

# Программирование на языке Си++

## Модуль 8. УПРАВЛЕНИЕ ИСКЛЮЧЕНИЯМИ

- Применение операторов `try`, `catch`, `throw`
- Динамическая идентификация типов времени выполнения (RTTI)
- Операции приведения типа



# Понятие исключения. Обработка исключительных ситуаций (начало)

- **Исключение** (исключительная ситуация) — особая, часто аномальная, ситуация, возникающая в программе на этапе исполнения таковой и нарушающая обычный ход вычислительного процесса. Механизм взаимодействия компонентов программы в исключительных ситуациях носит название **обработки исключительных ситуаций**
- Обработка исключительных ситуаций в программе на языке Си++ предполагает включение в ее текст **контролируемых блоков и обработчиков исключений**

- **Контролируемый блок**

```
try { <список операторов>; }
```

- **Оператор возбуждения исключения** — формирует временный статический объект-исключение, тип которого определяется типом выражения в операторе

```
throw <выражение>;
```

# Понятие исключения. Обработка исключительных ситуаций

(окончание)

- **Специализированный обработчик исключения** — «настроен» на обработку исключений одного типа

```
catch (<имя типа> [<идентификатор>])  
{ <список операторов>; }
```
- **Универсальный обработчик исключения** — «настроен» на обработку исключений всех типов

```
catch (...) { <список операторов>; }
```
- За каждым контролируемым блоком обязательно расположен один или несколько обработчиков исключений
  - При наличии нескольких обработчиков они должны отличаться типами обрабатываемых исключений
  - Обработчик `catch (...)` должен идти последним
  - Исключение как объект не локализуется в контролируемом блоке и не исчезает, пока не будет должным образом обработано. Благодаря этому исключение может выступать средством передачи любого количества информации из точки возбуждения в точку обработки исключительной ситуации

## Динамическая идентификация типов (RTTI)

- **Динамическая идентификация типов (RTTI)** позволяет установить фактический производный тип объекта, адресуемого по указателю на базовый полиморфный класс. Для выяснения фактического производного типа служит унарная **операция** `typeid`, операнд которой в этом случае должен иметь тип класса хотя бы с одной виртуальной функцией

- В общем виде выражение с операцией `typeid` имеет вид

```
typeid(<имя типа>)
```

```
typeid(<выражение>)
```

Тип каждого из этих двух выражений — `const type_info&`.

- RTTI-операция `typeid` — это событие времени выполнения для классов с виртуальными функциями и событие времени компиляции для остальных типов [ЛЛ01]

# Список литературы

- [КР92] Керниган Б., Ритчи Д. Язык программирования Си / Пер. с англ. — М.: Финансы и статистика, 1992. — 272 с.
- [КР06] Керниган Б., Ритчи Д. Язык программирования С / Пер. с англ. — М.: Вильямс, 2006. — 304 с.
- [ЛЛ01] Липпман С., Лажойе Ж. Язык программирования С++. Вводный курс. — СПб.: Невский диалект, 2001. — 1104 с.
- [Под03] Подбельский В.В. Язык Си++: Учеб. пособие. — 5-е изд. — М.: Финансы и статистика, 2003. — 560 с., ил.
- [Под04] Подбельский В.В., Фомин С.С. Программирование на языке Си. — 2-е доп. изд. — М.: Финансы и статистика, 2004. — 600 с.
- [Стр08] Страуструп Б. Язык программирования С++. Специальное издание. — СПб.: Невский диалект, 2008. — 1104 с.