**ЛЕКЦИЯ 9**

**МЕТОДЫ ОБРАБОТКИ ИСКЛЮЧЕНИЙ**

1. Исключение – это непредвиденное событие, не позволяющие продолжить выполнения программы;

**ВИДЫ ИСКЛЮЧЕНИЙ**

1. Синхронные исключения: Такие исключения могут быть только в определенных местах программах;
2. Асинхронные исключения: могут возникать в любой момент программы (вторжение другой программы в память нашей, проблема железа).

**ОБРАБОТКА ИСКЛЮЧЕНИЯ**

1. Обнаружение исключения;
2. Передача управления обработчику исключений;
3. Выполнение обработчика;
4. Обработчик исключения – блок кода, которому передается управления в случае появления исключения;

**ВИДЫ ОБРАБОТКИ ИСКЛЮЧЕНИЙ**

1. Обработка с возвратом – обработчик ликвидирует проблему и возвращает управление в точку появления исключения;
2. Обработка без возврата – после устранения проблемы, вернет управление в определенную точку программы.

**МЕХАНИЗМ ОБРАБОТКИ ИСКЛЮЧЕНИЙ В C++**

try (содержит потенциально опасный код. Все переменные, объявленные в нем – локальны для try)

{

//исключение

throw () //генерирует объект для передачи значения (константа, переменная, выражение)

}

Catch (тип объекта, переданный throw) (… - для всех throw, если нету явного аналога. Если нету catch для нужного типа данных, будет ошибка)

{

//обработка исключения

}

Нельзя ставить явный catch после троеточного!

**ОБРАБОТКА ИСКЛЮЧЕНИЙ С ПОМОЩЬЮ ФУНКЦИЙ**

1. Если нет оператора – обработчика, то вызывается функция unexpected, которая терминирует всю программу;
2. Обе функции можно подменить своими реализациями. Подключаем директиву “exeption”. Определяем свой вариант, заменяем используя set\_unexpected(метод). Может закончить исполнение функции exit(int) или abort;

**ЗАЧЕМ?**

1. Функция, которая обнаруживает ошибку, может не знать, что с ней делать. Функция, знающая, как что делать, может её не обнаружить;
2. Стоит использовать, когда заранее известно, что программа не сможет в случае чего продолжить исполнение программы при некорректно вводе пользователя;