**1. Расскажите как генерируется исключение.**

**2. Расскажите методику обработки исключений.**

При возникновении исключения среда CLR ищет блок catch, который может обработать данное исключение. Если такого блока не найдено, то пользователю отображается сообщение о необработанном исключении, а дальнейшее выполнение программы останавливается. И чтобы подобной остановки не произошло, и надо использовать блок try..catch.

При использовании блока **try...catch..finally** вначале выполняются все инструкции между операторами try и catch. Если между этими операторами вдруг возникает исключение, то обычный порядок выполнения останавливается и переходит к инструкции сatch. Но если возникшее исключение не является исключением типа, указанного в инструкции сatch, то оно не обрабатывается, а программа просто зависает или выбрасывает сообщение об ошибке.

Далее в любом случае выполняется блок finally. Однако этот блок необязательный, и его можно при обработке исключений опускать. Если же в ходе программы исключений не возникнет, то программа не будет выполнять блок catch, сразу перейдет к блоку finally, если он имеется.

**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 обработке исключения?**

Catch, finally

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

Если такого блока не найдено, то пользователю отображается сообщение о необработанном исключении, а дальнейшее выполнение программы останавливается

**6. Что такое фильтры исключения? Приведите пример**

Общая идея довольно простая: рядом с блоком **catch** появляется возможность задать некоторый предикат, который будет определять, будет ли вызван блок **catch** или нет.

Генерация исключения в CLR происходит следующим образом:

1. Идет поиск ближайшего блока **catch**, который удовлетворяет типу генерируемого исключения.

2. Исполняется предикат фильтра исключения. Если предикат возвращает **true**, то данный блок **catch**считается подходящим и начинается раскрутка стека и выполнение всех блоков **finally** на пути от места генерации исключения к обработчику.

3. Если фильтр исключения возвращает **false**, то поиск подходящего блока **catch** продолжается.

**7. Могут ли исключения быть вложенными?**

Один блок try может быть вложен в другой. Исключение, генерируемое во внутреннем блоке try и не перехваченное в соответствующем блоке catch, передается во внешний блок try

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

Catch{}

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

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

It’s Ok

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

Если возникает необходимость снова сгенерировать исключения из блока, который обрабатывает исключения, можно сделать это путем вызова throw без указания исключения. В результате теку­щее исключение будет передано во внешнюю последовательность try/catch обработки исключений. Причиной для этого может послужить желание обрабатывать исключения несколькими обработ­чиками. Например, один обработчик может заниматься одним аспектом исключения, а второй обработчик — другим. Исключение может быть снова сгенерировано или изнутри блока catch, или из функции, вызванной в этом блоке. Когда повторно генерируется исключение, оно не бу­дет перехвачено той же самой инструкцией catch. Оно будет распространяться до следующей внешней инструкции catch.

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

**public** **virtual** Exception **GetBaseException**() ;

**public** **virtual** **void** **GetObjectData**(Serializationlnfо info, StreamingContext context);