**ПРАКТИЧНА РОБОТА №2**

**«ВИКОРИСТАННЯ КОЛЕКЦІЙ. КЛАСИ ARRAYLIST,QUEUE, STACK, HASHTABLE, SORTEDLIST»**

**ВАРІАНТ №3**

**Мета роботи:** засвоїти основи використання масивів і колекцій в мові C#, набути практичних навичок роботи з двовимірними масивами.

**Вправа 3:**

Повний лістинг програми:

using System;

namespace prac2\_console

{

internal class Program

{

static void task1(int[,] array)

{

bool rowHasNegative = false;

int product = 1;

int j = 0;

for (int i = 0; i < array.GetLength(0); i++)

{

while (j < array.GetLength(1) && !rowHasNegative)

{

if (array[i, j] < 0)

rowHasNegative = true;

else

product \*= array[i, j];

j++;

}

if (!rowHasNegative)

{

Console.WriteLine($"The product of row #{i+1}: {product}");

product = 1;

}

j = 0;

rowHasNegative = false;

}

}

static void task2(int[,] array)

{

int maxLength = 0;

int maxNumber = 0;

int Row = 0;

int currentNumber;

int currentLength = 0;

for (int i = 0; i < array.GetLength(0); i++)

{

for (int j = 0; j < array.GetLength(1); j++)

{

currentNumber = array[i,j];

for (int k = 0; k < 3; k++)

{

if (currentNumber == array[i,k])

{

currentLength++;

}

}

if (currentLength>maxLength)

{

maxLength = currentLength;

maxNumber = array[i, j];

Row = i + 1;

}

currentLength = 0;

}

}

Console.WriteLine($"Row with maximum sequence is #{Row}. Max squence is {maxLength}. Max number is {maxNumber}");

}

static void Main(string[] args)

{

int[,] array = { { 1, -3, 5 }, { 2, 1, 1 }, { 4, 4, 9 } };

task1(array);

task2(array);

}

}

}

Програма складається з класу Program в якому є 3 методи: основний метод Main та два методи task1, task2 що виконують дії, поставлені в завданні.

У статичному методі *Main* створюється та ініціалізується двомірний масив, викликаються 2 інших методи, в які передається попередньостворений масив:

static void Main(string[] args)

{

int[,] array = { { 1, -3, 5 }, { 2, 1, 1 }, { 4, 4, 9 } };

task1(array);

task2(array);

}

Статичний метод *task1* за допомогою циклу перевіряє кожен рядок на наявність негативних елементів, якщо таких намає, то перемножує едементи цього рядка і виводить на консоль номер рядка та добуток його елементів:

static void task1(int[,] array)

{

bool rowHasNegative = false;

int product = 1;

int j = 0;

for (int i = 0; i < array.GetLength(0); i++)

{

while (j < array.GetLength(1) && !rowHasNegative)

{

if (array[i, j] < 0)

rowHasNegative = true;

else

product \*= array[i, j];

j++;

}

if (!rowHasNegative)

{

Console.WriteLine($"The product of row #{i+1}: {product}");

product = 1;

}

j = 0;

rowHasNegative = false;

}

}

Статичний метод *task2* також за допомогою циклу проходить кожний рядок та в змінну maxLength записує максимальне значення серії та номер рядка, в якому ця серія знаходиться. Також, в змінну maxNumber записується максимальне значення цього рядка.

static void task2(int[,] array)

{

int maxLength = 0;

int maxNumber = 0;

int Row = 0;

int currentNumber;

int currentLength = 0;

for (int i = 0; i < array.GetLength(0); i++)

{

for (int j = 0; j < array.GetLength(1); j++)

{

currentNumber = array[i,j];

for (int k = 0; k < 3; k++)

{

if (currentNumber == array[i,k])

{

currentLength++;

}

}

if (currentLength>maxLength)

{

maxLength = currentLength;

maxNumber = array[i, j];

Row = i + 1;

}

currentLength = 0;

