МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М.Є. Жуковського

«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп’ютерних систем та інфокомунікацій

Кафедра комп’ютерних систем, мереж і кібербезпеки (503)

Лабораторна робота № *6*

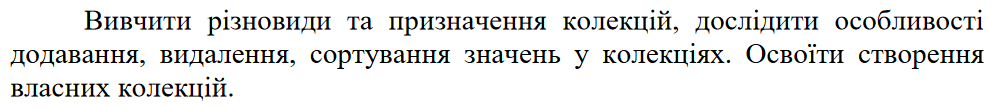
|  |  |
| --- | --- |
|  | *Колекції* |
|  | (назва лабораторної роботи) |
| з дисципліни | *Технології програмування* |
|  | (шифр)  ХАІ**.**503**.**525a**.**03О**.**123-Комп'ютерна інженерія**,** ПЗ №9629619 |

|  |  |  |
| --- | --- | --- |
| Виконав студент гр. | 525а | *Литвиненко А.В.* |
| 28.11.2022 | (№ групи) | (П.І.Б.) |
| (підпис, дата) |  |  |
| Перевірив | канд. техн. наук, доцент | |
|  |  | *Є. В. Бабешенко* |
| (підпис, дата) |  | (П.І.Б.) |

Харків – 2022

**Тема роботи:** колекції

**Мета роботи**:

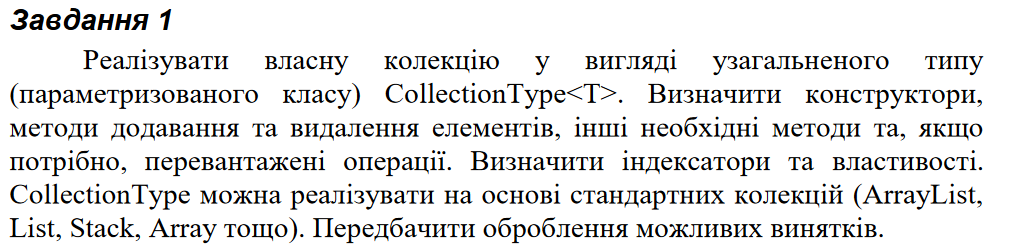


**Варіант 5**

**Задача 1**

**Частина 1**. Постановка завдання

**Умова:**



**Частина 2**. Схема класів

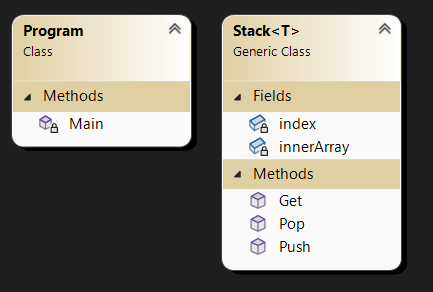


Рисунок 1 - Схема класу

**Частина 4**. Текст програми

Program.cs

using System;

class Stack<T>

{

int index = 0;

T[] innerArray = new T[100];

public void Push(T item)

{

innerArray[index++] = item;

}

public T Pop()

{

return innerArray[--index];

}

public T Get(int k)

{

return innerArray[k];

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Task 1");

Stack<int> intStack = new Stack<int>();

Stack<string> stringStack = new Stack<string>();

Console.WriteLine("\n< Int stack >");

intStack.Push(5);

intStack.Push(4);

intStack.Push(3);

intStack.Push(2);

intStack.Push(1);

for(int i = 0; i <= 4; i++)

{

Console.WriteLine(i.ToString() + " : " + intStack.Get(i));

}

Console.WriteLine("\n< String stack >");

stringStack.Push("Atom");

stringStack.Push("Molekula");

stringStack.Push("Element");

stringStack.Push("Object");

stringStack.Push("Planet");

for (int i = 0; i <= 4; i++)

{

Console.WriteLine(i.ToString() + " : " + stringStack.Get(i));

