

Strategy Design Pattern

Vad, varför och hur i C#

Vad är Strategy Pattern?

- Strategy-mönstret är ett beteendemönster.
- Tillåter dynamiskt byte mellan olika algoritmer eller beteenden.
- Inkapslar algoritmer i separata klasser med en gemensam gränssnitt.

Varför använda Strategy Pattern?

- Undvik långa if-else eller switch-satser.
- Möjliggör att byta algoritm vid körning.
- Följer Open/Closed Principle (OCP): Algoritmer kan ändras utan att påverka klientkoden.
- Enkel att underhålla och utöka.

Komponenter i Strategy Pattern

1. Strategy Interface: Definierar en gemensam struktur för alla algoritmer.
2. Concrete Strategies: Implementerar specifika algoritmer.
3. Context Class: Har en referens till en strategi och använder den.

C# Kod: Strategy Interface

C# Kod: Concrete Strategies

C# Kod: Context Class

Sammanfattning

- Strategy Pattern separerar algoritmer och gör dem utbytbara.
- Undviker långa if-else-satser och gör koden mer flexibel.
- Perfekt för dynamiska beteenden och algoritmer.
- Följer goda designprinciper som OCP.