

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

з дисципліни	<i>Колекції</i>
	(назва лабораторної роботи)
	<i>Технології програмування</i>
	(шифр)

ХАІ.503.525а.03О.123-Комп'ютерна інженерія, ПЗ №9629619

Виконав студент гр.	<u>525а</u>	<u>Литвиненко А.В.</u>
28.11.2022	(№ групи)	(П.І.Б.)

(підпис, дата)

Перевірів	<u>канд. техн. наук, доцент</u>
-----------	---------------------------------

_____ (підпис, дата)	<u>Є. В. Бабешенко</u> (П.І.Б.)
-------------------------	------------------------------------

Харків – 2022

Тема роботи: колекції

Мета роботи:

Вивчити різновиди та призначення колекцій, дослідити особливості додавання, видалення, сортування значень у колекціях. Освоїти створення власних колекцій.

Варіант 5

Задача 1

Частина 1. Постановка завдання

Умова:

Завдання 1

Реалізувати власну колекцію у вигляді узагальненого типу (параметризованого класу) `CollectionType<T>`. Визначити конструктори, методи додавання та видалення елементів, інші необхідні методи та, якщо потрібно, перевантажені операції. Визначити індексатори та властивості. `CollectionType` можна реалізувати на основі стандартних колекцій (`ArrayList`, `List`, `Stack`, `Array` тощо). Передбачити оброблення можливих винятків.

Частина 2. Схема класів

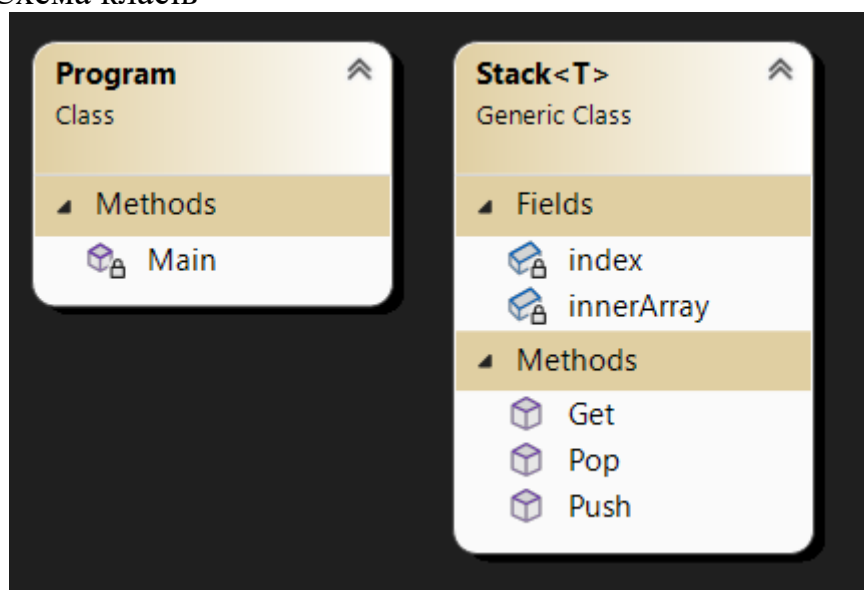


Рисунок 1 - Схема класу

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 >");

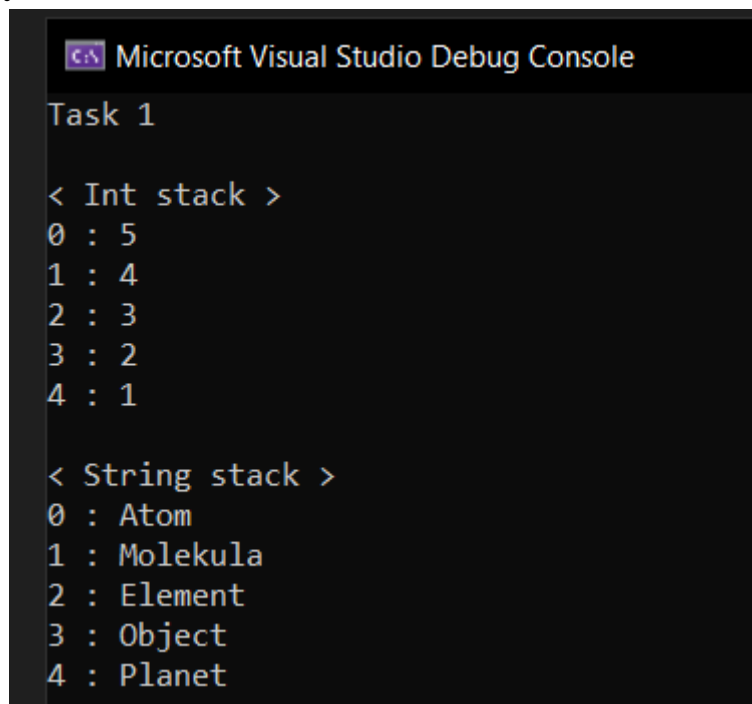
        intStack.Push(5);
        intStack.Push(4);
        intStack.Push(3);
        intStack.Push(2);
        intStack.Push(1);
        for(int i = 0; i <= 4; i++)
        {
            Console.WriteLine(i.ToString() + " : " + intStack.Get(i));
        }

