1. Что такое обобщение (generic)?

Термин *обобщение* означает параметризированный тип.

Обобщения позволяют решить проблему упаковки и распаковки а также повысить производительность приложения и справится с проблемой безопасности типов.

2. нету

3. Как можно наложить определенное ограничение на параметр?

С помощью ключевого слова where T : (class, struct, name\_class, name\_class<T>, new() – это что-то, что имеет конструктор без параметров).

4. Как можно наложить несколько ограничений на параметр?

Через ,

class Folf<T> where T: class, new()

5. Перечислите все существующие ограничения на типы данных обобщения?

class,struct, конкретный класс или интерфейс, new()

6.тип Т может быть только А либо его потомком B

7.тип Т можеть быть только классом

8.тип Т может быть только структурой

9. Приведите примеры, когда обобщенный класс может действовать как базовый или производный класс.

Class Dog<T, V>: Folf<T> (Folf<T> - базовый)

T – параметр передаваемый базовому классу

V – параметр характерный только для производного класса0

Class Folf<T> : Animal (Folf<T> - производный)

10. В каких случаях в обобщениях может использоваться оператор default?

Когда в обобщении нужно присвоить пере менным некоторое начальное значение, то на прямую это сделать нельзя, поэтому используется ключевое слово default(T)

Он присваивает ссылочным типам значение null, а значимым типам 0.

11. Поясните как использовать статические переменные в обобщенных классах.

В обобщённых классах определённым образом создаётся набор статических членов и его можно инициализировать без создания объекта

Car<T>

{

Public static T speed;

}

Car<string>.speed = “fast”;

Car<int>.speed = 50;

12. Приведите пример обобщенного интерфейса.

Interface Numbers<T>

{

Void Show();

Void Add();

}

13. В чем отличие обобщенных классов от обобщенных структур?

Они похожи но структуры нельзя наследовать.

14. Какие классы для работы с файлами вы знаете? Приведите пример

File, fileInfo

С их помощью мы можем создавать, удалять, перемещать файлы, получать их свойства и многое другое.

Некоторые полезные методы и свойства класса FileInfo:

* **CopyTo(path)**: копирует файл в новое место по указанному пути path
* **Create()**: создает файл
* **Delete()**: удаляет файл
* **MoveTo(destFileName)**: перемещает файл в новое место
* Свойство **Directory**: получает родительский каталог в виде объекта DirectoryInfo
* Свойство **DirectoryName**: получает полный путь к родительскому каталогу
* Свойство **Exists**: указывает, существует ли файл
* Свойство **Length**: получает размер файла
* Свойство **Extension**: получает расширение файла
* Свойство **Name**: получает имя файла
* Свойство **FullName**: получает полное имя файла

Класс File реализует похожую функциональность с помощью статических методов:

* **Copy()**: копирует файл в новое место
* **Create()**: создает файл
* **Delete()**: удаляет файл
* **Move**: перемещает файл в новое место
* **Exists(file)**: определяет, существует ли файл