

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Кафедра программных систем

**Лабораторная работа №8**

**Использование интерфейсов при реализации иерархии классов**

Выполнил: Мороз

Илья Олегович

Группа № K3120

Проверила: Казанова П.П.

Санкт-Петербург

2019

**Цель работы:**

Целью работы является знакомство с интерфейсами и обработчиками событий в языке C# и их применение на практике.

**Ход работы:**

**Упражнение 1.**

1. За основу был взят проект из предыдущей лабораторной работы.
2. Был создан интерфейс IPr для описания работы с объектами класса Book
3. В данном интерфейсе были описаны два метода SetPrice и PriceBook.
4. Класс Book унаследовал интерфейс IPr. Соответствующие изменения были внемены в класс Book.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass

{

interface IPr

{

void SetPrice(double price);

double PriceBook(int s);

}

}

Листинг 1 – интерфейс IPr

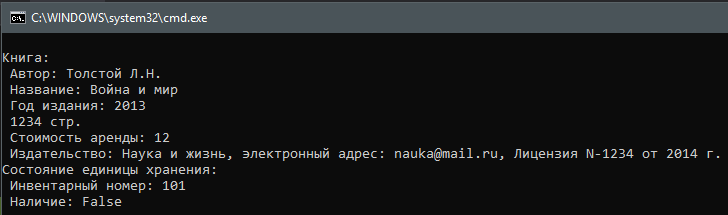


Рисунок 1 – изменения не повлияли на работоспособность программы

1. Для класса Magazine был создан интерфейс IPubs, позволяющий оформлять подписку.
2. В класс Magazine были внесены соответствующие изменения.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass

{

interface IPubs

{

void Subs();

bool IfSubs { get; set; }

}

}

Листинг 2 – интерфейс IPubs

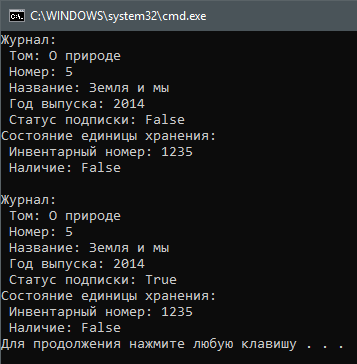


Рисунок 2 – проверка работоспособности программы

**Упражнение 2.**

1. Класс Item кнаследовал интерфейс IComparable, входящий в стандартный пакет C#.
2. В классе Item был переопределён метод CompareTo, позволяющий сравнивать между собой объекты класса Item.
3. В методе Main был создан List объектов класса Item, который позже был отсортирован по инвентарному номеру.

int IComparable.CompareTo(object obj)

{

Item it = (Item)obj;

if (this.invNumber == it.invNumber) return 0;

else if (this.invNumber > it.invNumber) return 1;

else return -1;

}

Листинг 3 – переопределение метода CompareTo для сравнения по инвентарному номеру

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass

{

class Program

{

static void Main(string[] args)

{

Publisher publ = new Publisher("Наука и жизнь", "nauka@mail.ru", 1234, new DateTime(2014, 12, 14));

Book b2 = new Book("Толстой Л.Н.", "Война и мир", publ, 1234, 2013, 101, true);

b2.SetPrice(12);

Magazine mag1 = new Magazine("О природе", 5, "Земля и мы", 2014, 1235, true);

List<Item> itList = new List<Item>();

itList.AddRange(new Item[] { b2, mag1 });

itList.Sort();

Console.WriteLine("Сортировка по инвентарному номеру: ");

foreach (Item item in itList)

{

item.Print();

}

}

}

}

Листинг 4 – класс Program

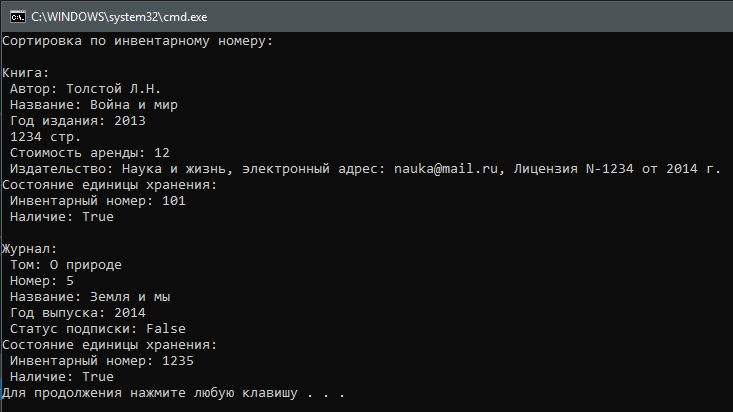


Рисунок 3 – отсортированный список

**Упражнение 3.**

1. Был создан класс Audit, который является приемником события Subscribe.
2. Было создано событие Subscribe. В классе Magazine был объявлен делегат и private событие Subscribe.
3. В классе Audit был реализован обработчик события Subscribe. Теперь при подписке на журнал, информация о нём записывается в файл на диске.
4. В методе Main был запущен обработчик события, а также была совершена подписка на журнал.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace MyClass

