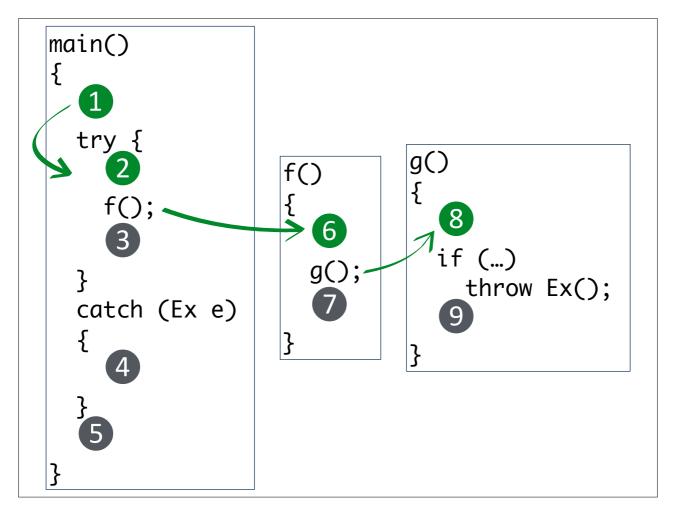
```
main()
{
  1
  try_{
                               g()
{
                    f()
                    {
    f();
                                  8
                       6
                                 if (...)
                      g();
                                    throw Ex();
  catch (Ex e)
                    }
                               }
}
```

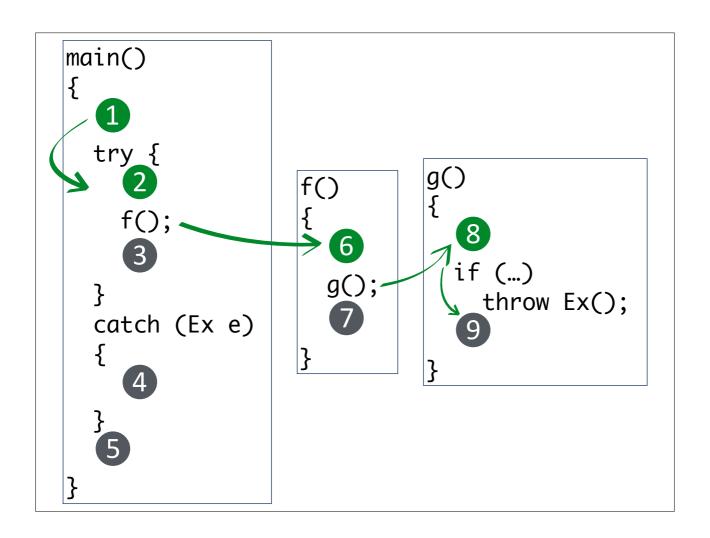
```
main()
{
   1
  try_{
                              g()
{
     2
                    f()
                    {
    f();
                                  8
                      6
                                 if (...)
                      g();
                                   throw Ex();
  catch (Ex e)
  {
                    }
```

