Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут”

Кафедра АСОІУ

**ЗВІТ**

про виконання комп’ютерного практикуму №4

з дисципліни

«Сучасні технології програмування»

Тема: «LINQ to Objects»

Варіант 2

|  |  |
| --- | --- |
| Прийняв: | Виконав: |
| Ліщук  Катерина  Ігорівна | студент 3-го курсу  гр. ІП-51 ФІОТ  Гладишко  Валентин  Юрійович |

Київ – 2017

**ЗМІСТ:**

1. Мета РОБОТИ 3
2. Постановка задачі 4
3. Виконання завдання 5
4. Висновок 13
5. ЛІСТИНГ КОДУ ПРОГРАМИ 14

# Мета РОБОТИ

Ознайомитися з обробкою даних з використанням бібліотеки LINQ to Objects.

# Постановка задачі

При виконанні комп'ютерного практикуму слід реалізувати наступні дії:

1. Розробити структуру даних для зберігання згідно варіантів, наведених нижче (можна запропонувати власні варіанти). У кожному з варіантів має бути як мінімум 3-4 класу. В рамках реалізації повинні бути продемонстровані зв'язку між класами: один-ко-многим і багато-до-багатьох.
2. Розробити як мінімум 7 різних запитів, використовуючи різні дії над множинами: угруповання, сортування, фільтрацію, об'єднання результатів декількох запитів в один (join, concat) і т.д.

Наприклад (предметна область Склад товарів):

* 1. Вивести перелік всіх продуктів
  2. Вивести список виробників, назва яких закінчується на 'e'
  3. Вивести список виробників та продуктів, які вони виробляють
  4. Вивести поставки згруповані по даті
  5. Вивести список виробників, в яких хоча б в одного продукту назва починається на 'C'
  6. Вивести всі продукти відсортовані по імені
  7. Вивести загальну ціну по продуктам на складі

1. Розробити програму, що реалізовує обробку даних з використанням бібліотеки LINQ to Objects.
2. Програма повинна бути розроблена у вигляді консольного застосування на мові C #.
3. Коротко описати архітектуру проекту
4. Побудувати діаграму класів.

Розробити структуру даних для зберігання інформації про товари на складі. Товар характеризується: назвою, вартістю, кількістю (в штуках), списком дат надходження на склад, виробником (найменування фірми).

# Виконання завдання

* 1. **Короткий опис архітектури проекту**

В програмі є 3 класи: “Product”, “Manufacturer”, “Supply”.

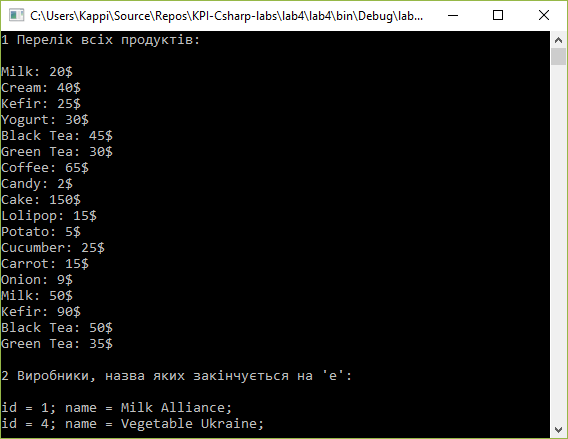
* Клас “Product” містить ID, назву продукту, ціну, та ID виробника.
* Клас “Manufacturer” містить ID та назву виробника.
* Клас “Supply” містить ID, дату поставки, ID продукту, та кількість продукту.
  1. **UML-діаграма класів**

****

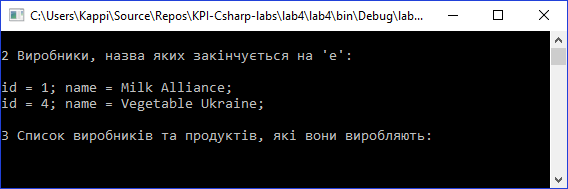
* 1. **Запити(предметна область Склад товарів)**

