1. Расскажите как генерируется исключение.

Ответ:

Исключение – это аномальное поведение во время выполнения, которое программа может обнаружить, например: деление на 0, выход за границы массива или истощение свободной памяти. Такие исключения нарушают нормальный ход работы программы, и на них нужно **немедленно отреагировать.**

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.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Под генерацией исключений подразумевается **создание объектов, описывающих исключительные ситуации, и информирование (с их помощью) системы о том, что произошла нештатная ситуация во время выполнения программы.**

Оператор **throw** информирует систему, прерывает выполнение программы, и «выбрасывает» объект специфического класса (класса «*Exception*» из пространства имен «*System»*, либо его наследника) с информацией о причине сбоя.

Изображение выглядит как текст

Автоматически созданное описание

2. Расскажите методику обработки исключений.

Ответ:

Иногда при выполнении программы возникают ошибки, которые трудно предусмотреть или предвидеть, а иногда и вовсе невозможно. Например, при передаче 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.**

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

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

Ответ:

Ключевое слово try.

Изображение выглядит как текст

Автоматически созданное описание

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

Ответ:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

**Поиск подходящего блока catch в CLR осуществляется сверху вниз, поэтому**

**наиболее конкретные обработчики должны находиться в начале списка.**

Изображение выглядит как текст

Автоматически созданное описание

Блок try может существовать и без блока finally, ведь иногда его код просто

не требует последующей очистки. **Однако если вы решили создать блок finally,**

**его следует поместить после всех блоков catch. И помните, что одному try может**

**соответствовать только один блок finally.**

**Запомните, что в блок finally помещается код для выполнения**

**завершающей очистки. И он должен выполнять только те действия, которые не-**

**обходимы для отмены операций, начатых в блоке try.**

Изображение выглядит как текст

Автоматически созданное описание

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

Ответ:

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

Если нужный блок catch не найден, то при возникновении исключения программа аварийно завершает свое выполнение.

6. Что такое фильтры исключения? Приведите пример

Ответ:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

7. Могут ли исключения быть вложенными?

Ответ:

**С одним оператором try можно связать несколько операторов catch**. И на практике это делается довольно часто. **Но все операторы catch должны перехватывать исключения разного типа.**

Изображение выглядит как текст, табличка

Автоматически созданное описание

**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.**

Изображение выглядит как текст

Автоматически созданное описание

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

Ответ:

Изображение выглядит как текст

Автоматически созданное описание

9. Чем следует руководствоваться при размещении обработчиков исключения?

Ответ:

**В блоках catch следует всегда упорядочивать исключения от более производных к менее производным.**

**Сначала следуют потомки с наибольшей глубиной наследования, потом —их базовые классы (если таковые имеются) и, наконец, —класс System.Exception**

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

Изображение выглядит как текст

Автоматически созданное описание

Ответ:

Будет выведено «It’s OK».

IndexOutOfRangeException - Исключение, которое выдается при попытке доступа к элементу массива или коллекции с индексом, выходящим за его границы.

В блоки catch выполнение никогда не зайдет. Мы создаем массив строк на 5 элементов. Индексация в массиве с 0, т.е 0, 1, 2, 3, 4 элемента. Обращаемся к элементу с индексом 4 – все хорошо.

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

Ответ:

**Оператор [throw](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/throw), включенный в блок catch**, позволяет заново вызвать исключение, перехваченное блоком catch.

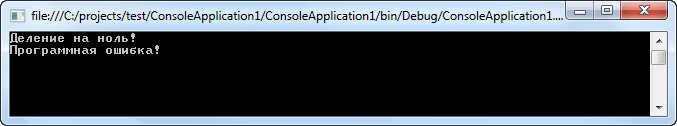
Изображение выглядит как текст

Автоматически созданное описание

Исключение, перехваченное в одном блоке catch, может быть повторно сгенерировано в другом блоке, чтобы быть перехваченным во внешнем блоке catch.

Изображение выглядит как текст

Автоматически созданное описание



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

Ответ:

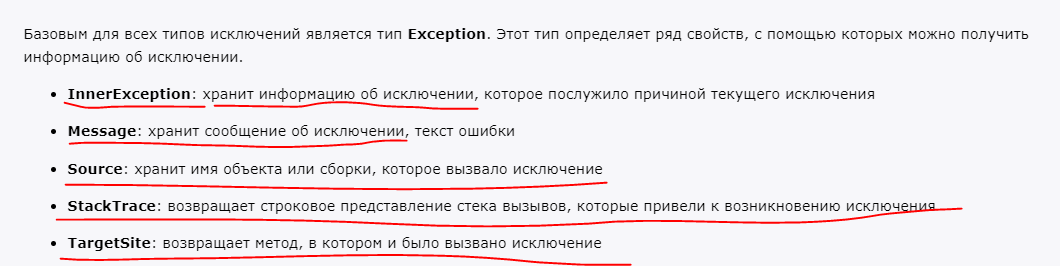
**Все определяемые на уровне пользователя и системы исключения в конечном итоге всегда наследуются от базового класса System.Exception, который, в свою очередь, наследуется от класса System.Object.**

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание



Изображение выглядит как текст

Автоматически созданное описание



Исключения, которые генерируются самой платформой .NET, называются **исключениями уровня системы**. **Эти исключения считаются неустранимыми фатальными ошибками.** Они наследуются прямо от базового класса System.SystemException, который, в свою очередь, наследуется от System.Exception (а тот — от класса System.Object):

Поскольку все исключения .NET представляют собой типы классов, вполне допускается **создавать собственные исключения, предназначенные для конкретного приложения.**

ОБЩИЙ МЕХАНИЗМ  
  
Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст, табличка, снимок экрана

Автоматически созданное описание