ЛАБОРАТОРНАЯ РАБОТА №6

ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ.

Содержание работы:

Написать консольное приложение в соответствии с вариантом задания. Для упрощения выполнение значимых операций заменить на вывод сообщения в консоль (например, сортировку конкретным типом заменить на сообщение «сортируем {тип сортировки}»).

Определения паттернов на языке C# и примеры реализаций можно посмотреть на https://metanit.com/sharp/patterns/

Построить диаграмму классов ВАШЕЙ реализации паттерна. Продемонстрировать работу приложения. Объяснить, какую проблему решает данный паттерн и какими средствами.

Варианты заданий:

1. *Оплата покупки в интернете (цепочка обязанностей)*

Вы хотите оплатить покупку в Интернете. Всего есть 3 способа оплаты: A (например, банковский счет), B (Paypal) и C (Bitcoin). Каждый из них располагает суммами в SUM\_A, SUM\_B и SUM\_C соответственно, а приоритетность способов снижается от A к C. Необходимо приобрести товар с заданной стоимостью, оплатив его подходящим способом. Вывести результаты операций в виде сообщений на экран.

1. *Прием платежей с разных типов счетов (адаптер)*

Необходимо принимать оплату с различных типов лицевых счетов: стандартной банковской кредитной карты (номер карты, фамилия, имя, дата), paypal (email, пароль), qiwi (номер тел, фамилия, имя, email, код защиты), не смотря на то, что их API различаются.

Принять оплаты с разных счетов и вывести результат в виде сообщений на экран.

1. *Использование пульта ДУ для просмотра телевизора (Команда).*

вы (Client) включаете, выключаете телевизор(Receiver), управляете программами (Command) с помощью пульта (Invoker). Реализовать симулятор управления телевизором: включить телевизор, переключить на следующую программу, вернуться на предыдущую программу, повторить переключение программы, выключить телевизор.

1. *Корзина для интернет-магазинов, работающая с любыми моделями товаров и способами доставки, задаваемыми владельцами интернет-магазинов (Abstract Factory).*

Создать библиотеку для работы с товарами интернет-магазина. товары и методы работы с ними определяются пользователями вашей библиотеки.

1. *Стоимость напитка (Декоратор)*

Есть базовый напиток – кофе, имеющий три разновидности: черный, без кофеина, эспрессо. К кофе можно заказать различные дополнения (молоко, шоколад и т.д.), да еще украсить сверху взбитыми сливками. Дополнения не бесплатны, поэтому они должны быть учтены в заказе. Необходимо подсчитать конечную стоимость заказанного кофе в зависимости от выбранных покупателем ингредиентов. Вывести на экран название и стоимость заказанного варианта.

1. *«Ленивая» загрузка файла (Прокси)*

Есть класс Image, который загружает файл с изображением с диска. Реализовать класс, который будет загружать файл только по необходимости, например при вызове функции «Показать изображение».

Продемонстрировать загрузку напрямую через класс Image и через прокси.

Загрузку и отображение файла заменить на вывод сообщений в консоль.

1. *Логгер (стратегия)*
2. Реализовать класс логгер, записывающий сообщения.
3. Реализовать 3 типа поведения (стратегий):

ConsoleLogger - выводит сообщение в консоль

SimpleFileLogger - выводит сообщение в файл

DBLogger - выводит сообщение в БД

1. Реализовать класс, который будет использовать одну из стратегий.

Запись в файл и в БД для упрощения заменить на вывод сообщения в консоль. В программе воспользоваться разными типами логгера.

1. *Сортировка разными методами (стратегия)*

Реализовать сортировку массива методом, выбранным пользователем. (Пузырьком, быстрая, вставками)

Реализацию сортировок заменить на вывод сообщения в консоль, воспользовавшись встроенным методом сортировки..

1. *Чат (посредник)*

Реализовать имитацию чата между пользователями.

Пользователи добавляются в окно чата и отправляют сообщения.

