1. Расскажите как генерируется исключение.

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.

 throw new Exception("Длина имени меньше 2 символов");

2. Расскажите методику обработки исключений.

Иногда при выполнении программы возникают ошибки, которые трудно предусмотреть или предвидеть, а иногда и вовсе невозможно. Например, при передачи файла по сети может неожиданно оборваться сетевое подключение. такие ситуации называются **исключениями**. Язык C# предоставляет разработчикам возможности для обработки таких ситуаций. Для этого в C# предназначена конструкция **try...catch...finally**.

При использовании блока **try...catch..finally** вначале выполняются все инструкции в блоке **try**. Если в этом блоке не возникло исключений, то после его выполнения начинает выполняться блок **finally**. И затем конструкция try..catch..finally завершает свою работу.

Если же в блоке try вдруг возникает исключение, то обычный порядок выполнения останавливается, и среда CLR начинает искать блок **catch**, который может обработать данное исключение. Если нужный блок catch найден, то он выполняется, и после его завершения выполняется блок finally.

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

Catch

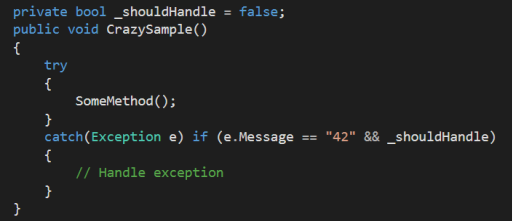
4. Какие ключевые слова используются для обработки и генерации исключений? Расскажите об механизме обработке исключения?

==2

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

Если в программе нет предложения catch, способного обработать исключение, оно остается необработанным. Но исключение – это настолько серьезная ошибка, что программа не может продолжать выполнение. Так что будет вызвано исключение автоматически.

6. Что такое фильтры исключения? Приведите пример



\*в финальной версии языка C# вместо **if** будет использоваться ключевое слово **when**.

7. Могут ли исключения быть вложенными?

Да, могут.

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

Try – Catch (???)

\*Исключения представляют собой типы, производные в конечном счете от System.Exception.

9. Чем следует руководствоваться при размещении обработчиков исключения?

Здравым умом, не генерировать исключения повторно (при вложенности внутренний try-catch может перехватить и проглатить внешнее исключение), стараться избегать исключений (Но многих методов, генерирующих исключение, можно избежать с помощью защитного программирования), создавать пользовательские исключения (для лучшей читаемости кода), логировать исключения.

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

Its OK

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



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

getMessage()

getStackTrace()

printStackTrace()

