1. Расскажите как генерируется исключение.

Искл. ситуация – состояние ошибки, обнаруженное в программе в ходе ее вып. (/0, невозм. выдел память…)

if (b==0)  
 throw new Exception(“Zero div”);

2. Расскажите методику обработки исключений.

try – контролируемый блок  
throw – генерация искл. ситуации внутри try  
catch – обработчик исключений (за try неск.)  
finally – код, очищающий ресурсы и др. д-вия (вып. 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.

Фильтр искл. – позв. указать доп. условия, при кот. исп. обработчик исключения  
Эти условия bool, перед ними **WHEN:**

catch (Exception ex) when  
 (ex.GetType() != typeof(System.FormatException))  
 … //обработка исключений  
 //кроме того, кот. называется FormatException

7. Могут ли исключения быть вложенными?

try может

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

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?

1) что такое исключения?  
  
- Исключение AC # - это ответ на исключительное обстоятельство, которое возникает во время работы программы, например попытка деления  
  
на ноль.  
  
- Исключения обеспечивают способ передачи контроля из одной части программы в другую.  
  
- Обработка исключений C # построена на четырех ключевых словах: try , catch , finally и throw .  
  
2) как его сгенироровать?  
  
- ИСКЛ генерируется автоматически, но их также можно генерировать вручную с помощью оператора throw  
  
- 1. Сначала, мы создаем класс с нашим пользовательским исключением (пр., SymbolsException), который наследуется от встроенного Exeption  
  
2. Потом, в нашем нужном классе, вызываем наше исключение при определенном условии. Пример:  
  
if (weight < 0)  
  
throw new SymbolsException("Длина имени меньше 2 символов");  
  
3. После чего оно автоматически вызовется в нашем Program при нарушнии условия weight < 0 или при  
  
catch (SymbolsException)  
  
3) Что мы используем в нашем классе исключения SymbolsException?  
  
- как правило, это конструктор, который в общем случае наследует какой-то метод или параметр у базового типа Exception  
  
class SymbolsException : Exception  
  
{  
  
string SymbolsException(string msg) : base(msg) {}  
  
}  
  
4) если несколько catch относятся к одному try, то вып-ся сначала более специфичные  
  
(пр., OverflowDivideByZero : Exception, потом Exeption)  
  
5) что такое класс Exeption  
  
- Класс Exception — базовый класс всех типов исключений в C#.  
  
6) спрашивает по коду  
  
7) блок try catch finally  
  
try - пробуем процедуру  
  
catch - ловим ошибку и обрабатываем исключение. catch (Exception ex) - параметр это тип обрабатываемого исключения  
  
finally - финальный блок к которому приводят try или catch.  
  
Блок finally можно использовать для высвобождения ресурсов  
  
8) какие catch могут быть?  
  
- хз, вроде обычные (пр., catch (DivideByZeroException) ) и универсальные (пр., catch (Exception ex))  
  
9) они могут идти подряд? - да  
  
10) catch when, что это?  
  
- фильтры исключений. Сначала выполняется when, потом обрабатывается исключение  
  
11) debug -исп-ся для взаимо-я с отладкой  
  
12) debugger исп-ся для взаимо-я с отладкой  
  
13) Debug.Assert(3 > 1) принимает параметр, если он возвращает false то тогда он остановит работу программы и будет функционировать,как точка останова. Если true то программа продолжит работу.

В **C#** **исключение** является объектом, который создается и «выбрасывается» (throw) в случае возникновения ошибки