# **C# Strategy**

Шаблон проектування **Strategy** визначає сімейство алгоритмів, інкапсулює кожен з них і робить їх взаємозамінними. Цей патерн дозволяє алгоритму змінюватися незалежно від клієнтів, які його використовують.

До класів та об'єктів, що беруть участь у цьому паттерні, відносяться наступні:

* **Стратегія (SortStrategy) -** оголошує інтерфейс, спільний для всіх підтримуваних алгоритмів. Контекст використовує цей інтерфейс для виклику алгоритму, визначеного **ConcreteStrategy**
* **ConcreteStrategy (QuickSort, ShellSort, MergeSort) -** реалізує алгоритм з використанням інтерфейсу Strategy
* **Context (SortedList) -** конфігурується об'єктом **ConcreteStrategy**, зберігає посилання на об'єкт **Strategy**, може визначати інтерфейс, який дозволяє об'єкту **Strategy** отримати доступ до його даних.

## **Structural code in C#**

using System;

namespace Strategy.Structural

{

/// <summary>

/// Strategy Design Pattern

/// </summary>

public class Program

{

public static void Main(string[] args)

{

Context context;

// Three contexts following different strategies

context = new Context(new ConcreteStrategyA());

context.ContextInterface();

context = new Context(new ConcreteStrategyB());

context.ContextInterface();

context = new Context(new ConcreteStrategyC());

context.ContextInterface();

// Wait for user

Console.ReadKey();

}

}

/// <summary>

/// The 'Strategy' abstract class

/// </summary>

public abstract class Strategy

{

public abstract void AlgorithmInterface();

}

/// <summary>

/// A 'ConcreteStrategy' class

/// </summary>

public class ConcreteStrategyA : Strategy

{

public override void AlgorithmInterface()

{

Console.WriteLine(

"Called ConcreteStrategyA.AlgorithmInterface()");

}

}

/// <summary>

/// A 'ConcreteStrategy' class

/// </summary>

public class ConcreteStrategyB : Strategy

{

public override void AlgorithmInterface()

{

Console.WriteLine(

"Called ConcreteStrategyB.AlgorithmInterface()");

}

}

/// <summary>

/// A 'ConcreteStrategy' class

/// </summary>

public class ConcreteStrategyC : Strategy

{

public override void AlgorithmInterface()

{

Console.WriteLine(

"Called ConcreteStrategyC.AlgorithmInterface()");

}

}

/// <summary>

/// The 'Context' class

/// </summary>

public class Context

{

Strategy strategy;

// Constructor

public Context(Strategy strategy)

{

this.strategy = strategy;

}

public void ContextInterface()

{

strategy.AlgorithmInterface();

}

}

}