Контрольні запитання

**1. Синтаксис оголошення інтерфейсу**

Інтерфейс оголошується за допомогою ключового слова interface:

public interface IExample

{

void DoWork();

int Value { get; set; }

}

Назви інтерфейсів зазвичай починаються з літери I.

**2. Що може містити інтерфейс**

Інтерфейс може містити:

* **методи** без реалізації (до C# 8.0);
* **властивості**, індексатори, події;
* **статичні або дефолтні реалізації** (з C# 8.0 і вище — за умови підтримки компілятором);
* **НЕ МОЖЕ містити** змінних або полів стану (але може — константи, якщо вони статичні й const або static readonly).

3. Відмінності інтерфейсу від абстрактного класу

|  |  |  |
| --- | --- | --- |
| **Порівняння** | **Інтерфейс** | **Абстрактний клас** |
| Наслідування | Можна реалізовувати багато | Тільки один базовий клас |
| Поля | Немає | Може містити поля |
| Реалізація методів | Немає (до C# 8.0) | Може містити реалізацію |
| Конструктори | Немає | Може мати конструктори |
| Призначення | Контракт дій | Частковий шаблон реалізації |

**4. Стандартні інтерфейси .NET**

Найуживаніші:

* IComparable / IComparable<T> — для сортування;
* ICloneable — для копіювання об’єктів;
* IEnumerable, IEnumerator — для перебору в foreach;
* IEquatable<T> — порівняння об’єктів;
* IDisposable — для вивільнення ресурсів (using);
* IFormattable — для кастомного форматування рядків;
* IList<T>, ICollection<T> — для колекцій.

**5. Коли використовуються is та as**

* is — перевірка, чи об'єкт належить до певного типу:

**if** (obj **is** IEngine engine)

{

Console.WriteLine($"Двигун: {engine.EngineType}");

}

* as — спроба приведення типу з поверненням null, якщо не вдалося:

IEngine engine = obj **as** IEngine;

**if** (engine != null)

{

Console.WriteLine(engine.Power);

}

is безпечніше в нових версіях C#, бо можна одразу отримати змінну.