1. Вивести перелік всіх продуктів
2. Вивести список виробників, назва яких закінчується на 'e'
3. Вивести список виробників та продуктів, які вони виробляють
4. Вивести поставки згруповані по даті
5. Вивести список виробників, в яких хоча б в одного продукту назва починається на 'C'
6. Вивести всі продукти відсортовані по імені
7. Вивести загальну ціну по продуктам на складі
   1. **Результати виконання програми**

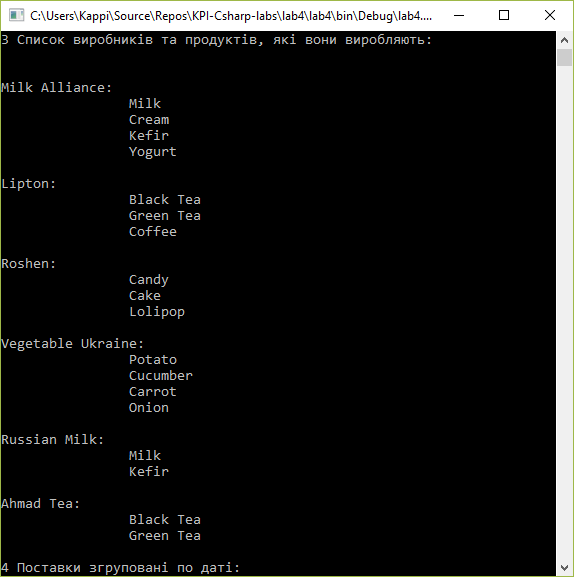
*Результат виконання 1:*



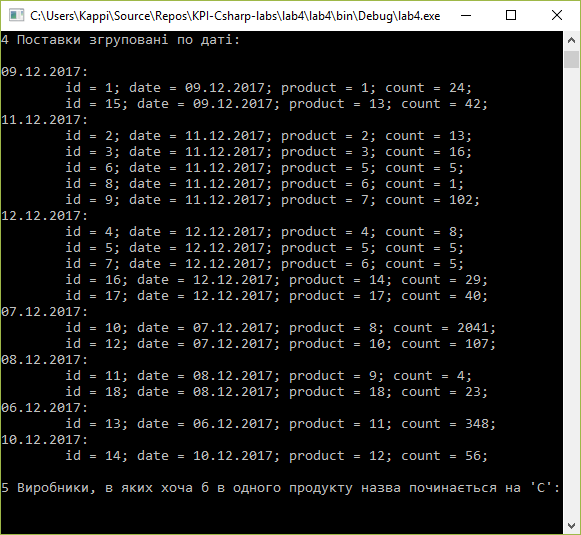
*Результат виконання 2:*



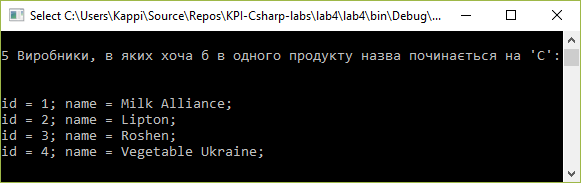
*Результат виконання 3:*



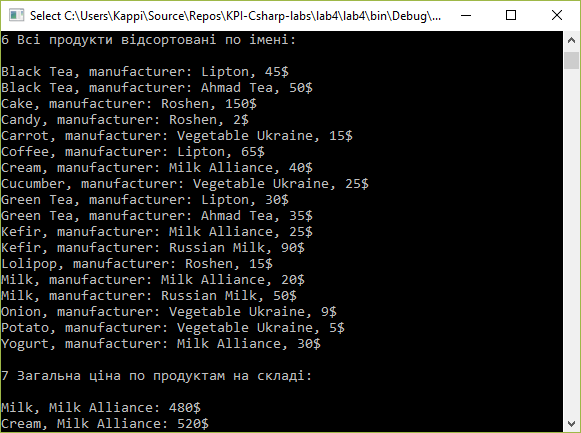
*Результат виконання 4:*



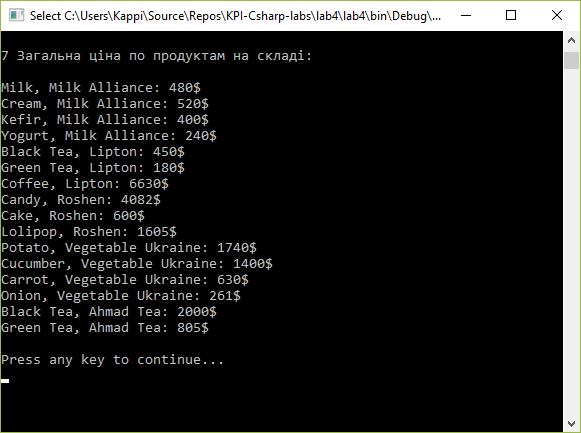
*Результат виконання 5:*



*Результат виконання 6:*



*Результат виконання 7:*



# Висновок

Отже, я ознайомився з обробкою даних з використанням бібліотеки LINQ to Objects. За допомогою LINQ я реалізував 7 різних запитів, з різними діями над множинами: угрупування (group by), сортування (orderby), фільтрацію (where), об'єднання результатів декількох запитів в один (join).

Посилання на виконану лабораторну:

<https://github.com/ValentineGladyshko/KPI-Csharp-labs/tree/master/lab4>

1. **ЛІСТИНГ КОДУ ПРОГРАМИ**

«Program.cs»

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab4

{

class Product

{

public int productID;

public string name;

public int price;

public int manufacturerID;

public Product(int productID, string name, int price, int manufacturerID)

{

this.productID = productID;

this.name = name;

this.price = price;

this.manufacturerID = manufacturerID;

}

override public string ToString()

{

return ("id = " + productID.ToString() + "; name = "

+ name + "; price = " + price.ToString()

+ "; manufacturer = " + manufacturerID.ToString() + ";");

}

}

class Manufacturer

{

public int manufacturerID;

public string name;

override public string ToString()

{

return ("id = " + manufacturerID.ToString() + "; name = "

