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

\_\_\_\_\_\_\_\_\_\_\_\_ Якоцуц Д. А.

(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.

namespace KR

{

interface IpropertyObserver

{

void Update(ISubjectObserverable subject);

}

internal class User: IpropertyObserver

{

Regex regex = new Regex(@"[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$");

protected string Name { get; set; }

public string SGEmail

{

get { return Email; }

set

{

if (regex.IsMatch(value))

{

Email = value;

}

}

}

protected string Email;

protected string Phone { get; set; }

public void Update(ISubjectObserverable subject) { }

}

}

namespace KR

{

interface ISubjectObserverable

{

void Add(IpropertyObserver observer);

void Remove(IpropertyObserver observer);

void Notify();

}

internal class Shop: ISubjectObserverable

{

private List<IpropertyObserver> \_observers = new List<IpropertyObserver>();

public void Add(IpropertyObserver observer)

{

Console.WriteLine("Пользователь добавлен");

this.\_observers.Add(observer);

}

public void Remove(IpropertyObserver observer)

{

this.\_observers.Remove(observer);

Console.WriteLine("Пользователь удалён");

}

public void Notify()

{

foreach (var observer in \_observers)

{

observer.Update(this);

}

}

}

}

namespace KR

{

internal class Program

{

static void Main(string[] args)

{

User user = new User();

user.SGEmail = "zhidkost@gmail.com";

Console.WriteLine(user.SGEmail);

Shop shop = new Shop();

User user1 = new User();

User user2 = new User();

User user3= new User();

shop.Add(user1);

shop.Remove(user1);

shop.Add(user2);

shop.Add(user3);

shop.Add(user);

shop.Notify();

}

}

}