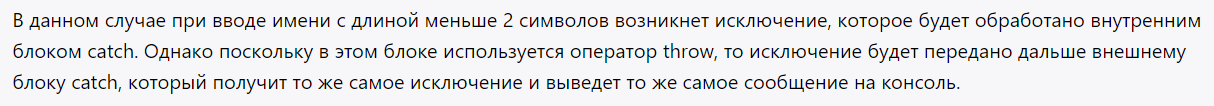
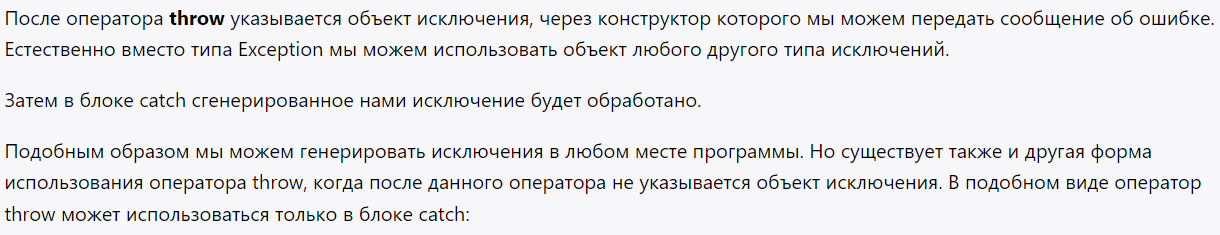
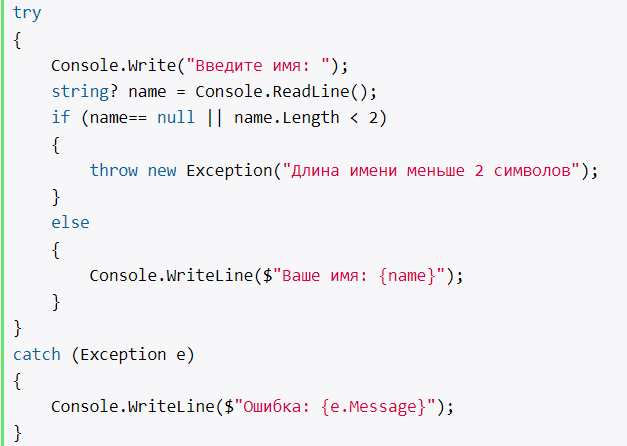
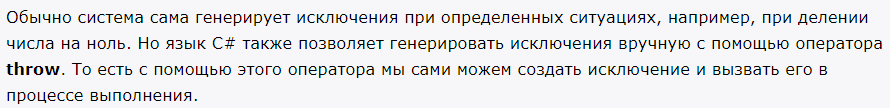
1. Расскажите как генерируется исключение.



2. Расскажите методику обработки исключений.

Обработка исключений позволяет программе корректно обрабатывать ошибки. Для этого используются блоки **try**, **catch**, **finally**. Блок **try** содержит код, который может вызвать исключение. В блоке **catch** указываются обработчики исключений для разных типов исключений. Блок **finally** содержит код, который будет выполняться в любом случае, независимо от того, было исключение или нет.

try – контролируемый блок  
throw – генерация искл. ситуации внутри try  
catch – обработчик исключений (за try неск.)  
finally – код, очищающий ресурсы и другие действия (выполняется всегда, 1 на 1 try)