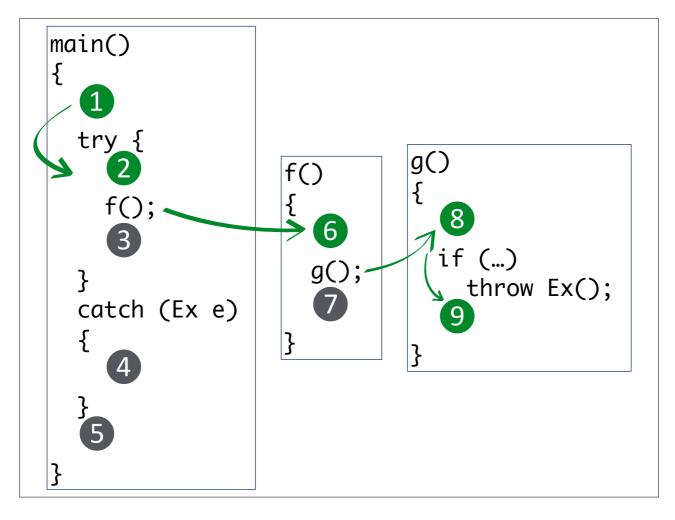


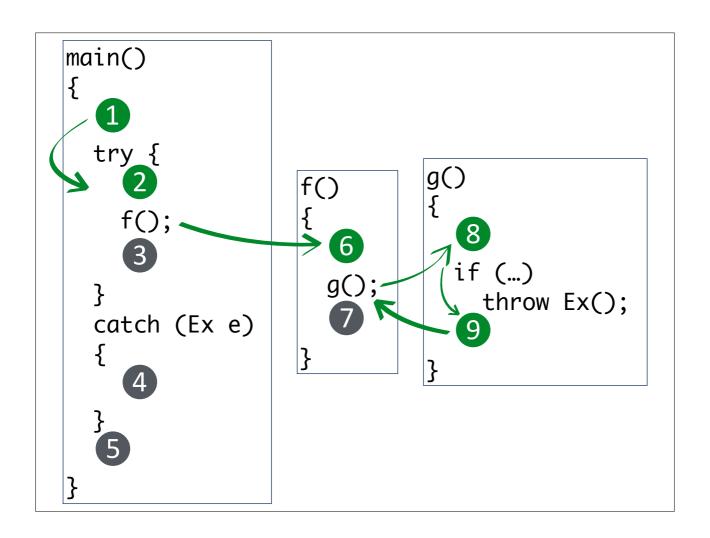


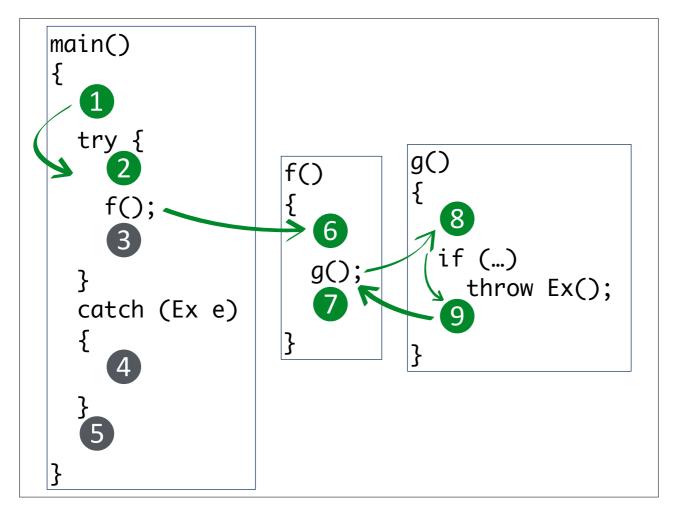


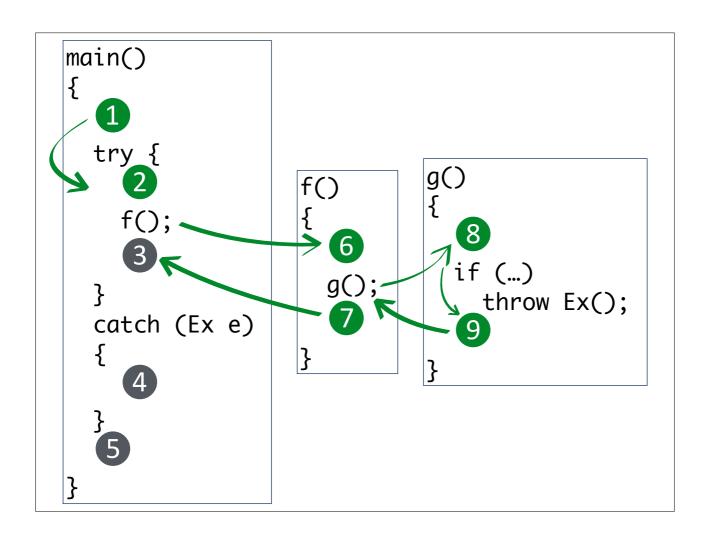


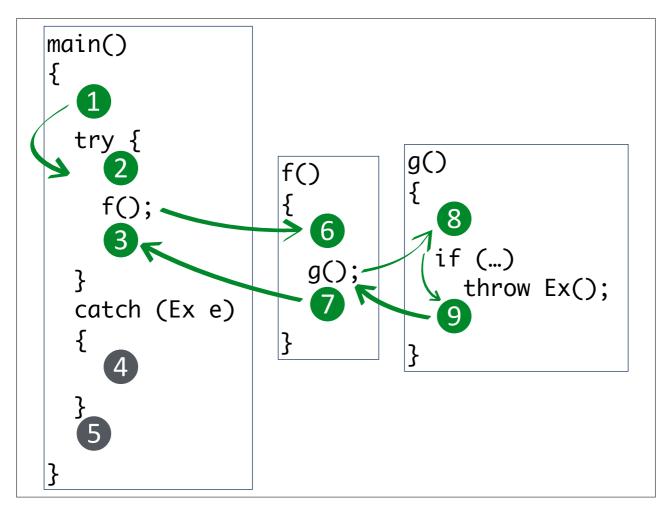


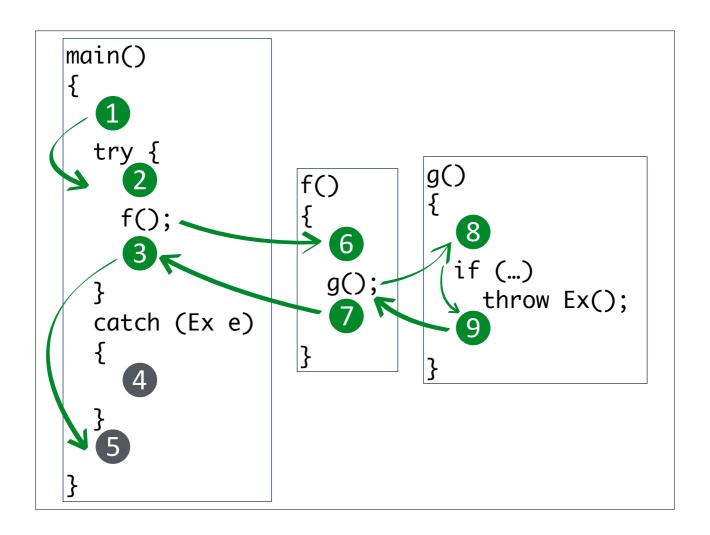


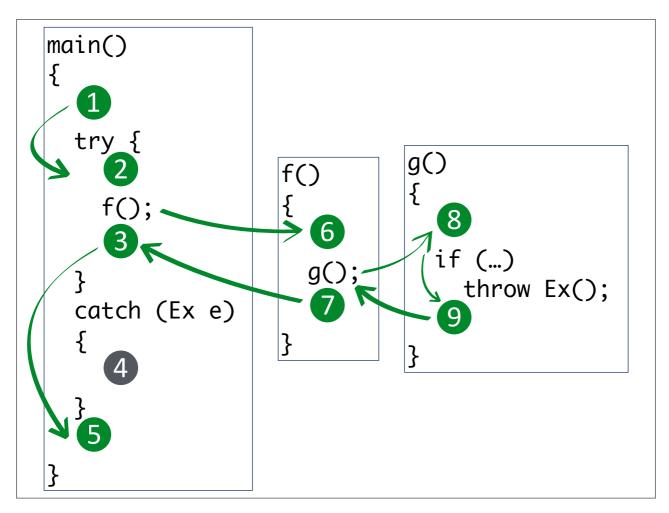


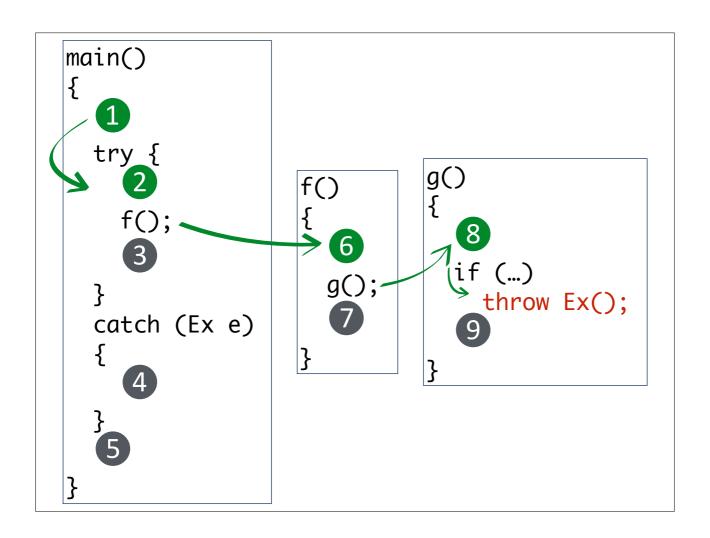


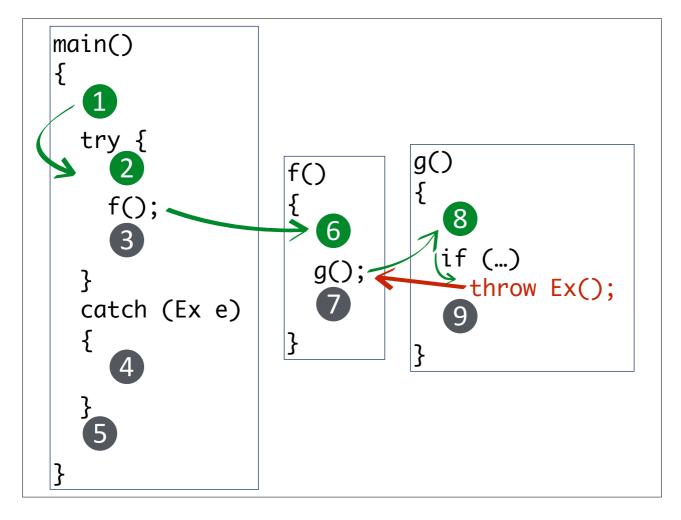


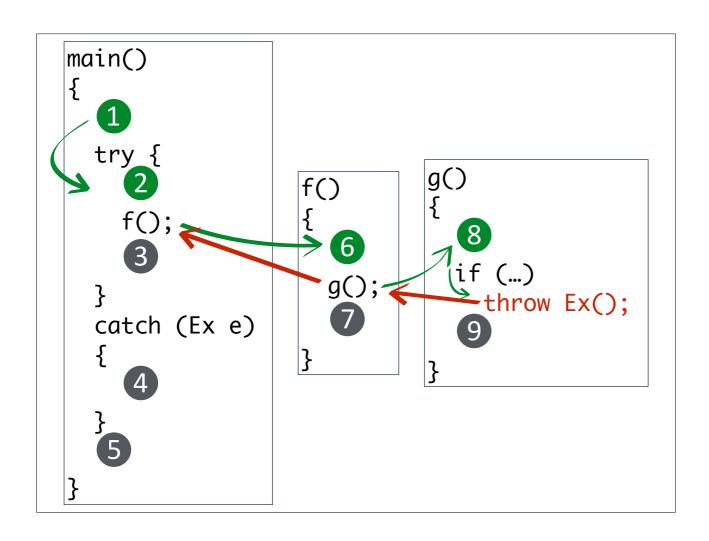


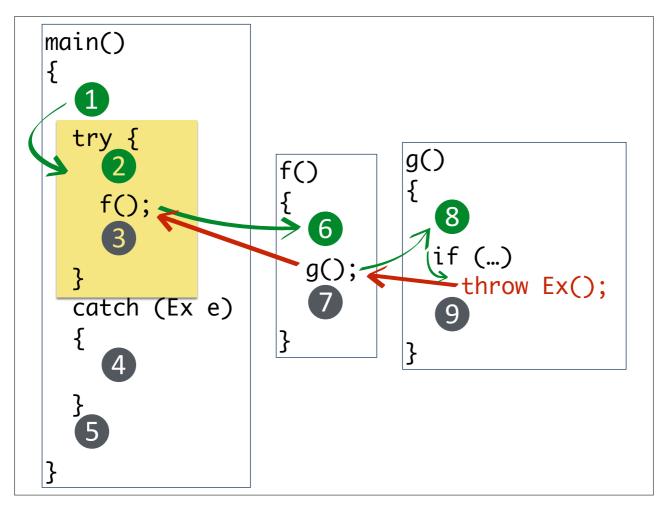


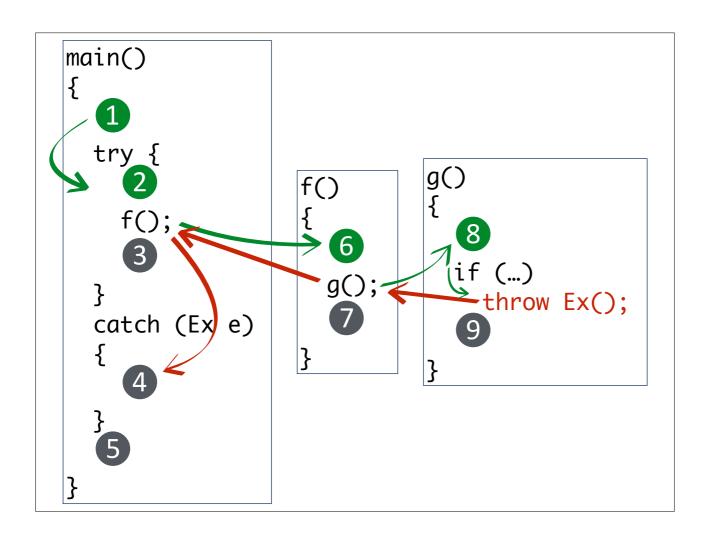


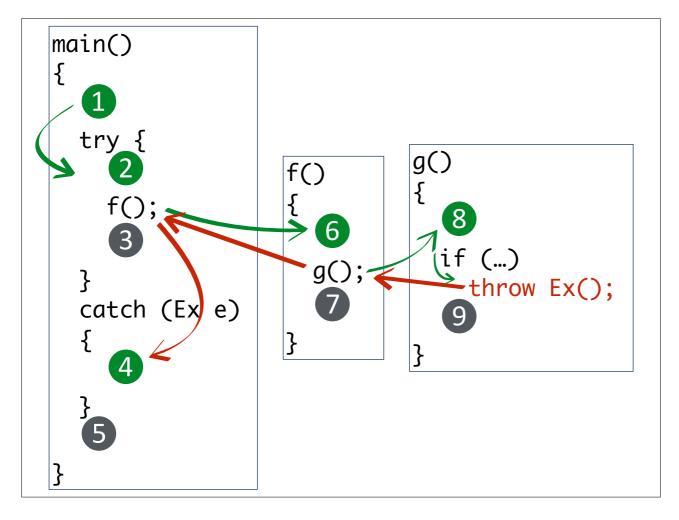


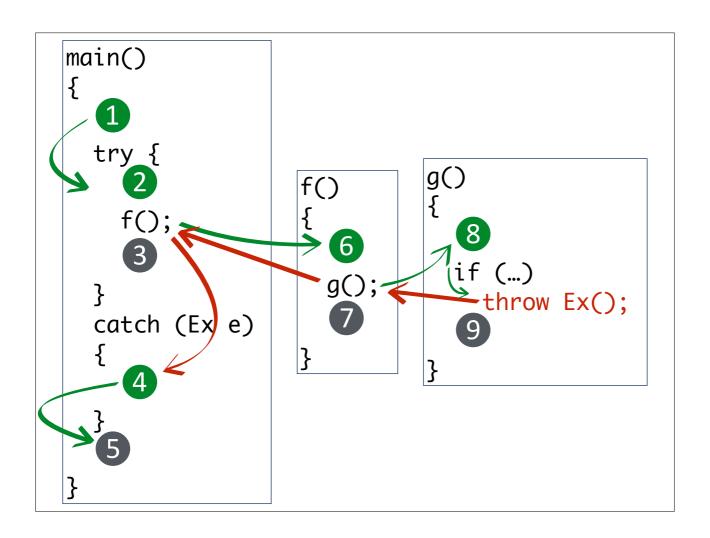


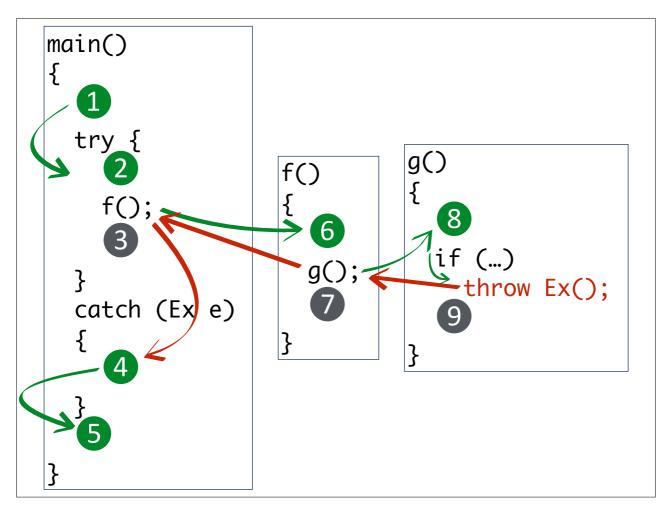












