

Визуальное программи- рование

ЛЕКЦИЯ 6

Содержание лекции

01

Обработка исключений

ОБРАБОТКА ИСКЛЮЧЕНИЙ

Понятие исключения

Исключениями, или **исключительными ситуациями**, обычно называются аномалии, которые могут возникать во время выполнения и которые трудно, а порой и вообще невозможно, предусмотреть во время программирования приложения.

В .NET существует **структурированная обработка исключений**, которая представляет собой методику, предназначенную для работы с исключениями, которые могут возникать на этапе выполнения.

Обработка исключений

Программирование со структурированной обработкой исключений подразумевает использование четырех следующих связанных между собой сущностей:

- тип класса, который представляет детали исключения;
- член, способный генерировать (throw) в вызывающем коде экземпляр класса исключения при соответствующих обстоятельствах;
- блок кода на вызывающей стороне, ответственный за обращение к члену, в котором может произойти исключение;
- блок кода на вызывающей стороне, который будет обрабатывать (или перехватывать (catch) исключение в случае его возникновения.

Обработка исключений

- Блоки **try** инкапсулируют код, формирующий часть нормальных действий программы, которые потенциально могут столкнуться с серьезными ошибочными ситуациями.
- Блоки **catch** инкапсулируют код, который обрабатывает ошибочные ситуации, происходящие в коде блока `try`. Это также удобное место для протоколирования ошибок.
- Блоки **finally** инкапсулируют код, очищающий любые ресурсы или выполняющий другие действия, которые обычно нужно выполнить в конце блоков `try` или `catch`. Важно понимать, что этот блок выполняется независимо от того, сгенерировано исключение или нет.

Класс Exception

Все определяемые на уровне пользователя и системы исключения в конечном итоге всегда наследуются от базового класса `System.Exception`, который, в свою очередь, наследуется от класса `System.Object`.

Специальные исключения

Если планируется создать действительно заслуживающий внимания специальный класс исключения, необходимо позаботиться о том, чтобы он соответствовал наилучшим рекомендациям .NET. Он должен:

- наследоваться от `ApplicationException`;
- сопровождаться атрибутом `[System.Serializable]`;
- иметь конструктор по умолчанию;
- иметь конструктор, который устанавливает значение унаследованного свойства `Message`;
- иметь конструктор для обработки "внутренних исключений";
- иметь конструктор для обработки сериализации типа.

ВОПРОСЫ ПО ЛЕКЦИИ