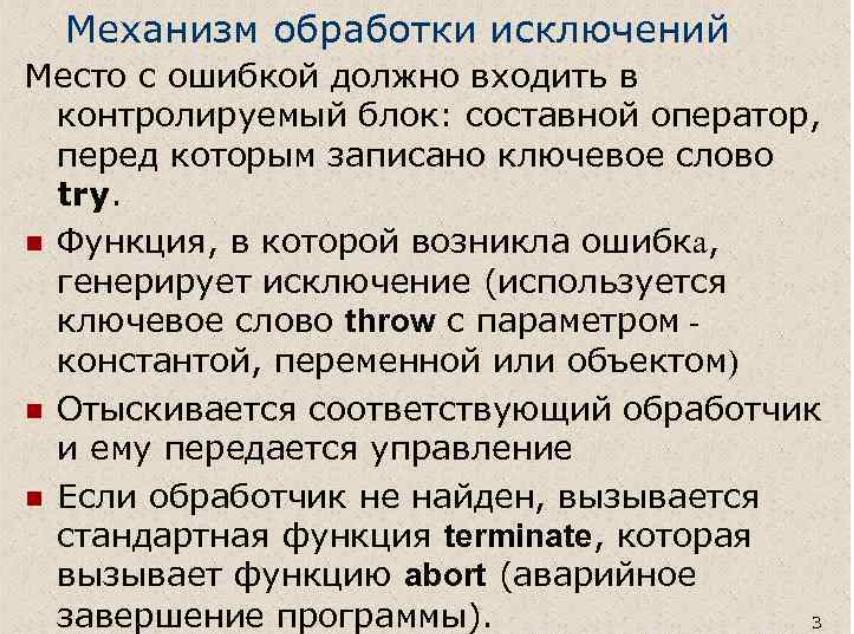
ЛАБОРАТОРНАЯ РАБОТА 7 ИСКЛЮЧЕНИЯ

1. Расскажите как генерируется исключение.

Обычно система сама генерирует исключения при определенных ситуациях, например, при делении числа на ноль. Но язык C# также позволяет генерировать исключения вручную с помощью оператора throw. То есть с помощью этого оператора мы сами можем создать исключение и вызвать его в процессе выполнения.

После оператора throw указывается объект исключения, через конструктор которого мы можем передать сообщение об ошибке.

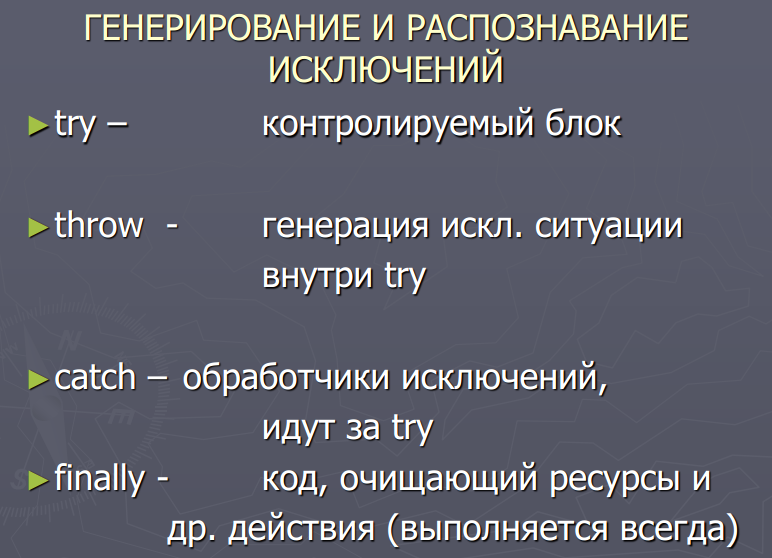
1. Расскажите методику обработки исключений.



1. Какое ключевое слово служит для обозначения блока кода, в котором можно генерировать исключение?

Throw

1. Какие ключевые слова используются для обработки и генерации исключений? Расскажите об механизме обработке исключения?

в к

* try-catch, try-finally, try-catch-finallyа

Когда внутри такого блока try возникает исключение, поток управления переходит к первому подходящему обработчику исключений в стеке вызовов. В C# ключевое слово catch обозначает обработчик исключений.

* Если для созданного исключения не существует обработчиков, выполнение программы прекращается с сообщением об ошибке.

1. Что будет, если в программе нет предложения catch, способного обработать  
   исключение?