}

}

}

**Частина 5**. Тестування

**Скриншот тестування:**

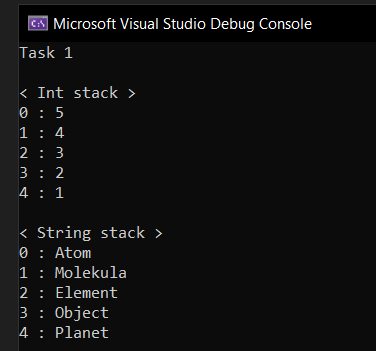
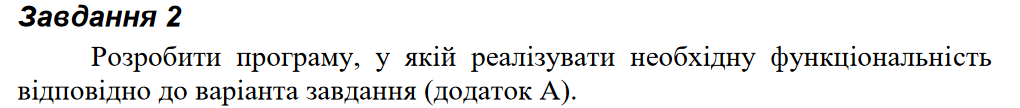


Рисунок 2 – скришот тестування

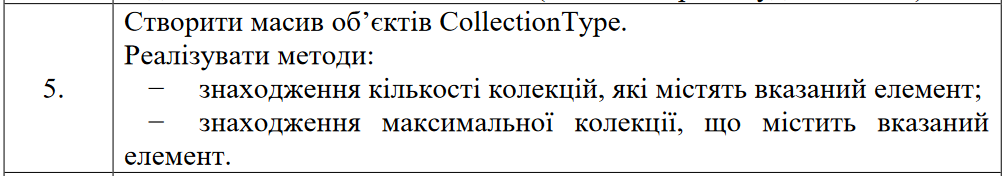
**Задача 2**

**Частина 1**. Постановка завдання

**Умова:**

****

**Умова з додатка:**



**Частина 2**. Схема класу

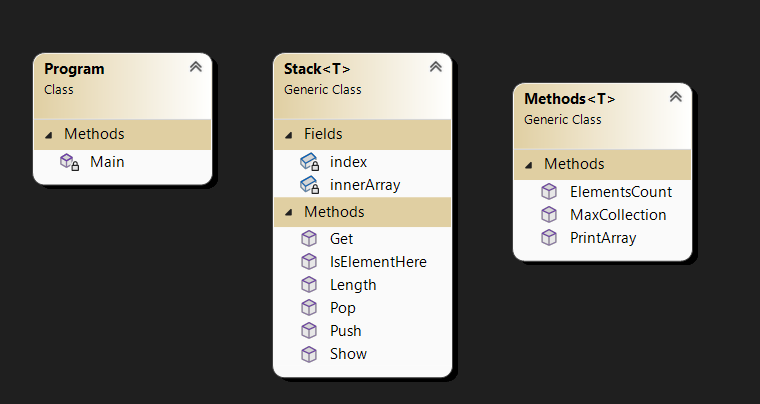


Рисунок 3 - Схема класу

**Частина 3**. Текст програми

Program.cs

using System;

using System;

class Stack<T>

{

int index = 0;

T[] innerArray = new T[100];

public void Push(T item)

{

innerArray[index++] = item;

}

public T Pop()

{

return innerArray[--index];

}

public T Get(int k)

{

return innerArray[k];

}

public bool IsElementHere(T el)

{

return innerArray.Contains(el);

}

public void Show()

{

for(int i = 0; i < Length(); i++)

{

Console.WriteLine(i.ToString() + " : " + innerArray[i]);

}

Console.WriteLine();

}

public int Length()

{

return innerArray.Count(s => s != null);

}

}

class Methods<T>

{

// Виведення всього масиву

public static void PrintArray(Stack<T>[] list, int length)

{

Console.WriteLine("Printing array\n");

for(int i = 0; i < length; i++)

{

list[i].Show();

}

}

// Знаходження кількості колекцій, що містять вказаний елемент

public static int ElementsCount(T el, Stack<T>[] list, int length)

{

int count = 0;

for (int i = 0; i < length; i++)

{

if (list[i].IsElementHere(el))

{

count++;

}

}

return count;

}

// Знаходження кількості максимальної колекції, що містить вказаний елемент

public static void MaxCollection(T el, Stack<T>[] list, int length)

{

int maxlength = 0;

Stack<T> tmp = new Stack<T>();

for (int i = 0; i < length; i++)

{

if (list[i].IsElementHere(el) && maxlength < list[i].Length())

{

tmp = list[i];

maxlength = list[i].Length();

}

}

Console.WriteLine("Max stack is : ");

tmp.Show();

Console.WriteLine("With length : " + maxlength.ToString());

