

ОБРАБОТКА
ИСКЛЮЧИТЕЛЬНЫХ
СИТУАЦИЙ

- 1. прервать выполнение программы;
- 2. вернуть значение ошибки;
- 3. вывести сообщение об ошибке в поток `cerr` и вернуть вызывающей программе некоторое приемлемое значение, которое позволит ей продолжить работу

- *Исключительная ситуация* или *исключение* – это возникновение непредвиденного или аварийного события, которое может порождаться некорректным использованием аппаратуры

Исключения позволяют логически разделить вычислительный процесс на две части:

- обнаружение аварийной ситуации;*
- обработка аварийной ситуации*

Общий механизм обработки исключений

Контролируемый блок – составной оператор, перед которым написано ключевое слово **try**

- Обработка исключения начинается с появления ошибки. Функция, в которой она возникла, генерирует исключение. Для этого используется ключевое слово **throw** с параметром, определяющим вид исключения. Параметр может быть константой, переменной или объектом и используется для передачи информации об исключении его обработчику
- Отыскивается соответствующий обработчик исключения и ему передается управление
- Если обработчик исключения не найден, вызывается стандартная функция **terminate**, которая вызывает функцию **abort**, аварийно завершающую текущий процесс. Можно установить собственную функцию завершения процесса

Определение исключений

- Выделить контролируемый блок – блок `try`
- Предусмотреть генерацию одного или нескольких исключений операторами `throw` внутри блока `try` или внутри функций, вызываемых из этого блока
- Разместить сразу за блоком `try` один или несколько обработчиков исключений `catch`

- `try { // фрагмент кода }`
- `throw` выражение

Тип выражения, стоящего после `throw`, определяет тип порождаемого исключения

- `catch (...) { // действия по обработке исключения }`

- catch (Type info)
 { // обработка исключения типа Type
 // с использованием значения info
 }
- catch (Type)
 { // обработка исключения типа Type}
- catch (...)
 { // обработка исключений всех типов}

Перехват исключений

- Создают копию параметра `throw` в виде статического объекта, который существует до тех пор, пока исключение не будет обработано;
- В поисках подходящего обработчика раскручивают стек, вызывая деструкторы локальных объектов, выходящих из области действия;
- передают объект и управление обработчику, имеющему параметр, совместимый по типу с этим объектом

Обработчик считается найденным, если тип объекта, указанного после `throw`:

- тот же, что и указанный в параметре `catch` (параметр может быть записан в форме `T`, `Const T`, `T&` или `const T&`, где `T` – тип исключения);
- является производным от указанного в параметре `catch` (если наследование производилось с ключом доступа `public`):
- является указателем, который может быть преобразован по стандартным правилам преобразования указателей к типу указателя в параметре `catch`.

Пример

```
#include <fstream.h>
// класс, информирующий о своем
// создании и уничтожении

class Hello
{Hello() { cout << "Hello!" << endl; }
  ~Hello() { cout << "Bye!" << endl; }
};
```

Пример

```
void f1( )
{ifstream ifs(name); // открываем файл
  if (!ifs)
    { cout << "генерируем исключение " << endl;
      throw "Ошибка при открытии файла";
    }
}

void f2( )
{Hello H; // создаем локальный объект
  f1(); // вызываем функцию, генерирующую исключение
}
```

Пример

```
int main( )
{ try
    { cout << “входим в try-блок” << endl;
      f2( );
      cout << “выходим из try-блока” << endl;
    }
  catch( int i)
    { cout << “вызван обработчик int, исключение - ” << i;
      cout << endl;
      return -1;
    }
}
```

Пример

```
catch( const char * p)
```

```
{ cout << “вызван обработчик const char *”;
```

```
  cout << “, исключение - ” << p << endl;
```

```
  return -1;
```

```
}
```

```
catch( ... )
```

```
{ cout << “вызван обработчик всех исключений ”;
```

```
  cout << endl;
```

```
  return -1;
```

```
}
```

```
return 0; // обошлось без исключительных ситуаций
```

```
}
```

Результаты выполнения программы

входим в try-блок

Hello!

генерируем исключение

Bye!

вызван обработчик `const char *`,
исключение – Ошибка при открытии
файла