        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:



```
Microsoft Visual Studio Debug Console

Task 1

< Int stack >
0 : 5
1 : 4
2 : 3
3 : 2
4 : 1

< String stack >
0 : Atom
1 : Molekula
2 : Element
3 : Object
4 : Planet
```

Рисунок 2 – скришот тестування

Задача 2

Частина 1. Постановка завдання

Умова:

Завдання 2

Розробити програму, у якій реалізувати необхідну функціональність відповідно до варіанта завдання (додаток А).

Умова з додатка:

5.	Створити масив об'єктів <code>CollectionType</code> . Реалізувати методи: <ul style="list-style-type: none">– знаходження кількості колекцій, які містять вказаний елемент;– знаходження максимальної колекції, що містить вказаний елемент.
----	--

Частина 2. Схема класу

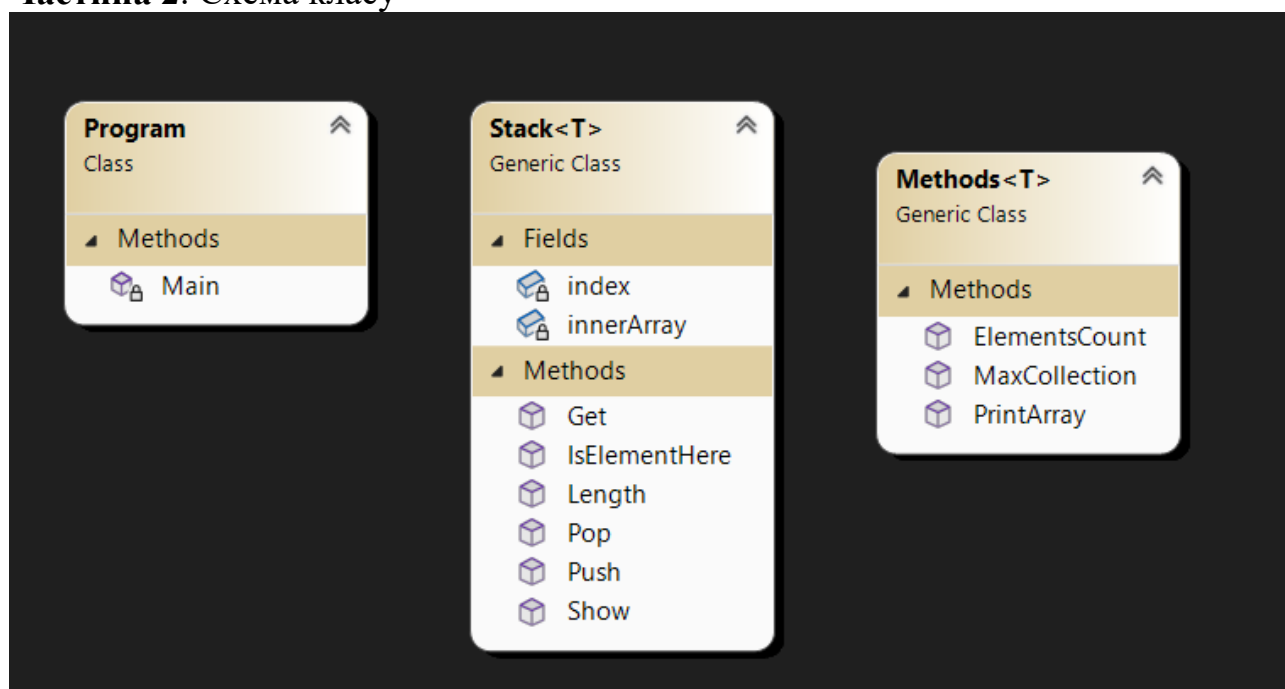


Рисунок 3 - Схема класу

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:

```

Task 2
< Stack 0 >
0 : Atom
1 : Molekula
2 : Element
3 : Object

< Stack 1 >
0 : Atom
1 : Molekula
2 : Object
3 : Planet

< Stack 2 >
0 : Atom
1 : Molekula
2 : Object
3 : Planet

Element 'Planet' : 2
Element 'Element' : 1
Element 'Atom' : 3

Max stack is :
0 : Atom
1 : Molekula
2 : Element
3 : Object

With length : 4

