Лабораторная работа 12

Паттерны проектирования

Задание 3: Реализация паттерна «Команда»;

Общее описание:

Паттерн «Команда»; (Command) инкапсулирует запрос в виде объекта,

позволяя передавать запросы как параметры, ставить их в очередь, отменять

и логировать. В каждом примере будут задействованы следующие

компоненты:

* Интерфейс команды (ICommand) с методом Execute().
* Конкретная команда (ConcreteCommand), которая реализует

ICommand и инкапсулирует запрос.

* Получатель (Receiver), который выполняет конкретное действие.
* Инициатор (Invoker), который вызывает команду.
* Клиент (Client), который связывает команды с получателями.

Файловые операции – копирование и перемещение

* Интерфейс: ICommand с методом Execute().
* Конкретные команды: CopyFileCommand и MoveFileCommand.
* Получатель: FileManager с методами CopyFile(string source, string destination) и MoveFile(string source, string destination).
* Инициатор: FileOperationInvoker управляет файловыми операциями.

Листинг программы

using System;

public interface ICommand

{

void Execute();

}

public class FileManager

{

public void CopyFile(string source, string destination)

{

Console.WriteLine($"Copying file from {source} to {destination}");

}

public void MoveFile(string source, string destination)

{

Console.WriteLine($"Moving file from {source} to {destination}");

}

}

public class CopyFileCommand : ICommand

{

private readonly FileManager \_fileManager;

private readonly string \_source;

private readonly string \_destination;

public CopyFileCommand(FileManager fileManager, string source, string destination)

{

\_fileManager = fileManager;

\_source = source;

\_destination = destination;

}

public void Execute()

{

\_fileManager.CopyFile(\_source, \_destination);

}

}

class Program

{

static void Main()

{

FileManager fileManager = new FileManager();

ICommand copyCommand = new CopyFileCommand(fileManager, "file1.txt", "file2.txt");

ICommand moveCommand = new MoveFileCommand(fileManager, "file2.txt", "archive/file2.txt");

FileOperationInvoker invoker = new FileOperationInvoker();

invoker.SetCommand(copyCommand);

invoker.ExecuteCommand();

invoker.SetCommand(moveCommand);

invoker.ExecuteCommand();

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| ICommand copyCommand = new CopyFileCommand(fileManager, "file1.txt", "file2.txt");  ICommand moveCommand = new MoveFileCommand(fileManager, "file2.txt", "archive/file2.txt"); | Copying file from file1.txt to file2.txt  Moving file from file2.txt to archive/file2.txt |

Анализ результатов:

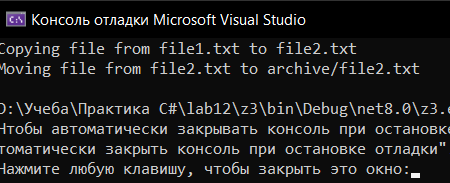


Рисунок 1 – Результат работы программы