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Task 2");

Stack<string>[] stringStack = new Stack<string> [10];

stringStack[0] = new Stack<string>();

stringStack[1] = new Stack<string>();

stringStack[2] = new Stack<string>();

int l = stringStack.Count(s => s != null);

stringStack[0].Push("Atom");

stringStack[0].Push("Molekula");

stringStack[0].Push("Element");

stringStack[0].Push("Object");

stringStack[1].Push("Atom");

stringStack[1].Push("Molekula");

stringStack[1].Push("Object");

stringStack[1].Push("Planet");

stringStack[2].Push("Atom");

stringStack[2].Push("Molekula");

stringStack[2].Push("Object");

stringStack[2].Push("Planet");

for(int i = 0; i < l; i++)

{

Console.WriteLine("< Stack " + i.ToString() + " >");

stringStack[i].Show();

}

Console.WriteLine("Element 'Planet' : " + Methods<string>.ElementsCount("Planet", stringStack, l).ToString());

Console.WriteLine("Element 'Element' : " + Methods<string>.ElementsCount("Element", stringStack, l).ToString());

Console.WriteLine("Element 'Atom' : " + Methods<string>.ElementsCount("Atom", stringStack, l).ToString());

Console.WriteLine();

Methods<string>.MaxCollection("Element", stringStack, 3);

}

}

**Частина 4**. Тестування

**Скриншоти тестування:**

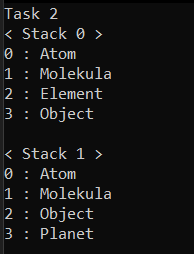
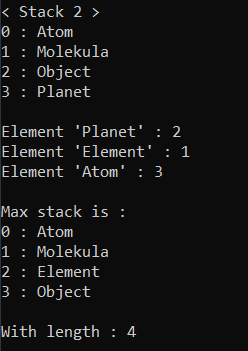
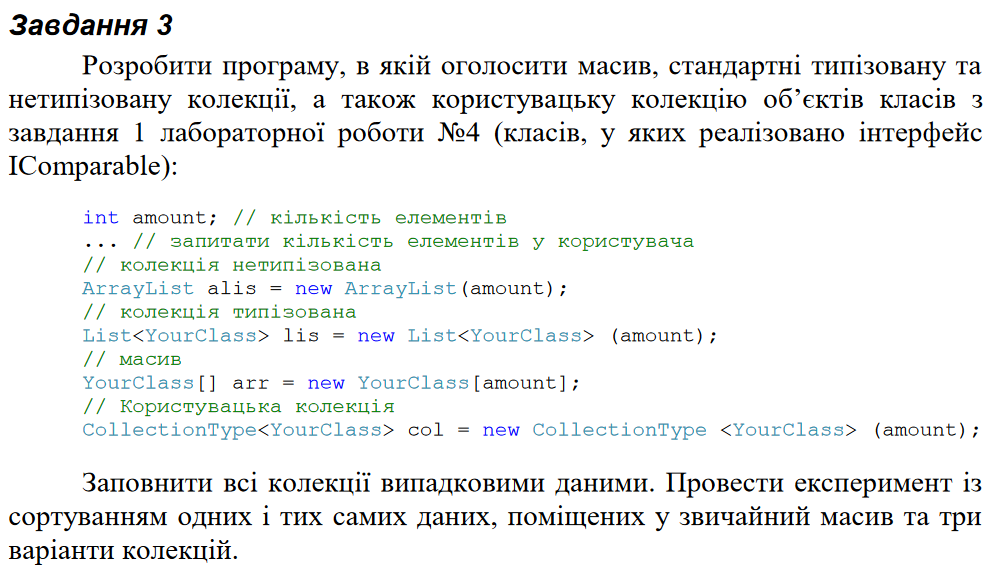
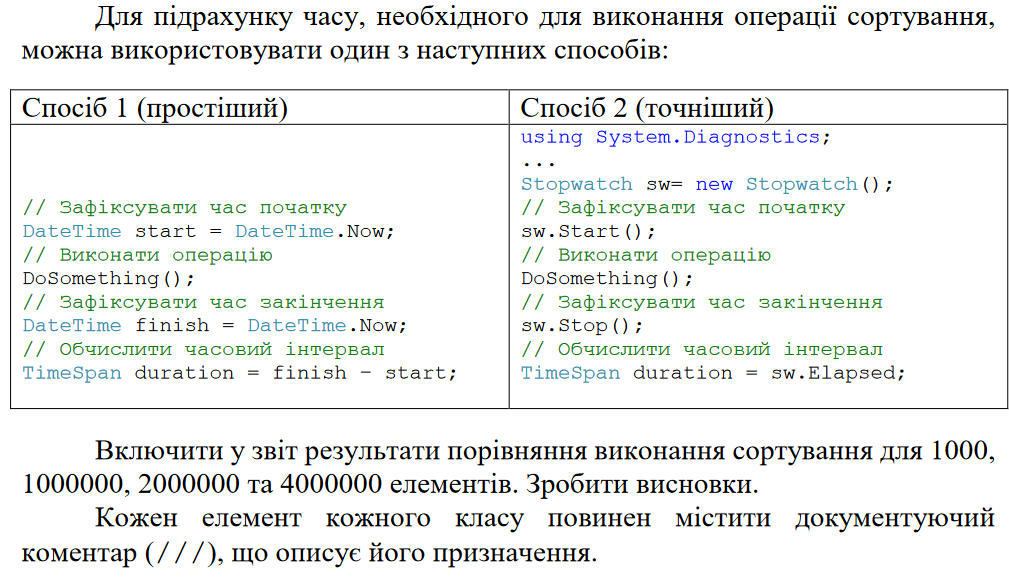
 

