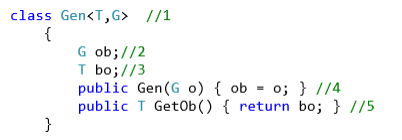
1. Что такое обобщение (generic)?

Обобщение (Generics) в C# — это механизм, который позволяет создавать обобщенные (generic) типы и методы, то есть такие, которые могут работать с данными любого типа. <T> является параметром типа и представляет собой тип данных, который будет определен или указан при создании экземпляра класса, вызове метода, или использовании интерфейса.

Формально мы говорим, что Stack <T>— это открытый (open) тип, а Stack <int>— закрытый (closed) тип.

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



В строке 5 вы пытаетесь вернуть значение переменной bo, которая не была инициализирована, что может вызвать проблемы.

3. Как можно наложить определенное ограничение на параметр?

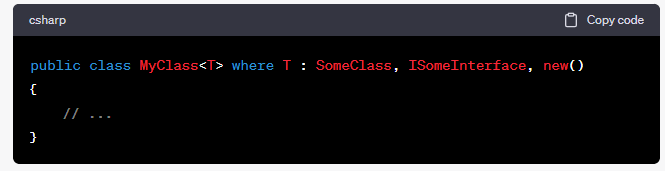
Ограничения методов указываются после списка параметров после оператора where:



После оператора where указывается универсальный параметр, для которого применяется ограничение. И через двоеточие указывается тип ограничения.

4. Как можно наложить несколько ограничений на параметр?

Вы можете комбинировать ограничения:



В этом примере T должен быть классом, реализующим ISomeInterface и имеющим открытый конструктор без параметров.

5. Перечислите все существующие ограничения на типы данных обобщения?

1. \*\*where T : **struct**\*\*:

- Это ограничение требует, чтобы тип T был значимым типом данных (структурой).

2. \*\*where T : **class**\*\*:

- Требует, чтобы тип T был ссылочным типом (классом).

3. \*\*where T : **new**()\*\*:

- Задает, что тип T должен иметь общедоступный конструктор без параметров.

4. \*\*where T : <base class name>\*\*:

- Указывает, что тип T должен быть производным от указанного базового класса.

5. \*\*where T : <interface name>\*\*:

- Требует, чтобы тип T реализовывал указанный интерфейс.

6. \*\*where T : U\*\*:

- Ограничивает обобщение T типом U, что подразумевает, что T должен быть таким же или производным от U.

7. \*\*where T : **notnull**\*\*:

- Добавлено в C# 8.0, указывает, что тип T не может быть значением null.

8. \*\*where T : unmanaged\*\*:

- Добавлено в C# 7.3, требует, чтобы тип T был неуправляемым типом

9. \*\*where T : enum\*\*:

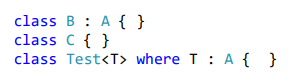
- Требует, чтобы тип T был перечислением.

10. \*\*where T : delegate\*\*:

- Требует, чтобы тип T был делегатом.

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





В данном фрагменте листинга ограничение на тип `T` задано с использованием ключевого слова `where` и выражения `where T : A`. Это означает, что параметр типа `T` должен быть либо самим классом `A`, либо его производным классом.

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



Тип `T` должен быть ссылочным типом данных. (объекты, строки, классы, интерфейсы, делегаты, массивы)

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



Тип `T` должен быть структурой или другим значимым типом.

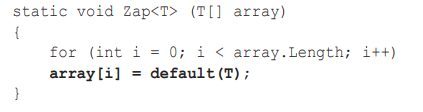
9. Приведите примеры, когда обобщенный класс может действовать как базовый или производный класс.

Обобщенные классы могут входить в иерархию классов аналогично необобщенным классам. Следовательно, обобщенный класс может действовать как базовый или производный класс. Главное отличие между иерархиями обобщенных и необобщенных классов заключается в том, что в первом случае **аргументы типа, необходимые обобщенному базовому классу, должны передаваться всеми производными классами вверх по иерархии .**

**В производном классе следует непременно указывать параметры типа, требующиеся его обобщенному базовому классу**, даже если этот производный класс не обязательно должен быть обобщенным. Разумеется, в производный класс можно свободно добавлять его собственные параметры типа, если в этом есть потребность.

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

Оператор default в обобщениях используется для получения значения по умолчанию для типа данных параметра обобщения. Это полезно в ситуациях, когда вы хотите инициализировать переменную обобщенного типа значением по умолчанию, независимо от того, какой тип используется.





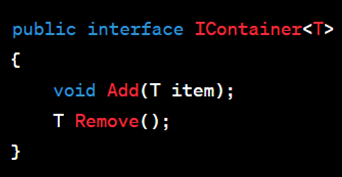
1. Для ссылочных типов (`T` - классы или интерфейсы) `default(T)` будет равен `null`.

2. Для значимых типов (`T` - структуры, числовые типы, enum и т. д.) `default(T)` зависит от конкретного значимого типа. Например, для числовых типов это будет эквивалентно `0`, для `struct` - это будет структура с нулевыми полями, а для `enum` - это будет первый элемент перечисления (если не указано явно другое значение).

11.Поясните как использовать статические переменные в обобщенных классах.

В обобщённых классах в C# статические переменные разделяются между всеми экземплярами обобщённого класса, основанными на одном и том же параметре типа. Каждый параметр типа приводит к созданию отдельной копии статических переменных.

12.Приведите пример обобщенного интерфейса.



13.В чем отличие обобщенных классов от обобщенных структур?

Обобщенные классы могут иметь наследование, поддерживать интерфейсы и так далее, по умолчанию null.

Обобщенные структуры не могут использовать наследование, более эффективны с точки зрения использования памяти, так как хранятся в стеке и не требуют сборки мусора.

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

1. \*\*`System.IO.File` и `System.IO.FileInfo`:\*\*

2. \*\*`System.IO.Directory` и `System.IO.DirectoryInfo`:\*\*

3. \*\*`System.IO.Path`:\*\*

4. \*\*`System.IO.StreamReader` и `System.IO.StreamWriter`:\*\*