+ name + ";");

}

public Manufacturer(int manufacturerID, string name)

{

this.manufacturerID = manufacturerID;

this.name = name;

}

}

class Supply

{

public int supplyID;

public DateTime date;

public int productID;

public int count;

public Supply(int supplyID, DateTime date, int productID, int count)

{

this.supplyID = supplyID;

this.date = date;

this.productID = productID;

this.count = count;

}

public DateTime RandomDate(Random random)

{

DateTime start = DateTime.Today.AddDays(-7);

return start.AddDays(random.Next(0, 7));

}

public Supply(int supplyID, Random random, int productID, int count)

{

this.supplyID = supplyID;

this.date = RandomDate(random);

this.productID = productID;

this.count = count;

}

override public string ToString()

{

return ("id = " + supplyID.ToString() + "; date = "

+ date.ToShortDateString() + "; product = " + productID.ToString()

+ "; count = " + count.ToString() + ";");

}

}

class Program

{

static void Main(string[] args)

{

Random random = new Random();

List<Product> products = new List<Product>()

{

new Product(1, "Milk", 20, 1),

new Product(2, "Cream", 40, 1),

new Product(3, "Kefir", 25, 1),

new Product(4, "Yogurt", 30, 1),

new Product(5, "Black Tea", 45, 2),

new Product(6, "Green Tea", 30, 2),

new Product(7, "Coffee", 65, 2),

new Product(8, "Candy", 2, 3),

new Product(9, "Cake", 150, 3),

new Product(10, "Lolipop", 15, 3),

new Product(11, "Potato", 5, 4),

new Product(12, "Cucumber", 25, 4),

new Product(13, "Carrot", 15, 4),

new Product(14, "Onion", 9, 4),

new Product(15, "Milk", 50, 5),

new Product(16, "Kefir", 90, 5),

new Product(17, "Black Tea", 50, 6),

new Product(18, "Green Tea", 35, 6),

};

List<Manufacturer> manufacturers = new List<Manufacturer>()

{

new Manufacturer(1, "Milk Alliance"),

new Manufacturer(2, "Lipton"),

new Manufacturer(3, "Roshen"),

new Manufacturer(4, "Vegetable Ukraine"),

new Manufacturer(5, "Russian Milk"),

new Manufacturer(6, "Ahmad Tea")