Рисунок 4 – скришоти тестування

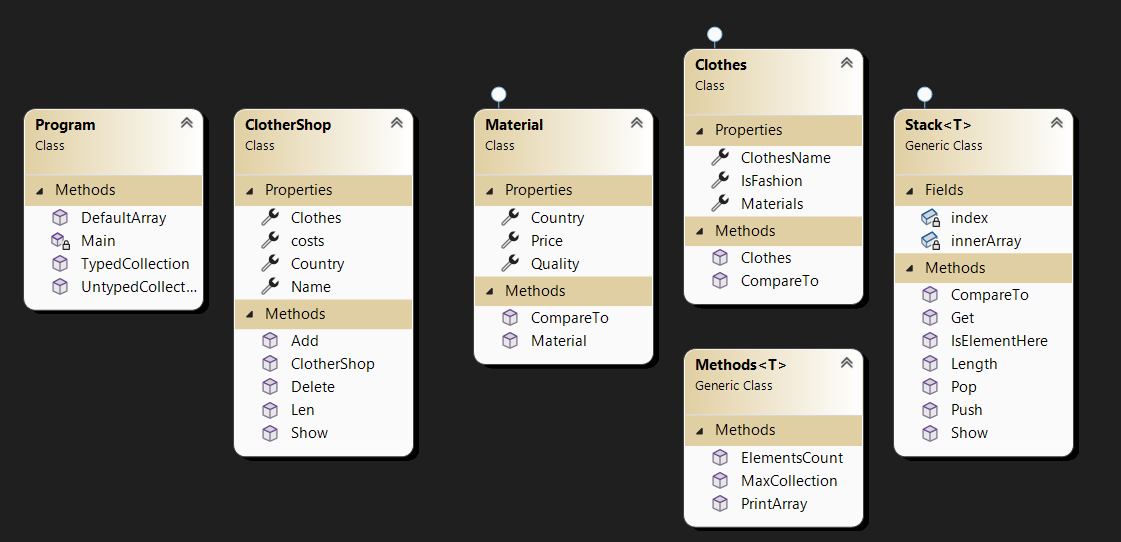
**Задача 3**

**Частина 1**. Постановка завдання

**Умова:**

****

**Частина 2.** Схема класів



Малюнок 5 – Схема класів

**Частина 3**. Текст програми

Program.cs

using System;

using System.Collections;

using Task\_3;

namespace Task\_3

{

class Program

{

public static void DefaultArray(int l)

{

ArrayList arr = new ArrayList(l);

for (int i = 0; i < l; i++)

{

if (i % 3 == 0)

{

arr.Add(3);

}

else if (i % 4 == 0)

