Практическое значение №10\_2

**«Обработка исключений»**

**Теория**

Исключительная ситуация, или исключение — это возникновение непредвиденного или аварийного события, которое может порождаться некорректным использованием аппаратуры. Например, это деление на ноль или обращение по несуществующему адресу памяти. Обычно эти события приводят к завершению программы с системным сообщением об ошибке. С++ дает программисту возможность восстанавливать программу и продолжать ее выполнение.

Исключения С++ не поддерживают обработку асинхронных событий, таких, как ошибки оборудования или обработку прерываний, например, нажатие клавиш Ctrl+C. Механизм исключений предназначен только для событий, которые происходят в результате работы самой программы и указываются явным образом. Исключения возникают тогда, когда некоторая часть программы не смогла сделать то, что от нее требовалось. При этом другая часть программы может попытаться сделать что-нибудь иное.

Исключения позволяют логически разделить вычислительный процесс на две части — обнаружение аварийной ситуации и ее обработка. Это важно не только для лучшей структуризации программы. Главной причиной является то, что функция, обнаружившая ошибку, может не знать, что предпринимать для ее исправления, а использующий эту функцию код может знать, что делать, но не уметь определить место возникновения. Это особенно актуально при использовании библиотечных функций и программ, состоящих из многих модулей.

Другое достоинство исключений состоит в том, что для передачи информации об ошибке в вызывающую функцию не требуется применять возвращаемое значение, параметры или глобальные переменные, поэтому интерфейс функций не раздувается. Это особенно важно, например, для конструкторов, которые по синтаксису не могут возвращать значение.

# **Общий механизм обработки исключений**

Место, в котором может произойти ошибка, должно входить в контролируемый блок — составной оператор, перед которым записано ключевое слово **try**.

Рассмотрим, каким образом реализуется обработка исключительных ситуаций.

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.

Синтаксис исключений

Ключевое слово **try** служит для обозначения контролируемого блока — кода, в котором может генерироваться исключение. Блок заключается в фигурные скобки:

**try{ }**

Генерация (порождение) исключения происходит по ключевому слову throw, кото­рое употребляется либо с параметром, либо без него:

throw [ выражение ];

Тип выражения, стоящего после throw, определяет тип порождаемого исключения. При генерации исключения выполнение текущего блока прекращается, и происходит поиск соответствующего обработчика и передача ему управления. Как правило, исключение генерируется не непосредственно в try-блоке, а в функциях, прямо или косвенно в него вложенных.

Обработчики исключений начинаются с ключевого слова catch, за которым в скобках следует тип обрабатываемого исключения. Они должны располагаться непосредственно за try-блоком.

Практика

1. Пример. Демонстрирует обработку исключений (lab10\_6.cpp)
2. Пример. Здесь задействовано несколько блоков обработки исключений. Срабатывает тот блок, который соответствует типу исключения. (lab10\_7.cpp)
3. Пример. Исключение может генерироваться в вызываемой программе, а перехватываться в вызывающей (lab10\_8.cpp)
4. Пример. Если мы хотим перехватить любые исключения, то должны написать catch(…). Этот блок имеет смысл размещать в конце всех выбранных блоков (lab10\_9.cpp)
5. **Задание.** Создать массив, состоящий из 10 чисел. Вывести первый отрицательный элемент с помощью обработчика исключений. Дополнительно – вывести все отрицательные числа.