Лабораторная работа 11

Паттерны программирования

Задание 3: Реализация паттерна &quot;Наблюдатель&quot; (Observer)

В каждой задаче должно быть минимум 3 класса:

* Издатель (Subject) – хранит список подписчиков и уведомляет их об
* изменениях.
* Интерфейс подписчика (IObserver) – задает метод для получения
* уведомлений.
* Конкретные подписчики (ConcreteObserver) – реализуют логику
* обработки уведомлений.

Управление акциями в интернет-магазине

Классы:

* PromotionManager – издатель, добавляет новые акции.
* ICustomer – интерфейс подписчика.
* LoyalCustomer, RegularCustomer – подписчики, получают уведомления

об акциях.

Листинг программы

public interface ICustomer

{

void Update(string promotion);

}

public class PromotionManager

{

private List<ICustomer> \_customers = new List<ICustomer>();

public void Subscribe(ICustomer customer)

{

\_customers.Add(customer);

}

public void Unsubscribe(ICustomer customer)

{

\_customers.Remove(customer);

}

public void AddPromotion(string promotion)

{

Notify(promotion);

}

private void Notify(string promotion)

{

foreach (var customer in \_customers)

{

customer.Update(promotion);

}

}

}

public void Update(string promotion)

{

Console.WriteLine($"Loyal Customer {\_name} notified first: New promotion - {promotion}");

}

}

public class RegularCustomer : ICustomer

{

private string \_name;

public RegularCustomer(string name)

{

\_name = name;

}

public void Update(string promotion)

{

Console.WriteLine($"Regular Customer {\_name} notified: New promotion - {promotion}");

}

}

class Program

{

static void Main(string[] args)

{

PromotionManager manager = new PromotionManager();

LoyalCustomer loyal = new LoyalCustomer("Alice");

RegularCustomer regular = new RegularCustomer("Bob");

manager.Subscribe(loyal);

manager.Subscribe(regular);

manager.AddPromotion("20% off on electronics");

manager.AddPromotion("Free shipping this weekend");

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| LoyalCustomer loyal = new LoyalCustomer("Alice");  RegularCustomer regular = new RegularCustomer("Bob"); | Loyal Customer Alice notified first: New promotion - 20% off on electronics  Regular Customer Bob notified: New promotion - 20% off on electronics  Loyal Customer Alice notified first: New promotion - Free shipping this weekend  Regular Customer Bob notified: New promotion - Free shipping this weekend |

Анализ результатов:

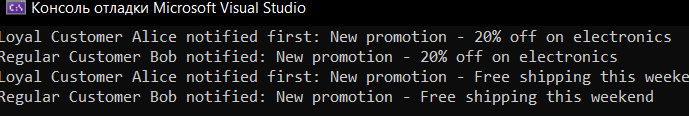


Рисунок 1 – Результат работы программы