1. Расскажите как генерируется исключение.

Исключение представляет событие, которое возникает при выполнении программы и нарушает ее нормальной ход. Например, при передаче файла по сети может оборваться сетевое подключение, и в результате чего может быть сгенерировано исключение. Если исключение не обработано, то программа падает и прекращает свою работу. Поэтому при возникновении исключений их следует обрабатывать.

1. Расскажите методику обработки исключений.

Для обработки исключений применяется конструкция try..catch..finally. В блок try помещаются те действия, которые потенциально могут вызвать исключение. Блок catch перехватывает возникшее исключение и обрабатывает его. Блок finally выполняет некоторые завершающие действия

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

try - для выделения фрагмента кода, в котором генерируется исключение.

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

Ключевое слово throw служит для генерации исключений, try - для выделения фрагмента кода, в котором генерируется исключение, а catch - для обработки исключения из предыдущего фрагмента try. При этом блок catch следует сразу после try, а самих блоков catch может быть сколько угодно. Так же есть необязательный блок finally в котором выполнится финальное действие.

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

Прекращение работы программы , а также сообщение об исключении.

1. Что такое фильтры исключения? Приведите пример

try { int result = x / y; } catch(Exception ex) when (y==0) { Console.WriteLine("y не должен быть равен 0"); } catch(Exception ex) { Console.WriteLine(ex.Message); }

1. Могут ли исключения быть вложенными?

Да

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

catch (Exception ex);

1. Чем следует руководствоваться при размещении обработчиков исключения?

try могут быть вложенные

более специфичные исключения обрабатываются первыми

Свои классы исключений должны наследоваться от System.Exeption или System.ApplicationException

Может иметь одну конструкцию catch без аргументов ( нежелатьельно)

Finally выполняется всегда ( не выполняется в случае выброса StackOverflowException или System.exit(0))

Может быть трансляция исключения

При использовании инструкций lock, using и foreach блоки try/finally создаются автоматически

Генерация исключений в finally (нежелательно - код восстановления или очистки будет выполнен не полностью)

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

static void Main(string[] args)

{ string[] str = new string[5];

try { str[4] = "anything";

Console.WriteLine("It's OK");

}

catch (IndexOutOfRangeException e)

{ Console.WriteLine("IndexOutOfRangeException");

}

catch (Exception e)

{ Console.WriteLine("Exception");

} } }

It is OK.

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

try

{

int result = x/y;

}

catch(DivideByZeroException)

{

Console.WriteLine(“Деление на ноль!”);

throw;

}

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

Data Свойство IDictionary, которое содержит произвольные данные в парах "ключ–значение".

HelpLink Может содержать URL-адрес (или URN) файла справки, который предоставляет подробные сведения о причине возникновения  
исключения.

InnerException Это свойство может использоваться для создания и сохранения последовательностей исключений во время обработки исключений. Его можно использовать для создания нового исключения, содержащего ранее перехваченные исключения. Исходное исключение может быть перехвачено вторым исключением в свойстве InnerException, что позволяет коду, обрабатывающему второе исключение,  
проверить дополнительные данные.

Message Предоставляет сведения о причине возникновения исключения.

Source Возвращает или задает имя приложения или объекта, вызывавшего ошибку.

StackTrace Содержит трассировку стека, которую можно использовать для определения места возникновения ошибки. Эта трассировка стека содержит имя исходного файла и, при наличии отладочной информации, номер программной строки.