```
read_student_info
void read_student_info(char* filename,
        vector<Student>& v)
  ifstream input(filename);
  int count;
 input >> count;
                                             read student
  for (int i = 0; i < count; i++) {</pre>
   Student s = read student(input);
                                           Student read_student(ifstream& input)
      v.push_back(s);
    } catch(runtime_error& ex) {
                                                                                         read date
                                               string name;
    input.clear();
string to_be_ignored;
                                               input >> name;
                                               Date bdate = read_date(input);
                                                                                       Date read_date(ifstream& input)
    \bigstar getline(input, to_be_ignored);
                                               return Student(name, bdate);
                                                                                            int d, m, y;
                                                                                           char ch:
  input.close();
                                                                                            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);
```

```
بروسز اجرار برناده، تعیییز کنیر
نتیجه فرافوانر () h چیست؟
void f(int i) {
    cout << "f1\n";
    if (i % 2)
         throw runtime_error("error");
    cout << "f2\n";
}
void g() {
    for (int i = 0; i < 5; i++) {
         cout << "g" << i << endl;</pre>
         f(i);
    }
}
void h() {
    try {
         cout << "h1\n";
         g();
         cout << "h2\n";
    } catch (runtime_error& ex) {
         cout \ll "h3\n";
    cout << "h4\n";
}
```

```
void f(int i) {
   cout << "f1\n";</pre>
      try {
           cout << "f2\n";
if (i % 2)
     throw runtime_error("error");
cout << "f3\n";
} catch (runtime_error& ex) {</pre>
           cout << "f4\n";
throw runtime_error("I insist!");
cout << "f5\n";</pre>
      cout << "f6\n";
void g() {
    try {
           cout << "g1\n";
           f(1);
cout << "g2\n";
      } catch (runtime_error& ex) {
           cout \ll "g3\n";
      cout << "g4\n";
}
void h() {
    try {
           cout << "h1\n";
            g();
           cout << "h2\n";
     } catch (runtime_error& ex) {
   cout << "h3\n";</pre>
      cout << "h4\n";
}
```

بروسز اجرار برنامه، تعییسز کنیر نتیجه فراخوانس () h چیست ؟