1. **類別與介面的角色**

**抽象類別 (abstract class)**: 是一個可以包含具體實現（method body）和抽象方法（沒有實現）的類別。抽象類別通常用來提供共通的基礎功能，並且可以被其他類別繼承來實現或重寫具體方法。

**介面 (interface)**: 是一種純粹的契約，定義了類別必須實現的行為（方法、屬性、事件等），但不提供具體實現。介面不能包含任何方法的實作，只能定義方法的簽名。

1. 繼承方式

**抽象類別**: 只能繼承一個抽象類別，這是因為 C# 不允許多重繼承。你可以繼承抽象類別並實現它的抽象方法。

**介面**: 類別可以實現多個介面。這允許在 C# 中進行多重介面實現，彌補了單一繼承的限制。(可以多重繼承)

1. 使用場景

**抽象類別:** 當你需要一個基礎類別，並且希望在多個類別中共享一些具體的實現邏輯時，應該使用抽象類別。抽象類別可以有一些實現方法，也可以包含抽象方法，這樣派生類別可以選擇覆寫或繼承這些方法。

**介面**: 當你需要定義一個跨多個不相關類別的契約時，應該使用介面。介面更適合定義不依賴於具體類別層次結構的功能。它強調的是行為，而非實現。

1. 建構子

**抽象類別**: 可以有構造函數，並且可以定義它的初始化邏輯。

**介面**: 不能包含構造函數，因為介面只是定義契約，並不直接創建對象

1. 特性

**抽象類別**: 可以擁有字段、屬性、事件和方法，並且可以在方法內提供實現。

**介面**: 只能定義方法簽名和屬性等，無法提供實現（除非是 C# 8.0 的預設方法）。

對比表

| **特性** | **抽象類別 (abstract class)** | **介面 (interface)** |
| --- | --- | --- |
| 目的 | 定義共有的基類和部分實現 | 定義行為契約 |
| 允許多重繼承 | 否 | 是 |
| 成員實現 | 可以有實現，也可以是抽象的 | 不能有實現（C# 8.0 開始可以有預設實現） |
| 成員存取修飾符 | 可以有不同的存取修飾符（public, private, protected） | 只有 public 成員 |
| 构造函數 | 可以有構造函數 | 不能有構造函數 |
| 可以包含字段 | 可以 | 不可以 |
| 支援多態 | 支援 | 支援 |
| 使用場景 | 用於需要共享實現邏輯的基類 | 用於定義跨多個類別的行為契約 |

心得

介面所謂的行為契約本質上是描述 (繼承)該介面的的類別應該做甚麼

（而不是如何做）

行為代表 這些功能或方法, 而契約代表 需要遵循這些約定來實現邏輯

而抽象則不只包含行為契約，還可以提供實現。