ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РЕСПУБЛИКИ ХАКАСИЯ

«ХАКАССКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 09.02.07

«Информационные системы и программирование»

контрольная работа №1

тема: наблюдатель, магазин

Руководитель:

\_\_\_\_\_\_\_\_\_\_\_\_Отыргашева А.А.

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка, дата)

Выполнил:

Студент группы ИС(ПРО) – 31

\_\_\_\_\_\_\_\_\_\_\_\_ Канатбеков Т. К.

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Абакан 2022 г.

Наблюдатель — поведенческий шаблон проектирования. Также известен как «подчинённые». Реализует у класса механизм, который позволяет объекту этого класса получать оповещения об изменении состояния других объектов и тем самым наблюдать за ними.

Ниже предоставлен код программы, с подробным описанием каждого пункта.

public interface IObserver

{

void Update(ISubject subject);

}

public interface ISubject

{

void Attach(IObserver observer); // Прикрепляем наблюдателя

void Detach(IObserver observer); // Отключаем наблюдателя

void Notify(); // Уведомление о различных действиях

}

public class Subject : ISubject

{

public int Chislo { get; set; } = 0; //Step, через него вывожу сумму к оплате

Regex reg = new Regex(@"^[^@\s][+@[^@\s]+\.[^@\s]+$](mailto:+@[%5e@\s%5d+\.%5b%5e@\s%5d+$)"); // Проверка вводимой почты по шаблону

private List<IObserver> \_observers = new List<IObserver>(); // Список наблюдателей, при вводе наблюдателя – добавляет, при откреплении – удаляет.

public void Attach(IObserver observer) // Прикрепление наблюдателя

{

Console.WriteLine("Subject: Подключение наблюдателя");

this.\_observers.Add(observer);

}

public void Detach(IObserver observer) // Открепление наблюдателя

{

this.\_observers.Remove(observer);

Console.WriteLine("Subject: Отключение наблюдателя");

}

public void Notify() // Уведомление о действии

{

Console.WriteLine("Subject: Произведено действие с наблюдателем ");

foreach (var observer in \_observers)

{

observer.Update(this);

}

}

public void Raschet() // Тут мы производим расчет суммы покупки, выводится число от 100 – 5000, после выводится уведомление о действии

{

Console.WriteLine("Subject: Подсчет суммы ");

this.Chislo = new Random().Next(100, 5000);

Console.WriteLine("Subject: Покупка на сумму " + this.Chislo);

this.Notify();

}

}

class ConcreteObserverA : IObserver // Условие, если число меньше 1000, оплата не пройдет

{

public void Update(ISubject subject)

{

if ((subject as Subject).Chislo < 1000)

{

Console.WriteLine("System: Оплата не произведена ");

}

}

}

class ConcreteObserverB : IObserver // Условие, при котором выдается сообщение об успешной оплате

{

public void Update(ISubject subject)

{

if ((subject as Subject).Chislo == 0 || (subject as Subject).Chislo >= 2)

{

Console.WriteLine("System: Оплата произведена ");

}

}

}

class ConcreteObserverC : IObserver // Сообщение о подтверждении почты, либо о том, что почта не прошла проверку

{

public void Update(ISubject subject)

{

if (reg.IsMatch(Email))

{

Console.WriteLine("Email подтевержден ");

}

else

{

Console.WriteLine("Email не подтвержден ");

}

}

}

class Program // Выводим данные в консоль

{

static void Main(string[] args)

{

User user = new User(); // Добавление пользователя

User.Email = “[baykalov1337@mail.ru](mailto:baykalov1337@mail.ru)”; // Почта пользователя

var subject = new Subject();

var observerС = new ConcreteObserverC();

subject.Attach(observerС);

subject.Raschet();

}

}

}