{

arr.Add(4);

}

else if (i % 5 == 0)

{

arr.Add(5);

}

else

{

arr.Add(0);

}

}

DateTime start = DateTime.Now;

Array.Sort(arr.ToArray());

DateTime finish = DateTime.Now;

TimeSpan duration = finish - start;

Console.WriteLine("Finished! Sort time : " + duration.ToString());

}

public static void TypedCollection(int l)

{

Material Cotton = new Material(50, 100, "Ukraine");

Material Silk = new Material(90, 50, "Poland");

Material[] forCoat = new Material[2];

forCoat[0] = Cotton;

forCoat[1] = Silk;

Clothes Coat = new Clothes(forCoat, "Premium Coat", true);

Clothes Jeens = new Clothes(forCoat, "Cool Jeens", false);

Clothes Pants = new Clothes(forCoat, "Pants from hell", true);

Clothes[] goods = new Clothes[l];

for (int i = 0; i < l; i++)

{

if (i == 0 || i % 3 == 0)

{

goods[i] = Coat;

}

else if (i == 1 || i % 4 == 0)

{

goods[i] = Jeens;

}

else if (i == 2 || i % 5 == 0)

{

goods[i] = Pants;

}

else

{

goods[i] = Coat;

}

}

DateTime start = DateTime.Now;

Array.Sort(goods);

DateTime finish = DateTime.Now;

TimeSpan duration = finish - start;

Console.WriteLine("Finished! Sort time : " + duration.ToString());

}

public static void UntypedCollection(int l)

{

Stack<string>[] stringStack = new Stack<string>[l];

for(int i = 0; i < l; i++)

{

stringStack[i] = new Stack<string>();

if(i == 0 || i % 3 == 0)

{

stringStack[i].Push("Molekula");

}

else if( i == 1 || i % 4 == 0)

{

stringStack[i].Push("Atom");

stringStack[i].Push("Molekula");

}

else if( i == 2 || i % 5 == 0)

{

stringStack[i].Push("Atom");

stringStack[i].Push("Molekula");

}

else

{

stringStack[i].Push("Molekula");

}

}

DateTime start = DateTime.Now;

Array.Sort(stringStack);

DateTime finish = DateTime.Now;

TimeSpan duration = finish - start;

Console.WriteLine("Sort time : " + duration.ToString());

}

static void Main(string[] args)

{

int[] elements = { 10, 50, 100, 200 };

Console.WriteLine("Default array");

Console.WriteLine("====================================================");

for (int i = 0; i < elements.Length; i++)

{

Console.WriteLine("Elements count : " + elements[i].ToString());

DefaultArray(elements[i]);

}

Console.WriteLine("====================================================\n");

Console.WriteLine("Typed collection");

Console.WriteLine("====================================================");

for (int i = 0; i < elements.Length; i++)

{

Console.WriteLine("Elements count : " + elements[i].ToString());

TypedCollection(elements[i]);

}

Console.WriteLine("====================================================\n");

Console.WriteLine("Untyped collection");

Console.WriteLine("====================================================");

for(int i = 0; i < elements.Length; i++)

{

Console.WriteLine("Elements count : " + elements[i].ToString());

UntypedCollection(elements[i]);

}

Console.WriteLine("====================================================\n");

}

}

}

Lab6\_1-2.cs

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task\_3

{

class Stack<T> : IComparable<Stack<T>>

{

int index = 0;

T[] innerArray = new T[5000000];

public void Push(T item)

{

innerArray[index++] = item;

}

public T Pop()

{

return innerArray[--index];

}

public T Get(int k)

{

return innerArray[k];

}

public bool IsElementHere(T el)

{

return innerArray.Contains(el);

}

public void Show()

{

for (int i = 0; i < Length(); i++)

{

Console.WriteLine(i.ToString() + " : " + innerArray[i]);

}

Console.WriteLine();

}

public int Length()

{

return innerArray.Count(s => s != null);

}

public int CompareTo(Stack<T> other)

{

return Length().CompareTo(other.Length());

}

}

class Methods<T>

{

// Виведення всього масиву

public static void PrintArray(Stack<T>[] list, int length)

{

Console.WriteLine("Printing array\n");

for (int i = 0; i < length; i++)