```

Рисунок 4 – скриншоти тестування

Задача 3

Частина 1. Постановка завдання

Умова:

Завдання 3

Розробити програму, в якій оголосити масив, стандартні типізовану та нетипізовану колекції, а також користувацьку колекцію об'єктів класів з завдання 1 лабораторної роботи №4 (класів, у яких реалізовано інтерфейс `Comparable`):

```
int amount; // кількість елементів
... // запитати кількість елементів у користувача
// колекція нетипізована
ArrayList alis = new ArrayList(amount);
// колекція типізована
List<YourClass> lis = new List<YourClass> (amount);
// масив
YourClass[] arr = new YourClass[amount];
// Користувацька колекція
CollectionType<YourClass> col = new CollectionType <YourClass> (amount);
```

Заповнити всі колекції випадковими даними. Провести експеримент із сортуванням одних і тих самих даних, поміщених у звичайний масив та три варіанти колекцій.

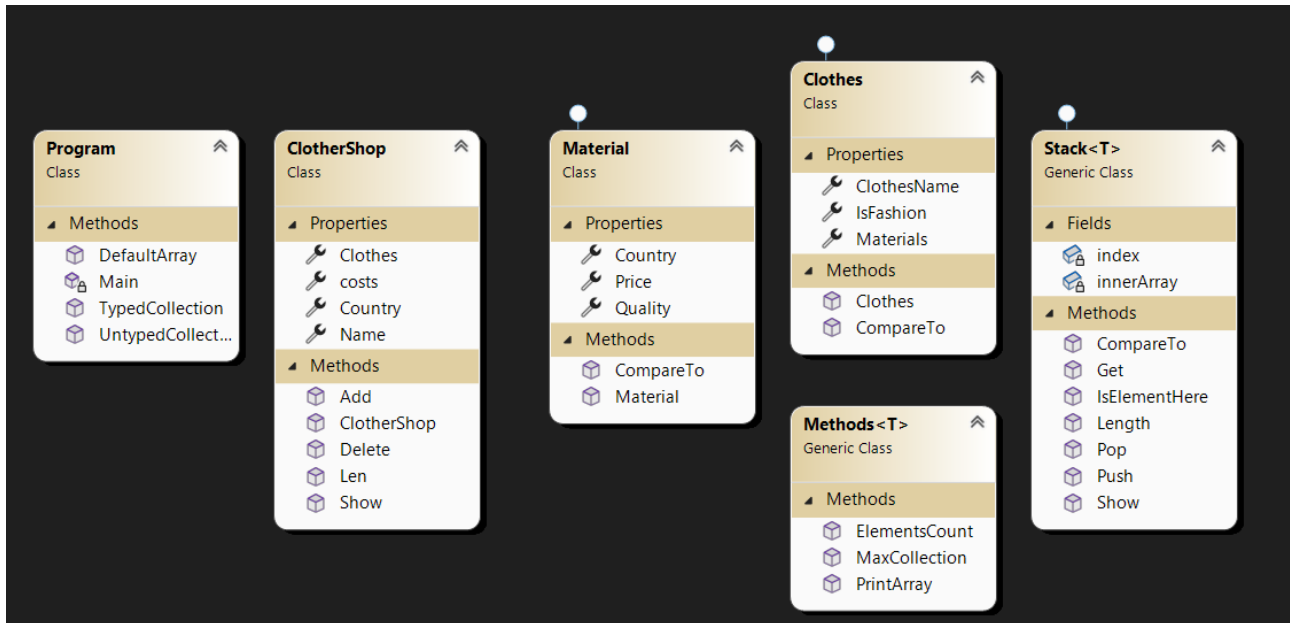
Для підрахунку часу, необхідного для виконання операції сортування, можна використовувати один з наступних способів:

Спосіб 1 (простіший)	Спосіб 2 (точніший)
<pre>// Зафіксувати час початку DateTime start = DateTime.Now; // Виконати операцію DoSomething(); // Зафіксувати час закінчення DateTime finish = DateTime.Now; // Обчислити часовий інтервал TimeSpan duration = finish - start;</pre>	<pre>using System.Diagnostics; ... Stopwatch sw= new Stopwatch(); // Зафіксувати час початку sw.Start(); // Виконати операцію DoSomething(); // Зафіксувати час закінчення sw.Stop(); // Обчислити часовий інтервал TimeSpan duration = sw.Elapsed;</pre>

Включити у звіт результати порівняння виконання сортування для 1000, 1000000, 2000000 та 4000000 елементів. Зробити висновки.