1. *Изменение шрифта в редакторе (Состояние)*

Реализовать имитацию текстового редактора, в котором изменяется шрифт. Реализовать несколько состояний стиля шрифта( например, «обычный», «прописные буквы» , «строчные буквы», разные цвета и т.п.). Вывести текст в разных состояниях на экран.

Пример изменения цвета в констоли: Console.ForegroundColor = ConsoleColor.DarkGreen;

1. *Кэш (стратегия)*

Необходимо:

1. Реализовать класс кэш. Примерный интерфейс:

**public interface** **IBaseCache** {

bool HasKey(string key);

string Read(string key);

void Write(string key, string value);

void Delete(string key);

};

1. Реализовать четыре типа поведения (стратегий)

* MemoryCache - данные хранятся в памяти
* FileCache - данные хранятся в файлах на диске. В файлах хранятся пары ключ-значения
* NullCache - кэш ничего не хранит, а запись происходит очень быстро
* PoorManMemoryCache - кэш хранит последние N записанных ключей

1. Реализовать класс, который будет использовать любую из стратегий
2. *Быстрый доступ к размеру изображения (Прокси)*

Реализовать класс, позволяющий получать информацию о размере изображения в BMP-файле со следующим предположением. Если имя файла заканчивается на \_WWWxHHH, где WWW и HHH – десятичные числа, то WWW – ширина, а HHH – высота изображения. В этом случае для получения размера необязательно открывать файл. Если формат не соответствует, модуль должен все же открыть BMP-файл и считать размер явным образом.

Получение размеров картинки (используется using System.Drawing;):

using (Stream stream = File.OpenRead(filename))

{

using (Image sourceImage = Image.FromStream(stream, false, false))

{

size = $"Width={sourceImage.Width}, height={sourceImage.Height}";

}

}

Парсинг строки WWWxHHH (using System.Text.RegularExpressions;):

//проверка окончания имени файла

Regex regex = new Regex(@"[\d]{3}x[\d]{3}");

string nameEnding = filename.Split('.')[0];

if (regex.Match(nameEnding).Length >0)

{

size = nameEnding.Substring(nameEnding.Length - 7);

}

1. *Смена темы оформления на вэб-сайте (мост)*

Есть сайт с разными страницами, и пользователь должен иметь возможность изменять тему. Создать 2 иерархии классов WebPage и Theme. Реализовать классы, возвращающие контент страницы в заданной теме. Создать страницы в разных темах.

1. *Фильтрация товаров онлайн-магазина по различным критериям (Стратегия)*

У вас есть товары, которые выставляются в интернет-магазине. Необходимо создать класс, применяющий различные фильтры к вашему списку товаров, в зависимости от выбора критериев поиска пользователем. Например, фильтр по максимальной стоимости товара и фильтр по производителю.

Отфильтровать товары разными способами.

1. *Инструмент для автоматического развертывания приложений. (шаблонный метод)*

Реализовать программный инструмент, позволяющий тестировать, проводить контроль качества кода, выполнять сборку, генерировать отчёты сборки, развёртывать приложение на тестовом сервере для разных платформ (Windows, Andriod, iOs).

В программе «запустить» инструмент для разных платформ. Действия по развертыванию заменить сообщениями в консоль.

1. *Рассылка уведомлений (наблюдатель)*

Люди, которые ищут работу, часто подписываются на сайты, где публикуются вакансии. Эти сайты уведомляют соискателей о подходящих должностях по мере их появления. Реализовать классы пользователя и сайта вакансий, осуществляющего рассылку уведомлений для подписанных на соответствующую вакансию пользователей. Предусмотреть возможность пользователя отписаться от рассылки.

1. *Элементы блок-схемы (Декоратор).*

Необходимо разработать класс Element, реализующий элемент блок-схемы. Есть три стиля его отображения, которые надо комбинировать: в рамке, зачеркнутый, выделенный. Для упрощения заменить рисование выводом на консоль. Вывести на экран разные виды элементов блок-схемы .