{

list[i].Show();

}

}

// Знаходження кількості колекцій, що містять вказаний елемент

public static int ElementsCount(T el, Stack<T>[] list, int length)

{

int count = 0;

for (int i = 0; i < length; i++)

{

if (list[i].IsElementHere(el))

{

count++;

}

}

return count;

}

// Знаходження кількості максимальної колекції, що містить вказаний елемент

public static void MaxCollection(T el, Stack<T>[] list, int length)

{

int maxlength = 0;

Stack<T> tmp = new Stack<T>();

for (int i = 0; i < length; i++)

{

if (list[i].IsElementHere(el) && maxlength < list[i].Length())

{

tmp = list[i];

maxlength = list[i].Length();

}

}

Console.WriteLine("Max stack is : ");

tmp.Show();

Console.WriteLine("With length : " + maxlength.ToString());

}

}

}

Laba4\_1.cs

using System.Data;

namespace Task\_3

{

public class Clothes : IComparable<Clothes>

{

public Clothes(Material[] materials, string clothesName, bool isFashion)

{

Materials = materials;

ClothesName = clothesName;

IsFashion = isFashion;

}

public Material[] Materials { get; set; }

public string ClothesName { get; set; }

public bool IsFashion { get; set; }

public int CompareTo(Clothes other)

{

return ClothesName.CompareTo(other.ClothesName);

}

}

public class Material : IComparable<Material>

{

public Material(int price, int quality, string country)

{

Price = price;

Quality = quality;

Country = country;

}

public int Price { get; set; }

public int Quality { get; set; }

public string Country { get; set; }

public int CompareTo(Material other)

{

return Price.CompareTo(other.Price);

}

}

public class ClotherShop

{

public ClotherShop(Clothes[] clothes, string name, string country, int costs)

{

Clothes = clothes;

Name = name;

Country = country;

this.costs = costs;

}

public int Len()

{

int length = 0;

for (int i = 0; i < Clothes.Length; i++)

{

if (Clothes[i] == null)

{

break;

}

length++;

}

return length;

}

public void Add(Clothes obj)

{

Clothes[Len()] = obj;

}

public void Delete()

{

Clothes[Len() - 1] = null;

}

public void Show()

{

// define length

for (int i = 0; i < Clothes.Length; ++i)

{

if (Clothes[i] == null)

{

break;

}

Console.Write(Clothes[i].ClothesName + "\n");

}

Console.WriteLine();

}

public Clothes[] Clothes { get; set; }

public string Name { get; set; }

public string Country { get; set; }

public int costs { get; set; }

}

}

**Частина 4**. Тестування

**Скриншот тестування:**

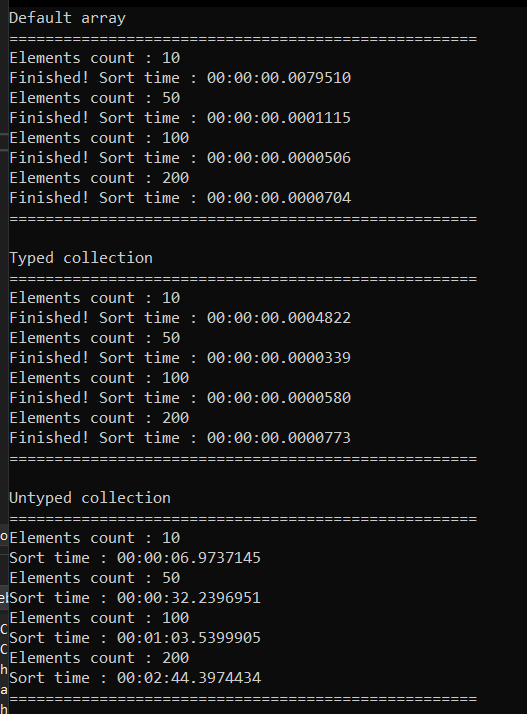


Рисунок 6 – скришот тестування

Графік 1 – графік порівняння швидкості сортування з різними типами даних

**Висновки**

Під час цієї лабораторної роботи я вивчив різновиди та призначення колекцій, дослідив особливості додавання, видалення, сортування значень у колекціяї, освоїв створення власних колекцій.