دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر

رسیدگی به خطاها — ۵

تایپ استثناءها

برنامه‌سازی پیشرفته — رامتین خسروی

بهار ۹۹

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);
}
```

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_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
      v.push_back(s);
    } catch(runtime_er
      input.clear();
      string to_be_ign
      getline(input, t
    }
  }
  input.close();
}
```

read_student

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

ate

```
_date(ifstream& input)
m, y;
> d;
> ch;
!= '/')
  throw 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,
{
  ifstrea
  int cou
  input >
  for (in
    try {
      Stu
      v.p
    } cat
      inp
      str
      get
    }
  }
  input.c
}
```

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

```
input)
…");
…");
}
```

طبق نحو زبان، ارسال هر عبارتی به عنوان استثناء ممکن است:

```
throw 1;

throw "Invalid date format";

throw runtime_error("Connection error");

throw f(i) * 2;
```

مدل مرسوم‌تر، استفاده از کلاس‌ها برای مشخص‌کردن نوع خطا است:

```
throw runtime_error("Connection error");
throw DatabaseError("Cannot connect to the database");
throw ImportException(input_filename, current_line);
```

بخشی از سلسله‌مراتب استثناءهای استاندارد C++

<stdexcept> header file

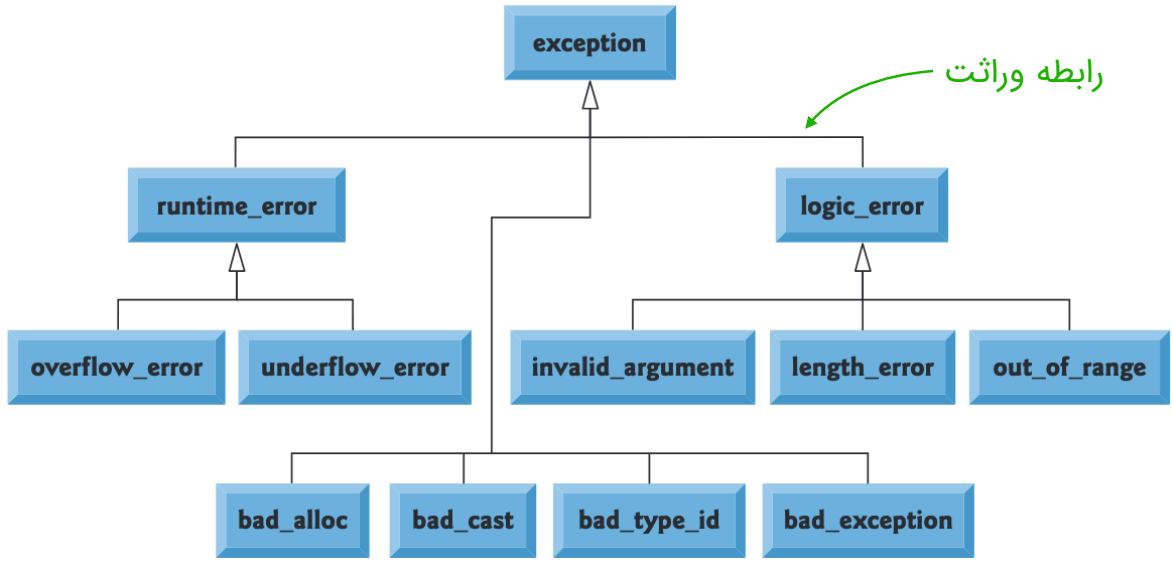

Image source: Deitel & Deitel, C++ How to Program

استفاده از شیء استثناء

```
throw runtime_error("Connection error");
```

```
catch (runtime_error& ex) {
    cerr << ex.what() << endl;
    //...
}
```

```
class ImportException {
public:
    ImportException(string filename_, int line_no_,
                    string description_ = "") :
        filename(filename_),
        line_no(line_no_),
        description(description_) {}

    string get_filename() { return filename; }
    int get_line_no() { return line_no; }
    string get_description() { return description; }

    string get_message() {
        ostringstream os;
        os << "Error when importing from " << filename
           << " at line " << line_no << ": "
           << description;
        return os.str();
    }
private:
    string filename;
    int line_no;
    string description;
};
```

تعریف تایپ استثناء جدید

استفاده از شیء استثناء

```
try {
    // ... code that might throw ImportException
} catch (ImportException& ex) {
    error_line_nos.push_back(ex.get_line_no);
    cerr << ex.get_message();
    // ... rest of exception handling code
}
```

شیء استثناء وظیفه منتقل کردن اطلاعات خطا را از محل بروز به محل رسیدگی برعهده دارد.

رسیدگی به انواع مختلف استثناءها

```
try {

    // ... code that might throw exception

} catch (ImportException& ex) {
    error_line_nos.push_back(ex.get_line_no);
    cerr << ex.get_message();
    // ... rest of exception handling code
} catch (DatabaseException& dbex) {
    // ... handle database exception
} catch (...) {
    // ... handle other types of exception
}
```


```
class ImportException : public runtime_error {
public:
    ImportException(string filename_, int line_no_,
                    string description_ = "") :
            runtime_error(description_),
            filename(filename_),
            line_no(line_no_),
            description(description_) {}

    string get_filename() { return filename; }
    int get_line_no() { return line_no; }
    string get_description() { return description; }

    string get_message() {
        ostringstream os;
        os << "Error when importing from " << filename
           << " at line " << line_no << ": "
           << description;
        return os.str();
    }
private:
    string filename;
    int line_no;
    string description;
};
```

```
try {

    // ... code that might throw exception

} catch (DatabaseException& dbex) {
    // ... handle database exception
} catch (runtime_error& rtex) {
    // ... handle runtime errors
} catch (...) {
    // ... handle other types of exception
}
```

will catch ImportException

تمرین‌های کوتاه

۱. اگر تایپ استثناء پرتاب‌شده با هیچ‌یک از catchها مطابقت نداشته باشد چه اتفاقی می‌افتد؟

۲. اگر تایپ استثناء پرتاب‌شده با بیش از یک catch مطابقت داشته باشد چه اتفاقی می‌افتد؟

۳. جای catch (...) در ترتیب بلوک‌های catch کجاست؟

٤. استفاده مستقیم از استثناءهای استاندارد C++ چندان توصیه نمی‌شود. به جای آن بهتر است زیرکلاس‌هایی از آن‌ها را تعریف و پرتاب کنیم. فکر می‌کنید دلیل هر یک از دو جمله قبل چیست؟

تمرین‌های کوتاه

۵. در متن مثال (موجود در گیت‌هاب)، استثناءهای پرتاب‌شده از سازنده `Date` و تابع `days_of_month` از نوع `runtime_error` تعریف شده‌اند. علت این امر ساده نگهداشتن `catch` در فایل `main.cpp` بوده. در این تمرین این استثناءها را به `invalid_argument` تغییر دهید و بلوک `try/catch` را نیز متناسب با آن تغییر دهید. دقت نمایید روش رسیدگی به تمام استثناءها کماکان ثابت خواهد بود.