{

class Audit

{

public static void RunAudit()

{

Magazine.Subscribe += new ProcessMagazineDelegate(Audit.MetodSubs);

}

public static void StopAudit()

{

Magazine.Subscribe -= new ProcessMagazineDelegate(Audit.MetodSubs);

}

public static void MetodSubs(Magazine mg, DateTime dt)

{

try

{

StreamWriter sw = new StreamWriter("infoSubscribe.txt", true);

sw.WriteLine(mg.ToString());

sw.WriteLine("Подписка оформлена {0}\n", dt);

sw.Close();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

}

}

Листинг 5 – класс Audit

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass

{

delegate void ProcessMagazineDelegate(Magazine mag, DateTime dt);

class Magazine : Item, IPubs

{

public string Volume { get; set; }

public int Number { get; set; }

public string Title { get; set; }

public int Year { get; set; }

public bool IfSubs { get; set; }

public static event ProcessMagazineDelegate Subscribe = null;

public void Subs()

{

this.IfSubs = true;

if (Subscribe != null)

{

Subscribe(this, DateTime.Now);

}

}

public Magazine(string volume, int number, string title, int year, long invNumber, bool taken) : base(invNumber, taken)

{

Volume = volume;

Number = number;

Title = title;

Year = year;

}

public Magazine() { }

public override string ToString()

{

string bs = String.Format("\nЖурнал:\n Том: {0}\n Номер: {1}\n Название: {2}\n Год выпуска: {3}\n Статус подписки: {4}", Volume, Number, Title, Year, IfSubs);

return bs;

}

public override void Print()

{

Console.WriteLine(this);

base.Print();

}

public override void Return()

{

taken = true;

}

}

}

Листинг 6 – изменённый класс Magazine

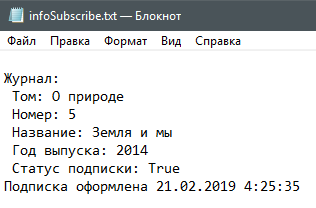


Рисунок 4 – содержимое файла после запуска программы

**Упражнение 4.**

1. На основе программы из предыдущей лабораторной работы Progression были рассмотрены возможности интерфейсов.
2. Абстрактный класс Progression был переделан в интерфейс IProgression.
3. Сщщтветствующие изменения были внемены в классы-наследники интерфейса.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Progression

{

interface IProgression

{

double GetElement(int k);

}

}

Листинг 7 – интерфейс IProgression

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Progression

{

class ArithmeticProgression : IProgression

{

public int a1 { get; private set; }

public int d { get; private set; }

public ArithmeticProgression(int a1, int d)

{

this.a1 = a1;

this.d = d;

}

public double GetElement(int k)

{

double element = this.a1 + this.d \* (k - 1);

return element;

}

}

}

Листинг 8 – класс ArithmeticProgression

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Progression

{

class GeometricProgression : IProgression

{

public int b1 { get; private set; }

public int q { get; private set; }

public GeometricProgression(int b1, int q)

{

this.b1 = b1;

this.q = q;

}

public double GetElement(int k)

{

double element = this.b1 \* Math.Pow(this.q, k - 1);

return element;

}

}

}

Листинг 9 – класс GeometricProgression

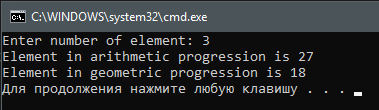


Рисунок 5 – изменения не повлияли на работоспособность программы

**Вывод:**

В ходе выполнения данной лабораторной работы были получены новые знания в языке C#. Были рассмотрены возможности интерфейсов и обработчиков событий. Полученные знания были закреплены при рефакторинге программы из предыдущей лабораторной работы.

**Ответы на контрольные вопросы:**

1. Для чего надо использовать интерфейсы?

Для описания взаимодействия с классами.

1. Где выделяется память под объекты классов?

Память под объекты классов выделяется в куче. Когда ссылка на объект пропадает, сборщик мусора удаляет его из кучи.