}

}

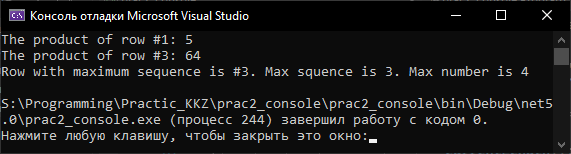


Рисунок 1 – демонстрація роботи програми

**Вправа 1.** Опис роботи класу ArrayList.

Лістинг до програми:

using System;

using System.Collections;

namespace ArrayList2

{

class Program

{

static void Main(string[] args)

{

ArrayList baseballTeams = new ArrayList();

baseballTeams.Add("St. Louis Cardinals");

baseballTeams.Add("Seattle Mariners");

baseballTeams.Add("Florida Marlins");

string[] myStringArray = new string[2];

myStringArray[0] = "San Francisco Giants";

myStringArray[1] = "Los Angeles Dodgers";

baseballTeams.AddRange(myStringArray);

foreach (string item in baseballTeams)

{

Console.Write(item + "\n");

}

Console.ReadLine();

}

}

}

В ході виконання програми створюється колекція ArrayList яка представляє собою список баскетбольних команд та в неї додаються 3 команди. Колекція ArrayList може зберігати в собі об’єкти різних типів. Далі створюється строковий масив та в нього додаються ще дві команди. Використовуючи метод AddRange ми можемо об’єднати данні з інших масивів або колекцій у об’єкті ArrayList. Далі використовуючи цикл foreach виводимо дані з колекції на екран. Використати цей цикл ми можем через те, що всі базові колекції поля імен System.Collections реалізують інтерфейс IEnumerable, що дозволяє інтексувати усі елементи колекцій. На рисунку 2 показаний результат роботи програми.

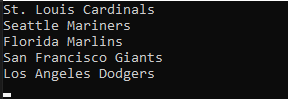


Рисунок 2 – результат виконання програми.

**Вправа 2.** Опис роботи HashTable

Лістинг до програми:

using System;

using System.Text;

using System.Collections;

namespace SocialSecurityNumbers

{

class MainEntryPoint

{

static void Main()

{

TestHarness harness = new TestHarness();

harness.Run();

}

}

class TestHarness

{

Hashtable employees = new Hashtable(31);

public void Run()

{

EMPLOYEEID idMortimer = new EMPLOYEEID("B001");

EmployeeData mortimer = new EmployeeData(idMortimer, "Mortimer", 100000.00M);

EMPLOYEEID idArabel = new EMPLOYEEID("W234");

EmployeeData arabel = new EmployeeData(idArabel, "Arabel Jones", 10000.00M);

employees.Add(idMortimer, mortimer);

employees.Add(idArabel, arabel);

while (true)

{

try

{

Console.Write("Enter employee ID (format:A999, X to exit)> ");

string userInput = Console.ReadLine();

userInput = userInput.ToUpper();

if (userInput == "X")

return;

EMPLOYEEID id = new EMPLOYEEID(userInput);

DisplayData(id);

}

catch (Exception e)

{

Console.WriteLine("Exception occurred. Did you use the correct format for the employee ID?");

Console.WriteLine(e.Message);

Console.WriteLine();

}

Console.WriteLine();

}

}

private void DisplayData(EMPLOYEEID id)

{

object empobj = employees[id];

if (empobj != null)

{

EmployeeData employee = (EmployeeData)empobj;

Console.WriteLine("Employee: " + employee.ToString());

}

else

Console.WriteLine("Employee not found: ID = " + id);

}

}

class EmployeeData

{

private string name;

private decimal salary;

private EMPLOYEEID id;

public EmployeeData(EMPLOYEEID id, string name, decimal salary)

{

this.id = id;

this.name = name;

this.salary = salary;

}

public override string ToString()

{

StringBuilder sb = new StringBuilder(id.ToString(), 100);

sb.Append(": ");

sb.Append(string.Format("{0-20}", name));

sb.Append(" ");

sb.Append(string.Format("{0:C}", salary));

return sb.ToString();

}

}

class EMPLOYEEID

{

private readonly char prefix;

private readonly int number;

public EMPLOYEEID(string id)

