[1. Расскажите как генерируется исключение.](#_1._Расскажите_как)

[2. Расскажите методику обработки исключений.](#_2._Расскажите_методику)

[3. Какое ключевое слово служит для обозначения блока кода, в котором можно генерировать исключение?](#_3._Какое_ключевое)

[4. Какие ключевые слова используются для обработки и генерации исключений? Расскажите об механизме обработке исключения?](#_4._Какие_ключевые)

[5. Что будет, если в программе нет предложения catch, способного обработать исключение?](#_5._Что_будет,)

[6. Что такое фильтры исключения? Приведите пример](#_6._Что_такое)

[7. Могут ли исключения быть вложенными?](#_7._Могут_ли)

[8. Какой синтаксис нужно использовать в C# для отлова любого возможного исключения?](#_8._Какой_синтаксис)

[9. Чем следует руководствоваться при размещении обработчиков исключения?](#_9._Чем_следует)

[10.Что будет выведено на консоль в результате выполнения фрагмента листинга?](#_10.Что_будет_выведено)

[11.Как повторно сгенерировать то же самое исключение в блоке обработчике catch?](#_11.Как_повторно_сгенерировать)

[12..Какие методы содержаться в классе Exception? Где и как их можно использовать?](#_12..Какие_методы_содержаться)

# 1. Расскажите как генерируется исключение.

1. Исключение генерируется в программе в случае возникновения ошибки или неожиданной ситуации. В языках программирования, поддерживающих исключения, таких как C#, исключение создается при помощи оператора throw.

# 2. Расскажите методику обработки исключений.

2. Методика обработки исключений включает в себя использование блока try-catch. Код, который может вызвать исключение, размещается внутри блока try, и соответствующий код для обработки исключения размещается в блоке catch. Если в блоке try происходит исключение, выполнение программы переходит к ближайшему подходящему блоку catch.

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

3. Ключевое слово для генерации исключения - throw.

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

4. В C# используются ключевые слова try, catch, finally для обработки и генерации исключений. Механизм обработки исключений позволяет программе перехватывать ошибки в рантайме и выполнять соответствующие действия.

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

5. Если в программе нет предложения catch, способного обработать исключение, исключение будет передано на уровень выше в стеке вызовов, и таким образом, будет произведено завершение программы с выводом сообщения об ошибке.

# 6. Что такое фильтры исключения? Приведите пример

6. Фильтры исключений в C# позволяют более точно указывать условия, при которых должен выполняться блок catch. Пример:

try

{

// код, который может вызвать исключение

}

catch (Exception ex) when (ex.Message.Contains("specific"))

{

// код для обработки исключения, удовлетворяющего условиям фильтра

}

# 7. Могут ли исключения быть вложенными?

7. Да, исключения могут быть вложенными. Внутри блока catch можно сгенерировать новое исключение, которое будет обработано в более внешнем блоке catch.

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

8. Для отлова любого возможного исключения в C# используется ключевое слово catch без указания конкретного типа исключения:

try

{

// код, который может вызвать исключение

}

catch

{

// код для обработки любого исключения

}

```

# 9. Чем следует руководствоваться при размещении обработчиков исключения?

9. При размещении обработчиков исключений следует руководствоваться принципом наименьшего уровня детализации (least specific to most specific). То есть, обработчики для более конкретных типов исключений должны идти перед более общими.

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

10. Фрагмент 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");

}

}

Если выполнение кода в блоке try приведет к IndexOutOfRangeException, будет выполнен соответствующий блок catch. Если это исключение не произойдет, но произойдет другое исключение типа Exception, то выполнится второй блок catch.

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

11. Для повторного генерирования того же самого исключения в блоке catch можно использовать ключевое слово throw без аргументов:

catch (Exception ex)

{

// обработка исключения

throw; // повторная генерация того же самого исключения

}

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

12. Класс Exception в C# содержит различные методы, такие как:

- GetMessage(): возвращает сообщение об ошибке.

- GetType(): возвращает тип исключения.

- ToString(): возвращает строковое представление исключения.

Эти методы можно использовать для получения информации об исключении в блоке catch:

```csharp

catch (Exception ex)

{

Console.WriteLine($"Message: {ex.Message}");

Console.WriteLine($"Type: {ex.GetType()}");

Console.WriteLine($"ToString: {ex.ToString()}");

}

```