**21 ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ**

Задание 1. Реализовать паттерны по примерам из теоретического материала согласно вариантам. Вариант высчитывается деление по модулю на 3 порядкового номера в журнале с прибавлением единицы. Паттерн проектирования – Стратегия.

Листинг программы:

using System;

using System.Collections.Generic;

namespace RefactoringGuru.DesignPatterns.Strategy.Conceptual

{

class Context

{

private IStrategy \_strategy;

public Context()

{ }

public Context(IStrategy strategy)

{

this.\_strategy = strategy;

}

public void SetStrategy(IStrategy strategy)

{

this.\_strategy = strategy;

}

public void DoSomeBusinessLogic()

{

Console.WriteLine("Context: Sorting data using the strategy (not sure how it'll do it)");

var result = this.\_strategy.DoAlgorithm(new List<string> { "a", "b", "c", "d", "e" });

string resultStr = string.Empty;

foreach (var element in result as List<string>) {

resultStr += element + ",";

}

Console.WriteLine(resultStr);

}

}

public interface IStrategy

{

object DoAlgorithm(object data);

}

class ConcreteStrategyA : IStrategy

{

public object DoAlgorithm(object data)

{

var list = data as List<string>;

list.Sort();

return list;

}

}

class ConcreteStrategyB : IStrategy

{

public object DoAlgorithm(object data)

{

var list = data as List<string>;

list.Sort();

list.Reverse();

return list;

}

}

class Program

{

static void Main(string[] args)

{

var context = new Context();

Console.WriteLine("Client: Strategy is set to normal sorting.");

context.SetStrategy(new ConcreteStrategyA());

context.DoSomeBusinessLogic();

Console.WriteLine();

Console.WriteLine("Client: Strategy is set to reverse sorting.");

context.SetStrategy(new ConcreteStrategyB());

context.DoSomeBusinessLogic();

}

}

}

Таблица 21.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| ConcreteStrategyA, ConcreteStrategyB | List |

Источник: собственная разработка

Анализ результата:

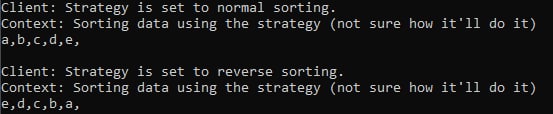


Рисунок 21.1 – Результат работы программы

Источник: собственная разработка