### Кам’янець-Подільський національний університет імені Івана Огієнка

### КАФЕДРА КОМП’ЮТЕРНИХ НАУК

### Навчальна дисципліна «Об’єктно-орієнтоване програмування»

### ЛАБОРАТОРНА РОБОТА #0102

## Тема:

## Статичні члени класу

Варіант №2

**Виконав**:  
студент 1-го курсу  
групи KNms1-B23  
Веретко С. І.

**Прийняла**:  
старший викладач,  
Мястковська М.О.

Кам’янець-Подільський – 2023

Хід роботи:

1. Теоретичні відомості

Модифікатор **static** може бути приписаний як до даних-членів так і до методів-членів класу. Якщо у класі декларується змінна із модифікатором **static**, то така змінна спільно використовується всіма екземплярами класу – фактично вона є глобальною для класу, а для доступу до неї вказується не ідентифікатор екземпляру, а ідентифікатор класу. Розглянемо перший приклад. В ньому визначений клас **MyClass**, в якому є член класу **num** та статичний член класу **count**. Звертання до **num** можливе лише через ідентифікатор екземпляру – у прикладі це **m1** або **m2** (тобто **m1.num** або **m2.num**), адже цей член класу **num** існує окремо та незалежно для кожного екземпляру класу. Звертання ж до **count** має відбуватись через ідентифікатор класу **MyClass** (тобто **MyClass.count**), оскільки **count** існує в єдиному примірнику та спільно використовується обома екземплярами **m1** та **m2**. Тобто, фактично статичні дані-члени класу можна сприймати як глобальні змінні у рамках класу.

Метод класу, визначений із модифікатором **static**, також є доступним на рівні самого класу, а не його екземплярів. Тобто для виклику такого методу непотрібний жодний екземпляр класу. Прикладом статичного методу є метод **Main()**, який викликається операційною системою. Зрозуміло, що в момент цього виклику жодного екземпляру жодного класу просто не може існувати. Іншим прикладом статичних методів, які ми неодноразово використовували у прикладах, є методи класу **Math** або **Console** – для звертання до них нам не було необхідності створювати відповідний об’єкт.

При використанні статичних методів слід пам’ятати про ряд обмежень, а саме:

1. Статичний метод може використовувати ***лише*** статичні дані-члени класу, адже статичний метод діє на рівні класу, не маючи доступу до екземплярів , а отже і до змінних екземплярів класу.
2. Статичний метод не може використовувати посилання **this**.
3. Статичний метод може викликати лише інші статичні методи класу. Щоб звернутись до нестатичного методу, потрібне посилання на екземпляр.

2. Завдання для виконання

## Завдання 1.

2 Варіант. Створити метод розширення для масиву цілих чисел, який сортує елементи масиву по спаданню.

Завдання 2.

2.1. Створити клас відповідно до варіанту. Клас також повинен містить статичне поле, для підрахунку кількості об’єктів.

