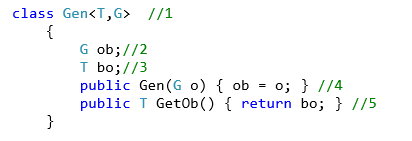
1. Что такое обобщение (generic)?

Обобщение — средство языка C#, позволяющее создавать программный код, содержащий единственное (типизированное) решение задачи для различных типов, с его последующим применением для любого конкретного типа (int, float, char и т.д.).

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



**В 5 строке (???)**

3. Как можно наложить определенное ограничение на параметр?

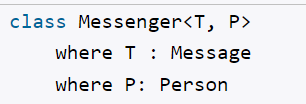
Ограничения методов указываются после списка параметров после оператора **where**:

|  |  |
| --- | --- |
|  |  |

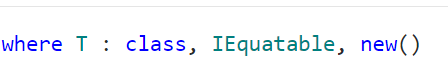
После оператора **where** указывается универсальный параметр, для которого применяется ограничение. И через двоеточие указывается тип ограничения - обычно в качестве ограничения выступает конкретный тип.

4. Как можно наложить несколько ограничений на параметр?

Если класс использует несколько универсальных параметров, то последовательно можно задать ограничения к каждому из них:



Так же мы можем на один параметр накладывать сразу несколько ограничений через запятую.



5. Перечислите все существующие ограничения на типы данных обобщения?

* where T: struct — ограничение типа-значения. Параметр типа должен быть унаследован от System.ValueType, то есть быть структурным типом;
* where T: class — ограничение типа-ссылки. Параметр типа должен быть ссылочного типа, то есть не должен быть унаследован от System.ValueType;
* where T: new() — параметр типа должен иметь конструктор без параметров (конструктор по умолчанию);
* where T: BaseClass — параметр типа должен быть классом BaseClass или классом, производным от него;
* where T: Interface — параметр типа должен реализовывать интерфейс с именем Interface.

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

class A { }

class В : A { }

class С { }

class Test<T> where T : A { }

**То есть внутри класса Test все объекты типа T можно использовать как объекты Message**

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

interface A { }

class Test<T> where T : class { }

**Ограничение типа-ссылки. Параметр типа должен быть ссылочного типа, то есть не должен быть унаследован от System.ValueType;**

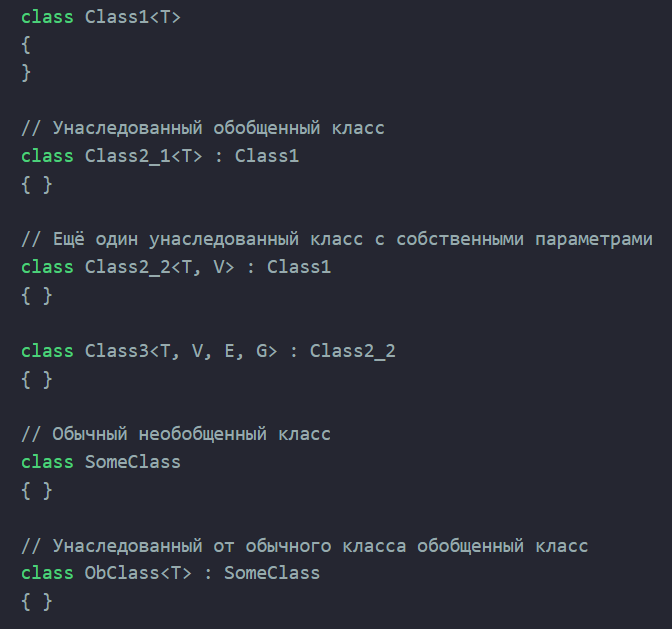
8. Какое ограничение на тип задано в следующем фрагменте листинга?

interface A { }

class Test<T> where T : struct { }

**Ограничение типа-значения. Параметр типа должен быть унаследован от System.ValueType, то есть быть структурным типом.**