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

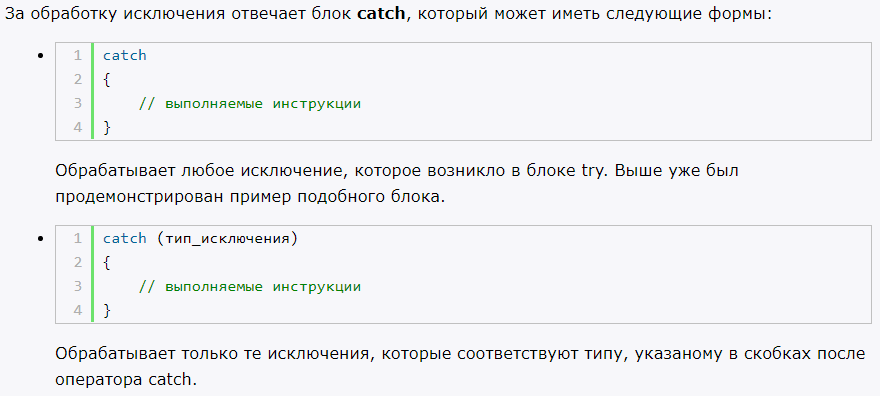
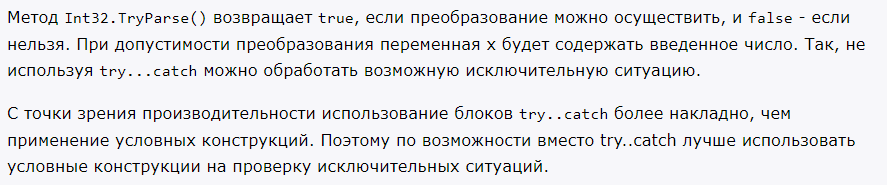
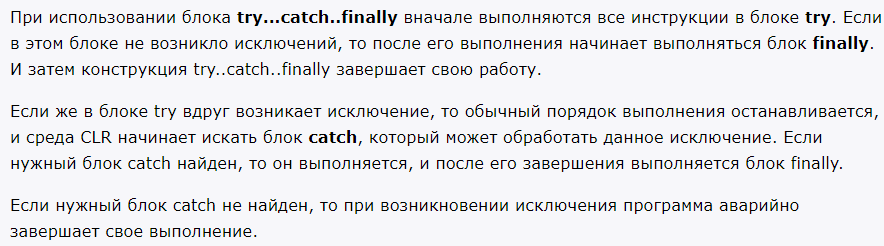
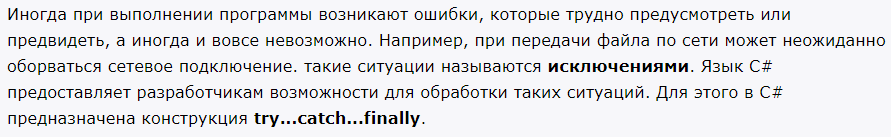
Try

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

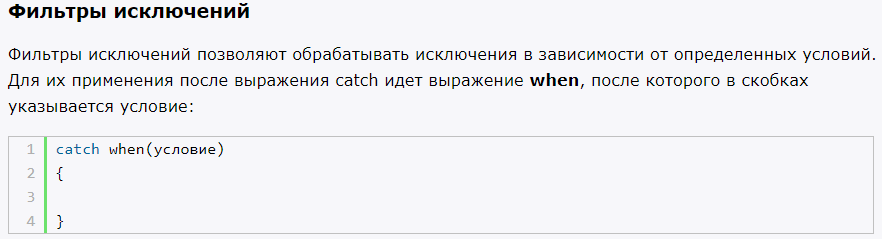
1) Исключения нет  
 try до конца  
 catch пропуск  
 finally вып.  
2) Исключение есть  
 try стоп  
 ищем catch (на соотв. по типу искл)  
\* есть catch:  
 ближ. catch  
 объект-искл в кач. параметра (если предусмотрено)  
 - finally  
 - нет finally -> вып. прогу после посл. catch данного try  
\* нет catch:  
 разматываем стек, лок. объекты выходят из обл. видимости  
 снова генер. искл  
 сообщ. «Необработанное искл», прога стоп

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

try – finally



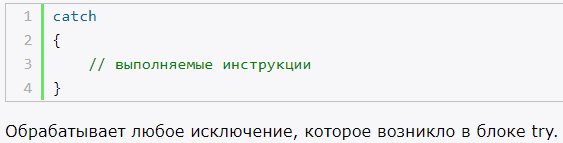
6. Что такое фильтры исключения? Приведите пример.



7. Могут ли исключения быть вложенными?

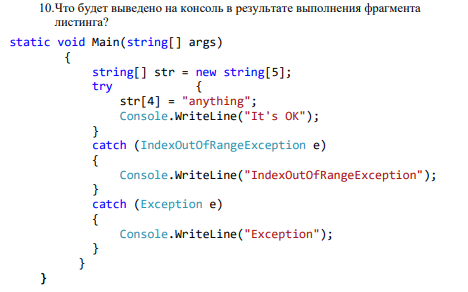
try может (да)

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



9. Чем следует руководствоваться при размещении обработчиков исключения?

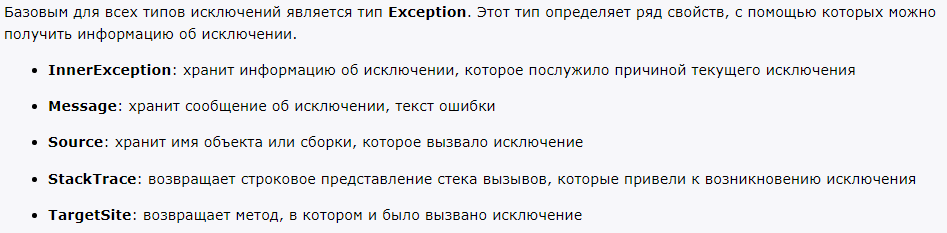
\* сначала более специф искл  
\* д. наслед от System.Exception или System.ApplicationException  
\* мб один catch без аргум (нежелательно)  
\* finally вып всегда  
\* при lock, using, foreach блоки try/finally созд. авто-  
\* генерация искл в finally нежелат