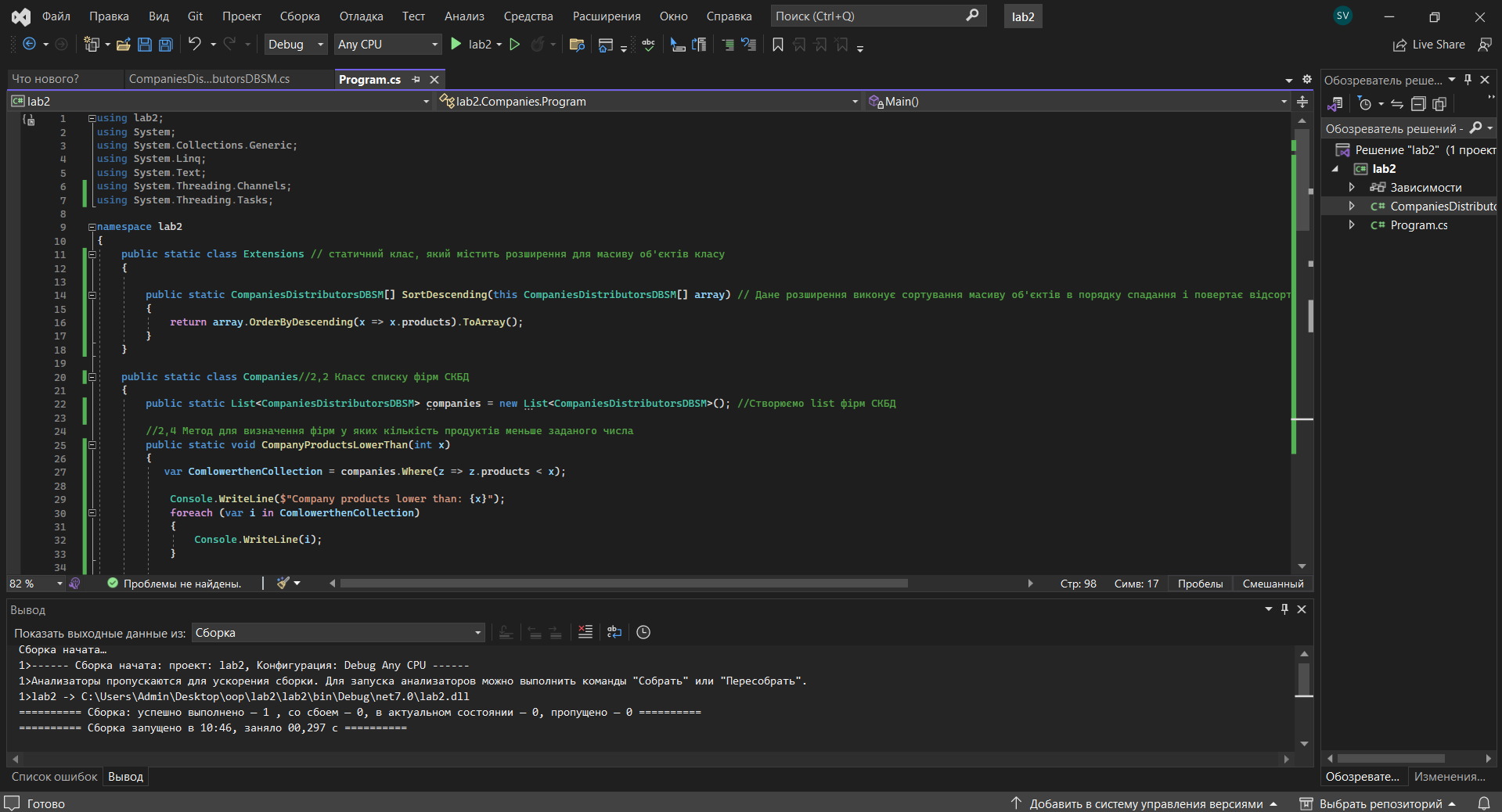
2.2. Створити масив об’єктів відповідного типу. Можна даний масив оформити у вигляді окремого класу.

2.3. Створити декілька об’єктів відповідного класу (не менше 10), і записати їх в масив.

2.4. Знайти об’єкти, які відповідають певному критерію. Критерій підібрати самостійно відповідно до варіанту. Наприклад, для першого варіанту: 1) види антилоп, чисельність яких менше заданого значення; 2) вид антилоп кількість яких є мінімальною.

2.5. Перевірити кількість існуючих об’єктів. Якщо це значення перевищує задане граничне значення1 – вивести повідомлення1, якщо значення менше заданого граничного значення2 – вивести повідомлення2.



3.Послідовність виконання завдань

4. Лістинг програми

using lab2;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Channels;

using System.Threading.Tasks;

namespace lab2

{

public static class Extensions //1 статичний клас, який містить розширення для масиву об'єктів класу

{

public static CompaniesDistributorsDBSM[] SortDescending(this CompaniesDistributorsDBSM[] array) // Дане розширення виконує сортування масиву об'єктів в порядку спадання і повертає відсортований масив

{

return array.OrderByDescending(x => x.products).ToArray();

}

}

public static class Companies//2,1 Класс списку фірм СКБД

{

public static List<CompaniesDistributorsDBSM> companies = new List<CompaniesDistributorsDBSM>(); //Створюємо list фірм СКБД

//2,4.1 Метод для визначення фірм у яких кількість продуктів меньше заданого числа

public static void CompanyProductsLowerThan(int x)

{

var ComlowerthenCollection = companies.Where(z => z.products < x);

Console.WriteLine($"Company products lower than: {x}");

foreach (var i in ComlowerthenCollection)

{

Console.WriteLine(i);

}

}

//2,4.2 Найменше продуктів

public static void CompanyMinProducts()

{

var MinProduct = companies.MinBy(x => x.products);

Console.WriteLine($"Company min product: {MinProduct}\n");

}

//2,5 Кількість компаній менших за значення

public static void CompanyCountLowerThan(int x)

{

if (companies.Count() > x) Console.WriteLine("Компаній більше заданого значення");

else Console.WriteLine("Компаній менше або рівно заданому значенню");