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

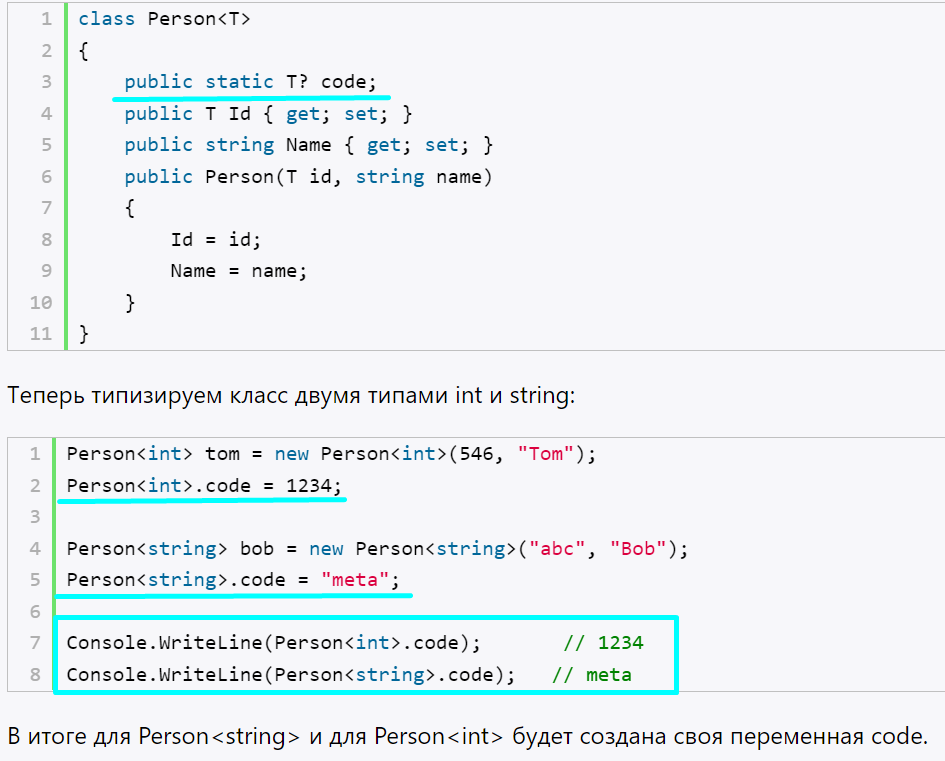


10. В каких случаях в обобщениях может использоваться оператор default?

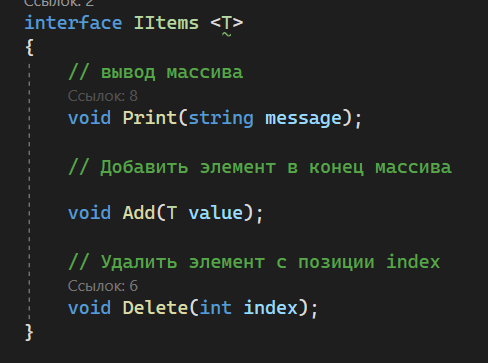
Получить обобщенное значение по умолчанию для параметра типа можно благодаря использованию ключевого слова default.

11. Поясните как использовать статические переменные в обобщенных классах.

Лучше всего поясняет это следующий фрагмент кода:



12. Приведите пример обобщенного интерфейса.

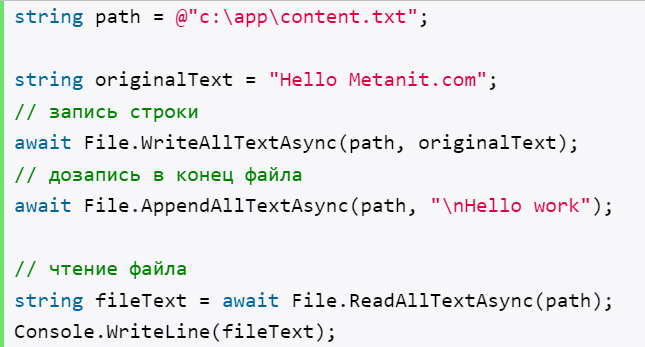


13. В чем отличие обобщенных классов от обобщенных структур?

Вероятнее всего, те же, что и между классом и структурой. Класс – является ссылочным типом (reference type), а структуры – значимый класс (value type). А следовательно, классы всегда создаются в, так называемой, “куче” (heap), а структуры создаются в стеке (stack).

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

* StreamWriter writer = new (path, false)// StreamReader reader = new StreamReader(path) (используя using)
* **File** и **FileInfo**.



* Класс **FileStream** представляет возможности по считыванию из файла и записи в файл. Он позволяет работать как с текстовыми файлами, так и с бинарными. *где fileMode – режим доступа*
* Класс file