Кожен елемент кожного класу повинен містити документуючий коментар (`///`), що описує його призначення.

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 Jeans = new Clothes(forCoat, "Cool Jeans", 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] = Jeans;
                }
                else if (i % 5 == 0)
                {
                    goods[i] = Pants;
                }
            }
        }
    }
}
```

```

        {
            goods[i] = Jeans;
        }
        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 ClothierShop
    {
        public ClothierShop(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.WriteLine(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

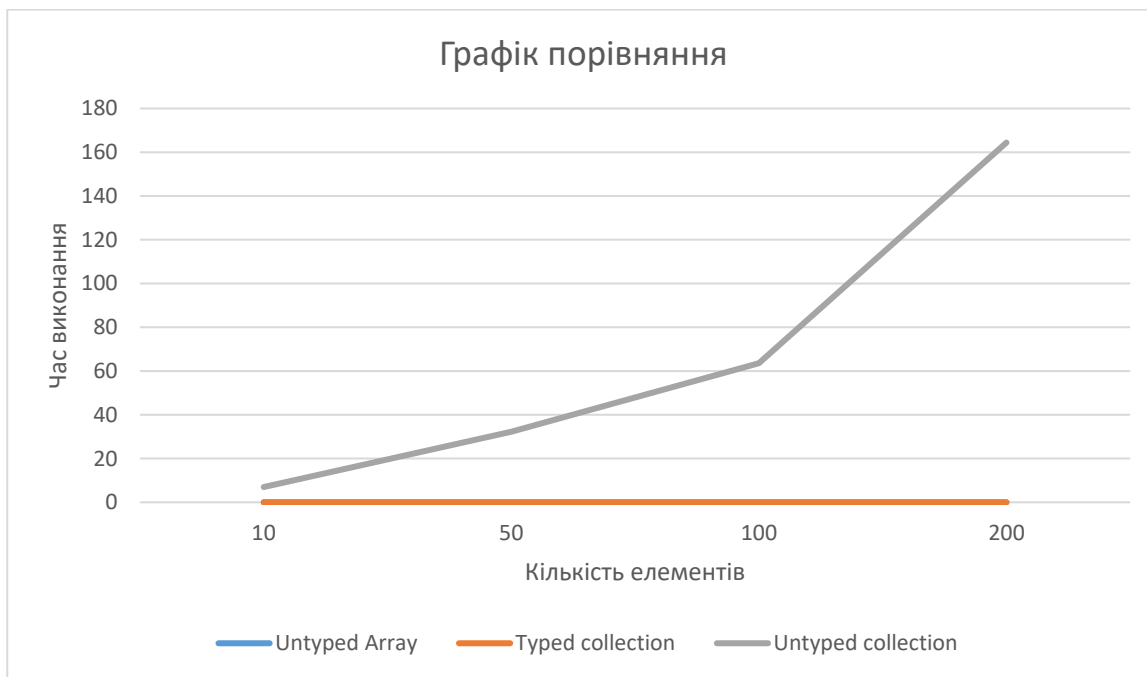
Скриншот тестування:

```
Default array
=====
Elements count : 10
Finished! Sort time : 00:00:00.0079510
Elements count : 50
Finished! Sort time : 00:00:00.0001115
Elements count : 100
Finished! Sort time : 00:00:00.0000506
Elements count : 200
Finished! Sort time : 00:00:00.0000704
=====

Typed collection
=====
Elements count : 10
Finished! Sort time : 00:00:00.0004822
Elements count : 50
Finished! Sort time : 00:00:00.0000339
Elements count : 100
Finished! Sort time : 00:00:00.0000580
Elements count : 200
Finished! Sort time : 00:00:00.0000773
=====

Untyped collection
=====
Elements count : 10
Sort time : 00:00:06.9737145
Elements count : 50
Sort time : 00:00:32.2396951
Elements count : 100
Sort time : 00:01:03.5399905
Elements count : 200
Sort time : 00:02:44.3974434
=====
```

Рисунок 6 – скриншот 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 колекції, освоїв створення власних колекцій.