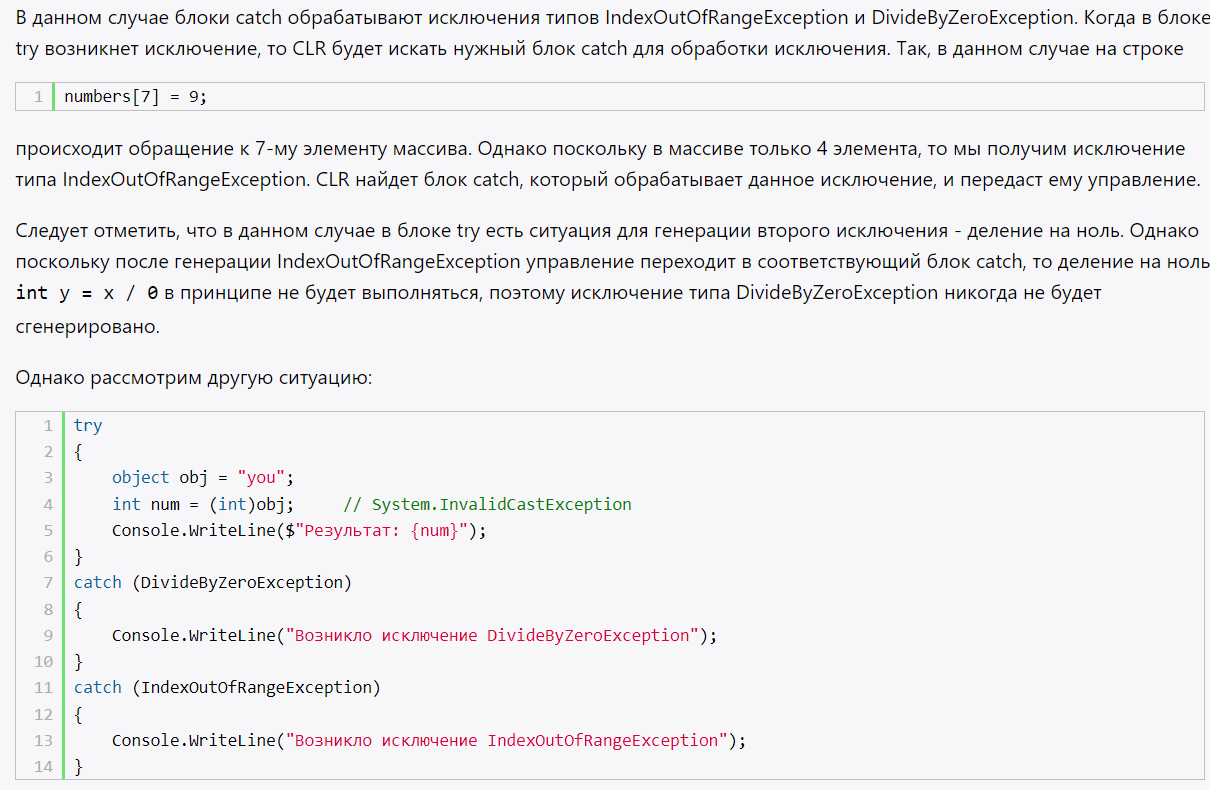
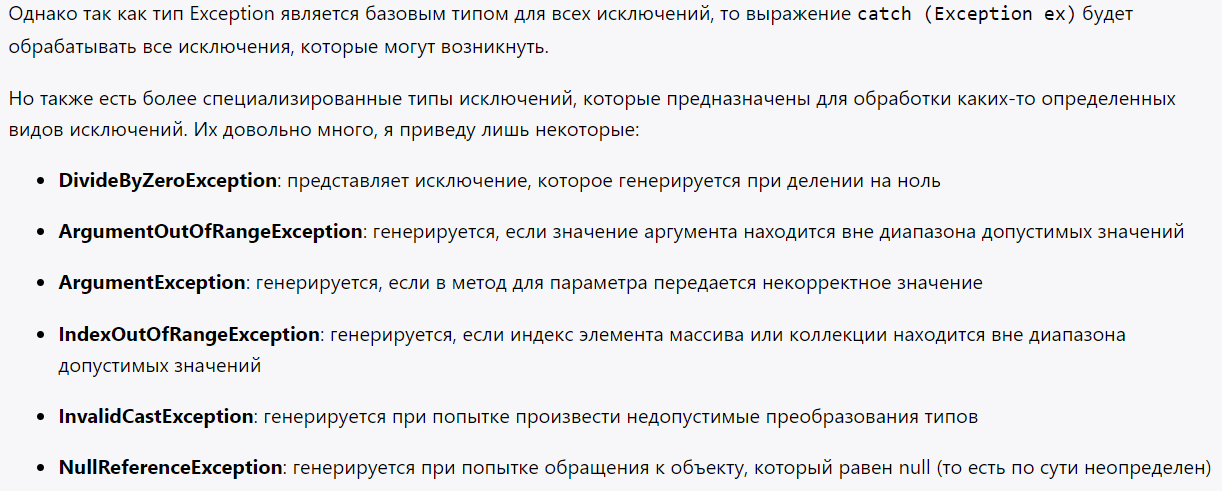
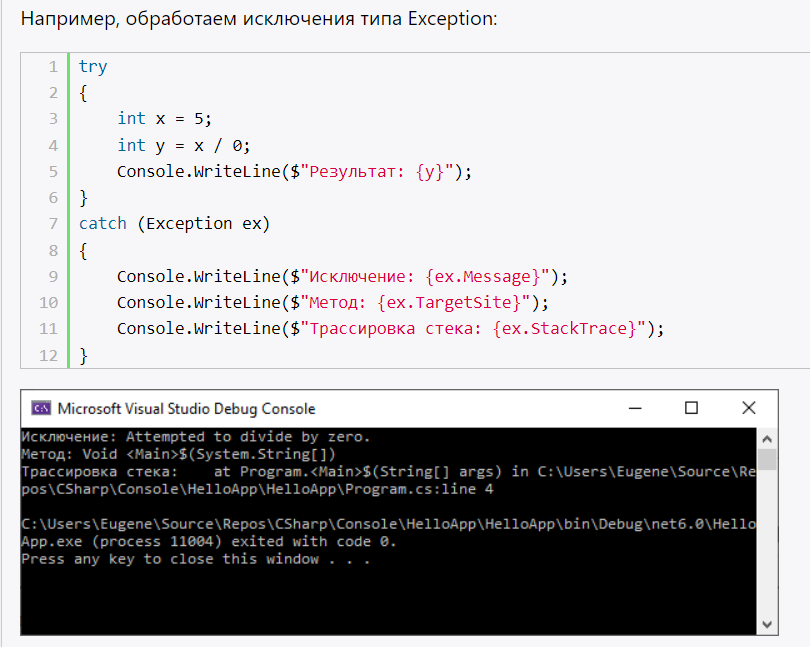