{

prefix = (id.ToUpper())[0];

number = int.Parse(id.Substring(1, 3));

}

public override string ToString()

{

return prefix.ToString() + string.Format("{0,3:000}", number);

}

public override int GetHashCode()

{

string str = this.ToString();

return str.GetHashCode();

}

public override bool Equals(object obj)

{

EMPLOYEEID rhs = obj as EMPLOYEEID;

if (rhs == null)

return false;

if (prefix == rhs.prefix && number == rhs.number)

return true;

return false;

}

}

}

У методі Main створюється еземпляр класу TestHarness та одразу викликається метод Run(), що ініціалізує роботу програми. У класі TestHarnes також був створений об’єкт класу колекції Hashtable та ініціалізований конструктором що приймає цілочислену змінну, що задає вмістимість колекції.

На початку методу Run() створюється еземпляр класу EMPLOYEEID та EmployeeData.

Лістинг цього класу:

class EMPLOYEEID

{

private readonly char prefix;

private readonly int number;

public EMPLOYEEID(string id)

{

prefix = (id.ToUpper())[0];

number = int.Parse(id.Substring(1, 3));

}

public override string ToString()

{

return prefix.ToString() + string.Format("{0,3:000}", number);

}

public override int GetHashCode()

{

string str = this.ToString();

return str.GetHashCode();

}

public override bool Equals(object obj)

{

EMPLOYEEID rhs = obj as EMPLOYEEID;

if (rhs == null)

return false;

if (prefix == rhs.prefix && number == rhs.number)

return true;

return false;

}

}

В цьому класі було створено 2 змінні prefix та number. Був створений конструктор з одним строковим параметром, що задає id робітника. Перший символ повинен бути символом, що відповідає латинському алфавіту та 3 цифри, що задають числову частину id. Потім був перевизначені усі методи, базового для всіх об’єктів, класу object. Метод ToString() задає правило перетворення еземплярів у строку, його перевизначення потрібне для зручного виводу потрібної інформації на екран. Метод Equals(object obj) який визначає правила порівняння 2 екземплярів між собою. Метод GetHashCode визначає правило отримання хеш коду для об’єктів. Хеш код це своєрідний ідентифікатор для об’єктів у цій програмі він потрібен для використання колекції Hashtable, бо вона використовує пари ключ/значення, а ці коди представляють собою ключі.

Далі був створений клас EmployeeData. В якому створені приватні поля імені, заробітної плати та шляхом агрегації реалізован об’єкт класу EMPLOYEEID. Також перевизначений метод ToString в якому використовуючи клас StringBuilder, що представляє динамічну строку, виведені данні про працівника на екран.

Створивши екземпляри цих класів у методі Run класу TestHarness, вони заносяться у колекцію Hashtable за правилом ключ – значення, де ключ це EMPLOYEEID, а значення це EmployeeData. Далі створюється цикл while та у конструкції try-catch Користувачу надається можливість здійснити пошук серед доданих у колекцію елементів шляхом введення id цих робітників. При введені некоректного значення id буде сгенерована помилка та її текст буде виведений на екран. Для того щоб вийти з циклу користувачу потрібно ввести «Х». На рисунку 3 показаний результат роботи програми.

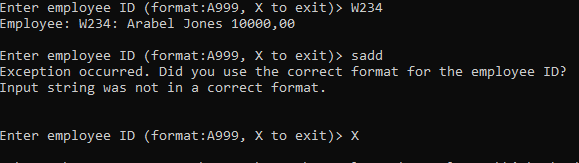


Рисунок 3 – результат виконання програми.

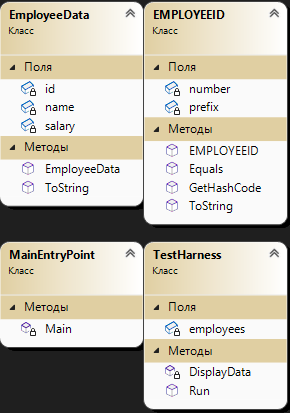


Рисунок 4 – UML діаграма класів