}

public class Program

{

static void Main() //У даному методі створюються об'єкти класу CompaniesDistributorsDBSM, які представляють різні компанії-дистриб'ютори, ініціалізуються відповідними значеннями

{

CompaniesDistributorsDBSM Oracle = new CompaniesDistributorsDBSM("Oracle", 1, 2500000000, 31.01);

CompaniesDistributorsDBSM IBM = new CompaniesDistributorsDBSM("IBM", 3, 2400000000, 29.25);

CompaniesDistributorsDBSM Microsoft = new CompaniesDistributorsDBSM("Microsoft", 2, 1000000000, 13.01);

CompaniesDistributorsDBSM KPNU = new CompaniesDistributorsDBSM("KPNU",1000, 950000000, 15.25);

CompaniesDistributorsDBSM Apple = new CompaniesDistributorsDBSM("Apple", 4, 2200000000, 28.50);

CompaniesDistributorsDBSM Google = new CompaniesDistributorsDBSM("Google", 5, 1800000000, 23.75);

CompaniesDistributorsDBSM Amazon = new CompaniesDistributorsDBSM("Amazon", 6, 1500000000, 20.50);

CompaniesDistributorsDBSM Facebook = new CompaniesDistributorsDBSM("Facebook", 7, 900000000, 12.25);

CompaniesDistributorsDBSM Tesla = new CompaniesDistributorsDBSM("Tesla", 8, 800000000, 11.50);

CompaniesDistributorsDBSM Netflix = new CompaniesDistributorsDBSM("Netflix", 9, 700000000, 10.75);

companies.Add(Oracle);

companies.Add(IBM);

companies.Add(Microsoft);

companies.Add(KPNU);

companies.Add(Apple);

companies.Add(Google);

companies.Add(Amazon);

companies.Add(Facebook);

companies.Add(Tesla);

companies.Add(Netflix);

foreach (var item in companies)

{

Console.WriteLine($"{item}\n");

}

CompaniesDistributorsDBSM[] companiesArray = { Oracle, IBM, Microsoft, KPNU, Apple, Amazon, Facebook, Tesla, Netflix };

Console.WriteLine("Сортування по спаданню за кількістю продуктів:");

var sortedArray = companiesArray.SortDescending(); // Об'єкти в companiesArray сортуються за допомогою методу розширення SortDescending та результат зберігається в sortedArray

foreach (var company in sortedArray) //У циклі кожен об'єкт у sortedArray виводиться на консоль за допомогою методу Print

{

Console.WriteLine(company);

Console.WriteLine();

}

CompanyCountLowerThan(5);

CompanyMinProducts();

CompanyProductsLowerThan(5);

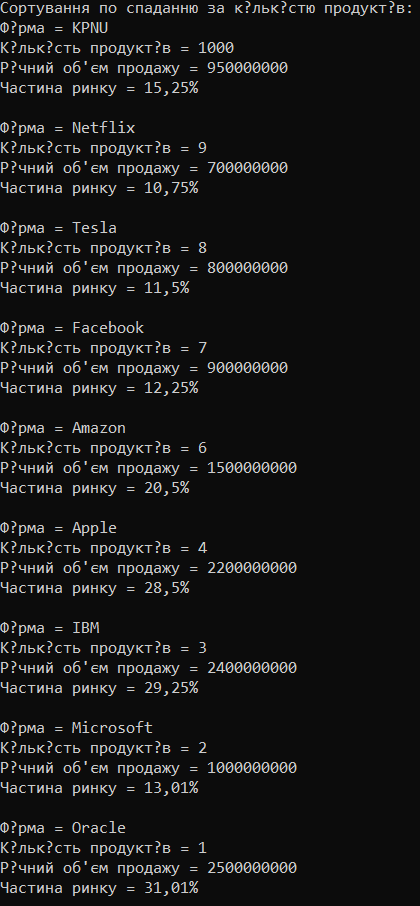
}

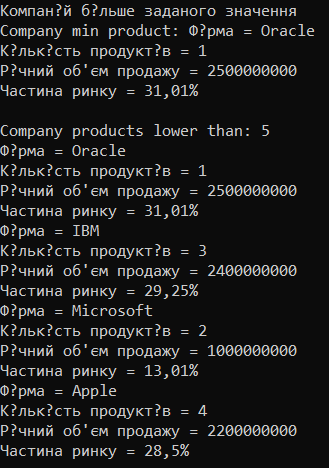
}

}

}

5. Результати роботи програми





6. [xanax4rl/lab (github.com)](https://github.com/xanax4rl/lab)