1. Что такое обобщение (generic)?

Обобщение – параметризированный тип. Обобщения позволяют решить проблему упаковки и распаковки, а также повысить производительность приложения.

1. Пусть дан фрагмент листинга. В какой строчке содержится ошибка?

Ни в какой.

1. Как можно наложить определенное ограничение на параметр?

При помощи ключевого слова where.

1. Как можно наложить несколько ограничений на параметр?

Class Prim<T> where T: class, new()

5. Перечислите все существующие ограничения на типы данных

обобщения?

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

6. Какое ограничение на тип задано в следующем фрагменте листинга?

class A { }

class В : A { }

class С { }

class Test<T> where T : A { }

Ответ: Т может быть либо объектом класса А, либо В

7. Какое ограничение на тип задано в следующем фрагменте листинга?

interface A { }

class Test<T> where T : class { }

Ответ: тип Т может быть только классом

8. Какое ограничение на тип задано в следующем фрагменте листинга?

interface A { }

class Test<T> where T : struct { }

Ответ: Тип Т может быть лишь структурой.

9. Приведите примеры, когда обобщенный класс может действовать как

базовый или производный класс.

Class Little<T,V>: Big<T>

Class Big<T>: Prim

10.В каких случаях в обобщениях может использоваться оператор

default?

Чтобы присвоить переменным некоторое начальное значение в обобщении.

Присваивает ссылочным типам null а значимым 0.

11.Поясните как использовать статические переменные в обобщенных

классах.

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?

Структуры нельзя наследовать.

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

File, Fileinfo.

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