};

List<Supply> stock = new List<Supply>()

{

new Supply(1,random, 1, 24),

new Supply(2,random, 2, 13),

new Supply(3,random, 3, 16),

new Supply(4,random, 4, 8),

new Supply(5,random, 5, 5),

new Supply(6,random, 5, 5),

new Supply(7,random, 6, 5),

new Supply(8,random, 6, 1),

new Supply(9,random, 7, 102),

new Supply(10,random, 8, 2041),

new Supply(11,random, 9, 4),

new Supply(12,random, 10, 107),

new Supply(13,random, 11, 348),

new Supply(14,random, 12, 56),

new Supply(15,random, 13, 42),

new Supply(16,random, 14, 29),

new Supply(17,random, 17, 40),

new Supply(18,random, 18, 23)

};

// 1 Вивести перелік всіх продуктів

Console.WriteLine("1 Перелiк всiх продуктiв:\n");

var Query1 = from product in products

select product;

foreach(var product in Query1)

{

Console.WriteLine(product.name + ": " + product.price + "$");

}

// 2 Вивести список виробників, назва яких закінчується на 'e'

Console.WriteLine("\n2 Виробники, назва яких закiнчується на 'e':\n");

var Query2 = from manufacturer in manufacturers

where manufacturer.name[manufacturer.name.Length - 1] == 'e'

select manufacturer;

foreach (var manufacturer in Query2)

{

Console.WriteLine(manufacturer);

}

// 3 Вивести список виробників та продуктів, які вони виробляють

Console.WriteLine("\n3 Список виробникiв та продуктiв, якi вони виробляють:\n");

var Query3 = from manufacturer in manufacturers

join product in products

on manufacturer.manufacturerID equals product.manufacturerID into table

select new { m = manufacturer.name, table };

foreach (var el in Query3)

{

Console.WriteLine("\n" + el.m + ":");

foreach(var el1 in el.table)

{

Console.WriteLine("\t\t" + el1.name);

}

}

// 4 Вивести поставки згруповані по даті

Console.WriteLine("\n4 Поставки згрупованi по датi:\n");

var Query4 = from supply in stock

group supply by supply.date into table

select new { Key = table.Key, table };

foreach (var el in Query4)

{

Console.WriteLine(el.Key.ToShortDateString() + ":");

foreach (var el1 in el.table)

{

Console.WriteLine("\t" + el1);

}

}

// 5 Вивести список виробників, в яких хоча б в одного продукту назва починається на 'C'

Console.WriteLine("\n5 Виробники, в яких хоча б в одного продукту назва починається на 'C':\n");

var Query5 = (from manufacturer in manufacturers

join product in products

on manufacturer.manufacturerID equals product.manufacturerID

where product.name[0] == 'C'

select manufacturer).Distinct();

foreach (var manufacturer in Query5)

{

Console.WriteLine(manufacturer);

}

// 6 Вивести всі продукти відсортовані по імені

Console.WriteLine("\n6 Всi продукти вiдсортованi по iменi:\n");

var Query6 = from product in products

join manufacturer in manufacturers

on product.manufacturerID equals manufacturer.manufacturerID

orderby product.name

select new { product, manufacturer.name };

foreach (var el in Query6)

{

Console.WriteLine(el.product.name + ", manufacturer: " + el.name + ", " + el.product.price + "$");

}

// 7 Вивести загальну ціну по продуктам на складі

Console.WriteLine("\n7 Загальна цiна по продуктам на складi:\n");

var Query7 = from supply in

(

from supply in stock

group supply by supply.productID into table

select new { table.Key, count = table.Sum(supply => supply.count) }

)

join product in products

on supply.Key equals product.productID

join manufacturer in manufacturers

on product.manufacturerID equals manufacturer.manufacturerID

select new { product.name, manufacturer = manufacturer.name, price = supply.count \* product.price };

foreach (var el in Query7)

{

Console.WriteLine( el.name + ", " + el.manufacturer + ": " + el.price + "$");

}

Console.WriteLine("\nPress any key to continue...");

Console.ReadKey();

}

}

}