Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Кубанский Государственный Технологический Университет»

Кафедра информационных систем и программирования

Отчет по лабораторной работе №4

По дисциплине «Рефакторинг и работа с унаследованным кодом»

«Сложные техники рефакторинга. Выделение представления»

Выполнил студент

Группы 21-КБ-ПР2

Жулькин Д.П.

Преподаватель:

Степанова Елизавета Владимировна

Краснодар

2024

Цель работы: Изучить техники рефакторинга позволяющие выполнять отделение представления от остального кода

Задание:

1) Использование внедрение зависимостей для поддержки динамического изменения поведения.

2) Проверить корректность работы нового метода и работоспособность формирования счета



Рисунок 1 – Созданные представления TxtView и HtmlView.

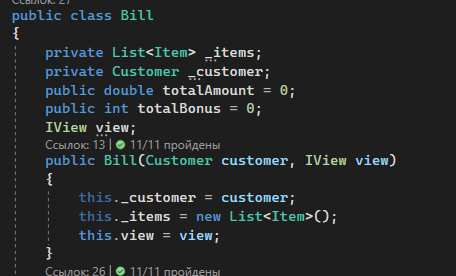


Рисунок 2 – Преобразование класса Bill

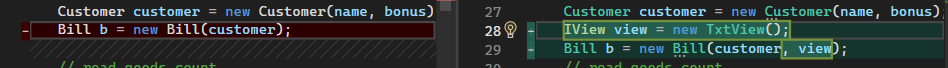


Рисунок 3 – Изменение всех вхождений создания класса Bill

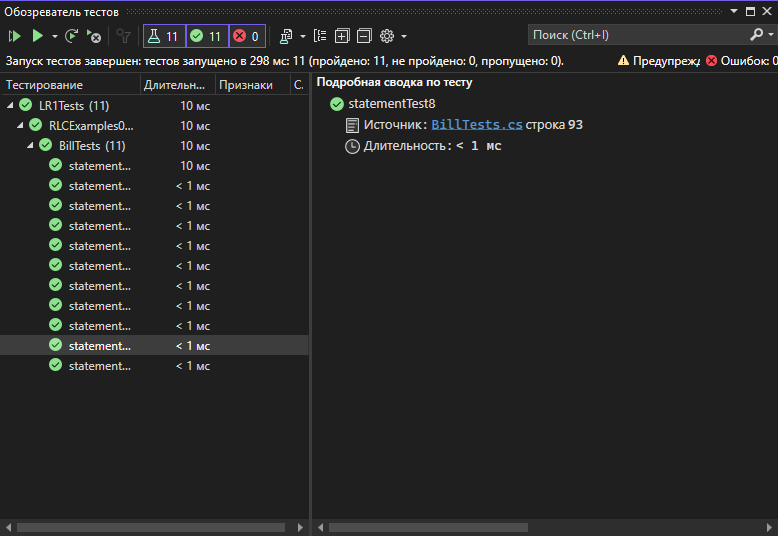


Рисунок 4 – Состояние тестов

Вывод: все задания выполнены успешно, изучены сложные техники рефакторинга позволяющие выполнять отделение представления от остального кода.

Листинг:

namespace RLCExamples01

{

public class Bill

{

private List<Item> \_items;

private Customer \_customer;

public double totalAmount = 0;

public int totalBonus = 0;

IView view;

public Bill(Customer customer, IView view)

{

this.\_customer = customer;

this.\_items = new List<Item>();

this.view = view;

}

public void addGoods(Item arg)

{

\_items.Add(arg);

}

public string statement()

{

List<Item>.Enumerator items = \_items.GetEnumerator();

string result = view.getHeader(\_customer.getName());

while (items.MoveNext())

{

Item each = items.Current;

(double discount, double usedBonus) = each.getDiscount(\_customer);

int bonus = each.getBonus();

double sumWithDiscount = each.getSum() - discount;

double thisAmount = sumWithDiscount - usedBonus;

totalAmount += thisAmount;

totalBonus += bonus;

result += view.getItemString(each, discount, thisAmount, bonus);

}

\_customer.receiveBonus(totalBonus);

result += view.getFooter(totalAmount, totalBonus);

return result;

}

}

}

namespace RLCExamples01 {

public interface IView {

string getHeader(string name);

string getFooter(double totalAmount, int totalBonus);

string getItemString(Item item, double discount,

double thisAmount, int bonus);

}

public class TxtView : IView {

public string getHeader(string name) {

return "Счет для " + name + "\n" +

"\t" + "Название" + "\t" + "Цена" +

"\t" + "Кол-во" + "Стоимость" + "\t" + "Скидка" +

"\t" + "Сумма" + "\t" + "Бонус" + "\n";

}

public string getFooter(double totalAmount, int totalBonus) {

return "Сумма счета составляет " + totalAmount.ToString() + "\n" +

"Вы заработали " + totalBonus.ToString() +

" бонусных баллов";

}

public string getItemString(Item item, double discount,

double thisAmount, int bonus) {

return "\t" + item.getGoods().getTitle() + "\t" +

"\t" + item.getPrice() + "\t" + item.getQuantity() +

"\t" + (item.getQuantity() \* item.getPrice()).ToString() +

"\t" + discount.ToString() + "\t" + thisAmount.ToString() +

"\t" + bonus.ToString() + "\n";

}

}

public class HtmlView : IView {

public string getHeader(string name) {

return "";

}

public string getFooter(double totalAmount, int totalBonus) {

return "";

}

public string getItemString(Item item, double discount,

double thisAmount, int bonus) {

return "";

}

}

}