

Az **interface-eket** gyakran használják **családi, típus-, vagy szerződés-szerű szerkezetek** (pattern-ek) megvalósítására, főleg ha több osztályt akarsz összekapcsolni vagy egységesíteni.

Konkréten az interface-ekre gyakori helyzetek:

- **Szerződésként (contract)**, hogy több osztály kötelezően megvalósítsanak bizonyos függvényeket, például `Comparable`, `Serializable`, stb.
- **Polimorfizmus** támogatására, hogy különböző osztályokat ugyanazzal a típussal kezelhess.
- **Családi minták**: ha van egy nagyobb rendszerelem (pl. járművek), és az alosztályok (autó, motor, bicikli) ugyanazt az interface-t implementálják, így garantált a közös viselkedés.

Példa interface-re:

```
interface Animal {  
    fun sound()  
}  
  
class Dog : Animal {  
    override fun sound() {  
        println("woof!")  
    }  
}  
  
class Cat : Animal {  
    override fun sound() {  
        println("Meow!")  
    }  
}
```

Ez az egyik leggyakoribb használati módja, amikor a **család** vagy **közös típus** egy szerződés mentén van definiálva.

Ha a kérdésed arra vonatkozott, hogy az interface-eket szokták-e "családmintáknál" vagy hasonló tervezési mintáknál alkalmazni, a válasz: igen, gyakran.