Обобщение (generic) — это механизм в C#, который позволяет создавать классы, структуры, методы, интерфейсы и делегаты с параметрами типа. Это повышает гибкость и повторное использование кода, сохраняя типобезопасность.

**Пример:**

csharp

Копировать код

class GenericClass<T>

{

public T Value { get; set; }

}

**2. Ограничения на типы данных обобщения (кратко):**

1. **class** — тип должен быть ссылочным.
2. **struct** — тип должен быть значимым.
3. **new()** — тип должен иметь конструктор без параметров.
4. **Базовый класс (BaseClass)** — тип должен наследоваться от указанного класса.
5. **Интерфейс (IInterface)** — тип должен реализовывать указанный интерфейс.
6. **Другой обобщенный параметр (TAnother)** — тип должен соответствовать другому обобщенному параметру.
7. **Как можно наложить определенное ограничение на параметр?**  
   Для наложения ограничения используют ключевое слово where с типом параметра:

csharp

Копировать код

class Test<T> where T : new() { }

Пример ограничения: тип T должен иметь конструктор без параметров (new()).

1. **Как можно наложить несколько ограничений на параметр?**  
   Используйте несколько условий через пробел:

csharp

Копировать код

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

{

// Ограничения: T — ссылочный тип и имеет конструктор без параметров.

}

1. **Перечислите все существующие ограничения на типы данных обобщения:**

* class: Тип должен быть ссылочным.
* struct: Тип должен быть значимым.
* new(): Тип должен иметь конструктор без параметров.
* Специфический базовый класс: Тип должен наследовать конкретный класс.
* Интерфейс: Тип должен реализовывать указанный интерфейс.
* Другой обобщенный параметр: Тип должен соответствовать другому обобщенному параметру.

1. **Какое ограничение на тип задано в следующем фрагменте листинга?**

csharp

Копировать код

class A { }

class B : A { }

class C { }

class Test<T> where T : A { }

**Ответ:** Ограничение T : A означает, что тип T должен быть A или наследником класса A (например, B).

1. **Какое ограничение на тип задано в следующем фрагменте листинга?**

csharp

Копировать код

interface A { }

class Test<T> where T : class { }

**Ответ:** Ограничение T : class означает, что тип T должен быть ссылочным типом (класс, интерфейс, делегат или массив).

1. **Какое ограничение на тип задано в следующем фрагменте листинга?**

csharp

Копировать код

interface A { }

class Test<T> where T : struct { }

**Ответ:** Ограничение T : struct означает, что тип T должен быть значимым типом (например, int, float, bool, DateTime и т.д.).

1. **Примеры обобщенного класса как базового или производного:**

* **Обобщенный класс как базовый:**

csharp

Копировать код

class Base<T> { }

class Derived<T> : Base<T> { }

* **Обобщенный класс как производный:**

csharp

Копировать код

class Base { }

class Derived<T> : Base { }

1. **В каких случаях в обобщениях может использоваться оператор default?**  
   Оператор default используется для получения значения по умолчанию для типа T:

* Для ссылочных типов — это null.
* Для значимых типов — это 0 (или эквивалент).  
  Пример:

csharp

Копировать код

T value = default(T);

1. **Как использовать статические переменные в обобщенных классах?**  
   Каждый тип T в обобщенном классе создает свою копию статической переменной:

csharp

Копировать код

class GenericClass<T>

{

public static int Counter;

}

GenericClass<int>.Counter = 1;

GenericClass<string>.Counter = 2;

1. **Пример обобщенного интерфейса:**

csharp

Копировать код

interface IRepository<T>

{

void Add(T item);

T GetById(int id);

}

class UserRepository : IRepository<User>

{

public void Add(User item) { /\* Реализация \*/ }

public User GetById(int id) { /\* Реализация \*/ return new User(); }

}

1. **В чем отличие обобщенных классов от обобщенных структур?**

* Классы являются ссылочными типами, а структуры — значимыми.
* Обобщенные структуры используются для небольших наборов данных, тогда как классы лучше для сложных объектов.
* При использовании структуры с обобщением передача данных осуществляется через копирование, а для классов — через ссылки.

1. **Какие классы для работы с файлами вы знаете? Пример:**

* File: Статические методы для операций с файлами.
* FileStream: Чтение/запись потоков байтов.
* StreamReader / StreamWriter: Чтение/запись текстовых данных.
* BinaryReader / BinaryWriter: Чтение/запись бинарных данных.
* Пример:

csharp

Копировать код

using (StreamWriter writer = new StreamWriter("file.txt"))

{

writer.WriteLine("Hello, File!");

}