1. Расскажите как генерируется исключение.

https://metanit.com/sharp/tutorial/2.31.php

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.

1. Расскажите методику обработки исключений.

https://learn.microsoft.com/ru-ru/dotnet/csharp/fundamentals/exceptions/

При обработке исключений используются ключевые слова try, catch и finally для действий, которые могут оказаться неудачными. Это позволяет обрабатывать ошибки так, как кажется разумным, а также правильно высвобождать ресурсы

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

Try

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

Try{}catch(){}finally{} throw new Exeptione(“Error”);

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

Если блок catch не находится, то среда CLR отображает пользователю сообщение о необработанном исключении и останавливает выполнение программы

1. Что такое фильтры исключения? Приведите пример

Catch() when (y == 0){}

1. Могут ли исключения быть вложенными?

Да, это вполне нормальная практика. Чаще всего код не должен заниматься обработкой исключений - у него обычно на это нет соответствующих полномочий.

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

Exception

1. Чем следует руководствоваться при размещении обработчиков исключения?

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");

}

}

}

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

Catch(Exeption ex){

…

Throw;//повторная генерация исключения

}

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

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

Для отладки кода вам **нужны** методы или свойства, которые предоставляются **классом** **Debug** в **C#**. **Класс** **debugger** используется для установки связи с отладчиком

1. Работа с классном Debugger

Нужен для отладки при запуске вне студии. Запускается при помощи команды: Debugger.Launch()

* Метод Debugger.Break()

**Break** работает по правилам:

* Если отладчик подключен, то происходит остановка при вызове методе, сравнимая остановке на breakpoint.
* Если же отладчик не подключен и вызывается метод **Break()**, тогда будет показано окно с выбором отладчика, но при этом закрыв его приложение продолжит работу.

1. Работа с классом debug

**Класс** **Debug** используется для вывода отладочной информации. Информация выводиться только при запуске приложения в режиме **отладки**, т.е. если приложение скомпилировано с константой **DEBUG**. По умолчанию сообщение отображается в окне вывода Visual Studio. Есть возможность организовать вывод сообщений в другие места, например в консоль или журнал Windows.