Its OK

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

try  
 int res = x / y  
catch (DivideByZeroException)  
 Console.WriteLine(“Деление на ноль!”);  
 throw; //при повт. вызове исключения с пом. throw  
. //инфа о нач. точке ост. в стеке

12. Какие методы содержатся в классе Exception? Где и как их можно использовать? 



/\*

Добавьте код в одной из функций макрос Assert. Объясните что он

проверяет, как будет выполняться программа в случае не выполнения

условия. Объясните назначение Assert\*/

//Assert проверяет условие, если оно не выполняется, то программа завершается с ошибкой

// все про исключение в этом блоке не будет выполняться

//1. Расскажите как генерируется исключение.

// Исключение генерируется при помощи оператора throw и объекта исключения.

//2. Расскажите методику обработки исключений.

// Обработка исключений происходит при помощи блоков try, catch, finally.

//3. Какое ключевое слово служит для обозначения блока кода, в котором можно генерировать исключение

// throw

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

// try, catch, finally, throw

//5. Что будет, если в программе нет предложения catch, способного обработать исключение?

// Программа завершится с ошибкой

//6. Что такое фильтры исключения? Приведите пример

// Фильтры исключения позволяют обрабатывать исключения по определенным условиям.

//. Могут ли исключения быть вложенными?

// Да, исключения могут быть вложенными.

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

// catch (Exception ex)

//9. Чем следует руководствоваться при размещении обработчиков исключения?

// Руководствоваться следует порядком от наиболее специфичных исключений к наименее специфичным.

//1.Как повторно сгенерировать то же самое исключение в блоке обработчике catch

// throw

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

// Message, Source, StackTrace, HelpLink, TargetSite, Data, InnerException, GetObjectData, ToString, GetType, Equals, GetHashCode, Finalize, MemberwiseClone

//исключение это объект, поэтому можно использовать все методы объекта и класса Exception

//Отличие Parse от TryParse

//Parse - преобразует строку в число, если строка не является числом, то выдает исключение

// TryParse - преобразует строку в число, если строка не является числом, то возвращает false

//как в vs автоматически отформатировать код в vs 2022

// нужно скачать плагин CodeMaid