При возникновении исключения среда CLR ищет блок catch, который может обработать данное исключение. Если такого блока не найдено, то пользователю отображается сообщение о необработанном исключении, а дальнейшее выполнение программы останавливается. И чтобы подобной остановки не произошло, и надо использовать блок try..catch.

1. Что такое фильтры исключения? Приведите пример.

Фильтры исключений — это механизм, который позволяет блоку catch объявить предусловия,

которым должно удовлетворять исключение, дабы быть пойманным данным блоком.

public void Main()

{

try

{

throw new Exception("E2");

}

catch(Exception ex) when(ex.Message == "E1")

{

Console.WriteLine("caught E1");

}

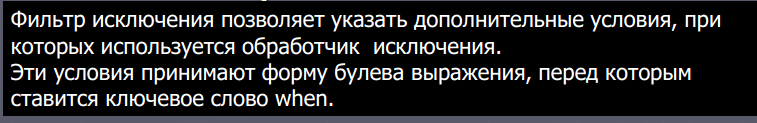
catch(Exception ex) when(ex.Message == "E2")

{

Console.WriteLine("caught E2");

}

}



1. Могут ли исключения быть вложенными?

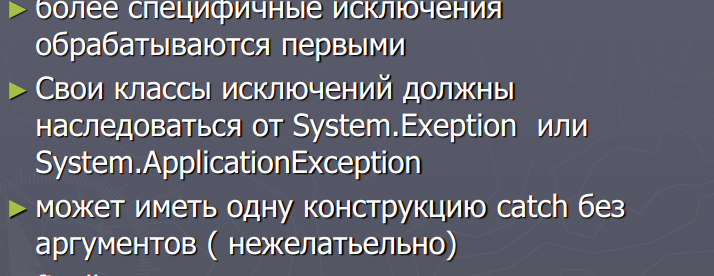
Могут.

Операторы try могут **быть** **вложенными**. Если **вложенный** оператор try не имеет своего обработчика catch для определения **исключения**, то идёт поиск обработчика catch у внешнего блока try и т.д. Если подходящий catch не **будет** найден, то **исключение** обработает сама система (что никуда не годится)

1. Какой синтаксис нужно использовать в C# для отлова любого возможного исключения?

Блок catch, отлавливающий исключения типа System.Exception. Также можно просто использовать catch{}, не указывая тип исключения.

1. Чем следует руководствоваться при размещении обработчиков исключения?



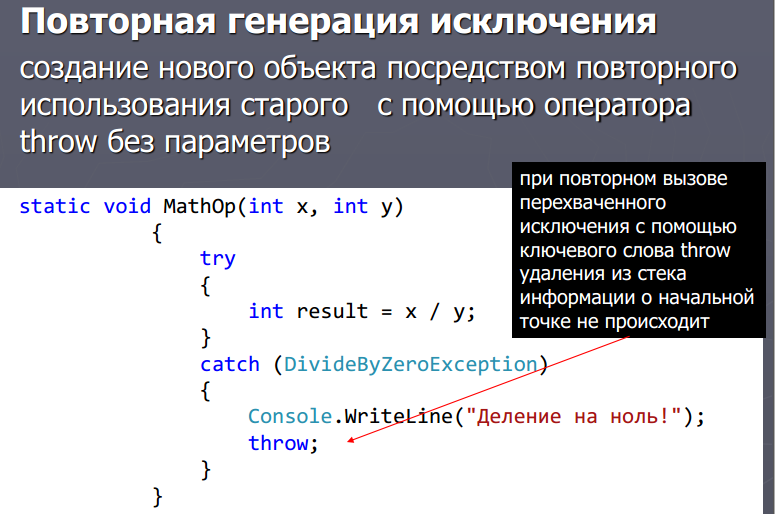
1. Что будет выведено на консоль в результате выполнения фрагмента листинга?

Ответ: It s OK

Массив строк, где 5 элементов

Мы в блоке try вводим 4ый элемент. Выхода за пределы массива нет!+-

11. Как повторно сгенерировать то же самое исключение в блоке обработчике catch?



1. Какие методы содержаться в классе Exception? Где и как их можно использовать?

Базовым для всех типов исключений является тип **Exception**.

ряд методов, с помощью которых можно получить **информацию об исключении**.

* **InnerException**: хранит информацию об исключении, которое послужило причиной текущего исключения
* **Message**: хранит сообщение об исключении, текст ошибки
* **Source**: хранит имя объекта или сборки, которое вызвало исключение
* **StackTrace**: возвращает строковое представление стека вызывов, которые привели к возникновению исключения
* **TargetSite**: возвращает метод, в котором и было вызвано исключение